Diary of a Technocratic Anarchist

by Forrest Higgs

Introduction

This document collects the blog posts from 3dreplicators.com and technocraticanarchist.blogspot.com so as to provide easy off-line access and also to preserve an archival copy of the work.

Permission to publish and distribute this document was given by Forrest Higgs.

The original text and images have been extracted and converted by means of a script. This may have resulted in formatting errors or incorrectly transferred text. Please note that dead links may result in missing images. Comments have not been extracted, these can be found on the original blog site, a link to each blog post is given at the top of each post in this document. Please report any errata to contact@garyhodgson.com

Gary Hodgson, 2012

Contents

December 2006

January 2007

February 2007

March 2007

April 2007

May 2007

June 2007

July 2007

August 2007

September 2007

November 2007

January 2008

February 2008

March 2008

April 2008

May 2008

June 2008

July 2008

August 2008

September 2008

October 2008

November 2008

December 2008

January 2009

February 2009

March 2009

April 2009

May 2009

June 2009

November 2009

December 2009

January 2010

February 2010

March 2010

April 2010

May 2010

June 2010

July 2010

August 2010

September 2010

October 2010

August 2011

December 2011

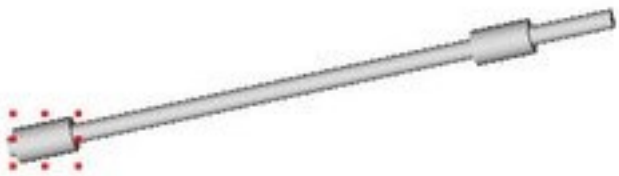
Design study for the 6.35 mm filament extruder

Wednesday, 8th March 2006 by Forrest Higgs

I put in a few hours using AoI to see if I could make all the pieces for the quarter inch filament extruder work together. I'd like you to walk through the design with me and comment if you can see problems with it that I've missed. I put in a few hours using AoI to see if I could make all the pieces for the quarter inch filament extruder work together. I'd like you to walk through the design with me and comment if you can see problems with it that I've missed.

First, I'm using a quarter-inch diameter wood auger bit that is just over seven inches long. It has eight flights in the screw for a length of four inches, two inches of bushing behind that and a shank that extends for an inch and a quarter beyond that. I didn't take time to develop the screw portion of the bit on the left on AoI and placed a thrust bearing where the bushing meets the shank on the right. The thrust bearing is made of a drilled out 3/8ths or 1/2 inch bolt.

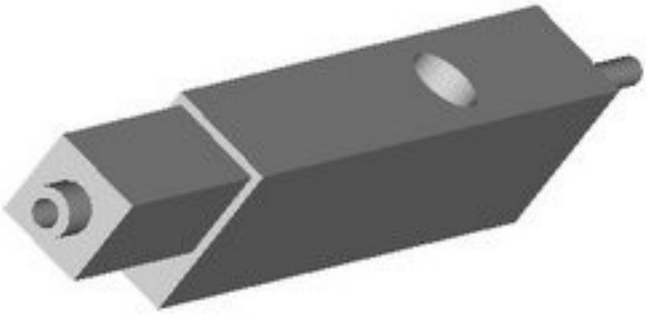
From my previous experiments I determined that the auger should extend slightly beyond the end of the pumping section to preclude polymer packing in the pump barrel.



You can see the end of the auger extend just beyond the end of the connector plug that I've made from another one of those drilled bolts from which I've sawn off the head.

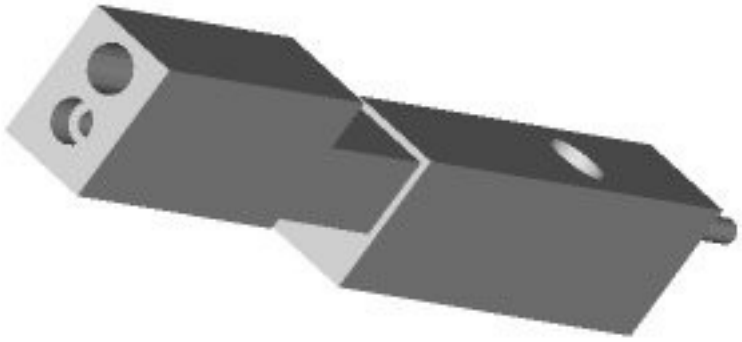
Now I include the body of the polymer pump.

You can also see the polymer feed into the top of the pump and the feed chamber where it intersects with the auger.



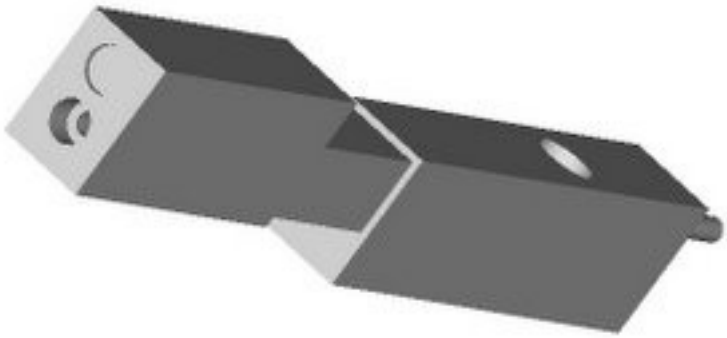
Next we screw the PTFE thermal barrier onto the connector plug. I've shown the PTFE bar, which has been drilled to a quarter inch to accomodate ploymer flow and tapped at each end to accomodate the connector plugs, as slightly smaller in cross section than it is so that you can see where the steel polymer pump section and the PTFE thermal barrier join.

We can now screw the heated extruder barrel onto the second connector plug seated in the end of the PTFE thermal barrier.

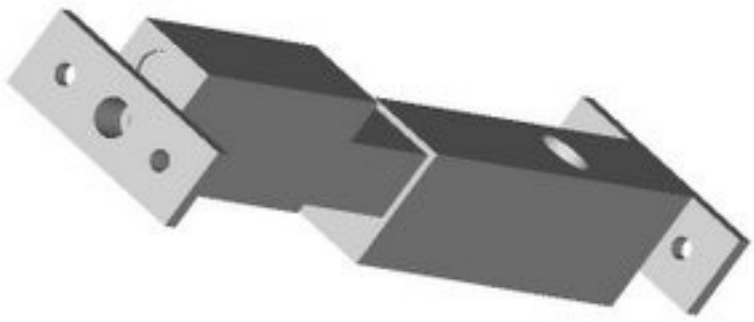


You can see that the heated barrel is drilled not only to allow passage of polymer as it is heated and is pumped towards the extruder tip but also to accomodate a 300 watt cartridge heater.

Now let us insert the cartridge heater.

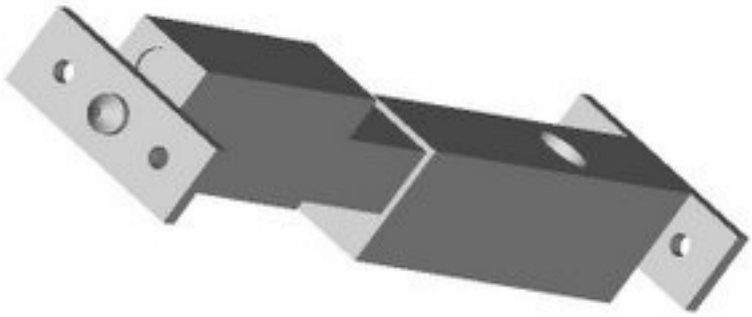


There is a quite inexpensive anti-lock compound that will prevent the cartridge heater from freezing up in its housing after a number of heating cycles. You will have to imagine the wires for the cartridge heater extending out the back of its housing towards the rear of the extruder. These cartridge heaters use regular lines electricity.You can also quite easily acquire bimetallic thermostats which can be used to control the temperature of the system. Like the cartridge heaters, these thermostats are very old technology and are also quite inexpensive.

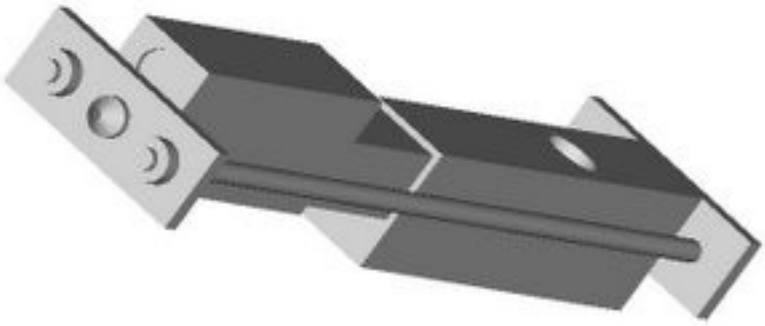


From this point we can apply the front and back retainer plates.

These are held in place by screwing the extruder tip into the heated extruder barrel on the front side and screwing the thrust bearing into the pump barrel on the back side of the assembly.

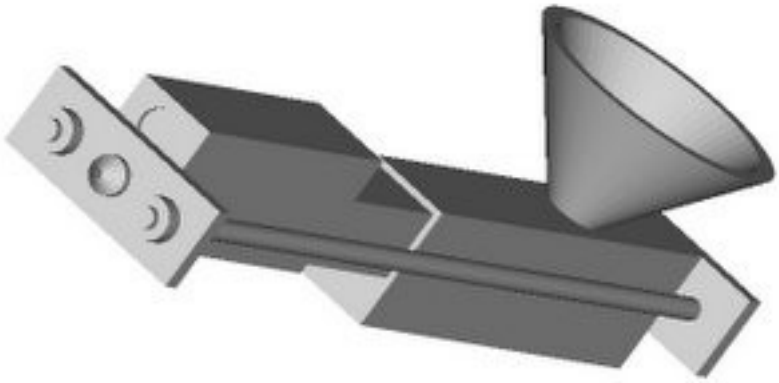


From there you secure the whole device with retaining rods made from threaded studding which are secured with lock nuts on both sides of the extruder.

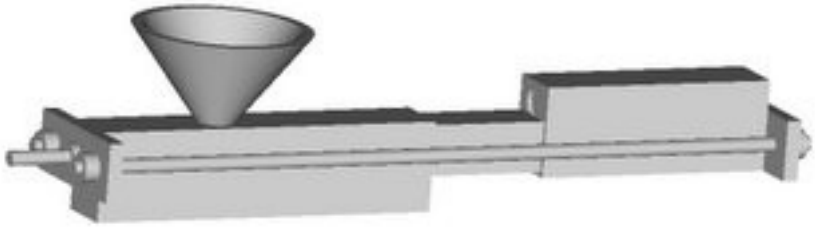


This arrangement hopefully will preclude the extruder from coming apart from the internal pressure of some 40 atmospheres created by the pumping action of the polymer pump. I've calculated that this pressure will create an axial thrust of about thirty pounds.

Once that is done you simply add the feed funnel for directing polymer powder resin into the polymer pump feed chamber and you should be good to go.



I am planning on taking the newly conservative approach that Adrian has taken of using an electric screwdriver to drive the system. I am guessing that since Adrian has already demonstrated that an electric screw driver will drive his 13.5 mm extruder it should be more than enough for my quarter inch device.
I was going to use the gear motor for the Mk II, but decided that there was too much drama in accomodating such a small piece of equipment into what is a rather large (about 1 foot long and weighing 5-10 lbs) piece of equipment.



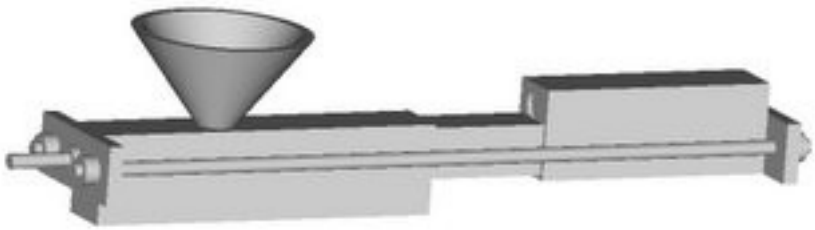
Using an electric screwdriver will, I think, make the question of securing the system to a mounting block considerably easier.
I should also be able to run it without having to resort to complicated drive and control schemes. As for operating the system, I already own an infrared thermometer which will allow me to measure the surface temperature of the heated extruder barrel rather well. Initially, I plan to let the system heat up slowly whilst empty by adjusting the thermostat (which I will tap mount into the chin of the extruder barrel, not shown) until I get the temperature I want. At that point I will start the electric screwdriver and start feeding polymer into the system. The thermal capacity of the system is very much larger than the that of the polymer flow, so I doubt that the system will much notice the polymer being extruded from it energetically.
This design largely emerged as an effort to salvage some of the materials bought for the old Gingery extruder but not used. It was also designed with an eye towards being easy to break down and clean in case of jams and allows for the performance of different polymers and also different diameter extruder tips to be surveyed. The cartridge heater is also magnitudes more robust than the hair-think nichrome wire that we are currently using in the Mk II. Adrian noted that dipping his extruder head in hot water to get polymer out wouldn't be good for his device. I can demount the extruder barrel by loosening the retainer rods, slide the heater cartridge heater out, remove the retaining plate and extruder tip and immerse the extruder barrel into boiling water with no danger whatsoever. :-)
Now... objections... observations?

Building the 6.35 mm filament extruder

Wednesday, 8th March 2006 by Forrest Higgs

The weather held this weekend so I was able to almost finish the 6.35 mm filament extruder prototype.

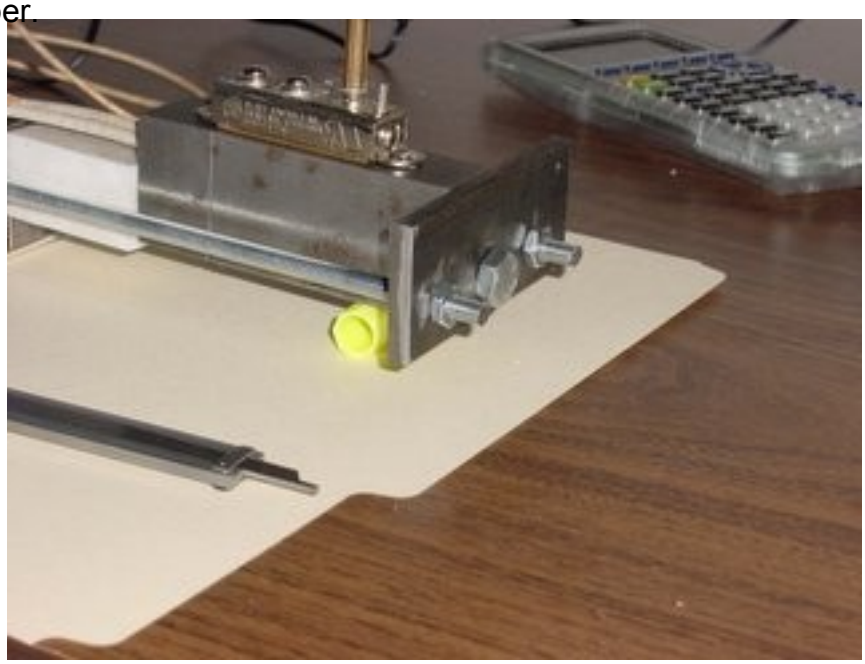
I am using Adrian's size convention in setting the calipers open to 20 mm for scale.
You can see that it bears a close resemblance to what my design study conception looked like.

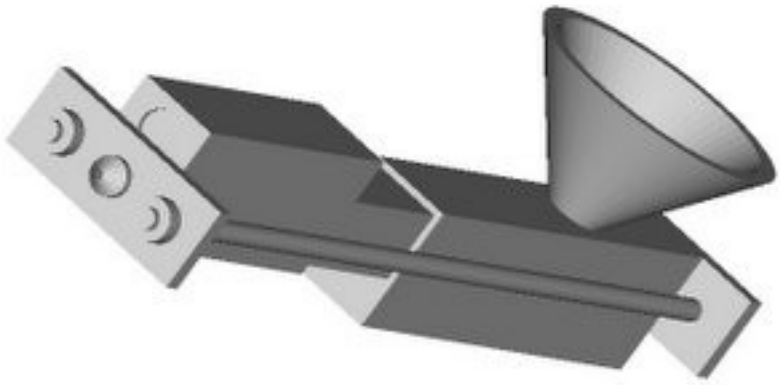


I oversized the PTFE sleeve in the thermal barrier so the design sags a bit right now. Following Brett's suggestion I will put a third support plate between the polymer pump and the PTFE thermal barrier to counteract any tendency towards buckling and more importantly, to provide a bit more support to the heavy steel extruder barrel to at the right hand side. That plate will support the retaining rods via PTFE sleeves so that heat moving down the retaining rods won't be transferred into the front of the polymer pump.
I had originally thought to use a 1 x 1 x 2 inch PTFE thermal barrier with two PTFE cylindrical plugs to connect it to the polymer pump at the left and the heated extruder barrel at the right. While I was building the extruder I realised that that would make four potential gaps in the polymer flow path between the pump and the extruder tip. I brought that down to two by drilling out a 1/2

inch cylinder of PTFE and then drilling a 1/2 inch hole in the 1 x 1 x 2 inch PTFE bar so that it acted as a sleeve for the cylinder. That worked quite nicely.

Here is a detail of the polymer pump. I got too tired to finish the thrust bushing for the auger bit so the auger is seen lying in the foreground of the pump. I am using an ordinary plastic funnel for a polymer feed hopper.





Finally, here is a detail of the extruder block and a mockup (3/8 -24 1/2 inch bolt) of the extruder tip. Again, I was too tired at the end of the day to attempt to drill out an extruder tip. I've done trial runs on both the extruder tip and the thrust bushing, both made from 3/8 inch bolts and found that to be very finicky work. I've got a handful of the bolts so I can afford a lot of errors. :-)

You can see the wires from the cartridge heater in the photo and also the thermostat sitting atop the heated extrusion barrel.
I have to tap in the thermostat and lock the extruder down on a block. I also have to mate the shank of the auger bit with the electric screwdriver you've seen in the pictures. That all will take another morning, I expect.
It's going to be another two-tylenol night. Right now I am for some supper, a shot of MacAllan 12 year old, a movie, a hot bubble bath and a good sleep. It's supposed to begin to storm in about an hour. We've got another pineapple express storm that's come up from Hawai'i this evening.

Monday tests on the 6.35 mm filament extruder

Wednesday, 8th March 2006 by Forrest Higgs

I decided to run a few preliminary tests on the 6.35 mm filament extruder before I did further work on it.

I used semolina as a proxy for polymer resin to check to see if the assembly would pump. There was no problem with that. I then inserted the PTFE thermal barrier into the assembly to see if I could pump semolina through that.

The semolina immediately jammed in the thermal barrier passage. It formed a 15 mm plug almost exactly like that I encountered when using the same auger bit in 1/4 inch steel pipe except the steel pipe only made a 5 mm plug before jamming.

The plug was easy enough to clear once I detached the thermal barrier from the polymer pump assembly.

Think that the problem might have been specific to the semolina I then used a few grams of the CAPA resin in powder form that I recently acquired as samples. The same thing happened.

Any ideas?

Tests on the 6.35 mm filament extruder

Wednesday, 8th March 2006 by Forrest Higgs

After reviewing Vik and Brett's comments, I decided to see if I could pump polymer through a PTFE barrel with an auger bit. It turns out that contrary to what I'd read, viz, that the wall's friction coefficient had to be higher than the auger's isn't so. CAPA pumped through a drilled length of PTFE with no trouble at all. That means that the whole pump and thermal break assembly could be made out of a single piece of PTFE.

There are two practical problems with doing that, though. First, PTFE is expensive at US$0.15/cm^3 (US$2.50/in^3). That wouldn't be so bad if I weren't worried that you'd be replacing it regularly because of wear between the auger and the barrel walls.

Vik makes another, more intriguing suggestion though. What if we applied a water jacket to the pumping section. My mind immediately went back to the old water cooled Browning heavy machine guns my grandfather used in WWI. I seem to remember that they worked by allowing the water to boil off. That would create a liming problem, but those are relatively easy to sort out.

Hmmm.... Adrian! How is that prototype of yours going?

20:22 PDT AKA 04:22 ZULU TIME 7 March...

Hit the hardware store again and had to choose between two German bits. One was half again longer than I needed and was made of some mad alloy of titanium and vanadium. I figured that I'd never be able to cut the carbide tip off of it, never mind mill the shank end down a bit to accomodate a thrust bushing.

I settled for a more modest German masonry bit that will require that I shorten the thermal barrier just a touch so that the tip end of it will extend into the melt chamber. I sawed off the end of that one and tested it with the pump barrel housing and thermal barrier and it happily pumps CAPA and even pumps it in the PTFE barrel. I suspect that it would have pushed the polymer completely out of the extra 25 mm of PTFE barrel beyond the end of the bit and out of the existing PTFE thermal barrier except that I was having to resist the thrust of the drill bit manually.

While I was at the hardware stockist I got the bits of power cord and taps necessary to seat the thermostat on the extruder barrel and get the melt going. I also bought a vise so that I can try drilling extruder tips here rather than having to trek out to my sister and brother-in-laws every time I need to put something in a vise.

22:35 PDT AKA 06:35 ZULU TIME 7 March...

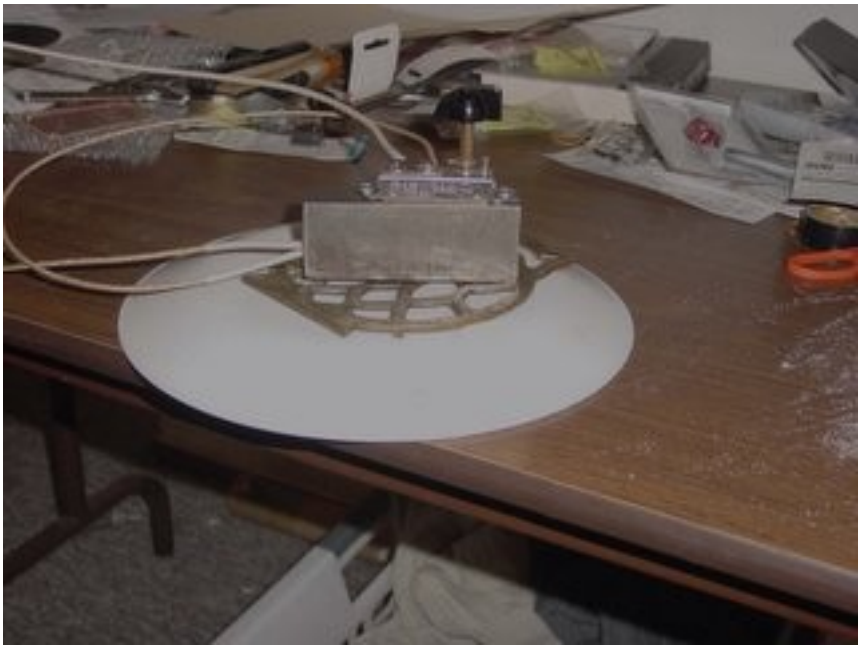
Successfully pumped polymer through the polymer pump, through the PTFE connecting barrel and into the heated extruder chamber with the new, longer masonry bit. Set up to drill a better aligned PTFE connecting barrel in situ and marked it. Drilling awaits a more godly hour.

0623 PDT AKA 14:23 ZULU TIME 7 March...

Managed to get a true hole through a PTFE cylinder on the second try. Double checking the polymer pump test when the batteries on the electric screwdriver ran down.

Time to make some breakfast. :-P

07:05 PDT AKA 15:05 ZULU TIME 7 March...
Repeated polymer pump test with recharged batteries. Interestingly, this time the pump filled the entire heated extruder barrel, 89 mm long and .89 cm^3 volume, before jamming. The additional flights provided by the masonry bit apparently makes for a more efficient pump for the amount of torque that we have. Will be drilling an extruder tip and wiring the heater barrel after breakfast. Hope to run a full test this morning.
09:43 PDT AKA 17:43 ZULU TIME 7 March...
Thermostat and power cables rigged correctly. No fires, electrocutions or explosions so far. Have my fire extinguisher on hand just in case. :-o
 Presently running tests to get a rough idea where the thermostat setpoints are. This takes time because of the thermal mass of the heater block. Taking temperature readings in the extruder mouths with the IR non-contact thermometer. The mass of the steel will predominate with this system so I don't have to run it loaded with polymer since the heater barrel weighs a couple of pounds and a full charge of polymer in the barrel less than a gram. :-)



Temperature in the barrel is at about 60 Celsius just now and the thermostat setting is at 12:00.
10:16 PDT AKA 18:16 ZULU TIME 7 March...
Thermostat setting of 12:00 stabilises at 127 Celsius. Shifted thermostat to 09:00.

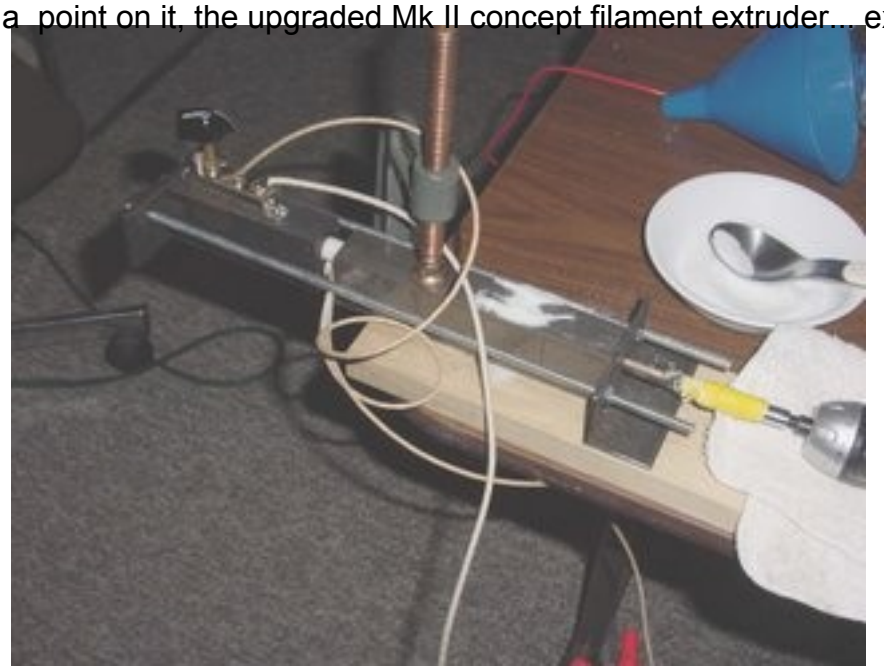
10:30 PDT AKA 18:30 ZULU TIME 7 March...
Thermostat setting of 09:00 stabilises at 94 Celsius. Shifted thermostat to 11:00. The indicator knob on that bad boy gets HOT!
10:58 PDT AKA 18:58 ZULU TIME 7 March...
Thermostat setting of 11:00 stabilises at 116 Celsius. It looks like the thermostat is pretty linear, which is what I'd hoped for. Now, I wonder what the deadband looks like.
Adrian got a bit of extrusion going at 120 Celsius, but Solvay took their melt flow index at 160, so I'd better check to see what higher settings on the thermostat gets me. The data sheets that Jeff's people in the UK so kindly sent along with the powder samples indicates that CAPA decomposes at 200 Celsius. Fortunately, there are no halogens or nitrogen in the formula for CAPA so if I screw up and burn some it's unlikely to kill me immediately.
Oh well, now to get some food and drill a extruder tip. :-)
13:30 PDT AKA 21:30 ZULU TIME 7 March...
Thermostat setting of 15:00 stabilises at 132 Celsius. I'm going to crank the thermostat up to max, about 17:00 and see what happens.
By the way, the deadband is about 5 degrees. Not bad. :-)
13:50 PDT AKA 21:50 ZULU TIME 7 March...
Thermostat setting of 17:00 (max setting) stabilises at 160 Celsius. How convenient! :-D
Looks like I'm going to have to develop a graph for the thermostat performance. That certainly isn't linear. I also need to go back and check 15:00 again.
14:15 PDT AKA 22:15 ZULU TIME 7 March...
Thermostat setting of 15:00 (max setting) stabilises at 137 Celsius. Looks like last time I caught it at the bottom of the dead band. Ok, fair enough. :-)
So far so good! :-)

We appear to have another successful filament extruder concept...

Wednesday, 8th March 2006 by Forrest Higgs

20:45 PDT AKA 04:45 ZULU TIME 8 March...

Not to put too fine a point on it, the upgraded Mk II concept filament extruder... extrudes filament.

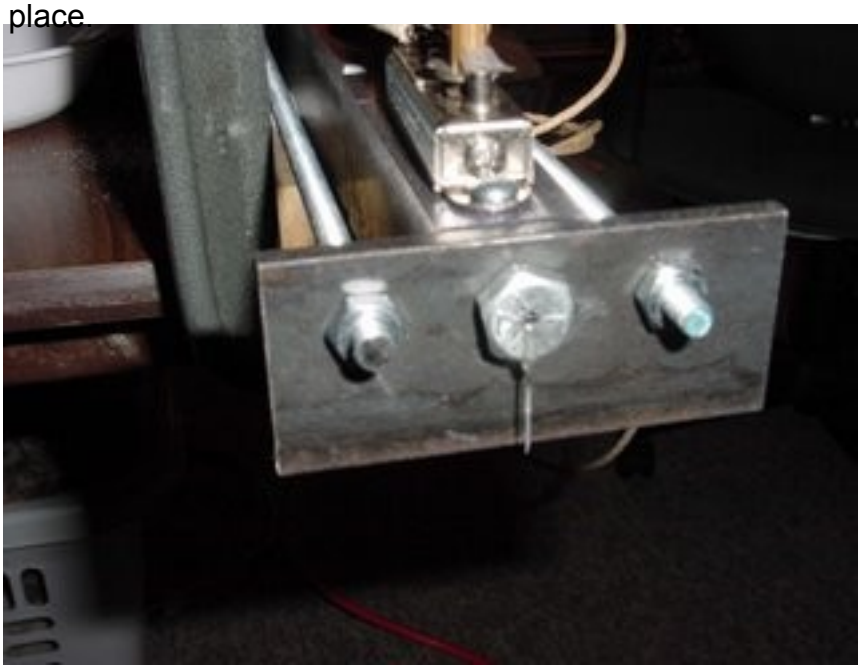


I decided to do the first run hot so that I could avoid problems with jamming. I set the thermostat on the heated barrel to 150 Celsius and let the system heat up for fifteen minutes keeping track of the temperature. When it hit 150 I began priming the polymer pump. Within 5 minutes molten polymer began to be extruded. I caught the output on a ceramic plate. The filament was too hot, as I expected and although it emerged at with a diameter of approximately 3 mm gravity quickly stretched it thin.

The product landing on the plate had no coherence in diameter.

Because I was working alone and was standing in for a thrust bearing for the test rig and spooning

polymer into the feed chamber as the experiment was underway I was unable to photograph the extrusion as it took place.



Here is a photo of the extrusion tip after I shut the system down. You can see a thread of filament remaining on the cool tip.

The first few feet of filament was dirty with grit and grease from the machining of the barrels. That soon cleared, however, and the last filament extruded was clear. CAPA turns translucent when it cools. It feels rather rubbery and is quite strong even in thin cross sections.

In the next days I will endeavour to configure the rig so that I can operate it with hands off. I will also have to figure out a way to connect the electric screwdriver to the extruder with something a little less informal than duct tape. :-)

Interestingly, the electric screwdriver provided more than enough torque to operate the system. There was no jamming of the system for any reason. As well, I was able to manually resist the axial thrust against the screw pump. Apparently, the larger diameter extrusion orfice means that the operating pressure in the heated extruder barrel is considerably less than I expected. That is wonderful! :-D

I think as well that it would be good to slightly incline the extruder so that the filament doesn't try to hang onto the extruder tip. Having a water tray directly under it would also be a good idea. Godet stations and a takeup reel system are something that somebody might want to think about reprapping before too long.

I will also do a series of test runs at lower temperatures to find a more optimal operating regimen.

I'd like to find a less informal gear motor to integrate into the system. I don't get the impression that the plastic gear motor that we use on the Mk II is going to be up to the job. I think that we will need just a little more torque than that motor can deliver. On the other hand, I get the impression that this system will produce filament at quite a good clip.

It would appear that the only reason for going to a larger diameter auger would be to be able to handle polymer in larger granules.

Using the masonry bit appears to have made all the difference at the last.

I would like to thank everybody on the team for their advice and observations. Thanks goes to Dr.

Bowyer who also ran with the concept using several different assumptions and whose example prodded me to work a lot harder than I would have on my own.
Finally, especial thanks go to Vik and Brett, whose timely last minute suggestions and encouragement yesterday got me past the jamming problems that I encountered. I was very discouraged at that point. I don't think I would have attempted to run the auger through the thermal barrier and into the extruder barrel had Brett and Vik not suggested it.
Thanks guys. This has been and continues to be a LOT of fun.

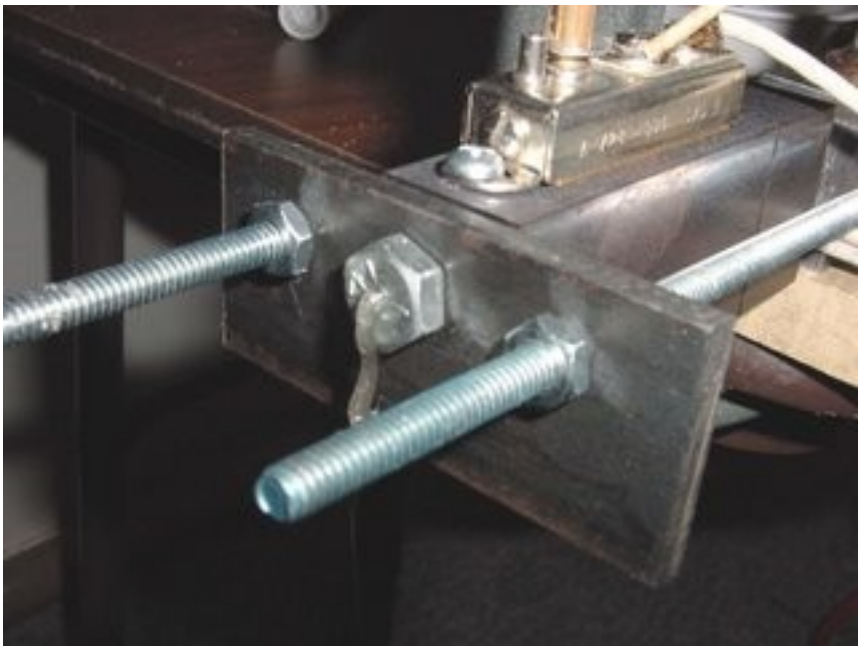
Wednesday morning tests...

Wednesday, 8th March 2006 by Forrest Higgs

07:45 PDT AKA 16:45 ZULU TIME 8 March...

This morning I did another test run of the extruder, this time with the barrel temperature set to 110. I taped down a lot of things and installed a funnel so that the feed of polymer into the pump would require less attention.

Before I powered up I noticed that the auger bit was frozen from having its tip embedded in solidified polymer in the extruder barrel. I turned on the cartridge heater as usual but this time with the extruder barrel full instead of empty. After a few minutes thermal expansion of the melting polymer forced a bit of filament out of the extruder tip before the electric screwdriver was turned on.



I tapped the on-button for the screwdriver and discovered that it rotated freely now.

At 110 Celsius the extrusion rate was considerably slower, but the filament tended to hold its

shape better. The axial thrust on the auger bit seemed to be considerably stronger, too. The emerging filament still tends to foul on the extruder tip and thread as the length of the filament increases. Beveling the extruder tip like us done on the MK II design would be a great idea. I'll pursue that this weekend when I can get out the the drill press at my sister's house.
I took note of Adrian's experience with the filament swelling after leaving the extruder tip. The tip diameter to filament diameter proportion implied in Adrian's report was about 0.79. I worked backwards from this and determined that drilling a 3/32nds diameter hole in the extruder would result in a 3 mm filament. In practice that works out fairly close from the measurements I took this morning.
It seems obvious at this point that if we are going to extrude CAPA in this temperature range we are going to have to cool and support the filament almost immediately after it leaves the extruder tip. Otherwise it is going to draw into a thin thread because of the weight of the cooled filament end acting on the emerging polymer which, because it is molten, has a negligible tensile strength. Perhaps I can fabricate a little PTFE trough to provide that support.
08:45 PDT AKA 16:45 ZULU TIME 8 March...
It occurred to me that I had a little PTFE trough already made in the reject misdrilled bits of PTFE cylinder that I had made. I sawed one of those in half and taped it into place. You can see the result. (hmmm... pics don't seem to be uploading right now)

The filament extruded into the trough all right but promptly adhered to the PTFE surface IN the trough. It seems that drilled PTFE surfaces are quite rough, a circumstance which probably explains how we can pump polymer through the PTFE thermal barrier in spite of the fact that the friction coefficient of PTFE is so low.
I also tried dripping water onto the filament as it emerged from the extruder tip.

That worked to an extent but I had to be careful not to drip it on the extruder tip which chilled it and caused flow problems.

A note on polymer fouling of the auger...

Wednesday, 8th March 2006 by Forrest Higgs

I disassembled the polymer extruder for the first time since it was assembled and tests began. As usual the auger, whose extreme end resides in the heated extruder barrel was frozen. Turning on the heater for the extruder barrel I was able to get the auger to rotate freely after a few minutes. Extracting it from the device proved, however, to be surprisingly hard.

When I recovered the auger I discovered that molten polymer had migrated through between two and three flights of the auger back towards the polymer pump of the mechanism. This melting was restricted to the PTFE thermal barrier part of the mechanism. Cleaning the auger was a trivial exercise. No polymer remained in the PTFE thermal barrier barrel.

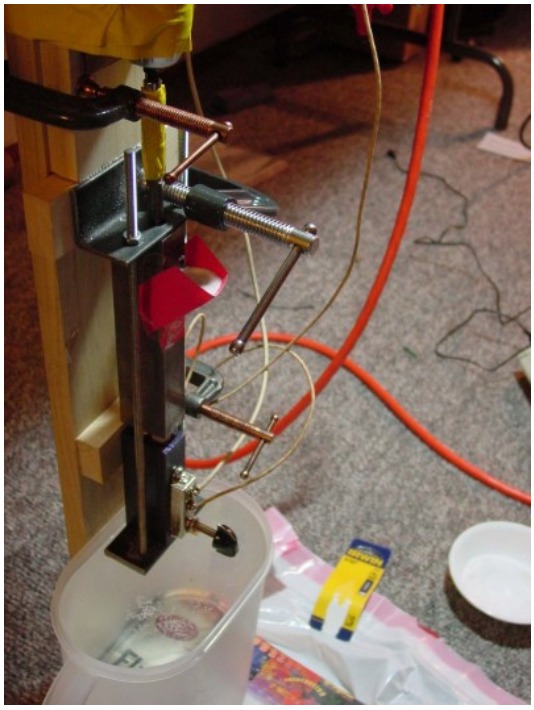
This mechanism may explain why the extrusion of filament has been somewhat irregular during the tests carried out on the extruder. During the tests I've been rather cavalier in leaving the extruder barrel heated with the polymer pump turned off. I suspect that this is when the molten polymer migration occurred.

It would appear to me that extracting and cleaning the auger before making filament would be considered a good practice rule for operating this device.

Closing on producing usuable filament...

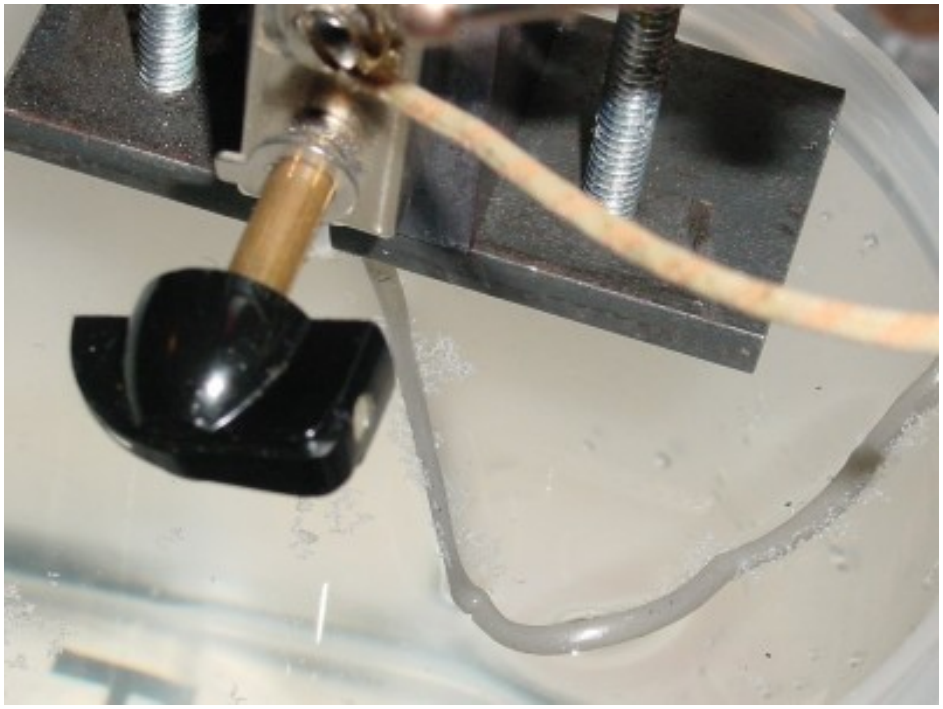
Wednesday, 8th March 2006 by Forrest Higgs

I realised late this afternoon that I need not build a framework as such to hold the quarter-inch extruder vertical. Ten minutes with a few pieces of scrap lumber and C-clamps gave me a solid platform for operating the extruder vertically. A water bath was created with a fruit juice jug and a piece of construction paper formed a makeshift polymer bin.



The water bath was situated about 2-3 mm below the extruder tip. I heated the extruder barrel to 120 Celsius and started the system.

This was the result.



The filament emerging from the extruder was instantly quenched in the water bath. The diameter of the resulting filament varied between 3.2 and 3.9 mm. The kinks that you see occurred when I had to manually poke the polymer feed bin with a thin bamboo stick to keep it filled.
Interestingly, the filament floated. This might indicate that we are entraining air in the filament since the density of the CAPA is marginally higher than that of water. No obvious bubbles were visible, however.
The practicality of Brett's advice that we need to think about designing a polishing station to straighten and regularise the dimensions of the raw filament now becomes obvious. That should be a perfect RepRapable product.
The extruder produced about 125 mm of filament in approximately 90 seconds. For these settings that is an output level of approximately 50 cm^3/hr. I had designed the system to produce about 15 times what a Mk II could consume per unit of time. We're producing about 18 times which is very close to our design goal. I currently have no way of knowing what the duty cycle of the bimetallic thermostat was, but given the setting chosen even if the cartridge heater had been on full-time it would have had a power demand of no more than 150 watts.
What we have is a prototype that proves the concept for a small polymer extruder. We need to take the concept through several generations now to achieve a useful system. I see several developments as being needed to make this a start-and-forget system.

- an improved polymer feed hopper that makes sure that the auger receives a steady supply of resin.
- roll godets
- heating oven
- takeup reel

It would be nice if this system could be self-threading. We obviously need to design a hands-off control strategy now that we have a way of making filament. Given the work that has already been done by the team with controlling steppers and the Mk II I have no doubt that we have the talent already on board to do this.
Here is the system that we are replicating.
It costs US$85,000 used. We're going to do it for under $150 tops as best as I can see.

Post extrusion treatment of filament...

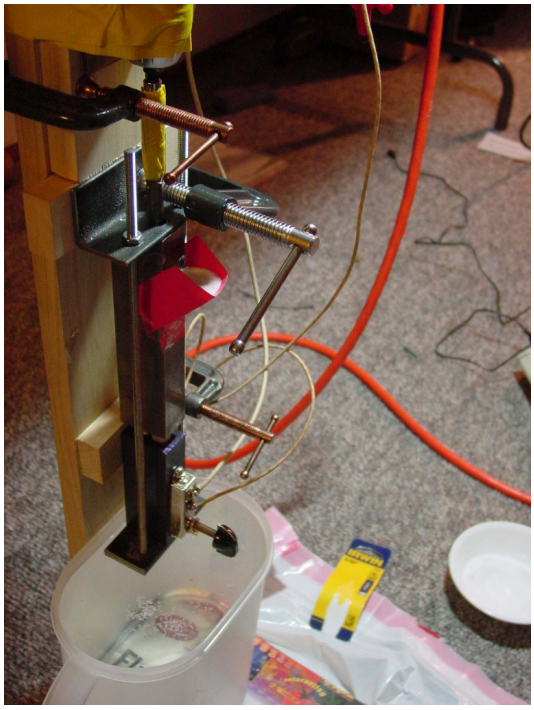
Wednesday, 8th March 2006 by Forrest Higgs

Reviewing the on-line literature about producing monofilament it would appear that for CAPA and polymorph at least we could use a heated quenching bath that would keep the filament very near the melting point of the polymer. If we regulated the temperature of the bath with enough care we might well be able to straighten and stretch the polymer to a proper diameter in one go. That would certainly reduce the complexity of the system.

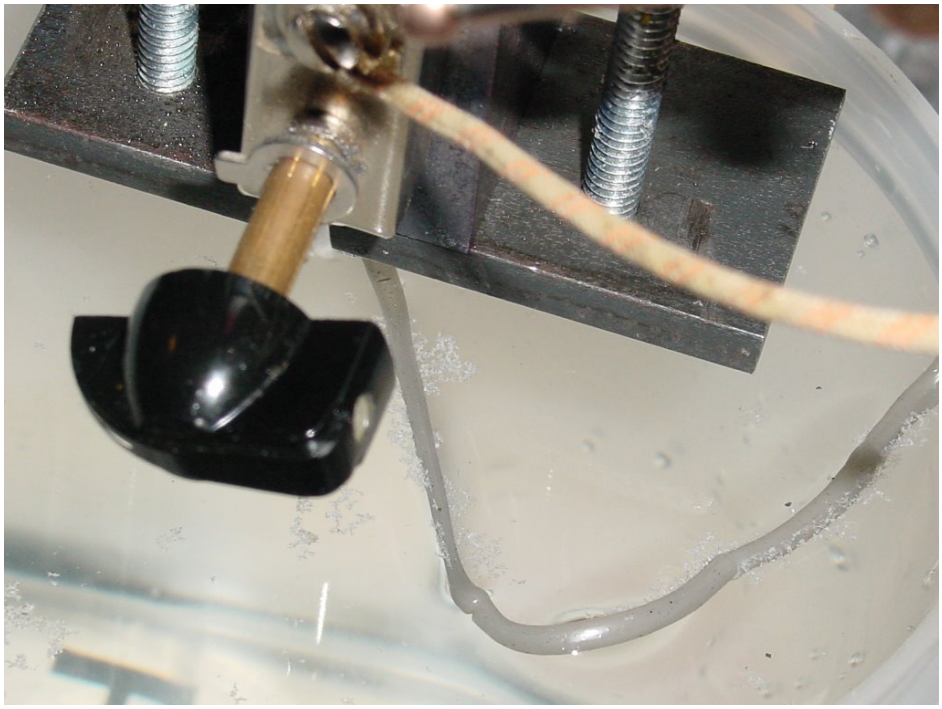
Closing on producing usuable filament

Thursday, 9th March 2006 by Forrest Higgs

I realised late this afternoon that I need not build a framework assuch to hold the quarter-inch extruder vertical. Ten minutes with a fewpieces of scrap lumber and C-clamps gave me a solid platform foroperating the extruder vertically. A water bath was created with afruit juice jug and a piece of construction paper formed a makeshiftpolymer bin.



Thewater bath was situated about 2-3 mm below the extruder tip. I heatedthe extruder barrel to 120 Celsius and started the system.
This was the result.



The filament emerging from the extruder was instantly quenched inthe water bath. The diameter of the resulting filament varied between3.2 and 3.9 mm. The kinks that you see occurred when I had to manuallypoke the polymer feed bin with a thin bamboo stick to keep it filled.
Interestingly, the filament floated. This might indicate that we areentraining air in the filament since the density of the CAPA ismarginally higher than that of water. No obvious bubbles were visible,however.
The practicality of Brett's advice that we need to thinkabout designing a polishing station to straighten and regularise thedimensions of the raw filament now becomes obvious. That should be aperfect RepRapable product.
The extruder produced about 125 mm offilament in approximately 90 seconds. For these settings that is anoutput level of approximately 50 cm^3/hr. I had designed the system toproduce about 15 times what a Mk II could consume per unit of time.We're producing about 18 times which is very close to our design goal.I currently have no way of knowing what the duty cycle of thebimetallic thermostat was, but given the setting chosen even if thecartridge heater had been on full-time it would have had a power demandof no more than 150 watts.
What we have is a prototype thatproves the concept for a small polymer extruder. We need to take theconcept through several generations now to achieve a useful system. Isee several developments as being needed to make this astart-and-forget system.

- an improved polymer feed hopper that makes sure that the auger receives a steady supply of resin.
- roll godets
- heating oven
- takeup reel

Itwould be nice if this system could be self-threading. We obviously needto design a hands-off control strategy now that we have a way of makingfilament. Given the work that has already been done by the team withcontrolling steppers and the Mk II I have no doubt that we have thetalent

already on board to do this.
Here is the system that we are replicating.
It costs US$85,000 used. We're going to do it for under $150 tops as best as I can see.

More post extrusion filament treatment...

Thursday, 9th March 2006 by Forrest Higgs

Solvay lists the melting point of CAPA at 58-60 Celsius. I created a water bath of that temperature on my stove and dumped in the filament produced with yesterday's experiment.

The filament instantly became soft and pliable. Straightening it was no problem at all, though it did break at one of the weak points where the extrusion rate had slowed down when it was being extruded.

One error in yesterday's observations was apparently in reporting that CAPA floats. Today it sank immediately, which is in keeping with it's listed density. I am wondering now whether the "floating" behaviour that it exhibited yesterday had to do with its almost neutral buoyancy coupled with the fact that it was still attached to the extruder tip.

In terms of post extrusion "polishing" of the filament, a heated quenching bath seems to be a good way to go for now.

More on polymer fouling of the auger...

Saturday, 11th March 2006 by Forrest Higgs

I removed the auger from the extruder again prior to setting up for a new extrusion test, this time with a heated quenching bath. Polymer melt was again found to have formed as far back as 50 mm from the end of the auger. This means that the meltdown is confined to the PTFE thermal barrier zone and does not extend into the zone of the steel polymer pump barrel.Just as a side note, the German masonry auger is double flighted and is supplied by Artu USA.

It is made of chrome vanadium steel and appears to be titanium coated if the colour is any guide.

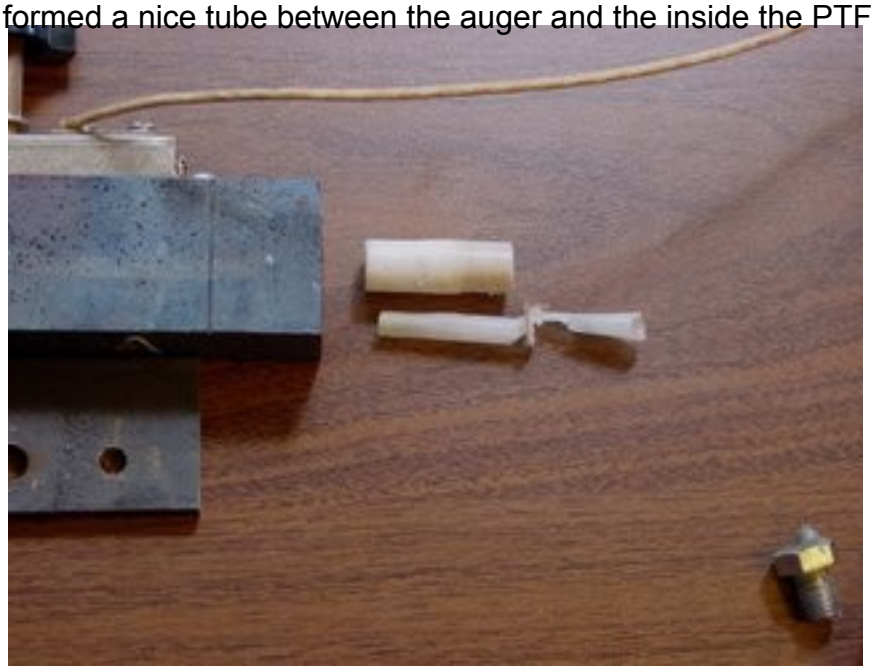
Results of a complete breadown of the quarter-inch extruder...

Saturday, 11th March 2006 by Forrest Higgs

After having removed the auger and cleaned it I left the extruder barrel heater on at 120 Celsius to cook out the rest of the polymer out of the barrel. Afterwards, I disassembled the system completely.

The first thing I noticed was that the half-inch PTFE thermal barrier between the polymer pump barrel and the extruder barrel showed signs of swelling. Whether this swelling is an artifact of the heat that has been applied to the extruder barrel or the pressure that has been developed in the upper reaches of the extruder pump, or both is not known.

What was more interesting was the observation that polymer had not only coated the hot end of the auger but also formed a nice tube between the auger and the inside the PTFE thermal barrier.



Not only had it done that, but it had also migrated another 24 mm up the polymer pump barrel even though molten polymer had not previously been observed on the auger that far back from the extruder barrel.

Given that the system had not been broken completely down for cleaning since it began to be operated it is not clear whether this polymer tube was formed all from the get-go or later on.

Two interesting observations came from looking at this CAPA tube. First, the swelling of the PTFE barrier comes from the inside and is expressed in the tube wall thickness. Second, and even more interestingly, the tube is a clear, clean white all the way down to the end of where it exits from the PTFE barrier into the extruder barrel. Extruded CAPA, on the other hand, has been a light silver-grey. Further, there are no flakes in the tube whereas you will occasionally see a flake of dark matter in the filament.

This argues that the CAPA goes from white to silver-grey *in the extruder barrel*.

I'm drawing two initial conclusions.

• we are developing pressures in the extruder barrel sufficient to cause a bulging of the PTFE

thermal barrier

- we are seeing either a purely thermal transformation of the CAPA colour or a thermally driven interaction between the CAPA and the mild steel walls of the extruder barrel that causes the colour change

Making Version 2

Sunday, 12th March 2006 by Forrest Higgs

Brett has kindly agreed to knock out a few kits for version 2 of the filament extruder and made some very valuable contributions towards its design. He has access to metal scrap the quality of which I can only dream about.

Here are the first two extruder barrels that he's fabricated. They're made of a good quality stainless steel so we should be in better shape with the colouring of the filament if my guesses about how they're getting coloured is right.

First off, version 2 will have a 1" diameter extruder barrel. The reason for this was my serendepitous discovery of a high quality, extremely cheap barrel heater used by the plastics industry in conventional extruders.

Its a sleeve heater designed to fit over cylindrical bits of plastic extruders to keep molten polymer inside molten. This bad boy prices in at about US$6-7 for a 300 watt heater. It can be bought for either 110 v or 230 v AC mains power. The cartridge heater it replaces costs closer to US$10-15. The only drawback to this kind of heater is that it makes the use of a bimetallic thermostat as we were able to do with version 1 impossible for the simple reason that there is no place on an extruder barrel covered by this heater to mount one. That is not a big problem in that the controller for the Mk II extruder that is being used on the RepRap itself already makes use of a thermistor temperature sensor for control. I am sending along a kit of pieces for version 2 to Simon who has kindly agreed to morph the Mk II extruder controller board design into one that can handle version 2.

The polymer pump will be made, according to Brett, from 1.25"x1.25" stainless steel bar stock. Overall, version 2 will weigh about half of what version 1 did. This is especially important in the extruder barrel in that it will have a much smaller thermal intertia than the version 1 did. The sleeve heater should also make for a much more evenly distributed thermal input to the polymer. I am expecting great things from version 2. :-)

First Microchip PIC16F628 programmed...

Sunday, 12th March 2006 by Forrest Higgs

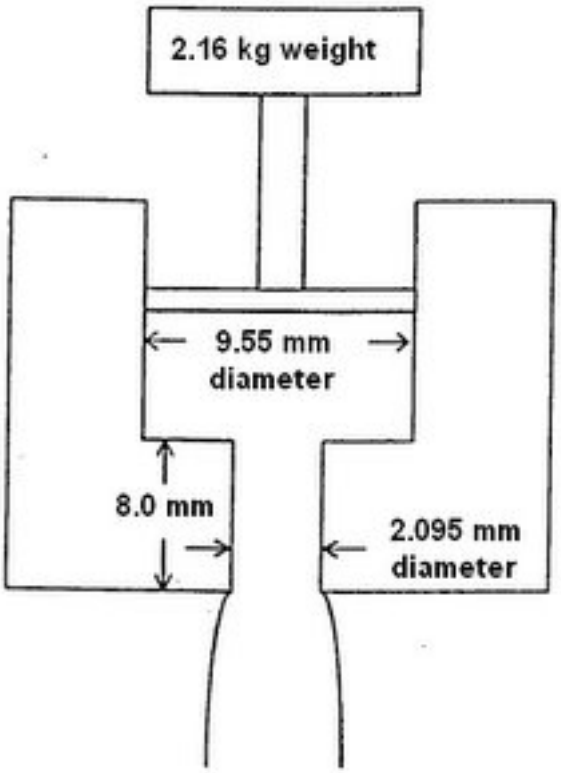
This ought to keep everyone laughing at my expense for a week. Have fun and welcome to it! :-D
I built the JDM PIC chip programming board more than a month ago, but just got around to trying to program a 16F628 chip on it this weekend. This is an important thing to be able to do because the 16F628 is the core of all the controller boards for the Godzilla prototyping machine. Simon, Vik and Adrian have been moving towards a final design for all the boards for the past month and it is looking like I'll be able to start building them in a week or two.
Anyway, the first bit of drama was finding the proper software to drive the board and program the 16F628's. As usual it was right in front of my nose and Simon kindly showed me the link. He then told me the settings to use to get it to work.
I plugged the board into my PC, fired up IC-Prog and had a go... nothing. For the next several hours I played chimpanzee on a keyboard trying things to see if I could get it rolling... nothing. Simon began to suspect that my ancient skills with soldering iron and PC board had maybe slipped a bit. Given that it had been 25 years since I'd last personally built one it sounded reasonable but annoying all the same. I considered just buying a premade, cheap programmer board.
Simon finally ran me through the troubleshooting protocol on the wiki site. The voltage numbers that I got made absolutely no sense whatsoever. I was going blind with fatigue when I finally crawled under my desk and had another look at the back of my PC. 25 pin cannon female connector for Com1, 9 pin cannon male connector for Com2... I'd rigged mine to use the 25 pin connector, just like the old days.
Wait a minute. Where was the printer connector? In the old days we used a Centronics connector, but those had fallen by the wayside some years back and NOW we used a... 25 pin cannon connector... female. It was still confusing because when I checked mode I had 2 serial ports. If that 25 pin cannon female connector was a printer cable where the hell was the other serial coms port? I've been using USB ports for peripherals for some years now so this was all remembering how to read ancient Egyptian for me.
I finally decided that I didn't know where the other coms port was but that the one I had had to be a 9 pin male cannon connector. This morning I went down to Radio Shack and bought a 9 pin female connector. I hooked it up ran IC-prog with the settings Simon talked about yesterday.
It wrote to the 16F628 and verified perfectly the first time out. I read it back into buffer 2 and compared buffer 1 and buffer 2 and it gave me a thumbs up.
Anyway, have a laugh. I deserve it. :-p

A new wrinkle on CAPA filament production...

Thursday, 20th July 2006 by Forrest Higgs

REPOST FROM 20 JULY 2006 POSTING OF MINE IN THE REPRAP BLOG

I went back over the filament production issue again and may have justhit upon a new wrinkle. I looked again at the ASTM standard capillaryviscometer that is used to measure the melt index of polymers.



Previously,I had made something like this out of plumbing parts. While my systemworked after a fashion there was a problem in fabricating a piston thatwould force the polymer out of the die. As well, there was a problem inscaling the device up in a way that would let us process a meaningfulcharge of polymer (~1 kg).

I think that I may have hit on a bitof a breakthrough in that regard by revisiting the diagram of thecapillary viscometer and then getting the melt index of CAPA 6800.

http://www.solvaycaprolactones.com/docroot/capro/static_files/attachments/capa_6800.pdf

At160 degrees Celsius CAPA 6800 caprolactone's melt index is 3 grammes/10minutes. What that means is that this device, heated to 160 degreeswill extrude 18 grammes per hour or 0.4 kg/day of filament which isroughly 2.4-2.5 mm in diameter.

When you think about it that'sawfully close to the size filament that we need and the dailyproduction is within shouting distance of what a fully functionalGodzilla scale RepRap could use.

Making the die a bit wider sothat we could make 3 mm filament would actually make the device workfaster, so that's not a problem. As well you could reduce the depth ofthe die channel which would also speed things up.

So what are the problems?
Thebig problem is the size of the melt chamber. A chamber diameter of 9.55mm (3/8ths inch) is too small. The melt chamber is sized to hold nomore than 15 cm^3 of polymer. If we keep the diameter as is andincrease the height of the melt chamber we can keep the force requiredthe same. With such a small diameter, though, the practical limits ofthis approach are reached very quickly. If we increase the diameter aswe must, however, the force applied to the piston in the melt chamberincreases as the square of the diameter. The practical limits of that,for a device that we wouldn't mind having around are also reached veryquickly.
Lets take a look at that force for a moment. We areapplying 2160 grammes (I know the units are ideosyncratic, but bearwith me for a moment) of force to 71.6 mm^2 of piston area. That meansthat we have about 30 grammes/mm^2. Since 1 atmosphere of pressure isabout 10 grammes/mm^2 we are applying about 3 atm of pressure to thetop of the melt to get that highly desireable flow rate.
Apressure of 3 atmospheres in US units is about 44 psi. Being anAmerican who has spent much of his life in SI only countries I carry a ***lot*** ofconversion figures in my head. Last weekend I took Zach down to theKragen auto parts shop and snooped around while he bought severalthings that he needed to repair his VW Kombi. I was quite taken with aUS$199 generator set rated at 1200 watts continuous. I also looked at asimilarly cheap shop compressor and noted that it operated at twopressure ranges, viz, 45 psi and 90 psi, or 3 and 6 atmospheres.
Click!
We're using plumbing parts, right? A pressure of 3-6 atmospheres for plumbing parts should be a walk in the park.
Let's look again at my little filament experiment from a few weeks back.

What'sto stop us from screwing another brass cap on top of this assembly andtapping a very standard compressor connector to it and then hooking itup to a standard shop compressor to provide the pressure that I wasapplying manually with great difficulty.
The short answer is... nothing.
Onceyou fire up the extruder barrel enough to plug the bottom of the barrelyou can apply pressure. A 200 mm length of 100 mm pipe would easilyhold 1 kg of CAPA 6800 granules.
About the only objection isthat a standard shop compressor is massive overkill. We require verylittle actual air volume, only the pressure.

With that in mind I went shopping and quickly found this...

Thisis a tiny compressor that you plug into your car's cigarette lighter.It costs US$8.24 and very conveniently runs on 12 volt power.
http://www.nationalonlinesales.com/index.asp?PageAction=VIEWPROD&ProdID=193
It should provide more than enough air volume to do the job.
Heatingthe barrel is probably more than we can reasonably expect from a 12 vsystem, though.
Happily, our friends at IMS have a solution.

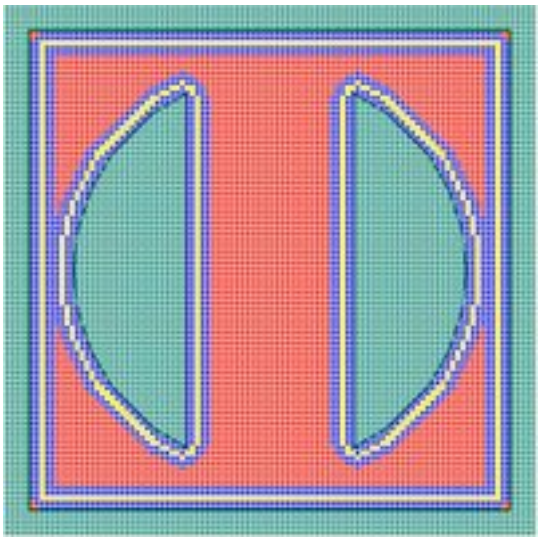
Theirmica insulated band barrel heaters are widely used in the plasticsextrusion industry. IMS will sell a 100 mm diameter x 100 high unitrated at 750 watts, far more than you need, to you for about US$30.

This should be fun. :-D

Grids rock

Tuesday, 8th August 2006 by Forrest Higgs

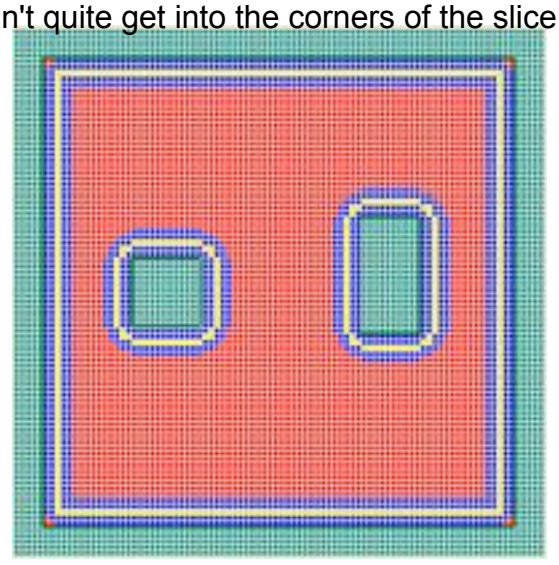
Finally got my head around how to solve the border problem. Grids rock!



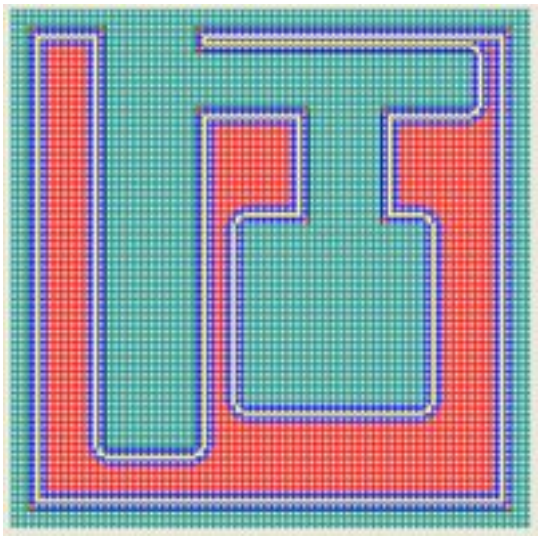
Not only do they track the perimeters properly, but they also show you the path for the centre of the MK II extruder head, won't let you overlap and shows you the corners and blind pockets that the Mk II isn't going to be able to get into.

In this pic the background is cyan, the perimeter is blue with the extruder path done in yellow. The red bits comprise the rest of the slice of the object. I haven't done an infill routine yet.

You can see how the Mk II can't quite get into the corners of the slice.

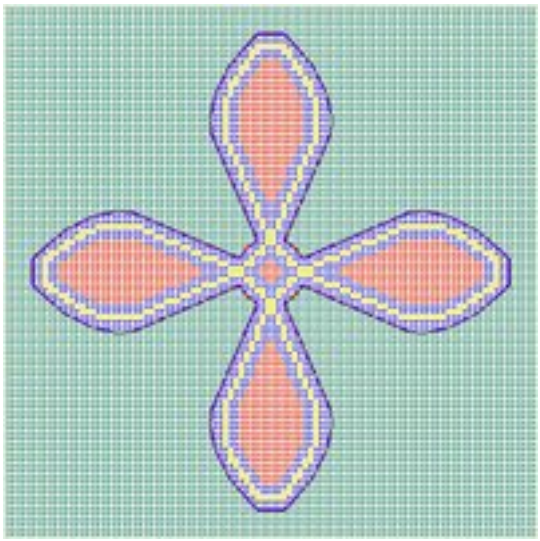


Here is a slice of a rectangular box with two holes in it. Looks pretty good for a start. :-D



Here is a really, really non-convex slice especially for Adrian. :-D

After looking at Adrian's non-convex polygon, which took a long time to do, I felt sufficiently confident in the approach to migrate some more of the coding from the old slice and dice programme that I wrote a few days ago into the grid. This improved the execution speed of the routine by about 1.5-2 magnitudes.



There are a few other tricks that I can use if need be to kick that speed up another 1.5-2 magnitudes. I'm not going to do it unless execution speed really gets to be an issue again. Because PC's always get lots faster if you wait about 18 months I tend to value code reliability and implicity a lot more than I do efficiency.

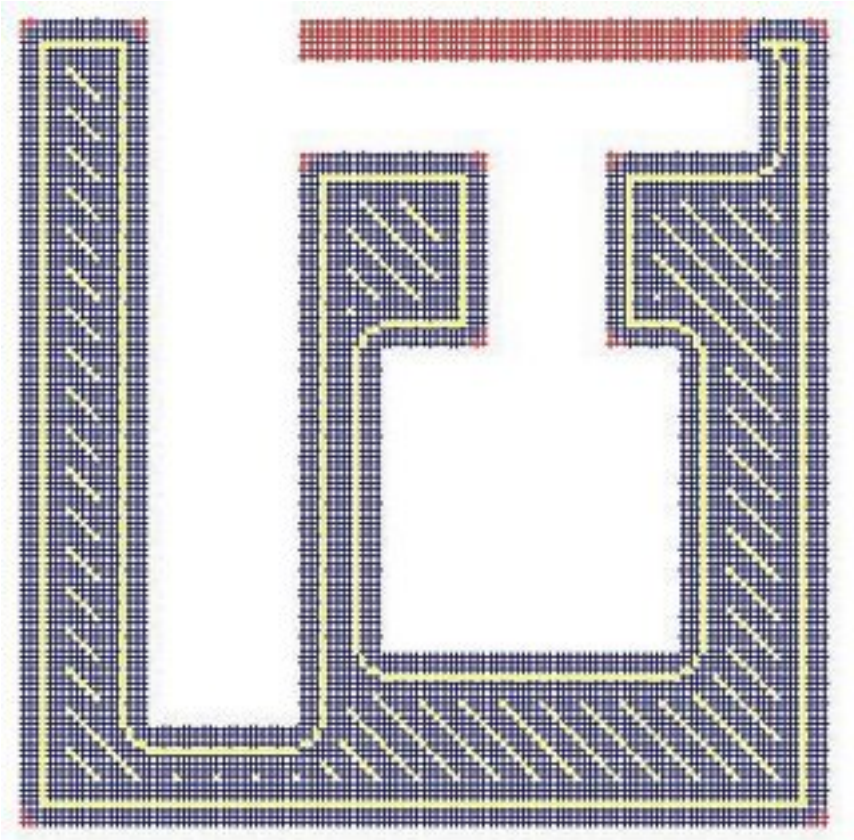
Finally, here is a perimeter trace for a 4-toothed involute profile gear. For this kind of shape I am beginning to wonder if our one-size-fits-all notion of using diagonal, 90 degree offset infills is really a good idea for all the sorts of things we'd like to make. If you look at the inner spot of infil required for this gear it would seem that a quick doubling of the perimeter depth would be more appropriate, ditto with the infills in the teeth. This requires some more thought, I suspect.

I am particularly excited about the prospect for leveraging this method with some heuristics that give the user some useful graphics feedback about whether the RepRap machine can actually make the part the user is designing. I suspect this is going to be a very big issue for the sorts of RepRap users who are using their machines to create new devices rather than simply making salad bowls and napkin rings.

Diagonal Infilling

Saturday, 12th August 2006 by Forrest Higgs

After Adrian's warning about the dangers of overlaying the same pattern slice after slice, I went back and developed an alternating diagonal infill routine. One of the issues that I had to confront was what to do when the perimeter routine couldn't deal with a feature of a slice that was narrower than a perimeter extrusion track. That required a little routine that determined which parts of the slice were outside of the perimeter after the perimeter had been developed.



You can see the results here. That red bit at the top is a feature that is narrower than a single trace of the Mk II. You can also see some red bits on the corners that represent parts of the slice that are too fine for the Mk II extrusion to reach.

Yesterday, Adrian suggested that I should be a bit more forthcoming with my developments both in documenting them and making them available while they are in the development phase. His point is very well taken in that I consider what I do almost perpetually "in development".

On the other hand, I am well aware of the fact that I am a very untidy person and am hesitant to let that untidiness flood into places like RepRap's subversion system. I'm going to try something that might be the best of both worlds. To that end I've opened up a directory on my own server...

http://www.sanma12.com/reprap

...as a repository for my extensive, messy, non-standard "developments", such as they are. For those of you working in a Wintel, Visual Studio.NET development environment my current code can be acquired at...

http://www.sanma12.com/reprap/software/VB.NET PC-Side Control System/Grid 09 upload 120806.zip

For the more orthodox RepRappers who are using Linux and Java, still feel free to grab my code and give it a look if you suspect that there may be something useful to what you want to do. VB.NET isn't very hard to read if you know where to look in the half-dozen files that a Visual Studio project generates.

In this case all of the actual BASIC code can be found in Form1.vb. Just open that file up in some sort of text editor and search for #End Region. The actual code follows that statement which is at the end of a bunch of statements that tell Visual Studio how to set up the form with buttons, labels and the like. The code proceeds from #End Region to the end of the file in plain ASCII text.

Going a bit further, the routine has three procedures that are taken in the following order...

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

- reads in the STL file
- generates the slice

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

- sets up the grid and determines what part of the grid lies within the grid
- develops the perimeter extrusion track
- determines what part of the slice lies within the perimeter

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

- does the infill

Those three buttons call several other little utility routines which can also be easily found in the Form1.vb file using your standard text search facility.

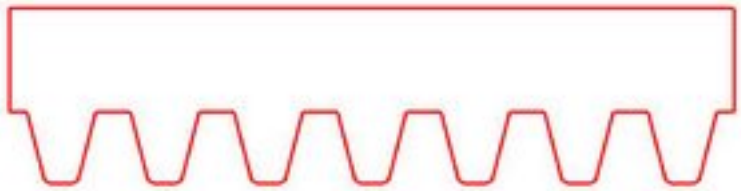
When I actually get something that is more or less complete, I'm talking about subsystems, you can be sure that it will be documented and placed in the Wiki and/or the subversion files.

Rack and Pinion

Sunday, 13th August 2006 by Forrest Higgs

I converted Kiplinger's maths to code and tested them. His pinion gear routine seems to be nothing more than an involute profile gear like we already have. Mind, the Swedish approach seems more robust than Kiplinger's. His routine for generating rack gear strips is trivial and appears to be very robust. I've got that going in a VB.NET test programme and will sort out the problems, such as they are, in it before shifting it over to a Java script that will work in Art of Illusion. Most of the "problems" are making sure that the routine generates a proper loop of line segments and resolving a terms difference between the input data for the Swedish involute profile gear programme and the rack programme. I don't think that is going to be any big deal.

Here is a sample rack I generated with the VB.NET code prototype routine.



It is a 14.5 degree pitch angle rack.

Just for variety here is a 20 degree angle rack. Pitch angles of 14.5 and 20 degrees are pretty much the industry standard for this kind of gear technology.

Rethinking everything

Monday, 14th August 2006 by Forrest Higgs

Posts by Sebastien and Simon got me to thinking this morning about RepRap.

Sebastian's post led me to recalculate a number I did way back last November and Simon's post made me mad, something I've always found to be a good spur to creative thought. Thanks go to you both. :-)

Sebastien's post was to Vik asking him what the extrusion rate of Da Witch was. Vik guessed at 1 cc/min. I responded doing an envelope calculation which indicated that Vik's calculation was just short of a magnitude off. From the message...

Figuring a 0.8 mm extrusion thread and running at the 4 mm/sec that Mk II was designed for you're getting about 2 mm^2/sec. or 7.24 cc/hr which gives you a modest, but not unreasonable 5.2 kg/month on a constant duty cycle.

Way back last November I made a calculation of extrusion rates when Adrian first published the Mk II extruder. The number I came up with then was 2.83 cc/hr or roughly 2 kg of polymer/month if operated full-time.

It was always obvious to me that 2 kg/month of polymer extrusion for 24x7 operation was just not practical. This morning I realised, however, that my calculation did not allow for the non-Newtonian swelling of the extruded polymer thread. I was figuring that the thread that came out of the Mk II was the same diameter as the orifice. WRONG! Vik has seen that a 0.5 mm orifice yields a 0.8 mm polymer thread.

Now extruding 5.2 kg/month of polymer is a whole different ball game.

That got me to thinking about Godzilla, which I have targeted 20-40 mm/sec extrusion rates. I've learned a lot designing and building Godzilla and at this point I am sure that I can get it to work, probably at about 20 mm/sec. That would extrude something like 26 kg of polymer a month.

There are some practical problems with Godzilla, though.

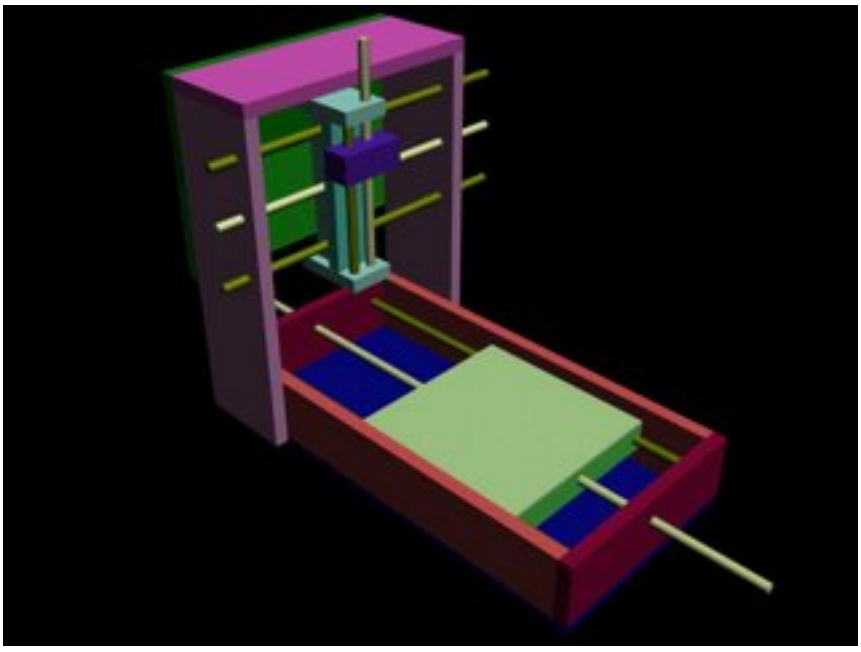
It's big. It's got a 700x700x350 mm work volume. It's footprint is about 1.5 square meters. It's also loud. Frankenmotors don't scream, but they are full-throated. Two of them running at once will make about as much noise as an electric blender. If you were married you couldn't keep Godzilla in the house with your wife unless she was as committed to 3D prototyping as you are. That's probably an unreasonable filter for guys wanting to marry to put on potential mates. It just wouldn't work. I'm crazy, live by myself, divorced and have a bit of hearing loss from my misspent youth, so Godzilla doesn't bother me. Most people would be bothered, though. As a practical matter Godzilla would have to live in somebody's workshop and it wouldn't be a machine that you'd want to leave running overnight or over the weekend. As a practical matter you're not going to be able to run it for more than about 8 hours/day.

That means that you're only going be extruding about 8.7 kg of CAPA a month with it, that's only 2/3rd's more than Vik's machine. Now it can make lots bigger things than Da Witch, but realistically, how often are you likely to be making huge things with your RepRap?

That's a bitter pill for me, but there it is all the same.

Now let's get to Simon's message that made me mad, but in a creative way. He indicated that servos aren't on for now, that it has to be steppers. Well, I've probably got more hours working

with turning brushed DC motors into servos than anybody on the team and I couldn't disagree more. Now Frankenmotors need some work and a lot of thinking to be used properly. The little yellow plastic G4 gearmotor servo is a very different story. Here's why.
You can get steppers surplus for about US$5-10/unit. The moment that RepRap takes off, however, surplus steppers are going to evaporate and people will be paying list price for them, which is about US$50-75/unit. That means that if you are shooting for a US$400 RepRap the steppers are going to cost you 40-60% of that. Parts for stepper boards are also going to cost you another US$15 or so in parts when all is said and done. That means that you're talking about 50-70% of your RepRap budget going for just motors and control boards before you even try to build the positioning stages and extruder. That's just not on.
Now I can buy a yellow plastic G4 gear motor retail for between US$5-6/unit.
http://www.pololu.com/products/solbot/0181/
Add a shaft encoder to that and you've got US$10. That's 20% of the real cost of a stepper. As well, the much lower power draw means that the chips you have to buy to drive the motor are cheaper. Finally, most of the mass and cost of the G4 and it's siblings is in the plastic gear box. The motor itself can be had for maybe US$1.50/unit retail. The rest you can reprap if you want.
What all this means is that the cost of your motors and control boards for gearmotor servos for a Vik-sized RepRap is going to be maybe US$60, and a bit less if you make your own gear boxes.
Now let's do a thought experiment. Let's take Reiyuki's pretty conventional (in CNC terms) gantry design for a wooden RepStrap.



He thoughtfully provides us with a materials quantity survey.

24x48" of 1/8"
48" of 1x2
16" of 1x3
74" of 1x4
78" of 1x6

Doing the numbers you get roughly 17250 cm^3 of materials volume. In wood that gives you about 10 kg of material. In CAPA it would be more like 17.5 kg.
If you redesigned Reiyuki's system for CAPA from the get go you can bring that number down to about 3.5 kg. Pushing it harder you could probably shave another kg off of that, but why bother?
What it all means is that a Reiyuki RepStrap could make a full polymer copy of itself in 20 days.
What it also means is that you could make one for under US$200, everything included... if you went the G4 gearmotor servo route. Double that if you use steppers.
A Reiyuki RepStrap has a 300x600 mm footprint. That's not much worse than most ink-jet printers. Put hoods over those G4's and you will have something that is probably considerably quieter than your ink-jet. Run the whole shebang on 12v power and you're not going to worry about letting it run nights and weekends. Reiyuki's design, when you look at it, isn't all that different than Vik's, which we know works. It uses threaded rods and we don't stress them to the extent Godzilla does. Same general configuration. It scales, too.
If we take my own proclivity for dynamic extruder heads we can probably either reduce the footprint or increase the capacity for a set footprint. I tend to think that G4's running positioning platforms using RepRapped rack and pinions (yes, I've just about got the script running) could make a lighter system still with fewer hard metal parts to buy.
You're can naturally make spare parts for a design like this when you aren't actually using the system for something else. There will be a natural tendency to make spares and keep maybe two systems going for reliability. Two systems not all that much bigger than Vik's have a hell of a lot of extrusion capacity, viz, about 11 kg/month.
The present paradigm has your PC running the RepRap. Put a bigger PIC18F chip in the controller and your PC could handle several RepRaps.
Having spares around is going to give you a natural tendency to help other people get theirs going, too.

Tommelise takes shape...

Wednesday, 27th September 2006 by Forrest Higgs

I'm beginning to get clear of the billable hours that I'd agreed to sothat I have a little bit of time to think about Tommelise.

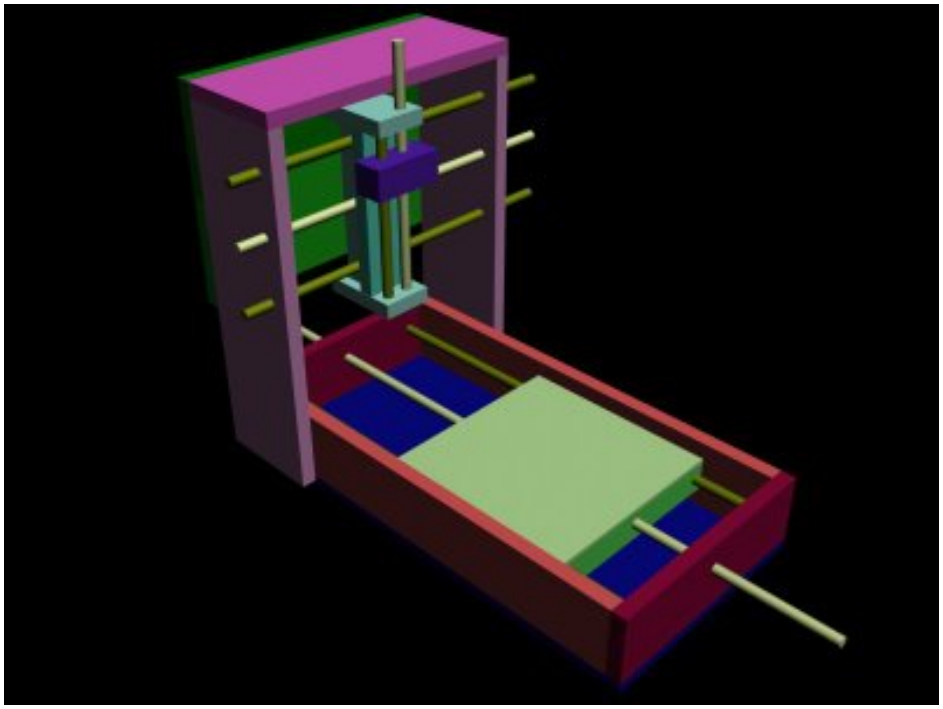
Thefirst obstacle that I hit was the discovery that the GM8 gearmotor hasmounting holes too small for any readily available bolt. Neither M3'sor 3-48's are small enough to fit them so it looks like I have to getcreative with ideas about how I mount the motor and encoder chip. Atleast I have a motor that otherwise makes that possible.

The first thing I had to do was bite the bullet and take down Godzilla.

Irealised that having to do that was at least partially responsible forthe fact that I hadn't done more work on Tommelise earlier.

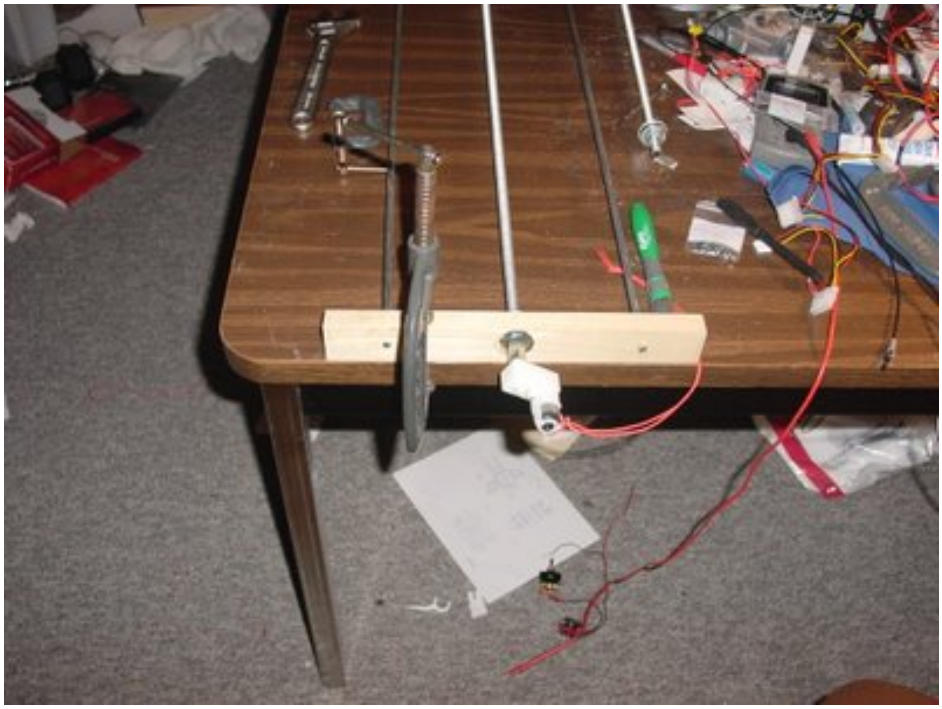
Oddly,once I'd taken it down I completely lost any sentiment I had for it andstarted looking at it as a source of poplar and parts for Tommelise.Doing that should save me a bunch of trips to Orchard Supply.

After a lot of dithering I decided to go with the Reiyuki/Luberth/CNC approach.



I laid out a basic horizontal z-axis

And then checked to see if the 12v GM8 would turn it.

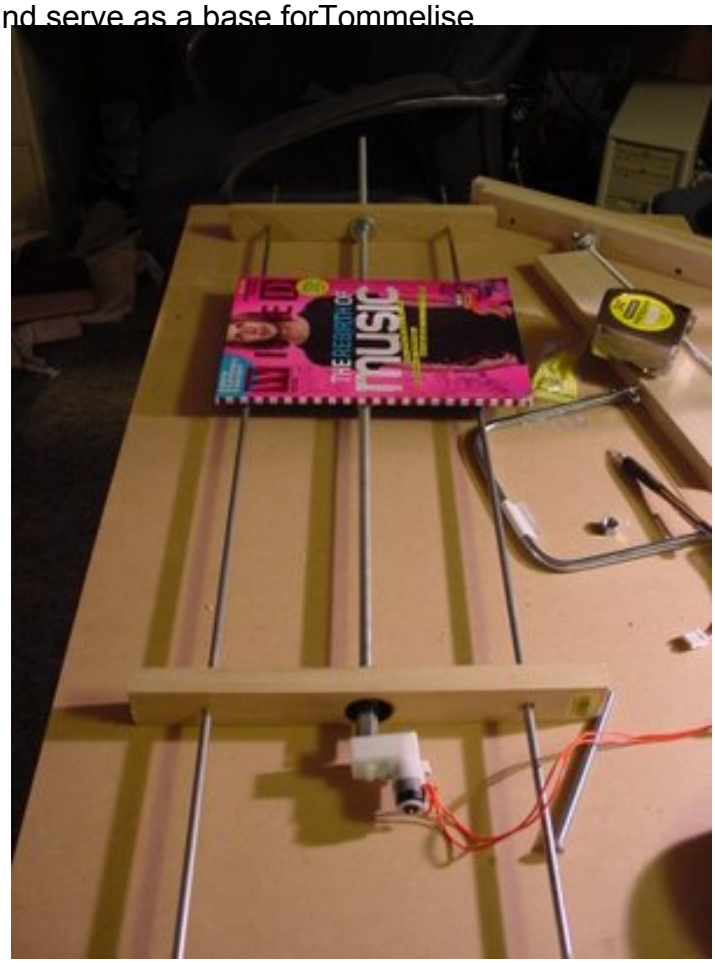


It did with no effort whatsoever.
Atthis point I'm using c-clamps to hold things together. I haven't cutthe 3/8ths inch threaded rod down to 2 feet yet and won't till I decidehow I'm going to integrate the x/z-axes. Mind, I know what it supposedto look like from Lubreth and Reiyuki's models. Getting the clearancessorted out and the rest integrated is what I'm not completely clear onyet, though.

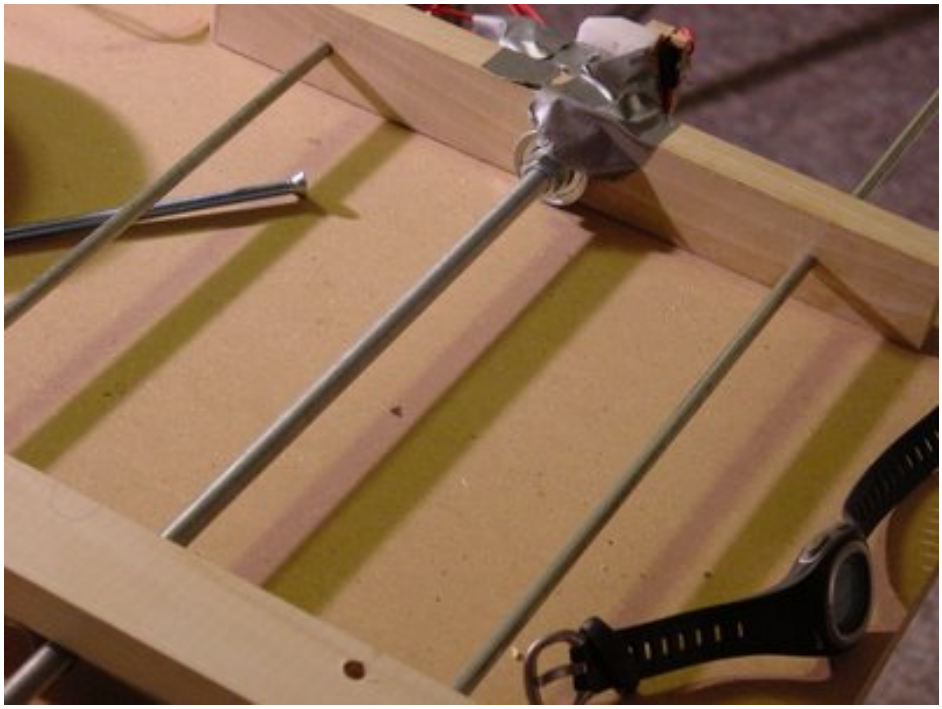
A bit further...

Thursday, 28th September 2006 by Forrest Higgs

I got a 2x4 foot piece of 3/4-in fibreboard from Orchard today andsawed it down to 2x3 to fit on my work table and serve as a base forTommelise.



I also shortened down the span of the z-axis to 650 mm as you see here.
Ibought a coping saw, which you can see in the photo, so that I can cutsome small pieces to make mounts for the shaft encoders for the GM8.
Justas a test I slapped the current edition of Wired magazine (~425grammes) onto the coupling nut to see if the GM8 would have any troublemoving it around. You can see a short video clip of the GM8 not having any trouble.
Aboutthe only obstacle to progress is straightening out the threaded rod tolimit lateral movement that you can see happening in the clip.
Idecide to take things a step further before turning in for the nightand strapped the 6v GM3 motor onto the rig since it was alreadyequipped with a shaft encoder and controller board.



Duct tape is a wonderful prototyping tool. :-)
The pseudostepper control regime for the GM3 gearmotor worked fine. Here is a small video clip of the running at 0.484 seconds/step with a 12 degree step size. Withthis thread pitch that works out to just over 0.03 mm/step.
I'm currently running it at 4 seconds/step and 22.5 degree step size. That works out to 0.06 mm/step.

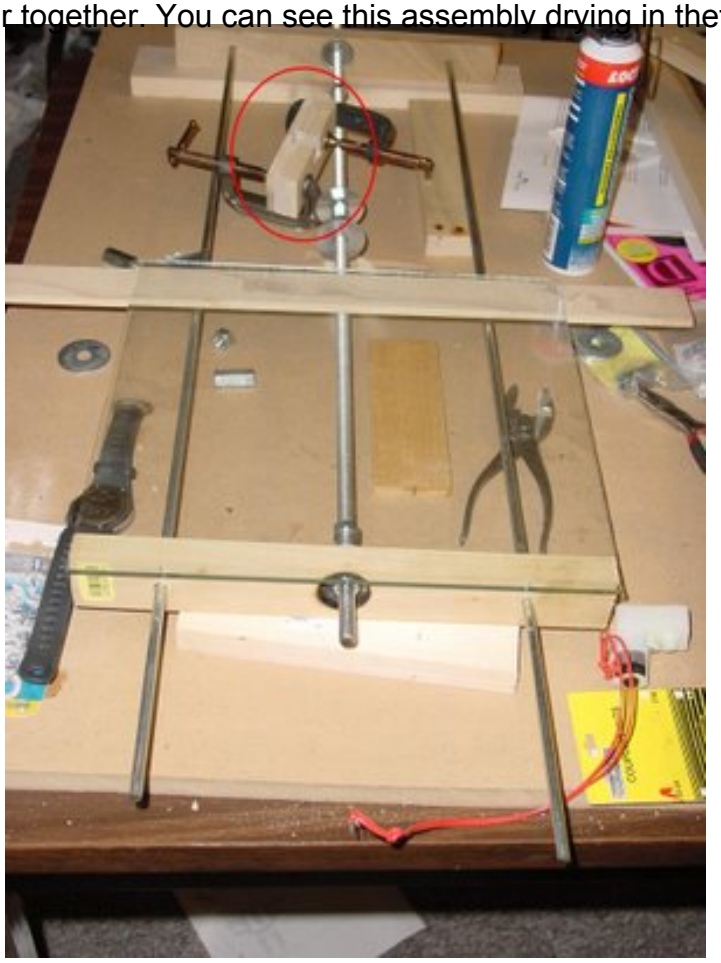
Tommelise sliding thrust collar under construction...

Saturday, 30th September 2006 by Forrest Higgs

I reached the conclusion that if early generation repraps are going tobe made with ordinary threaded studding we'd better get used to thefact that when you buy it from the hardware store it often isn'ttremendously straight. I got the piece that I was using for Tommelise'sz-axis (horizontal) straightened to a point where over 640 mm it hadabout +/-2 mm play. There are two ways that I could have dealt withthat and kept the accuracy on spec. The first was that I could have hadthicker guide rods (I used 1/4 inch) and used full sleeve bushings.That would have let the drive motor straighten the threaded drive rod.That, of course, would have eaten up torque. The other way was to letthe thrust collar move freely in the x-y plane. I decided to give thatone a go for this design exercise.

I designed a crude thrustcollar which would let the coupling nut move in the x-y plane. Thisconsisted of a poplar sleeve that let the coupling nut move freely inthe y direction and guides, also done in poplar, that let that sleevemove in the x direction.

The thrust collar had to more or lessmatch the width of the coupling nut (1 in) which entailed me gluing twostrips of poplar together. You can see this assembly drying in thefirst photo.



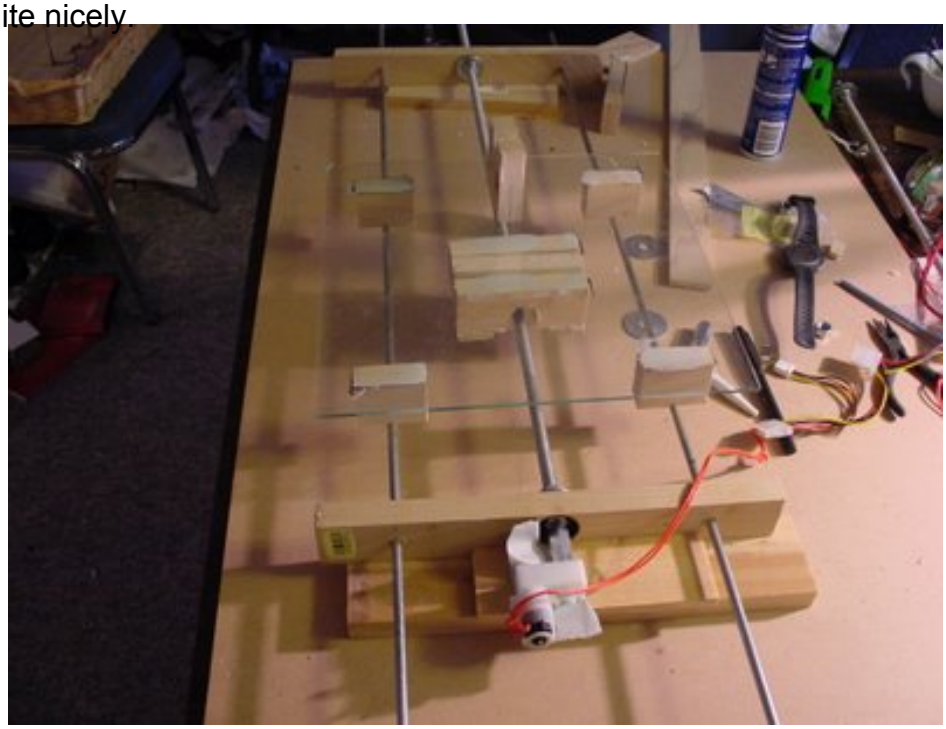
Atthat point I carved seatings for the guide rods out of poplar and gluedthem to the 320x320 mm double strength glass work surface.

Inthe second photo you can see them glued in place along with the guidesfor the thust sleeve. Fender washers keep the coupling nut fromslipping along the z-axis.
I got a bit overexcited and tried torun it this evening instead of letting the glue dry overnight. Thethrust collar appears to do what I'd intended, that is, to keep lateralmovements by the theaded drive rod from reaching the guide rods andrequiring dissapation as friction heat.
I was able to run it forabout a minute before one of the glued joints came loose. I regluedeverything and resolved that video clips of the z-axis platform willwait until tomorrow morning. :-s

Sliding thrust collar works!

Sunday, 1st October 2006 by Forrest Higgs

It looks nasty, but it does the job. It absorbs the periodic play inthe x-y plane without having to turn it into friction quite nicely.



I'vegot to shim the slide in the x-direction a bit more but for now it willallow me to make a CAPA replacement pretty easily. I also know how tomake it better for the x-axis, though the studding rod I'm using thereis very straight indeed.

I've taken a few video clips of it in operation.

In the first first clip you can see the play in the threaded drive rod. In the second clip you can see that the z-axis platform can handle some load and doesn't shake it around. The third clip gives you a feeling for how smooth the platform is moving. In this last clipyou can see that the x-axis slide could use a little extra shim tocover for the fender washers. I'll design that into the CAPAreplacement. It should be easier that way.

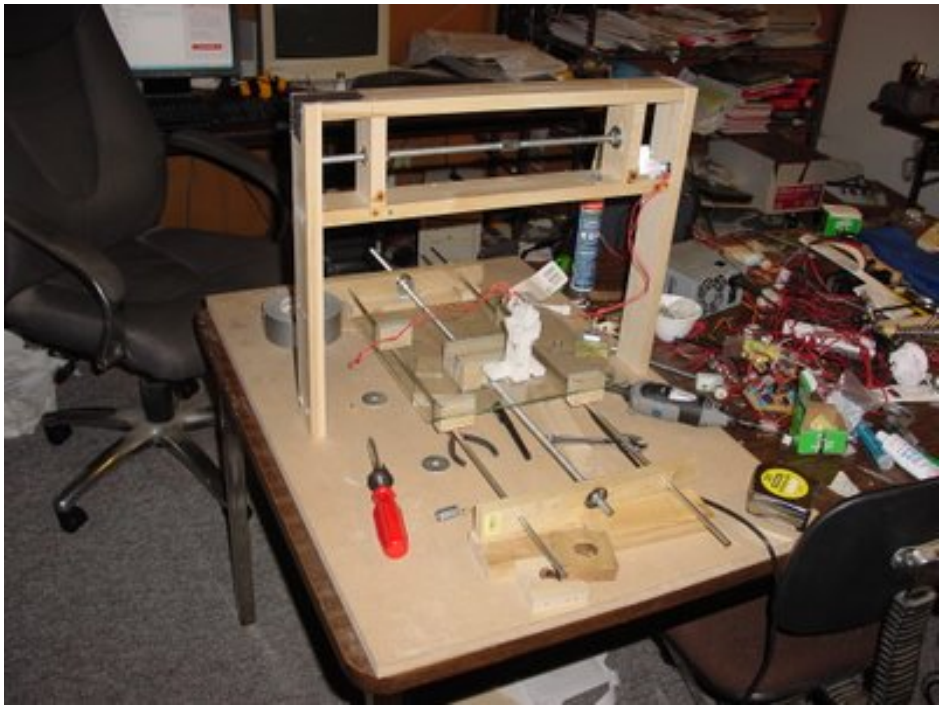
Most of the movement you see in the clips is me not holding the camera perfectly still. I do need to invest in a tripod. :-p

Gotto do some billable work for a few days, after which I'll do thesliding motor mount and encoder chip block. Then I can bolt the z-axisdown and go on to the x-axis. I'll be using the same techniques for thex and y axes so that should go a lot faster.

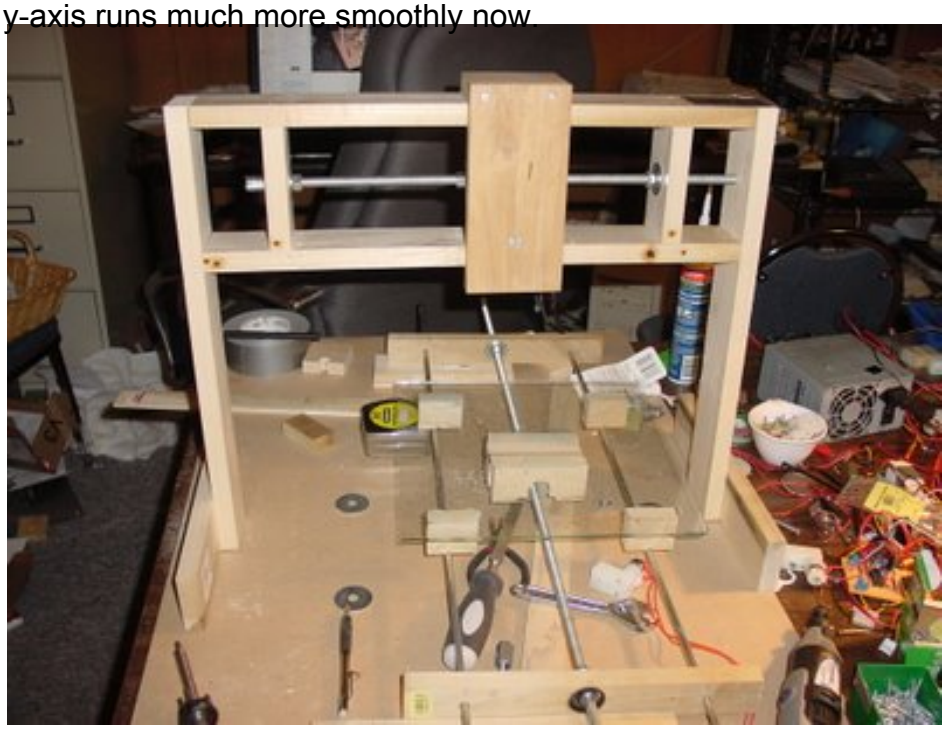
Tommelise x-axis underway...

Friday, 6th October 2006 by Forrest Higgs

I was able to salvage one of the z-axis towers from Godzilla and recycle it as an x-axis for Tommelise.



I've included a partially assembled Mk II and the GM8 for scale.
Idisassembled Godzilla's x-axis and salvaged the lumber and 3/8-24threaded rod. It turned out that the salvaged threaded rod was muchstraighter than the one I had in Tommelise's y-axis so I did a swap.The y-axis runs much more smoothly now.

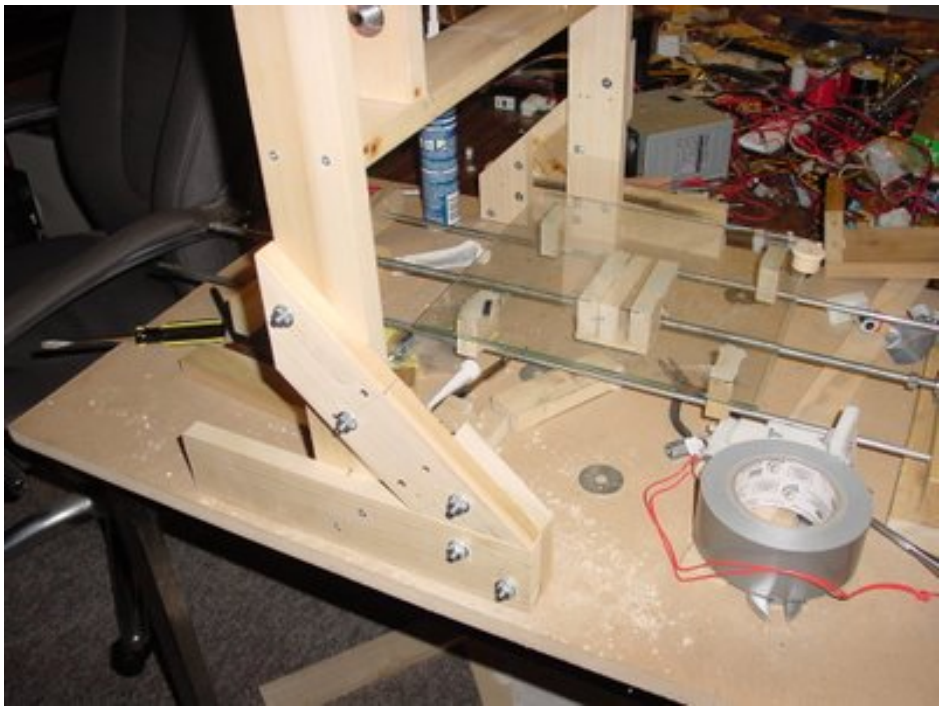


Lateron I fabricated a mockup of the z-axis mounting plane so that I couldsee if using the frame of the x-axis for a guide rod was going to beviable.
It is. The mounting plane will have to be deeper than shown here.

Locking down Tommelise...

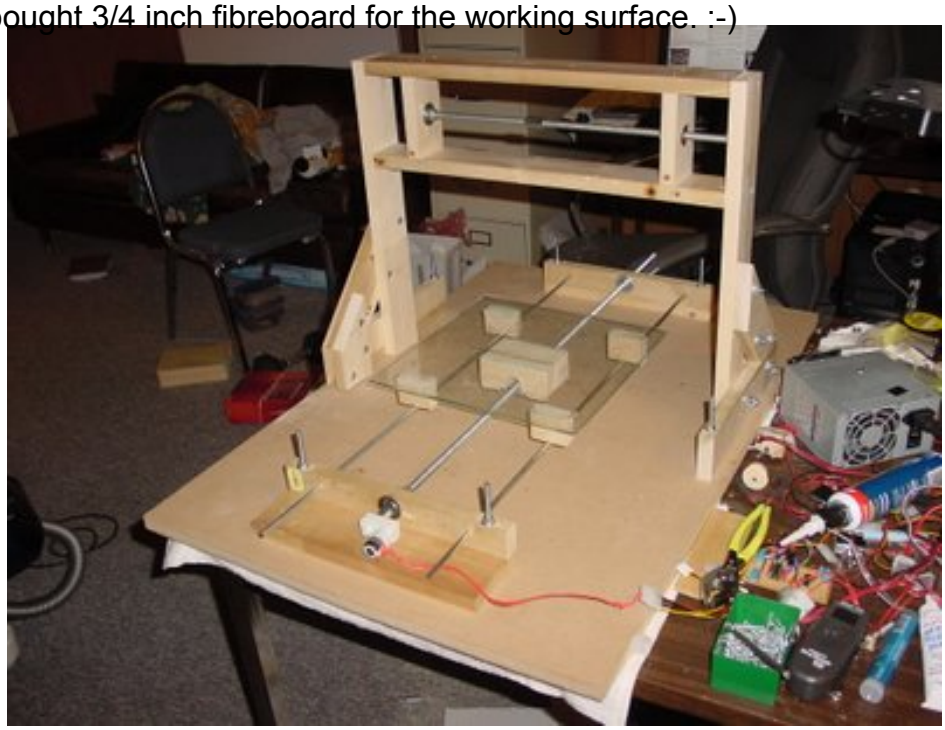
Sunday, 15th October 2006 by Forrest Higgs

I finally managed to get out to Carmel Valley and brought back a goodspirit level and carpenter's square. I had bought what I thought were abunch of 1/4-20 bolts and I guess I did. When I set out to align theelevated x-axis, however, I discovered that I had only barely enough todo the job.



First, I had to get the gantry mounting the x-axis properly vertical. I cross-braced and bolted the vertical members.

Afterthat I discovered that the x-axis guideway was somewhat warped from oneside of the working surface to the other. That will entail bolting thex-axis gantry to the fibreboard working surface. I'm glad now that Ibought 3/4 inch fibreboard for the working surface. :-)



AfterI'd got the x-axis gantry done it was a short jog to getting both thex-axis gantry and the y-axis stage locked down on the base plate.
There is a little residual warping in the gantry, but nothing I can't shim out.
Ididn't have the mounting blocks for the y-axis stage square before so Iam going to have to realign the guide blocks for the rails. That shouldbe no big deal.

12v motors arrive for Tommelise...

Friday, 20th October 2006 by Forrest Higgs

Ordering direct from Solarbotics seems to be the way to go.

I got a GM9 for the x-axis, which is a bit space constrained, and replacement motors for all the 5v GM8's plus some spares.

Success!

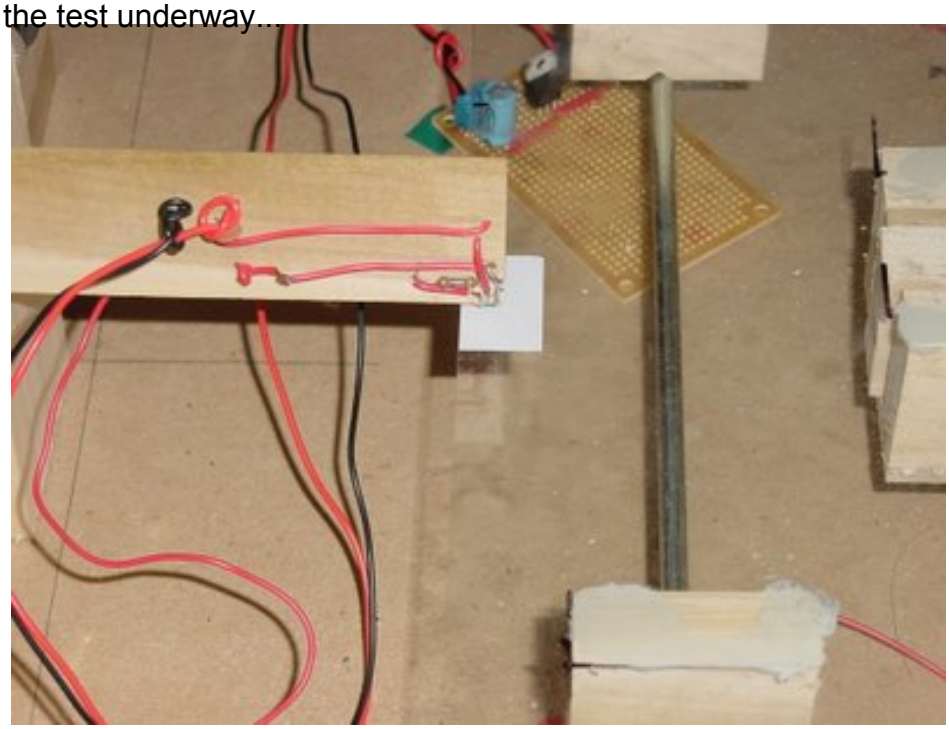
Friday, 17th November 2006 by Forrest Higgs

I've had a lot of troubles getting the Hamamatsu P5587 limits detectorchip to operate reliably as a PortB input. In theory the circuit it iswired into is supposed to put out a 0v and 5v signal when it crosses anIR opaque/transparent boundary. A lot of the time it does that ratherwell. After it warms up, however, you'll often start getting a lowvoltage of 1.74v and a high at 4.95v. Those don't trigger an interrupton PortB.

After a week of foolingaround trying to make it work the obvious way I finally decided toshift the input from the chip over to one of the the analog pins andmeasure the voltage during the 1 ms clock cycle interrupt and see ifthat works better. Sure enough it worked like a champ!

It tookabout 30 minutes to prove out the A/D code snipped and adapted out ofexample off of the Oshonsoft site, replace the PortB code with it andtest the thing on Tommelise. It's been running on a 50 mm work areaboundary to give it a lot of cycles per minute for about 45 minutes notwithout so much as a single hiccough.

Here is a pic of the test underway...



...and a short video of the y-axis positioning stage making the turn.

WhatI do with the analog input is simple. If I get over 2.5 volts I figurethat it's over an opaque surface. If it is less than that, I figure itis over a transparent surface. Super easy.

Now I can go aheadand wire in the x-axis to the 16F877A board with reasonable confidencethat extending the code that I have running the y-axis is going to dothe job for the x-axis.

I was counting port pins and I'm prettysure that I can get all four gearmotors (3 axes plus the extruder)running on this board slaved to the same 1 ms clock cycle. I'm going tobuild up a little test board around a 16F628A this weekend along with aTip 120 to get used to running the extruder barrel heater. There's noobstacle to putting that onto the 16F877A board, too, except that Iwant to test out the code in isolation first.

With a little luckI should be able to do everything I need to run Tommelise off of one16F877A using a 20 MHz clock crystal. :-)

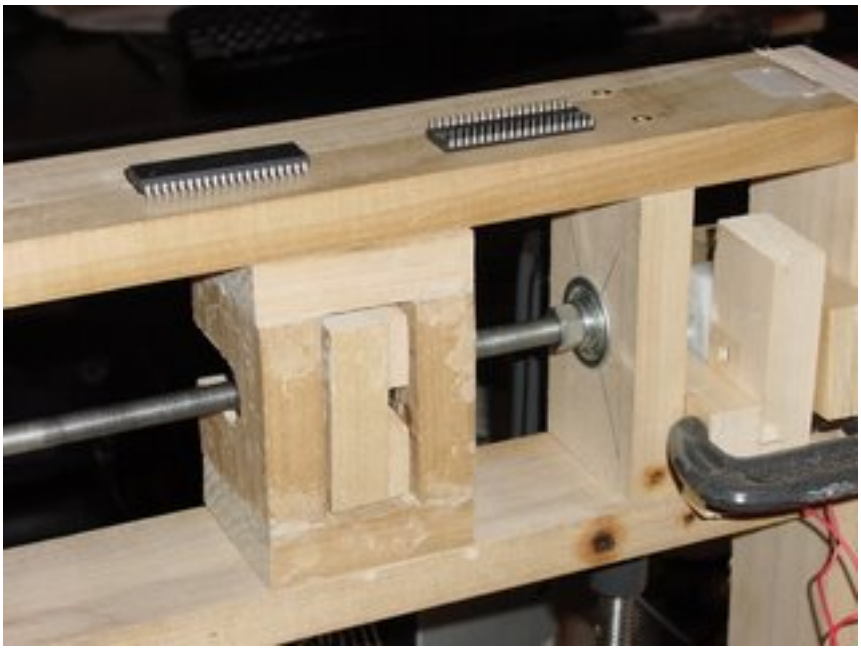
So far the limitsdetection firmware has been running constantly for over eight and ahours at a variety of stepping rates with no problems. The y-axis motoris staying very cool as well, right at 42 degrees. The 754410 driverchip is running about 40.

Doing the x-axis

Saturday, 18th November 2006 by Forrest Higgs

Igot the x-axis sliding joint working. It will be a lot less sloppy thanthe one on the y-axis. Here it is c-clamped to a fare-thee-well waitingfor the wood glue to dry. I got a really close tolerance fit on thisbeast, though. I suspect that if I want I can hang two extruders off ofeach side of the x-axis gantry.

I had to go down and get some light spackling compound and a few sheets of sandpaper to clean up the x-axis floating joint.



It seated with less trouble than I had expected.

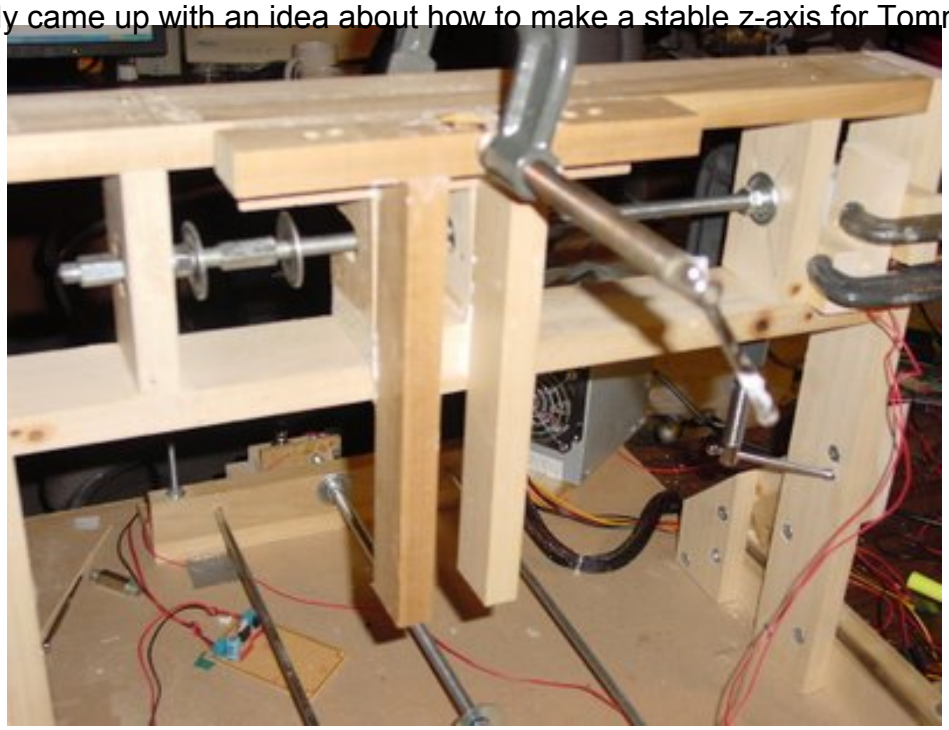
Here is a little videoof the joint traversing the x-axis. It's awfully smooth consideringthere are no guide bars save the poplar boards on the top and botton tokeep it straight.

It will be really nice being able to makenasty parts like this on a reprap machine instead of going through thewhole glue and woodworking gambit.

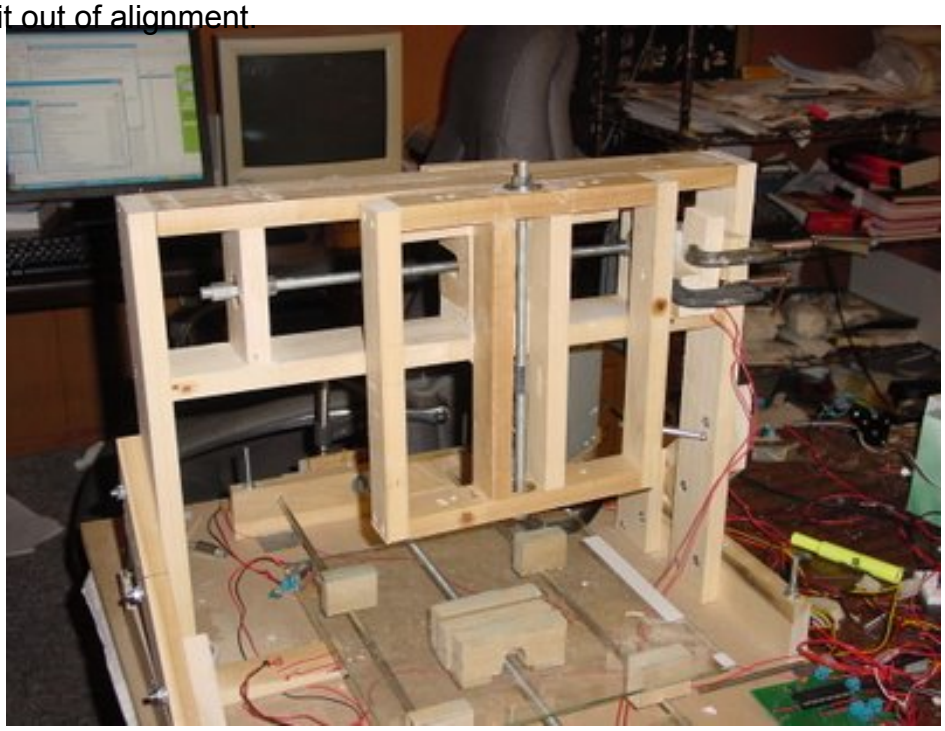
Z-axis takes shape

Monday, 20th November 2006 by Forrest Higgs

I finally came up with an idea about how to make a stable z-axis for Tommelise.



Here you see some of the bits come together in situon the x-axis platform. I'll probably have to rebuild it again after Iget this try at it put together. I was originally going to use 1/4"steel guide rods like Cat did, but changed my mind after I gotacquainted with how easy that arrangement jams if it is the slightestbit out of alignment.



Hereis the full z-axis stage without the carriage for the extruders inplace. As designed this one can accomodate two Mk II sized extrudersgrouped in towards the threaded thrust rod in the centre with theirelectronics cards mounted near the outer edge. This ensemble looks bigand bulky. It is bulky, to be certain, but surprisingly light at about750 grammes. It is also quite stable.

I haven't mounted all ofthe guide tabs and it tends to try to rotate about the y-axis abouthalf a degree while doing an x-axis traverse. The guide tabs shouldcure that. I was able to stabilise it by putting one fingertip whereone of the tabs will go.
Here is a video of the z-axis stage doing an x-axis traverse.
I'mthinking that it might be nice to have a separate z-axis for eachextruder so that we don't have to confront the problem of having thetip of one dragging over the work of another one.
BTW, the effective working volume for Tommelise is now 250x250x90 mm

Wire gauge drill bits

Tuesday, 21st November 2006 by Forrest Higgs

One of the most finicky bits of making a reprap in the States isdrilling the extruder barrel orifice. About the finest gauge drill bitthat you can get in most hardware stores is a wire gauge #60 which isjust a shade over 1 mm in diameter. I've been gritting my teeth aboutbuying one over the internet because it will cost about $3 and costmaybe $6-7 to ship. That sort of thing claws a big bloody hole in mystingy Scotts-Irish soul.

Tonight, though I was down at OrchardsSupply and there was an old guy there who really knew his tools. Heshowed me a set of #60-#80 wire gauge drill bits for $30. A #80 is .343mm in diameter. I snapped it right up.

Ifanybody else needs such a set you can get it over the web directly fromthe manufacturer who is in the Lower Hudson River Valley.

http://www.gyrostools.com/products.asp

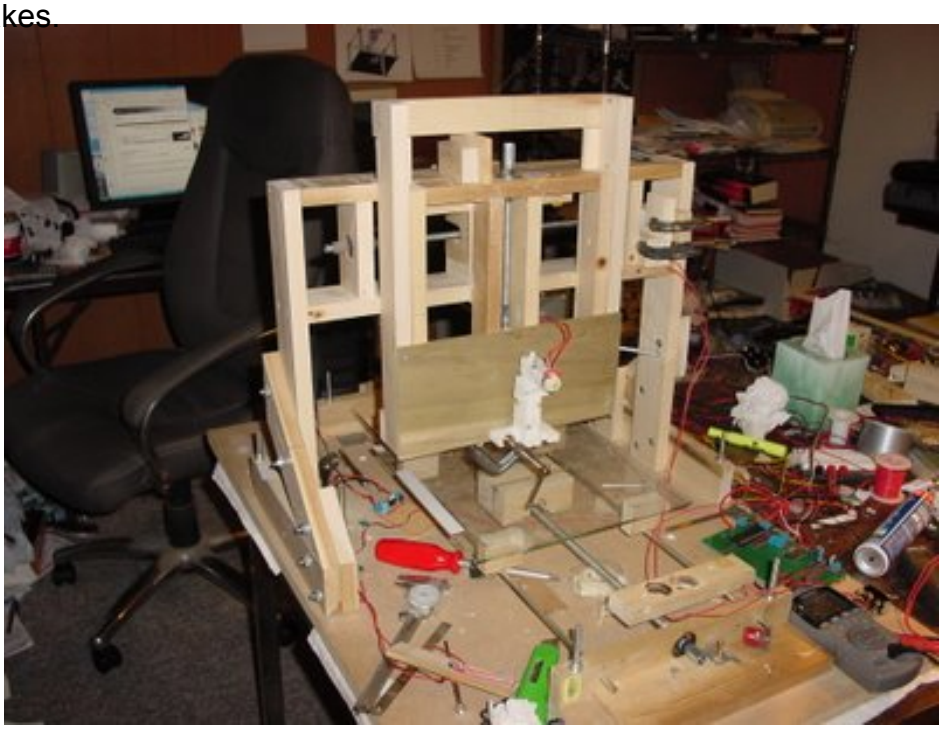
Theywork in my Dremel. Now to see if I can hold my hand steady enough toavoid breaking them. It will be interesting to see if 0.5 mm is reallythe smallest extruder orifice. :-)

Z-axis positioning stage mounted

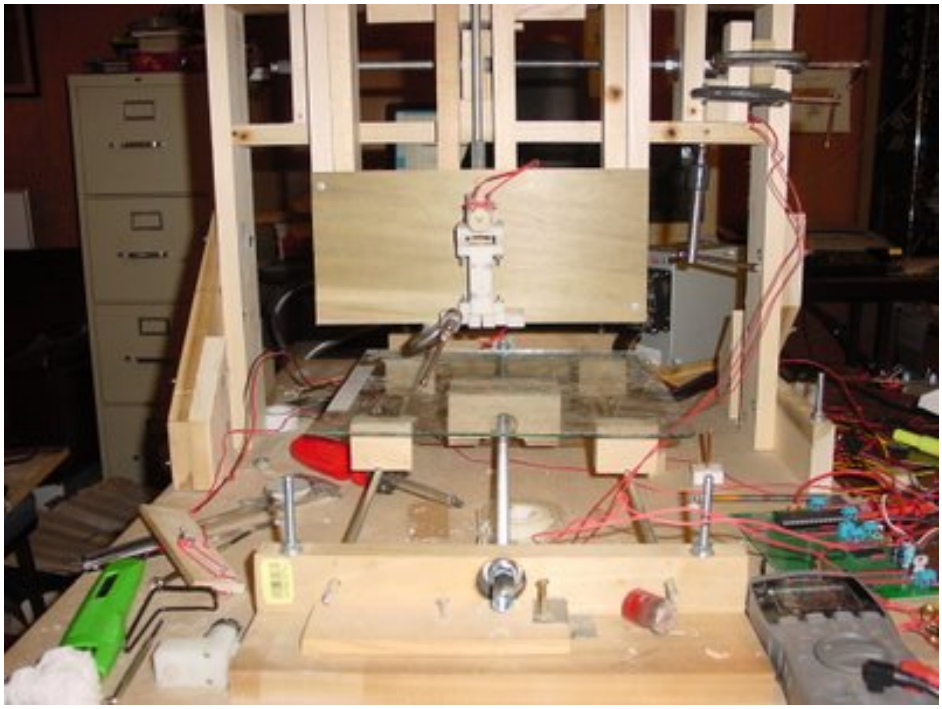
Tuesday, 21st November 2006 by Forrest Higgs

I finally decided to make the z-axis positioning stage like an inversedrawer, that is to say that it slides on the outside of and surroundsthe axis rather than the other way around.

Itseems to work rather well so far. I made it a little large so that thewood could swell slightly when the rainy season starts. I've got someadjustable clamps planned to get rid of the small amount of play thatthat makes.



I've clamped the Mk II where it will ultimately be mounted on the stage for scale.

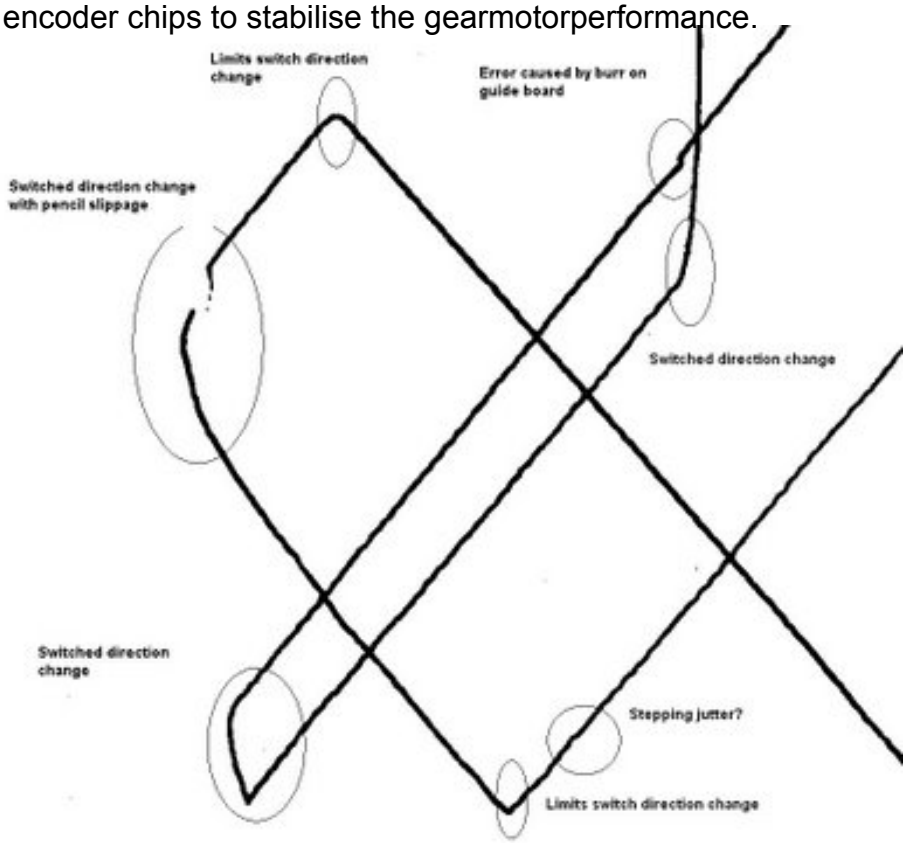


Idid a few x-axis traverses with the z-axis stage on-board. The weightof it doesn't slow anything down or seem to put any undue torquerequirements on the gearmotors. It does, otoh, smooth out the movementquite nicely, however.
I've been thinking about mounting aHamamatsu P5587 limits detector next to the extruder tip at a measureddistance to one side to act as a IR altimeter. :-)

First open-loop coordinated XY-axes pencil tests

Tuesday, 21st November 2006 by Forrest Higgs

This morning I reprogrammed the 16F877A to run both x and y-axessimultaneously so that I could get an idea of what problems I hadremaining to solve in the mechanical design. The gearmotors were run inpseudostepper mode but with open loop speed control, that is, there wasno feedback from the encoder chips to stabilise the gearmotorperformance.



Thetests were done with a 0.5 mm mechanical pencil on paper taped to theglass work surface. The paper was then removed and scanned in black andwhite at very high resolution so that it would be easier to seenon-linearities in the drawn lines that my eyesight might miss lookingat the paper in its original state.

When studying the outputkeep a few things in mind. First, the 0.5 mm mechanical pencil was notrigidly connected to the Mk II. Part of the nonlinearity that you seeat the turns is due to that. It also accounts for the break in the pathof the upper left hand direction change on the picture.

As well, some of the direction changes were done with mechanical switches rather than software and are marked as such.

Asbest as I can see there is a little bit of stepping non-linearity, alittle bit of non-linearity caused by burrs and unevenness on thepoplar guide boards for the x-axis and probably some othernon-linearities whose causes I haven't begun to understand.

Thenext step will be to sand off the x-axis guide boards a bit morecarefully and install the x-axis limits switch and the x-axis shaftencoder.

Odd bits of know-how picked up along the way

Saturday, 25th November 2006 by Forrest Higgs

I'm getting ready to mount a second 754410 onto my 16F877A motherboardin order to try to accomodate the z-axis pseudostepper motor and the MkII extruder polymer pump. Before buckling down to do that, though, Ijust swapped over the circuitry for the x-axis to test the motor mountfor the z-axis.

I finally got around to mounting those guidetabs after mounting the GM8 gearmotor. They smoothed out the movementof the x-axis rather dramatically. Feeling ambitious I decided to seeif I could get a thrust collar built for the z-axis positioning stageas well. The threaded drive rod for the z-axis is quite short. As aresult I decided to see if I could get by without the sliding jointthat have proved necessary in the y and x-axes.

While I had acoupling bolt I thought it might be nice to see if I could make athrust collar out of something else. I first thought to simply make oneout of a blob of CAPA in the manner of Vik. After a while, however, Irealised that I had a drill and tap for 3/8-24 studding, so I decidedto see if I could make one out of poplar.

Infact, I could. The poplar thrust collar was quite a nice fit and had nobacklash whatsoever. I then began working out the excess friction outof the collar with the GM8.

While the GM8 had sufficient torquefor the task the torque loading of the rather small drive shaft on thecoupling between the GM8 and the threaded drive rod generated aconsiderable amount of shear stress in the CAPA liner of the coupling.The stepping action of the GM8 induced what amounted to hysteresis inthe CAPA which resulted in heating of the liner.

Eventuallythe liner melted. Given that the GM8 only generates about 3200gramme-centimeters of torque, that this could happen is a worryingprospect for the use of CAPA to make parts for drive trains such ascouplings and gears. The GM8's torque output is about the same as oneencounters with a bipolar NEMA 17 running at slow stepping rates.
It would seem that we can look forward to using extremely conservative design parameters in gear trains made of CAPA.

Z-axis operational

Sunday, 26th November 2006 by Forrest Higgs

I got the GM8 moving the Z-axis positioning stage. You can see the video here.

Tommelise xy positioning stage

Sunday, 26th November 2006 by Forrest Higgs

My interpretation of Reiyuki's CNC gantry design appears to be veryrobust. I've been hammering the xy positioning stages at full speedsfor a full 8 hours now and nothing has shaken loose. Given what alashup much of the GM8/9 motor mounts are that is nothing short of amiracle. The GM8/9's are staying cool and their sound is staying thesame. The control board is staying cool as well.

I'm beginning to thing that I'm going to be able to extrude some stuff with this thing. I wasn't so sure before.

Coordinated xy-axis traverses

Sunday, 26th November 2006 by Forrest Higgs

After I finished the new Hamamatsu limits detector probe I tested it onthe y-axis. It would appear from the voltages that I could use it on aninput port pin. I've already got the software working on analog todigital pins for now, though, so I'm not going to take advantage ofthat. I may build an extra one and replace the old lashup I did.

Hereyou can see the new probe with it's tiny piece of circuit board toaccomodate screw terminals for power input and signal output.

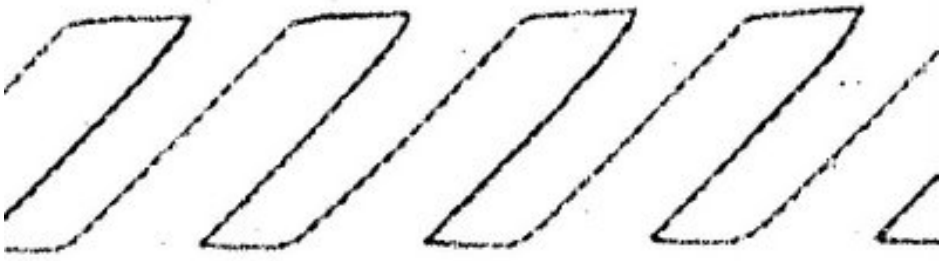
Hereyou can see both operational limits detectors while the x and y axesare doing traverses. I've been running pencil tests on the axes open inopen loop mode.

Ha! Here's another little bit of know-how that Iused to run into with discrete analog to digital chips which I'dforgotten about and which turns out still to be true for analog todigital channels on the 16F877A.
I'd put in a two pole connectorwhich sits on the AN0 and AN2 pins. When I did the firmware programmingI forgot that the connector skips a pin and programmed the x-axis forAN0 and the y-axis for AN1 instead of AN2. I connected the limitssensors up to the two poles. The x-axis sensor worked perfectly and they-axis sensor would respond properly to itslimits detector. After I watched it for a while I noticed that they-axis was changing direction both when its own sensor told it to andwhen the x-axis sensor signaled a change in direction.
What wasgoing on was that the AN1 channel, which the firmware was reading wasgetting voltage bleed from both the AN0 and AN2 pins and respondingaccordingly. This tendency for an open analog channel to get "bleed"from adjacent channels if it hasn't an input of its own is quite usual.Unless you know of this bleeding effect you can get some very bizarre "bugs" in your firmware.

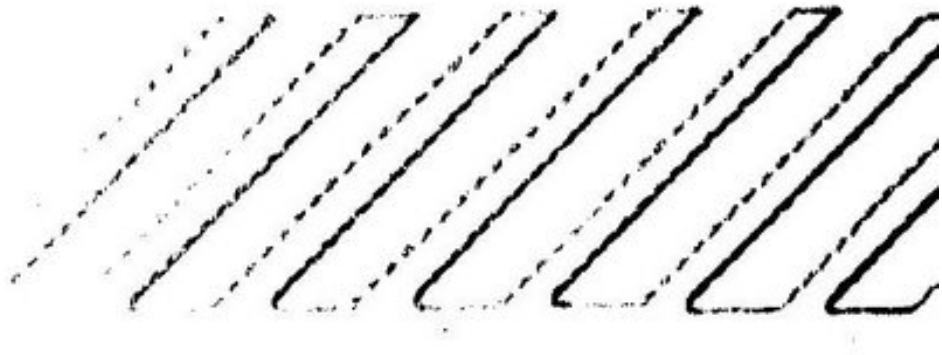
First cross-hatch

Wednesday, 20th December 2006 by Forrest Higgs

Well, I've got about +/-2 mm play in my pencil holder.



Thevertical dimension is just at 20 mm and the spacing of the cross-hatchlines is a bit over 8 mm. It's pretty easy to see the slewing at thecorners. I definitely need a more rigid pencil holder.

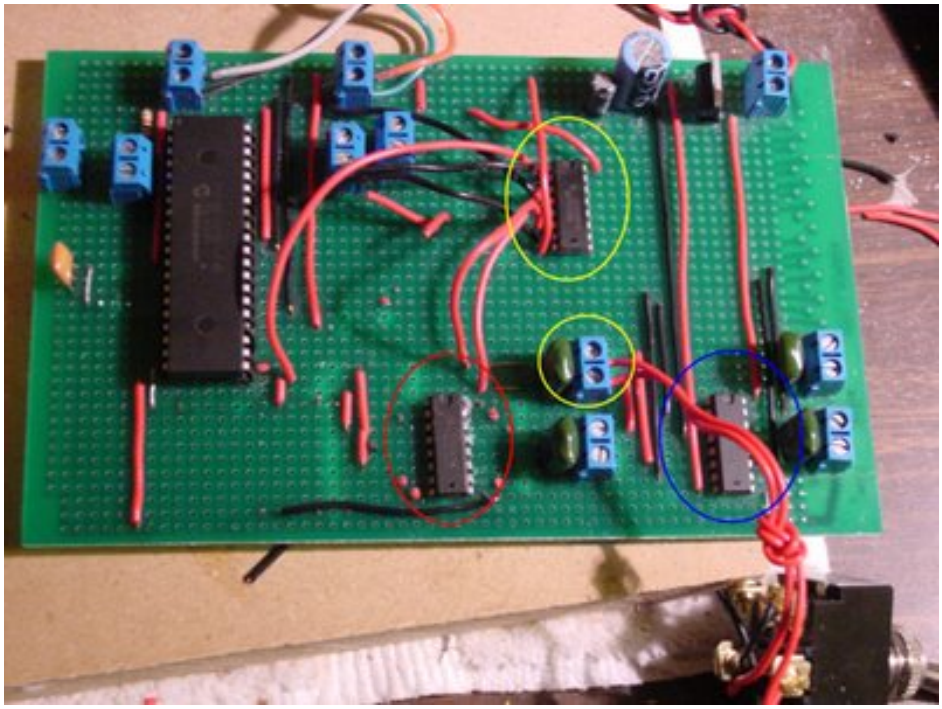


Hereis a slightly different cross-hatching pattern where I locked thepencil down a bit more solidly. Much less pencil slewing, though theline quality leaves something to be desired.

Z-axis enabled on the controller board...

Sunday, 24th December 2006 by Forrest Higgs

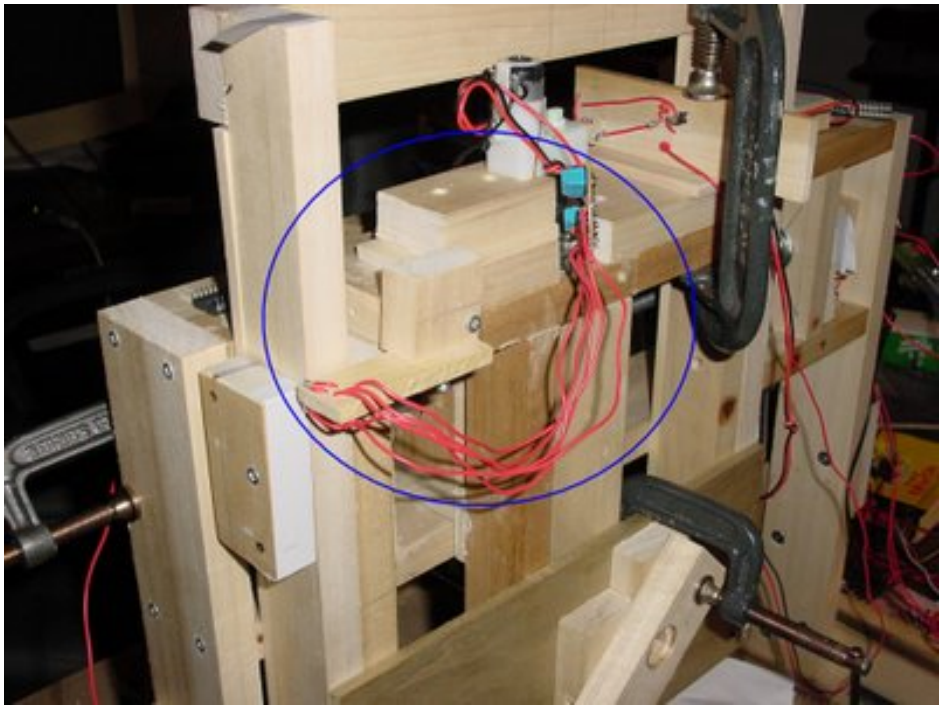
I tried installing a second 754410 dual darlington chip to power the z-axis gearmotor. What a mess!



First,I installed a 754410 for the z-axis and the Mk II (circled in red) tothe left of the one driving the x and y axes (circled in blue). Tryingto be neat I managed to miswire the new chip to the point of creating ashort within the chip.

I started to dig it out of the board (youcan see the cut pins on the right side of the chip) and then realisedthat I had some spare space on the board. I put a third 754410 (circledin yellow) in and did a quick lashup of the wiring which is very easyto see. You can also see the two terminal connector that runs thez-axis (circled in yellow as well) That mare's nest worked first timeout. There's a lesson in there somewhere, I expect.

I tested out the wiring with the {HZ+} and {CZ+} commands and it worked perfectly.



Thelimits detector went together fairly smoothly and the firmware formanaging the x and y axes checks out for the z-axis as well. The {RZ}command now works.
Now I have to wire up another shaft encoder and the z-axis will be good to go.

USB for Tommelise

Tuesday, 26th December 2006 by Forrest Higgs

Vladimirat Oshonsoft offered a discount on the purchase of the USB supportmodule for his IDE for the PIC 18F family of chips and I went ahead andgave myself a late Christmas present. :-)

That's not something I plan on dealing with with Tommelise real soon, but it's definitely on my "to do" list.

He also mentioned a new product that may be of interest to the rest of you builders, viz,

I am working intensively on the new software package - integrated development environment with simulator and basic compiler for ATtiny and ATmega microcontrollers from Atmel. It will be as much as possible compatible with my PIC IDEs. If you use these chips, please let me know what models you would like to see supported in the first release, planned for the end of January 2007.I'vebeen using his compilers for PIC's for half a year now. It's got meinto development of firmware with a minimum of frustration and timeloss. Vladimir provides excellent support for his products and ismeticulous about addressing bugs detected in his compilers. I have nodoubt that his new compiler for Atmel chips will be as good as his PICcompilers.

More on USB...

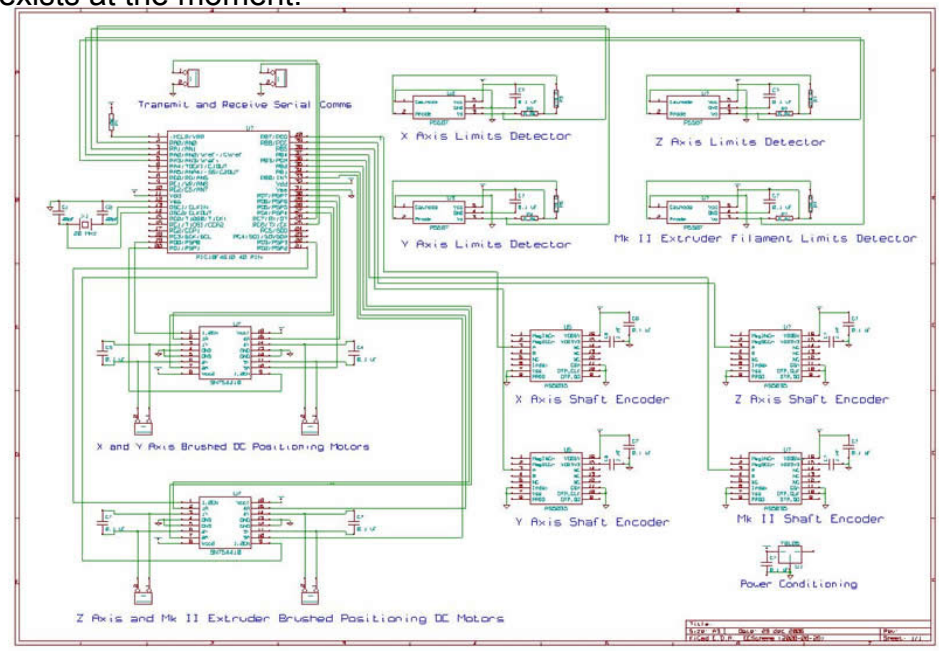
Wednesday, 27th December 2006 by Forrest Higgs

Vladimir sent along the USB upgrade for the PIC 18F IDE this morning. It looks really slick. About 8 lines of code gets Windows to recogniseyour PIC as a USB peripheral so you don't need to get into the drama ofwriting drivers. I may shift over to USB communications sooner than Ithought. It will be nice not needing a comms card. That will really bring down the components count. :-)

Schematic for the Controller Electronics for Tommelise (in progress)

Friday, 29th December 2006 by Forrest Higgs

I decided that I had better document the mare's nest of wires that are on the controller board that runs Tommelise this evening so I rolled out KiCad and got reacquainted with it. Here is the schematic as it exists at the moment.



Although it looks rather intimidating when you really look at the schematic you discover there are really only a few unique elements that are used in set configurations over and over. It is also worth noting that the encoder and limits detector modules are located on small, independent boards located quite near to where they are needed. That keeps the controller board a lot simpler, though it does load it up with connectors. :-)

Abusing IC's

Friday, 29th December 2006 by Forrest Higgs

I'm often amazed at how much abuse some integrated circuits can takeand how little others will be put up with. Since I'm blessed with about15 thumbs and a lifelong case of ADD I get to find out more about thatsubject that most, I suspect.

WhenI first started doing my own microcontroller board designs I was usingthe RepRap standard 16F628A PIC microcontroller. It's a solid littleCPU, at $1.75 ridiculously cheap, easy to program and utterly reliablewhen properly used. When you are prototyping boards you always use asocket to seat it since you want to be able to pull it from the boardand reprogram it in your PIC programmer board many, many times.

Ifyou are a little fuzzy in the morning when you are trying to make somecorrections to the firmware program you've written for the 16F628A,however, it is rather easy to put it into the board socket backwards.With the 16F628A, if you do that then power up the board the whole chipshorts out and in a few seconds gets hot enough to give you seconddegree burns if you chance to put your fingertip on it. When it doesthat your $1.75 PIC chip is cooked.

If that was all thathappened it wouldn't be too bad. It gets worse, though. If you power upyour board and then walk back to your PC to try to work with yourmicrocontroller board from your PC the chip will literally melt thesocket you've seated it in. When that happens you're faced with thetedious job of carefully removing the cooked PIC AND the socket out ofyour board. In practice what it means is that you've wrecked yourprototype board, too. That is especially annoying in that yourprototyping board is often as expensive as the IC chips and othercomponents you've put on it.

When I shifted over to the 16F877APIC chip a few months ago it was a whole different world. The 16F877Ais a much bigger and more powerful PIC chip than the 16F628A. It's a 40pin beast and costs about $5-6.

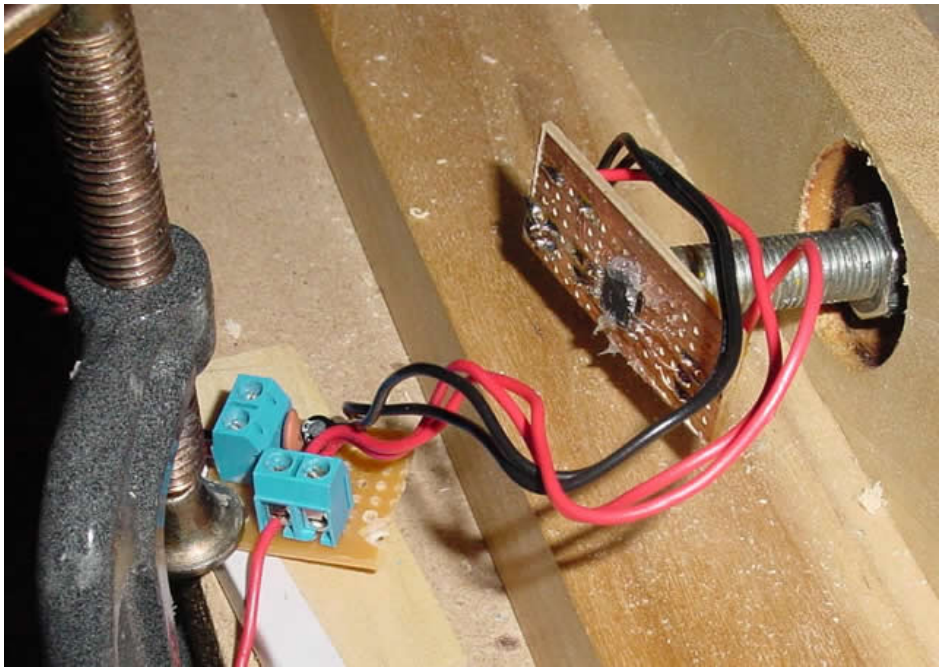
Whoever designed the 16F877A hadnitwits like me in mind. If you reverse a 16F877A in its socket thechip doesn't fry and you don't melt your socket, either. Yourcontroller board won't boot up, but who cares? Pull the chip and seatit properly and everything is fine. You don't lose your PIC chip andyou don't lose your prototype board. That's really cool.

The16F877A chip can handle more than that, though. I've even accidentlypulled one out of the board when the board was powered up. It didn'tmind a bit.

Indeed, the only time I've managed to fry one iswhen I was building a high amperage motor controller board around the16F877A. The dual Darlington chip that I was putting on the board was aL298N. You can safely handle up to about 4+amps with that big boy.Unfortunately, I had seated the L298N and had forgotton to cut all theunnecessary stripboard traces between the L298N and the 16F877A. Thatfried the PIC 16F877A. It still didn't melt the socket, though.

Justin passing, if you ever have need to run a fairly heavy motor the L298Nis an awesome chip. It might fry your other chips if you hook it upwrong, but I've yet to burn one out. It is a vertically mounted chipwith seating for a heat sink. It will run without one and seems happyto run properly at several hundred degrees. I found out about that whenI put a finger tip on one under load and burned off a fingerprint.Since then I've put heat sinks on them so that they only get hot enoughto give me first degree burns. That's a great improvement.

A fewweeks ago I shifted from the 16F877A to the even more powerful 18F4610.This bad boy costs about $9, has loads of Flash and RAM memory, a muchmore forgiving CPU, direct USB access and mirabile dictu is completelypin compatable with the 16F877A. All I had to do was recompile my oldfirmware for the new chip, plug it into the board with no alterationsand the thing ran like a dream. Indeed, the 18F4610 cured up a problemthat the 16F877A was giving me having to do with stack overflow causingrandom errors. I haven't had any stack overflows since shifting. The18F4610 looks to be as tough as the 16F877A. I haven't got onebackwards in the socket yet, but I have accidently pulled one from apowered up prototype board with no ill effects.
This morning Iwas testing the z-axis (vertical) on Tommelise to check to see if themagnetic shaft encoder chip was counting pulses properly and discoveredthat it wasn't. I have a bit of a love-hate relationship with theAustria Microsystems AS5035 magnetic shaft encoder chip. It is aWONDERFUL chip operationally. With most shaft encoders you have use acouple of infrared emitter detector chips to read an IR transparentdisc onto which you've printed radiating lines to give you your pulsesto tell you how far your shaft has rotated. IR shaft encoder assembliestend to be a couple of inches across as a result if you are looking fora decent pulses/revolution count. With the AS5035 you have a tinylittle chip about 3/8x3/8 inch big. You mount that facing the end ofyour rotating shaft and then either glue or, as I do, simplymagnetically stick a 1/4 inch pill-shaped magnet onto the end of therotating steel shaft. The chip reads the shifts in the field of thattiny little magnet and gives you a really clean set of pulses to workwith electrically. For 128 pulses/rotation (256 if you connect pins 3and 4 instead of just using 3 OR 4 for input) you pay about $4-5. Youcan get 4096 pulses/rotation for about $8-10. It really keeps the sizeof your shaft encoder way small and lets you put encoders in placeswhere you couldn't hope to seat an equivalent IR encoder assembly.That's the good news. The bad news is that the AS5035 is a surfacemount chip with mounting conductors (equivalent of the pins on a DIP ICchip) that you need to use a strong magnifying glass to see the gapsbetween. Ordinarily you'd need a proper surface mount solderingworkstation to attach those to anything. Fortunately, Dr. Bowyer ofRepRap fame over at the University of Bath in UK confected a methodwhich you can use to hook them up using a little piece of stripboard, abig magnifying glass and an ordinary soldering iron (plus a steady handand nerves of steel if you're a penny-pinching Scots-Irishman like Iam).
Back to the problem with the z-axis shaft encoder. Notunexpectedly I had miswired the board, though mercifully this time Ihad the chip connected up properly for a change. The chip was heatingup to about 120 degrees, warm enough that you could tell it was hotwhen you put your fingertip, the one without the fingerprint, on it.When I felt that it was warm I suspected that I'd fried it in that Ihad similarly fried one a few months back. That one, though, had got aLOT hotter. Just for luck, however, I rewired the tiny little boardthat went with the AS5035, slapped it on the 5v power supply andtweaked it with a pill magnet and got some pulses that looked right.Rather than slap it back on the z-axis for testing, which currentlyentails demounting the whole z-axis assembly I just popped it on to theend of the y-axis drive shaft and told it to calibrate the y-axis. Itdid so perfectly.



Hereyou can see the z-axis shaft encoder hooked up to the y-axis fortesting. If electronics intimidate you because you can't see yourselfdoing the magical things that such boffins seem to do on a daily basisyou are likely to find me very easy to relate to. I am probably theclumsiest and most error-prone amateur board designer in the western UnitedStates. If you've made a mistake, I've probably made the same one four or five times. :-(I have to depend on pure bullheadedness and persistence to getthings done.

Z-axis calibrated

Friday, 29th December 2006 by Forrest Higgs

I got the shaft encoder mounted on the z-axis threaded drive shaft. It measured 2.6250 inches which agrees with the working span I set it to. I also checked in the x-axis at 6.6595 inches and the y-axis at 6.3696 inches. I need to put a spot of glue on the pill magnet on the y-axis in that the shaft encoder's recess fits the magnet a bit tightly and tends to drag on it a bit too much for the magnetic attachment to the shaft to be sufficient.

I can expand the y-axis out to about 10 inches by simply relocating the limits sensor. It will take building up a narrower z-axis work platform to get a bigger x-axis and a minor rebuild of the system to get more than a 2 5/8 working depth.

What I've got, however, is more than adequate for the next phase of experimentation that I want to do. Here are the things I want to make in order of priority once I have the Mk II extruding...

- more robust housings for the shaft encoders
- a less ad hoc set of mounts for the gear motors
- a regularised sliding thrust collar design for the xy and xz working platforms
- a upgraded Mk II

Those are the little fiddly bits that make replicating a Tommelise really time consuming. One of those sliding joints alone takes a good morning to make by hand. That will set me up to do experimentation on on making the Mk II run HDPE and HPP polymers which are much cheaper and more readily available than the polycaprolactone that I am using now.

The very next thing that I will be doing, however, is creating a limits sensor to detect when we are approaching the xy working surface or anything laying on it. That will let me calibrate the vertical dimension of Tommelise's working volume on the fly. It will also give me a crude 3D scanner.

Speak of the Devil...

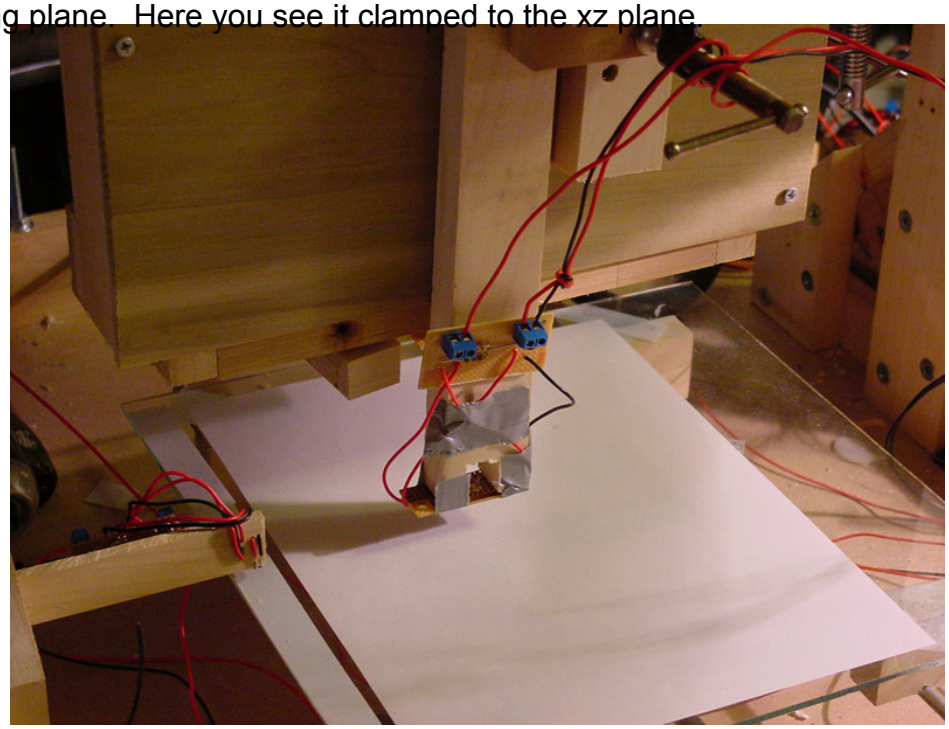
Friday, 29th December 2006 by Forrest Higgs

No sooner had I blogged the bit about planning to test HPP and HDPE than my sample of several pounds of 3 mm HPP filament arrived via UPS!

IR Altimeter installed...

Sunday, 31st December 2006 by Forrest Higgs

I salvaged a Hamamatsu P5587 Photoreflector chip, the kind I've been using to do limits detection, from a prototype board that I built to test the chip several months ago when I first acquired them and converted it into a lashup IR altimeter for Tommelise. Hopefully, it will let me map the flatness of the xy working plane. Here you see it clamped to the xz plane.



It looks nasty, but appears to work. I will be upgrading the firmware to accomodate it tomorrow morning.

Testing the IR Altimeter concept...

Sunday, 31st December 2006 by Forrest Higgs

This morning I wrote a new routine into the PIC firmware to test the IR altimeter that I built yesterday. although it works, it doesn't work the way I'd hoped that it would.

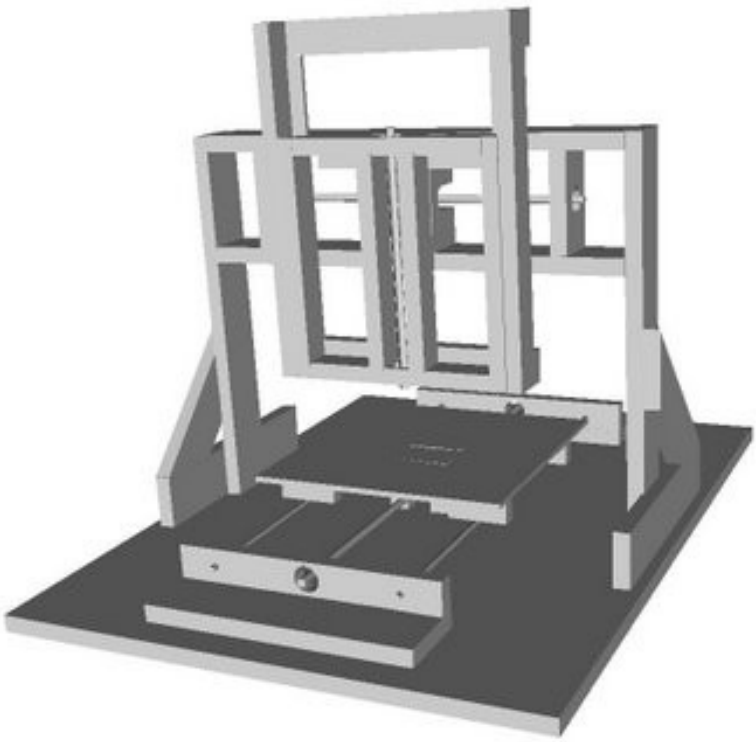
For an initial test I monitored the output of the Hamamatsu P5587 chip on the altimeter while the z-axis moved the xz extruder downwards towards the xy working surface. The PIC firmware was set to turn off the z-axis gearmotor when the chip "saw" the paper for the first time.

With white typing paper the P5587 chip "sees" the paper from about 1.25 inches away. I tried the test again with a piece of dark grey styrene plastic. In that case the analog response from the chip did not begin to rise until the chip was within 1/8" of the plastic surface. It appears that the dark styrene is also dark grey in the IR range.

Documenting the basic structure of Tommelise

Thursday, 25th January 2007 by Forrest Higgs

Recently, I built up the basic structural elements of Tommelise in the open source Art of Illusion (AoI) 3D modelling package. I took the opportunity to correct a few dimensions that I would have built somewhat differently, like making the xz vertical positioning stage a bit narrower, than I actually did in practice.



Other than a few things like that, it is shown in AoI pretty much as built. Mind, I am not particularly happy with the 1/4 inch steel guide rods that I used with the xy positioning stage, either. I may yet change them over for 1/2 inch ones. The 1/4 rods are entirely too flexible for my taste, though they don't seem to have caused trouble yet.

In any case, if you are interested in using the Tommelise design as a starting point for your own homebrew bootstrap 3D replicator, this AoI file will give you the dimensions of the basic structure. Keep in mind that AoI is not a CAD package so you have to set up a grid to get dimensions. The measurements are done in inches.

I will be building a more complete structural description of Tommelise in the Documentation section of the Clanking Replicator Project website in a few days.

First tries at code transfer

Sunday, 28th January 2007 by Forrest Higgs

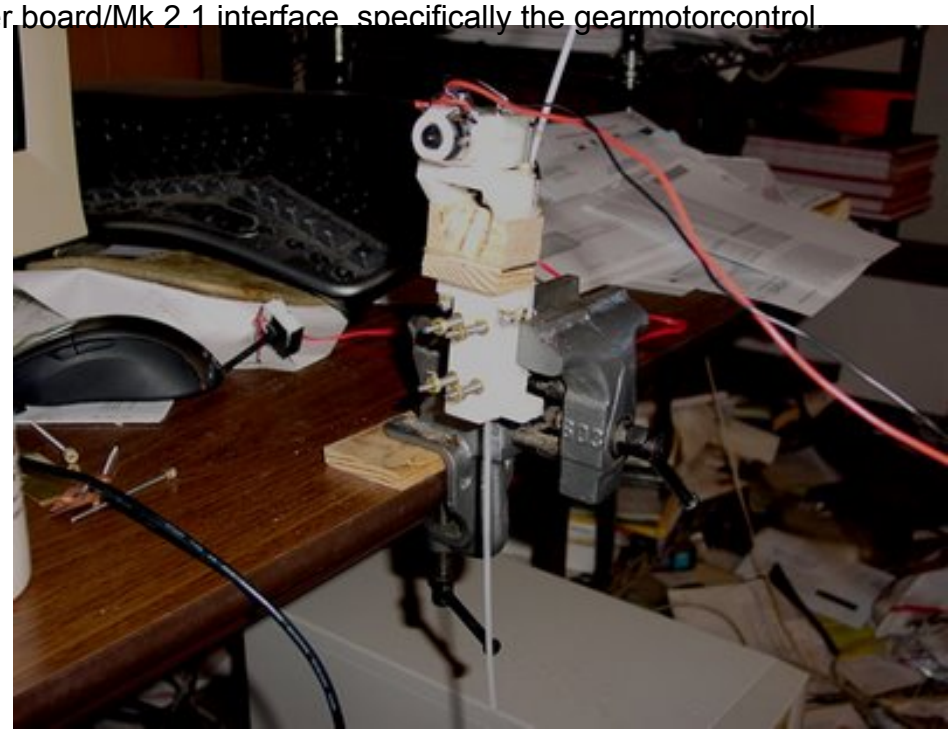
The people that are doing SplineScan said that their VB6 code was not too efficient. I've been trying to move it over to VB.NET 2005 and it's pretty obvious that it isn't. I'd forgotten that it's been several years now that I've written anything in VB6.

Mind, they have what look like some good mesh fitting routines that I'll most likely use to good effect.

Firmware for the Mk 2.1

Sunday, 28th January 2007 by Forrest Higgs

I was able to code the basic control for the gearmotor on the 2.1after breakfast with only one or two hilarious mishaps relating tospecifying the wrong I/O channels. Right now I'm checking out thePC/controller board/Mk 2.1 interface, specifically the gearmotorcontrol.



The Mk 2.1 has a number of changes in how it is operated. Probably themost important is the use of a channel on a 754410 dual Darlington chiprather than a transistor to run the gearmotor. What that means is thatI can feed filament into the heated extruder barrel AND pull it backout. I've just tested that feature and it works properly.

I'm thinking that bidirectional control of the filament will makeextracting tag ends of filament out of the Mk 2.1 easier and that itwill let me avoid the dribbling that Vik has reported with CAPA bybacking the filament up slightly when the extruder barrel is hot but weare not extruding.

The previously blogged low thermal inertia extruder barrel also makesheating response of the filament much more lively. I am alsocontrolling the temperature of the melting filament indirectly ratherthan using a thermistor as per present practice. I'm controlling powerinput to the nichrome 80 on the extruder barrel as a function offilament feed speed rather than using a thermostat function.

Using the slow pluse "pseudo stepping" rather than PWM motor controlalso lets me maintain full torque down to zero rpm. We shouldn't beseeing stalling absent a complete extruder barrel jam with the Mk 2.1.

Heated extruder barrel power circuit wired into the board

Sunday, 28th January 2007 by Forrest Higgs

I got the power transistor wired into Tommelise's control board. When I test it without a load it seems to be giving the voltages I'm looking for. However, when I hook it into the nichrome 80 heating element on the extruder barrel I don't get any heating. Further, I'm not able to measure the voltage I was when I had it disconnected.

All I can guess is that somehow I'm not getting any amperage. I can't see anything wrong with the circuit, so my suspicion is that I've got too tired to be doing this kind of work. Time for a cup of hot tea and "The Thin Man Goes Home".

Yes, so now you know. I'm an old movie buff when I'm too tired to do anything useful. Way back in the 1950's the little television station that served the little bit of the south Texas coast where I grew up showed a LOT of old movies for lack of any better programming. As a result I developed a taste for 1930's-1940's movies rather early on. I'm doing a "Thin Man" movie festival this month.

Mk 2.1 controls working on the Tommelise controller board

Monday, 29th January 2007 by Forrest Higgs

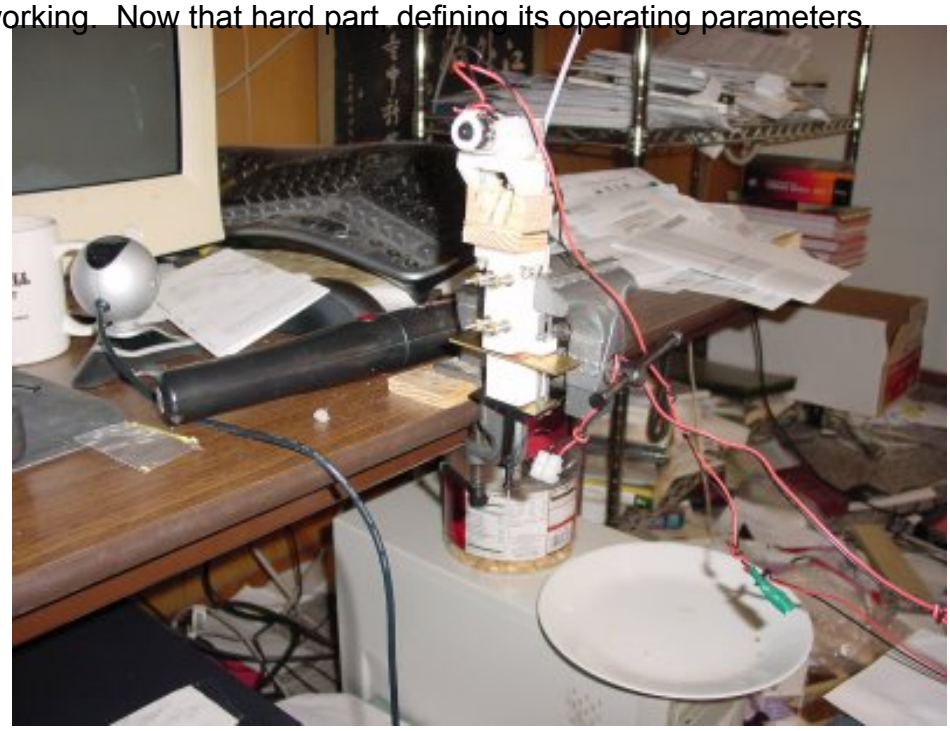
Getting the polymer pump control running on the Tommelise controller board was no problem at all. I hit a dead end, however, when I rigged the power transistor that runs the extruder barrel heater. I was trying to control that with PortC.3 on the PIC 18F4610. Turns out that that particular port bit defaults to something besides an output port even if you set Trisc.3 to 0. I didn't want to get into the drama of fiddling with the configuration for PortC so I just moved it to PortC.4 this morning and everything worked fine.

Now I can finish writing a control routine that coordinates extruder barrel thermal input and filament flow. Stay tuned! :-D

Mk 2.1: Defining the envelope

Monday, 29th January 2007 by Forrest Higgs

Okay. I got it working. Now that hard part, defining its operating parameters.



I first established that it would work at all

It did with HDPE. The extrusion rate wasn't nice, but these are early days.

First I did a series of tests with HDPE. It's a fairly tough, but softsurfaced engineering plastic. With hand feed experiments that I did awhile back it was slippery to a point where I had to grip it withpliers to feed it into the extruder barrel.

My first instinct was to really tighten the springs that press thefilament into the threaded polymer pump. You can do that, but if youtry to make the pump run too fast, that is, develop some

pressure inthe extruder barrel, it just excavates a groove in the side of thefilament and you get nowhere. That was discouraging.
I then shifted over to polypropylene (HPP). It is a harder surfacedplastic and the threaded pump drive engaged it much better than it didthe HDPE. After faffing around with HPP for a while I began to plan anexperimentation campaign.
Right now I'm running the polymer pump with HDPE at 2 rpm and 2 ampspower. I'm going to keep power constant and run the rpm up step bystep and see what happens.

Forensics on Tommelise's polymer pump

Tuesday, 30th January 2007 by Forrest Higgs

I finally concluded that the Mk 2.1 polymer pump isn't providing the thrust that I need to consistently extrude HDPE and HPP with Tommelise. Given that Vik down in Auckland has managed to rupture an extruder barrel with thrust from more or less the same Mk II pump down in Auckland, I can only conclude that the fault lies with how I've put the 2.1's together rather than with the design itself.

This morning I am going to do a breakdown of the polymer pump and do some forensics on it.

The Mk 2.1 extruder is operational

Tuesday, 30th January 2007 by Forrest Higgs

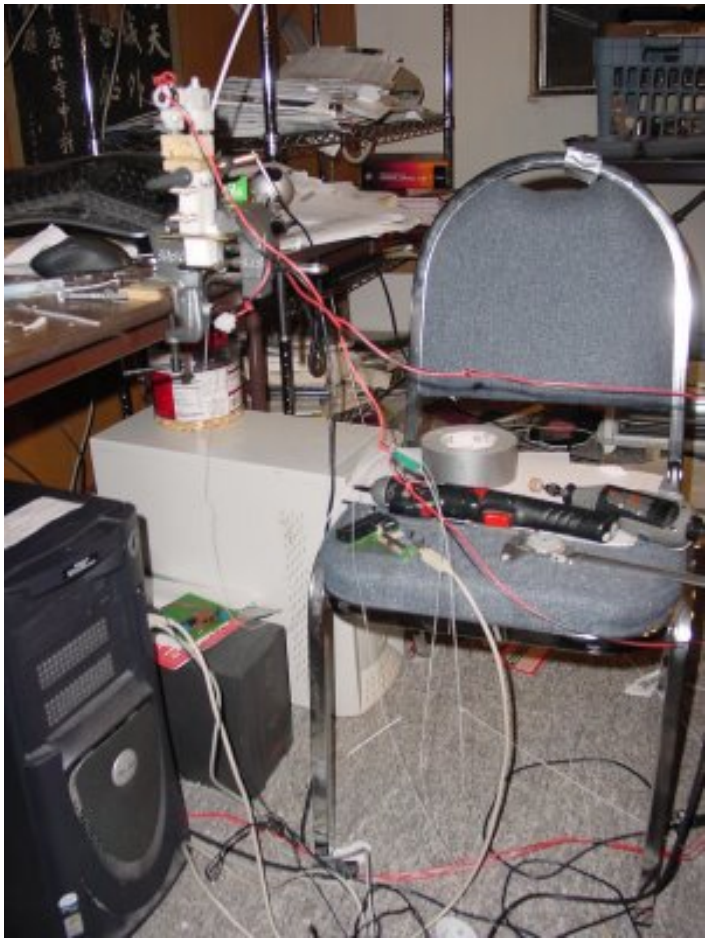
After a bunch of tweaks a complete rebuild of the polymer pump and asking Vik in Auckland some questions I've got the Mk 2.1 running properly ... with high density polyethylene (HDPE).

That is a first.

Bloody HPP and HDPE extruded thread everywhere

Tuesday, 30th January 2007 by Forrest Higgs

I've got both HPP and HDPE extruding out of the Mk 2.1 extruder at production rates (~3.4 mm/sec currently).



Iran 500 mm of 2.9 mm HDPE filament and 750 mm of 2.9 mm HPP filament. Now I've got to tidy up the Mk 2.1 so that I can mount it onTommelise's xz positioning table and then I can begin the long, painfulprocess of getting it to make something useful.

Tidying up the Mk 2.1 for Tommelise

Wednesday, 31st January 2007 by Forrest Higgs

The Mk 2.1 seems to be a very different beast from the Mk 2. Yesterdaybefore I got things working properly with the Mk 2.1, Vik mentionedthat he often put a clamp on his Mk 2 for a bit of extra pressure onthe polymer pump. I grabbed a c-clamp and tried that, applying justenough pressure to stall the GM3 (I don't worry about damage in that ithas an internal slip-clutch) and then backed off a touch.

That'swhen the Mk 2.1 started pumping like crazy. I had the GM3 seated on #40studding bolts not locked down. I noticed that the motor coupling wasriding up pushing the GM3 up on its studding bolts. That didn't hurtanything and I've had the motor coupling, made of a cut coupling nutand nut with a lock washer between the two, come loose before, so Ididn't worry too much about it. After I finished running the 400 mm ofHDPE, I had a rush of good sense to the brain and decided to take theMk 2.1 polymer pump apart before I tried running HPP.

That wasrare good sense. When I disassembled it I discovered that the motorcoupling hadn't broken loose at all. Instead, the thrust collar thatI'd made of two nuts with a lock washer between had broken looseletting the whole threaded polymer pump try to work its way out of thepump.

One thing wants noting at this point. I use a thrustcollar to absorb the tendency of the threaded drive rod to try to comeout of the pump rather than the two bushings that the Mk 2 has. I havethe a brass plate with a hole through it to admit the threaded rod atthe top of the pump before you get to the motor coupling. Thepreviously mentioned thrust collar butts against that. Basically, Iused the same idea that Vik did with his drawing that is in the Mk 2documentation on the reprap site.

Where the bottom bushing is inthe Mk 2 I have a steel bushing that is the same diameter as thethreaded thrust rod. This lets me avoid having to mill down the thrustrod since it seats right against the threads.

Back to the story.The threaded rod's thrust collar had come undone and let the threadedrod ride up in the pump till the bottom of the rod had slippedcompletely off of the bottom bushing.

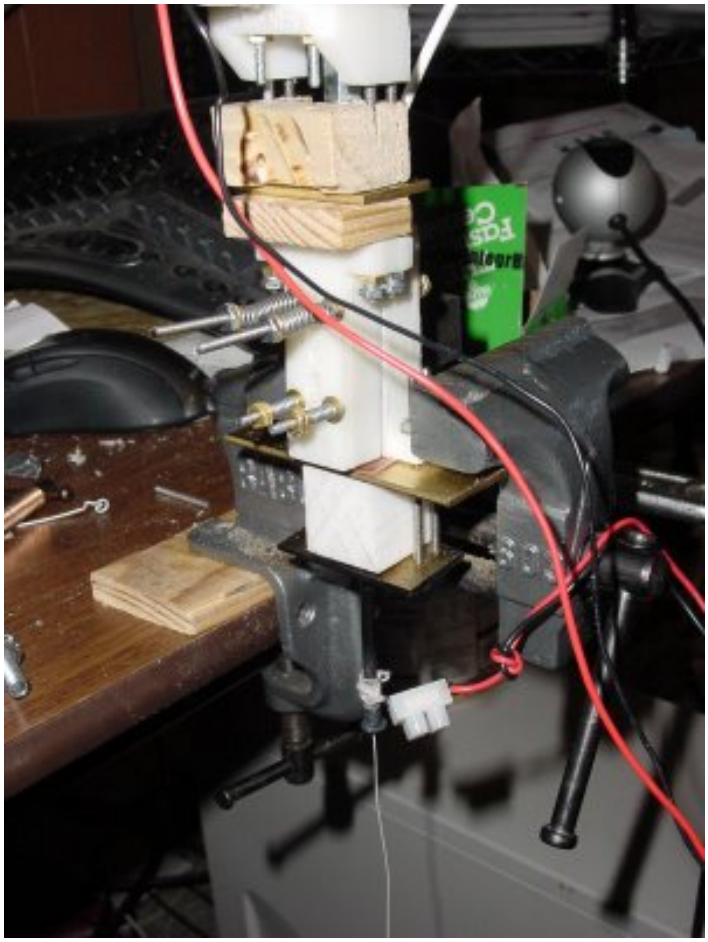
That's when the pump started working properly.

Whatthat meant was that with the configuration I was using all the bottompressure springs in the polymer pump were doing was adding friction tothe device for no good reason. The springs at the top where I wasapplying a touch of extra pressure with the c-clamp were effectivelythe only ones that mattered.

For the HPP I reseated the threadedrod in the polymer pump, really tightened the thrust collar and applieda bit of pressure to the top of the pump with the c-clamp in additionto what the springs could offer.

HPP pumped just fine with this arrangement.

Thismorning I sloped down to the hardware store and bought a more powerfulpair of springs for the top of the pump in the hope that I could getrid of the c-clamp. That worked. You can see the Mk 2.1 sans c-clampoperating this morning in this pic.

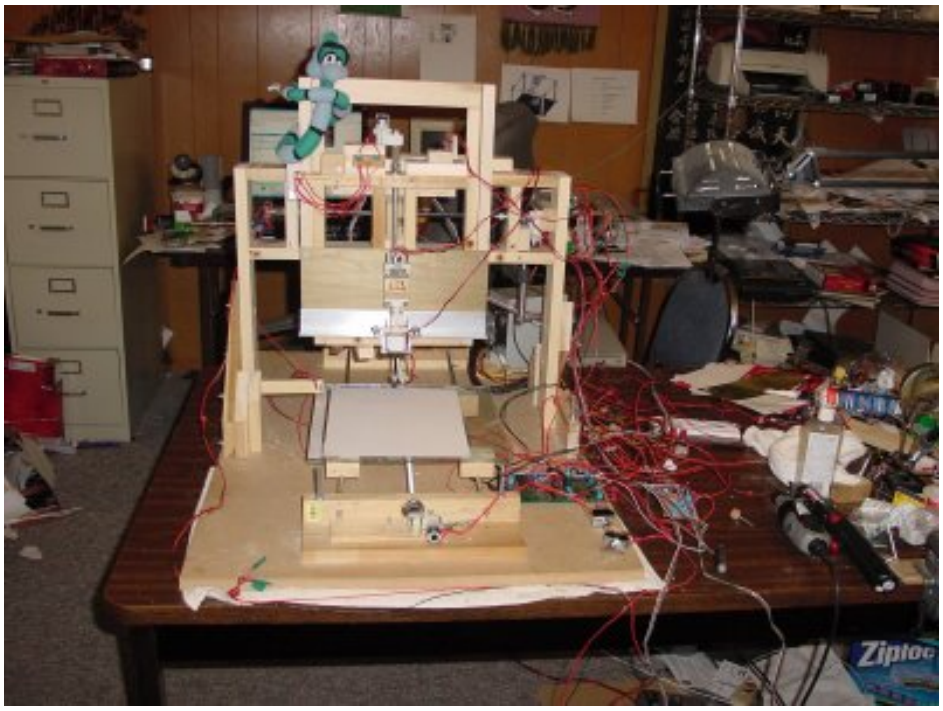


Thepicture shows you something else interesting that is happening. The BBQpaint that I used to tack down the insulated nichrome 80 heating wireis perishing at the top of the heater coil. This lets it delaminatefrom the extruder barrel and then causes the insulation to overheat andbegin to perish as well. The extruder barrel is still working, but Ican see how this is going to go.
My first try at a fix for thisproblem is to use more BBQ paint coats to tack down the heater coil.I've got some other options if that doesn't work.

Tommelise rollout

Wednesday, 31st January 2007 by Forrest Higgs

I put down the first piece of poplar on what has become Tommelise on27 September 2006. It's been just a shade over 4 months in the makingand now I'm ready to get it to make things.

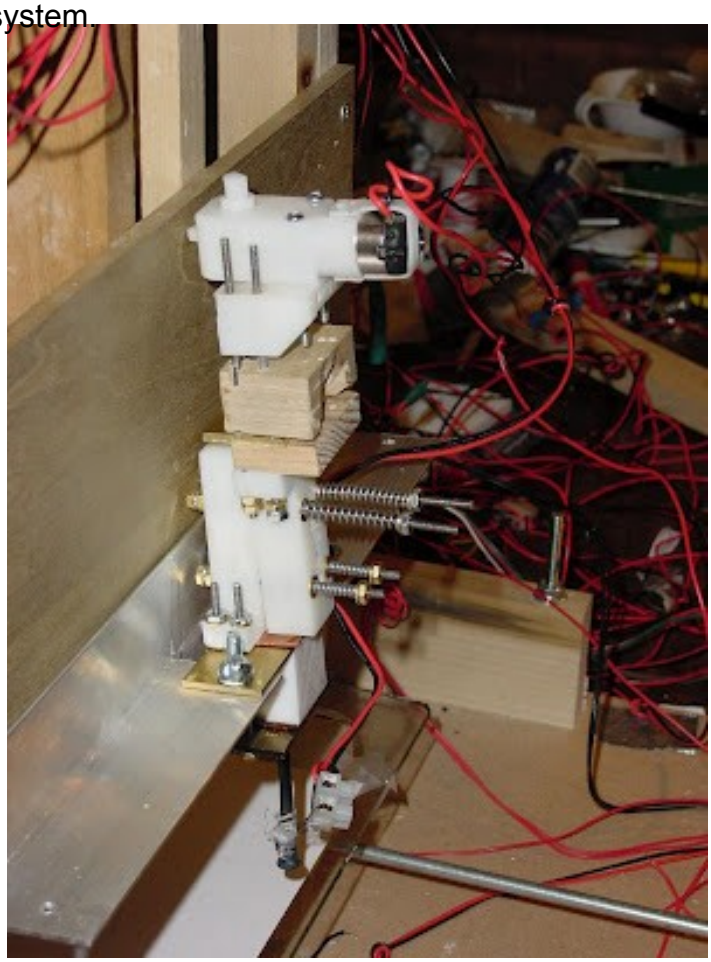


Allthe systems work. The trick now is to get them to all work together. Wish me luck. I have a feeling that the hardest part of the job ofmaking Tommelise a reality is ahead of me.

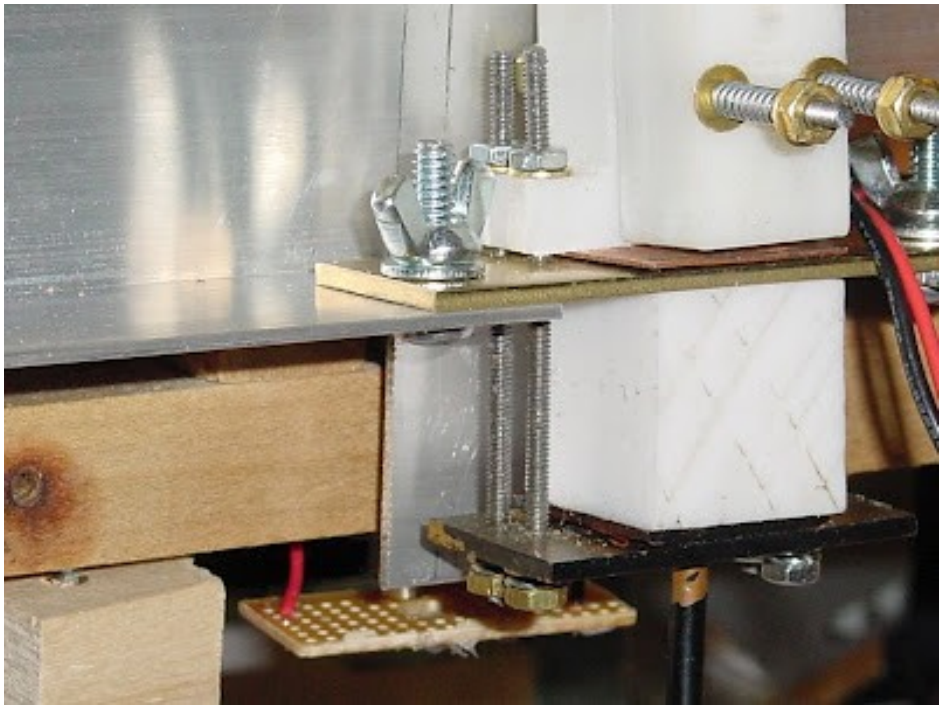
Mounting the Mk 2.1

Saturday, 3rd February 2007 by Forrest Higgs

I stopped and did a little documentation photography of the final mounting of the Mk 2.1 to the Tommelise positioning system.



I created a mounting bracket out of a foot-long piece 1-1/2 x 1-1/2x1/16 inch aluminum "L" section. I sawed a section out in the middle to accomodate the Mk 2.1. Notice that the spring loaded studding bolts that maintain pressure on the filament in the polymer pump stick out behind the polymer pump so that you can't mount it flatly on the xz positioning table, thus the need for the aluminum "L" section bracket. The Tommelise's Mk 3 extruder will have an integral mounting bracket on the back side of the the polymer pump which will eliminate the need for the "L" section.

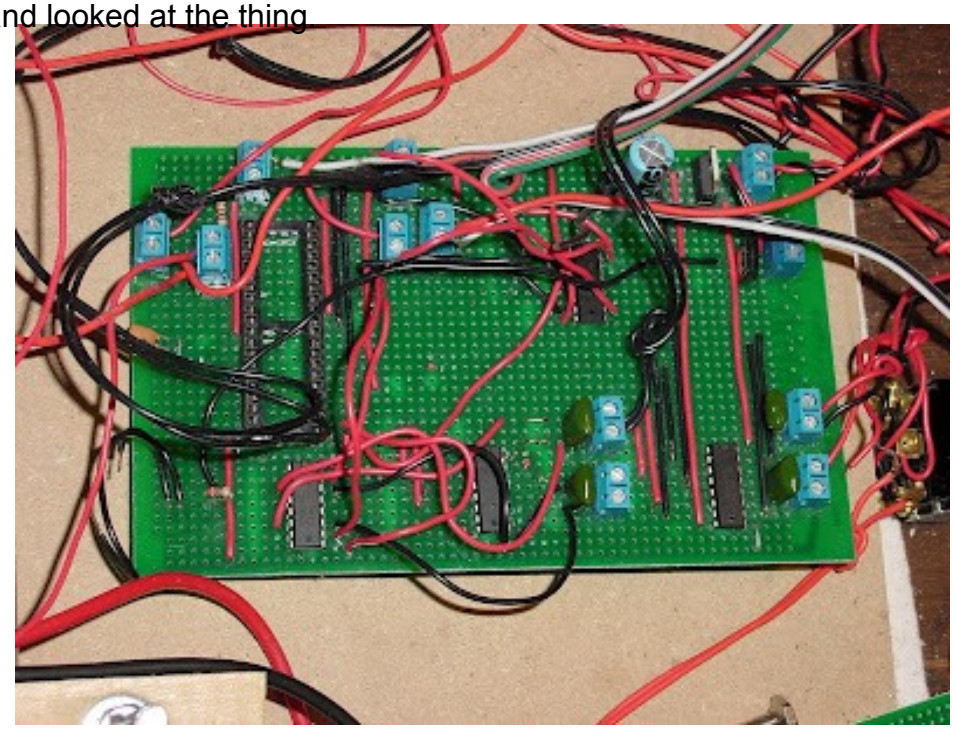


Here is a closer view of the mounting. You can see here that rather than saw the piece of flange off I simply bent it down. That was mere convenience on my part. You can also see the #10 machine bolts with wing nuts that I used to secure the brass mounting flange to the aluminum bracket. I have some longer #10's which I intend to spring load so that if there is a z-axis positioning mishap the Mk 2.1 extruder barrel will simply push upward against the springs rather than be thrust through the tempered glass xy working surface.

Time for a new controller board for Tommelise

Sunday, 4th February 2007 by Forrest Higgs

I was working at trying to find the bug in PIC-based controller board for Tommelise and finally stepped back and looked at the thing



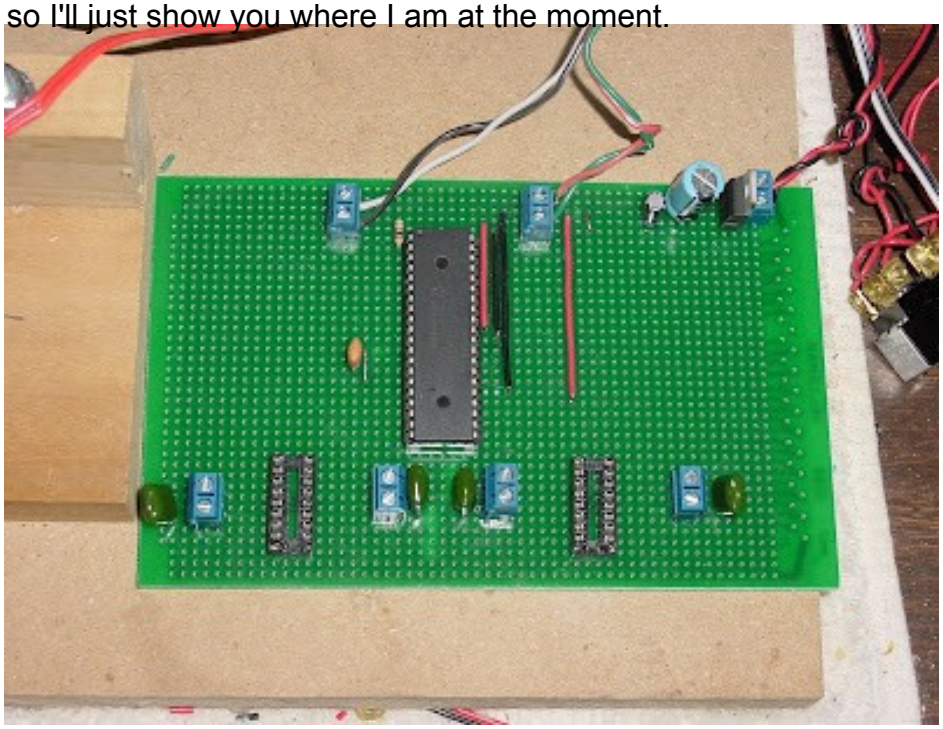
This is ridiculous. The whole thing is patches from various things that I tried that didn't work just right for one reason or another. Not only that, it's enough to scare the wits out of anybody contemplating making their own Tommelise. The Tommelise motherboard only has three integrated circuits (IC's), One Microchip PIC 18F4610 and two Texas Instruments 754410 dual Darlington chips. How simple can you get?

I don't make elegant boards, as you can see. The new one I'm building, however, ought to be a LOT less intimidating that this one.

Building the new Tommelise controller board

Sunday, 4th February 2007 by Forrest Higgs

I'm having to stay up fitting neural nets this evening. That leaves me a bit of free time while these things grind on my dual Xeon workstation. I've been slowly building up the new microcontroller board for Tommelise. I've taken a lot of pics for documentation but I've not the time to write that up this evening, so I'll just show you where I am at the moment.



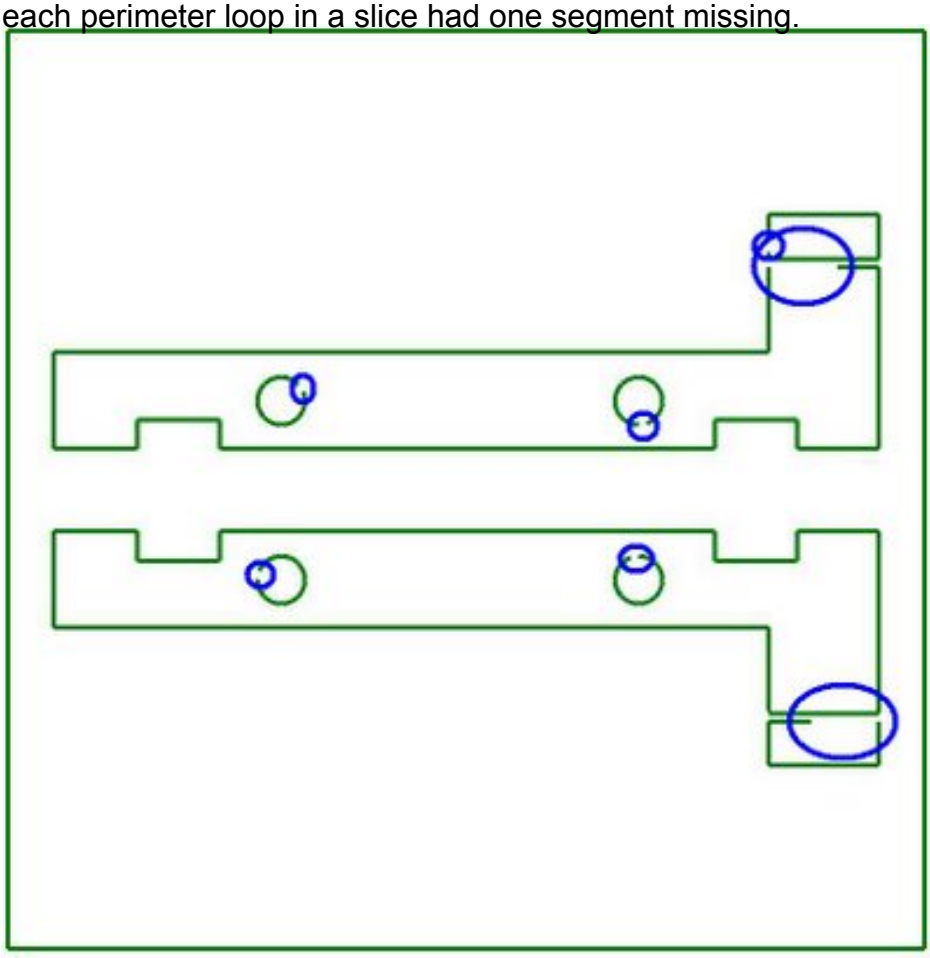
You can see the PIC 18F4610 in the middle of the board and sockets for two 754410 dual Darlington motor controllers at the bottom. The power conditioning voltage regulators and capacitors are at the upper right of the board with a two pole connector for 12v power.
To the left of the PIC chip is a 20 MHz timing crystal that sets its speed.
The little 1K Ohms resistor connects 5v power to pin 1 of the PIC chip and the black and orange lines just to the right of the PIC chip supply grounding and 5v power respectively.
The other two, longer black and orange lines carry serial comms to the two, two pole connectors at the top of the board that are connected to that multicoloured flat cable at. That cable is connected to a comms card that turns PIC serial comms into something my workstation's serial port can understand.
This PIC chip allows for direct USB communications with a PC. I haven't got around to getting that working just yet. Right now I'm more interested in getting Tommelise to actually make something useful than I am on guilding lillies, so to speak, so USB is not near the top of my priorities just now.
Add one power transistor and five more of those blue two pole connectors to attach all the sensor input lines and carry power to the Mk 2.1 extruder and the board will be complete. It's not too complicated a line, even for newbies. It does take a bit of patience and care in order to get it going properly, though. I'll be taking you through all that step by step in the documentation in a few months.
BTW, I've tested the board and it responds to serial comms properly. Now to get the dual Darlington chips hooked up and running the gearmotors. I got tired of wasting dual Darlington chips so I bought some cheap sockets for this board. Tommelise doesn't draw a lot of amperage to run its motors, so I can get away with this, I think. Ordinarily, putting dual Darlington chips in

sockets is not a very bright thing to do.

Chasing bugs in Slice and Dice

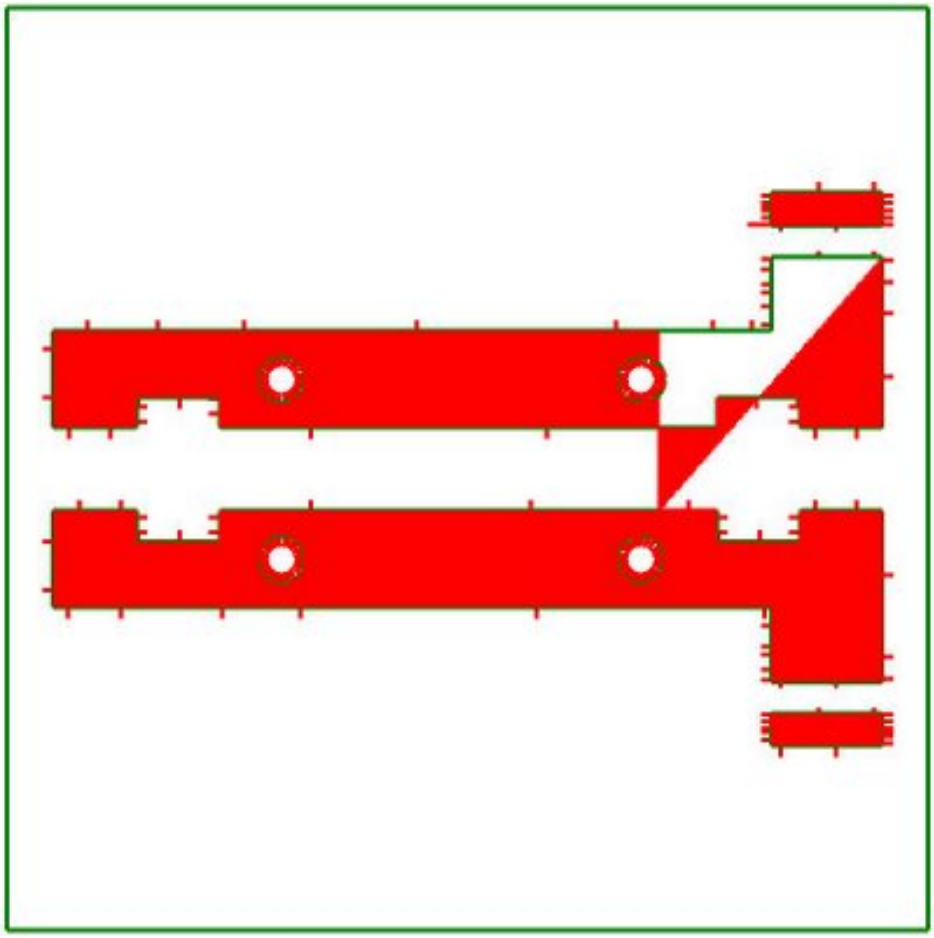
Friday, 23rd February 2007 by Forrest Higgs

When I got the Slice and Dice code running well enough to work withAdrian Bowyer's Mk 2 FDM extruder parts I noticed that on some slicesthings weren't as they should be. This evening I isolated such a sliceand started working my way through the code. The first thing I noticedwas that each perimeter loop in a slice had one segment missing.



I'vecircled the missing segments in blue. With the brute force method thatI am using this doesn't cause a lot of problems. It causes enough,however, that I cleaned that problem up.

I hoped that that wasthe only problem. Sadly, when I ran the Mk 2 polymer pump through awhole slice cycle I found that the fix got rid of most of the problemsbut not all. I was able to isolate another one before it got too lateto continue.



Thelittle spiky red line segments are normal vectors to the perimeter linesegments. Those can be ignored in that they are only a diagnostic tool.The problem is that diagonal line at the upper right of the slice thatbounds a real bit of confusion about what is the inside and what is theoutside of the perimeter. I find it suspicious that it ends at thelower left end on one corner of the starting line segment of anotherperimeter loop. I hope that means something.
I'll have to get back to this tomorrow night.

Still chasing bugs in Slice and Dice

Saturday, 24th February 2007 by Forrest Higgs

I've been at this since 0430 this morning and it's definitely time for a break. I got the perimeter fixed last night. Having that working reliably revealed that there was a bug in the Cramer's Rule routine that got intersections between the perimeter and the scan line. I finally got that fixed a few minutes ago and discovered that the routine that then decides which parts of the scan line are inside the perimeter seems to have a bug. Mind, 98 percent of it is working, but when you do as many calculations as Slice and Dice does that isn't anywhere near okay.

Time to make some lunch. The trouble I'm having brings home the old observation that coding is a young man's game. I'm always amazed at how well an app will run that turns out to be hideously flawed inside when you go looking for that "little bug".

Documenting the Mk 2.1: the polymer pump

Saturday, 24th February 2007 by Forrest Higgs

I got sick and tired of debugging Slice and Dice a few hours ago, sowhen Zach Smith, who is to the point of building up a Mk 2 FDMextruder, asked a few questions about how you build one without a metallathe I took the Mk 2.1 apart and took a few pictures, attached thepics and wrote him a few paragraphs about how I did it.

Once Idid that I realised that I had the basis for documenting part of howyou build a Mk 2.1, so I edited the pics and expanded what I'd toldZach for a less initiated audience. Here is what I've said so far.Eventually, this will go into the documentation section of this websiteafter a few more edits and cleanups resulting from feedback from youguys about the parts that aren't clear.

This is going to be a LONGblog entry, so I'm going to use the Extended Body feature of this blogso that those of you who want to look at earlier entries won't have toscroll half a mile past all of this entry to get to the other ones.

Mk 2.1 documentation section

Sunday, 25th February 2007 by Forrest Higgs

I've reworked the existing documentation about the Mk 2.1 in the Documentation section of this website to include the discussion about the polymer pump. You can take a look here.

Printing circuitry in the low-rent district

Sunday, 25th February 2007 by Forrest Higgs

I'm *always* inspired by Adrian's groundbreaking work. I guess I must see things with different eyes, though. I read his blog entryon working with low temperature eutectic alloys in the a little whileago. It seemed to me he'd gone to heroic lengths to make what was, forpractical purposes, a piece of 3 mm solder. Mind, being Wood's metal it's solder that will melt in a hot cup of coffee, but it's solder all the same.

His saying that he's wanted a filament that he could run through a Mk 2 FDM extruder put me on the hunt, I guess.

Yousee, I know that my low thermal inertia 2 amp extruder barrel ishitting somewhere around 190-210 degrees C (374-410 degrees F). I keepseveral diameters of solder. In regular acid core lead I keep heavy(1.27 mm) and fine (0.9 mm). Recently, I've been experimenting with aRoHS (no lead) compliant acid core solder in a fine grade.

Knowinghow hot my extruder barrel gets I began to wonder whether I could getit to melt in my extruder barrel. Problem was my extruder barrel is 3mm while my solder is 1.27 mm. As well, if I got it wrong it would belikely that I would jam the barrel.

Then I got to thinking aboutthe old 1 amp prototype extruder barrel that I'd built and tested backin late December and early January. Naah, it'd never get warm enough.On the other hand if it jammed I hadn't really lost anything.

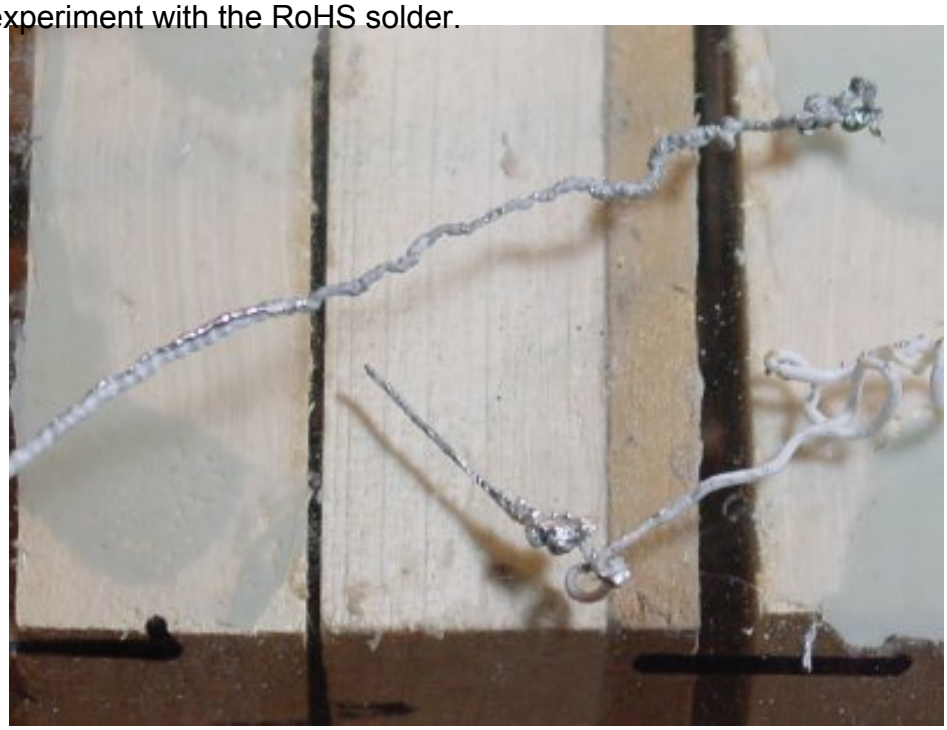
So, one thing leading to another I cut off a couple of inches of the heavy grade Sn63/Pb37 solder and folded it into a plug and stuffed it in the test extruder barrelwhich I'd locked into my vice and fired it up. I had a short length of3 mm HPP to use as a piston so I was good to go. The test extruder hada plug of either ABS or HPP in the end from the last time I used it, soafter the barrel heated up I inserted the HPP filament and began tofeed it into the cold end (~70-80 degrees C) of the extruder barrelwith a pair of pliers so that I wouldn't risk burning myself howeverslightly.

Afterthe plastic plug melted and began to clear as I fed the HPP filamentinto the extruder I began to see droplets of solder coming out with theHPP as you can see here.

Aftera few seconds of transition between plastic and solder the rest of theplug came out in a rush and fell into the HDPE sherbet container I'dthoughtfully placed under the extruder barrel to keep things off of thecarpet. The HDPE surface that they hit was relatively rough and theacid core solder adhered to them slightly.
Here was what was important though. The specific heatof metals is, on average, about 10% of that of organic materials likeplastic. What that means is that dropping small amounts of somethinglike solder on a plastic surface usually doesn't melt the plasticsurface significantly in spite of the fact that the molten solder isfar hotter than the melting point of the plastic. The plastic simplyabsorbs the heat like a sponge and chills the solder instantly.
Here is a closeup of one of the solder splashes that I peeled off ofthe HDPE surface laid on the tempered glass xy working plane ofTommelise.

I got my micrometer out and measured it at a very consistent 0.1 mm inthickness. Basically, I had

a very nice piece of tin-lead foil for mytroubles.
I repeated the experiment with the RoHS solder.



It behaved very differently. It came out in tiny beads with the HPPand in a few cases coated or partially coated the HPP. The two longthreads of solder shown here at the center and left of the pic areactually electrically conductive wires made of RoHS solder. RoHS isgoing to take some more thinking about.
I guess I'll have to head down to the hardware store and see if I canfind some hard copper tubing that can seat 1.27 mm acid core solder. Iwonder if I can talk Adrian into printing me up a Mk 2 parts kit thatcan handle 1.27 mm solder. It would be interesting to see if I couldprint an HDPE blank and then print the traces for a Tommelisemicrocontroller board in solder on it. :-D

Printing with conventional solder

Monday, 26th February 2007 by Forrest Higgs

I gave the extruding solder thing another go before it got too late.This time I tried to lay down a track with a HDPE surface within amillimeter of the extruder head. Keep in mind that I was pumping thesolder with one hand and moving the sherbet lid (my HDPE surface) withthe other.

Still, I did get one reasonable trace (circled in red)

There was enough energy in the solder trace to etch it into the HDPE toa point where it is solidly attached to the plastic surface. When Isay solidly, I mean that I can't peel it off with my fingernails. If I get time in the morning I'm headed down to the hardware store to see if they have electrician's solder (as opposed to electronics solder) in a diameter something like 3 mm.

Buggy as a Texas Summer Evening

Monday, 26th February 2007 by Forrest Higgs

I put some more hours in on Slice and Dice and did extensive texting on the work I was doing. I've arrived at the conclusion that the Slice and Dice concept is as bug-filled as an outdoor Texas Summer evening in a lit yard. It seems like the harder I work on it the worse it gets.

Anyway, I've got to take off a few days to do some billable hours and get the presentation that I'm doing on Tommelise for the San Diego O'Reilly Emerging Technology conference done that's happening on the 29th of next month. :-(

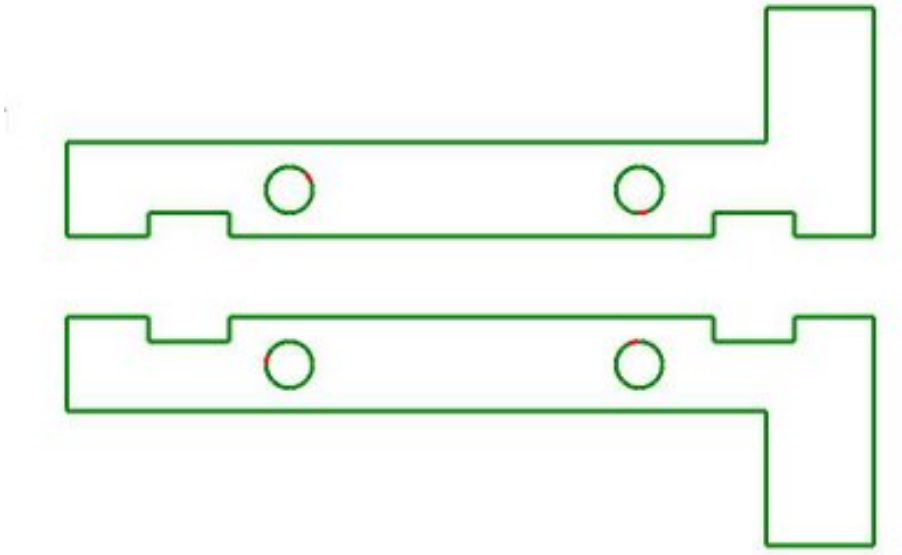
Different eyes

Tuesday, 27th February 2007 by Forrest Higgs

Many years ago in my misspent youth I worked for a IBM for a spellin the late 1960's doing 3D graphics programming. I used vector andscalar products and got pretty adept at solving the hidden line problemusing an even/odd technique that I neglected to publish. Oh well.

Inspite of the fact that I was treating this as a grid rendering problemI was trying to do the maths as vector loops. Vector loops are alwaysleaky at some level of granularity simply because numbers in computersalways have some limits to their resolution.

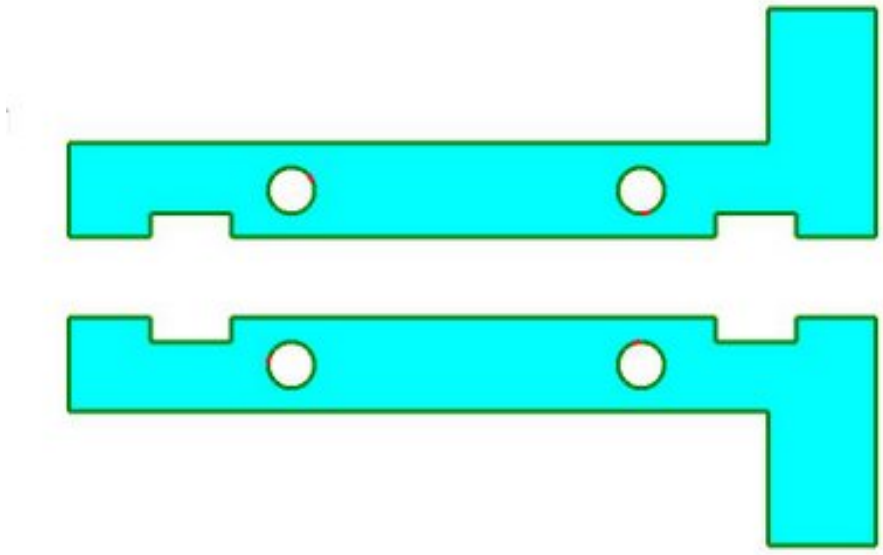
I stepped back and looked at my perimeter slice...



Whereaswhen treated as a vector loop it was inevitably going to leak, if Ilooked at it as an image composed of pixels I had throughfully usedthick enough lines to define the perimeter of the slice that there wasno way that it would leak ... if I treated it as an image instead of anabstract set of vectors.

What I wanted to do was to overlay a grid onto this perimeter and thendetermine which points on the grid were inside the perimeter and whichweren't. I'd then use the grid points inside the perimeter as astarting point to define the paths that the extruder had to take.

Before I got to that point, however, while I have the image of theperimeter what I really needed to do is a simple floodfill like you cando in Windows Paint.



Whatthis means is that I will have to draw the perimeter onto a bitmaprather than my form and use a floodfill routine to do the filling. Ican then overlay my grid over that image and ask what the colour of theimage is at every grid location. Very efficient. Very simple. Utterly reliable.
Gad, I feel so slow at times. I've been hammering this code forprobably two man days without realising that I'd gone back to what Iwas doing in the late 1960's instead of using my head and experience.

Inch by inch

Wednesday, 28th February 2007 by Forrest Higgs

I finally got Slice and Dice to recognise a bitmap of the slice's perimeter outline and I have got it able to return the color of a pixel given it's coordinates on the bitmap (I get those through a mouseclick). What that means is that I can recognise the perimeter of the slice because it is a different color that the background.

I've yet, however, to get it to change the color of a pixel on the bitmap. I thought that this part was going to be hard. I had now idea it was going to be this hard, however.
Oh well, I've got to stop doing this and get that presentation on Tommelise and 3D replication done for eTech. It's due by midnight tomorrow.

Got it finally

Wednesday, 28th February 2007 by Forrest Higgs

Finally found a code patch that lets me color pixels in a bitmap. Now to see if I can incorporate that into my floodfill routine. :-D

Extruding onto foamcore boards

Thursday, 15th March 2007 by Forrest Higgs

I've been using standard 5 mm foamcore of the sort that you mattpictures on for presentations as a base for extrusions. The .8 x 1metre board (yes, those are true metric measurements) It was obviouslypressed and cut to metric standards. A sheet cost me $7 retail.

I can not recommend this material enough. HDPE extrudes onto itwithout warping, shrinkage or peeling. The HDPE sticks very firmly tothe paper sheathing. You can peel it off with a mat knife, however,without a lot of drama and without seriously damaging the surface.

When I recently replaced the foamboard on the xy surface and was able to cut across some of the HDPE tracks that I extruded across the surface. Those that Ilaid down on the surface caused no perishing of the foam beneath thepaper. Those that I smeared across the paper with extruder head hadsignificant recession of the foam beneath the tracks. Places where Iparked the extruder head above the paper close by while it was hotshowed cratering in the foam beneath the paper.

Absolute vs relative measurements

Thursday, 15th March 2007 by Forrest Higgs

For the last few days I've been having trouble with gearmotor overruns causing cumulative errors in putting down a layer of extrusions. I first tried to correct for the overruns within the firmware. The fact that Oshonsoft Basic really doesn't have integers (negative and positive whole numbers) in the proper sense of the word but rather only cardinal (counting numbers) I had to go to considerable trouble to get them to behave like proper integers.

After not making too good a job of that I took a look at the size of the overruns and realised that they were far larger than they had any business being. I went back through my firmware archive and discovered that a dodge that I'd made to make the gearmotors move faster had introduced the larger overruns. I uncommented a few lines of basic and the size of the overruns decreased by over a magnitude.

All the same the overruns, however small, still cause cumulative errors. They're smaller, but they're still there. I want Tommelise to be as good as I can make it and this situation is simply not acceptable.

Right about midnight it hit me that I'd missed a rather large bet.

I have shaft encoders on the three axes on Tommelise. There is absolutely no reason whatsoever that I shouldn't be able to keep track of absolute position on the xy working plane. The way I set up the "moveto" command I use relative positioning. In hindsight, that is just stupid, especially since I can access absolute positioning without a lot of bother.

Mind, absolute positioning using shaft encoders is not perfect, but it is a LOT closer to perfect than what I have now.

I will have to be making some changes in Slice and Dice, the Control Panel and the Tommelise firmware to take advantage of absolute positioning. I think that it will be worth it.

Coding absolute positioning

Friday, 16th March 2007 by Forrest Higgs

I'm going to have a third run at this. The blog database seems to be hiccoughing a tiny bit.

I've got about halfway to having absolute positioning for extrusion on the xy working surface. I'm having to do it with 4 byte cardinals instead of 2 so I decided to move some of the calculation out of the interrupt routine on the PIC and into the control panel on the pc. That entailed me changing the format of the moveto command that controls the printing, so the job is a bit bigger than I had initially hoped it would be.

A few minutes ago I misaligned a PIC in it's socket and shorted it. The 18F4610 is pretty forgiving of this kind of abuse, but it eventually gets to be too much after a while. I took it as a sign that I was getting too tired to do this finicky sort of work and decided to call it an evening.

May have cracked the problem

Saturday, 17th March 2007 by Forrest Higgs

I was having a problem extruding the same perimeter defined by a large number of short extrusion paths verses one defined by only a few. The perimeter defined by the large number of short extrusion paths was always coming out significantly larger. This, of course, made for all sorts of frustration because the cross-hatching was defined with long extrusion segments, so there was always a substantial misfit between the perimeter and the infill.

I switched the firmware over from relative positioning to absolute and quickly found that that had little, if any effect on the problem I was getting pretty much the same position reporting for short segment perimeters as I was for long. You didn't have to measure to see the difference.

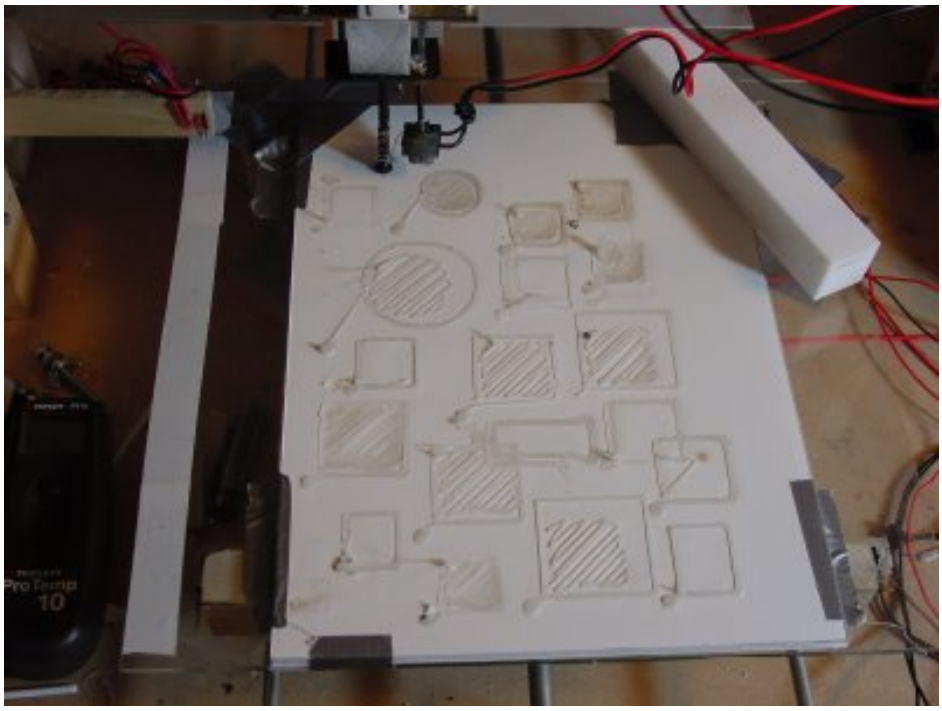
I started braking the gearmotors from the 754410 after the completion test said to stop rather than just letting them coast and the problem seems to have gone away.

There is still lots of work to do. I may be able to try multiple layers this weekend, though, if things go well.

A testament to Scots-Irishness

Saturday, 17th March 2007 by Forrest Higgs

I haven't been publishing much this last week, mostly because I'vebeen fighting with the Tommelise firmware to make it do what I want itto do. I've done a lot of hot extruder tests, as you can clearly see.



I've been primarily working on the perimeter/cross-hatching infillmisfit problem for the past few days as I'd mentioned earlier. I seemto have cracked that one for now so now I'm going on to get a degree ofcontrol over the extrusion rate. As you can see at the top of thefoamboard I was running some of the extrusion exercises both too hotand with the extruder head far too close to the board. I'll be workingwith those parameters and with trying multiple layers over the next fewdays.

Trying something different

Sunday, 18th March 2007 by Forrest Higgs

It hit me late last night that I am probably making the xy gearmotorcontrols a lot more complicated, and thereby unreliable, than they needto be.

Basic maths

At 5v the *Solarbotics GM8* hits 70 rpm at 5 volts. It has a gear ratio of 143:1
70x143 = ~10,000 rpm
It's probably rated at 12,000 but loses a bit in gearbox friction.
In any case that means that for every revolution that the gearbox turns, the motor turns 143.
Given that we can get 128 pulses out of one channel of the shaft encoder, we are getting ~1.1 motor revolutions per pulse.
Right now, I am using 3/8-24 studding (threaded rod). That says thatI'm needing 24 turns to traverse an inch. Converting to mm we have.
(24 turns/inch)/(25.4 mm/inch) = (24/25.4) turns/mm = ~0.945 turns/mm
Further, we have ~121 pulses/mm
To get 0.1 mm "accuracy" we'd need roughly 12 pulses.
The motor would turn 12x143 revolutions = 1716 revolutions to create 0.1 mm of movement.

Control

Now the minimum amount of information that is needed to controlmovement in the xy plane for this number of pulses is 4 bits, 2 todetermine the direction that the motors need to be turning and 2 to saywhether each gearmotor is on or off.
In essence, we need to create process 50 bytes of data for everycentimeter extruded. The .xml file being transmitted to Tommelise ismuch more compressed than that. The Tommelise firmware, however, needsto create this data stream from the .xml file and transmit it to the754410 controller chip.
Heretofore, I have been controlling the gearmotors on a 1 millisecondtime pulse and then checking the encoder pulsecounts at arbitrariallyselected times. Right now I am running the gearmotors for 10milliseconds and then checking the pulsecounts after another 10.
The simplist strategy, though by no means the fastest one, would be torun one gearmotor at a time for 0.1 mm. Turn it on till you get acount of 12 on the shaft encoder and then turn it off and brake it,then go do the other motor.
I'm going to try that out for a little while and see if I can get this interrupt code in the firmware under control.

Testing the new idea

Monday, 19th March 2007 by Forrest Higgs

I pretty much mapped out the logic for the simplified xy moveto command. While it is easier to conceive of and much more logical, at least at first blush, it could take a long time to debug using Oshonsoft.

Because of that, I've moved the code over into Visual Basic and am testing the logic there. I don't have to burn the program onto a PIC or use the IDE for testing if I do that. The IDE is good but it's a bit slow to run for such an interrupt scheme.

Problems with the z-axis shaft encoder

Thursday, 22nd March 2007 by Forrest Higgs

I seem to have got a short or loose connection in my z-axis shaft encoder. This is just a time consuming annoyance. I may have to build up another little shaft encoder if the AS5035 chip has gone south on me. :-(

Tooling up for eTech

Monday, 26th March 2007 by Forrest Higgs

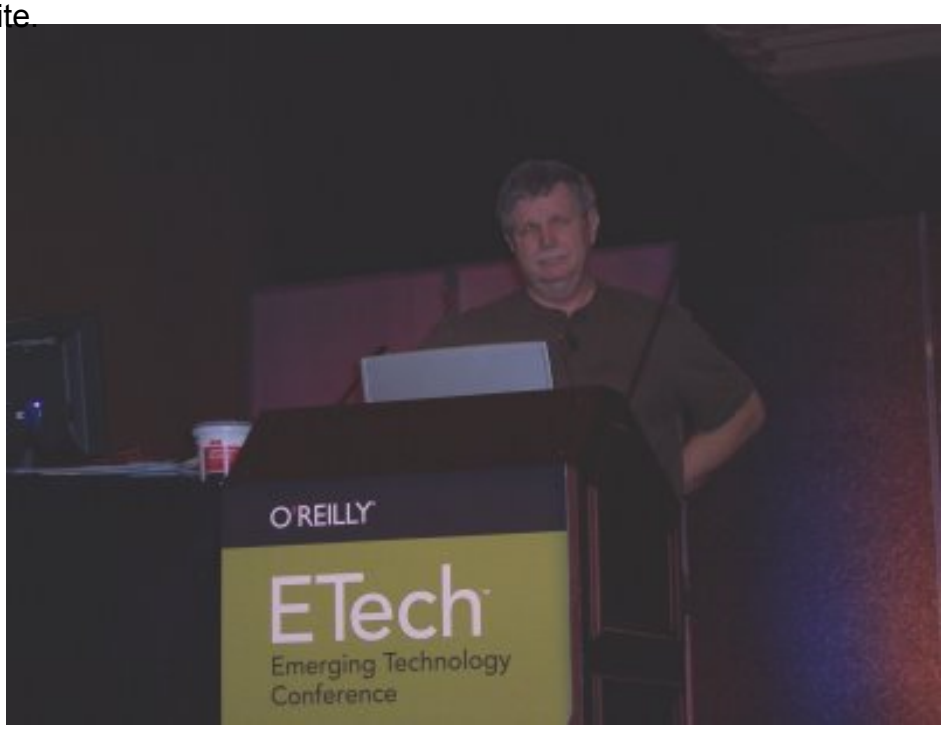
Well, I didn't get as far along at printing things with Tommelise as I'd hoped before my eTech presentation this Thursday. My z-axis encoder crashed which put me back a few days. I'm hoping to either have it fixed or to have built another one before tomorrow evening. It's a long, seven hour drive to San Diego. Mercifully, my son is on Spring break and will be doing some of the driving. Wish us luck and safe journey. It would be appreciated.

ETech 2007

Saturday, 31st March 2007 by Forrest Higgs

My son Adriaan and I just got back from San Diego. The presentation atETech Thursday afternoon went VERY well. The audience was between75-110 or so. The high ~$1,500-odd registration fee seemed to serve asa good filter for filling the audience with highly qualified, verymotivated people from just about everywhere.

Adriaan tried totake pics of the presentation with his digital camera. Unfortunately,the hall was quite dark for the presentation which made his attemptsdicey at best. This one is, however, my personal favourite.



I especially liked the vampire eyes thing. There were several businessand technical media folks there. One American who writes for aJapanese business journal had much better digital camera equipment thatwe did and did extensive photography of the presentation. I'm hopingthat I will be able to borrow a few of his pics for our archive.

Imade the pitch for self-replicating printers, RepRap and Tommelise andwe got good applause and very good questions in return. Afterward wehad about a dozen of the people who'd attended stay over. Most of themhad already been on both the RepRap and the Tommelise websites and oneBrit there had even been down to Bath to meet Adrian.

There werepeople who were selling 3D printers, design engineers, productengineers and the lot. We've got a couple more product design firmsthat are probably going to be hammering for Darwins.

I alsoheard an excellent and interesting rationale for why a professionalfirm would want a self-replicating printer. It goes like this. They'repaying $4.50/cubic inch for filament and if they don't use the filamentthat comes from the vendor they have their warrenty voided. They resentthat a LOT. As well, they'd like a printer that they could trydifferent things with and fix if it broke. What we're working on fitsthat to a "T". As it is, the machines they have typically can only dealwith one kind of plastic. The product design firms would like to doprinting with a machine that wasn't a black box like the commercialmachines.

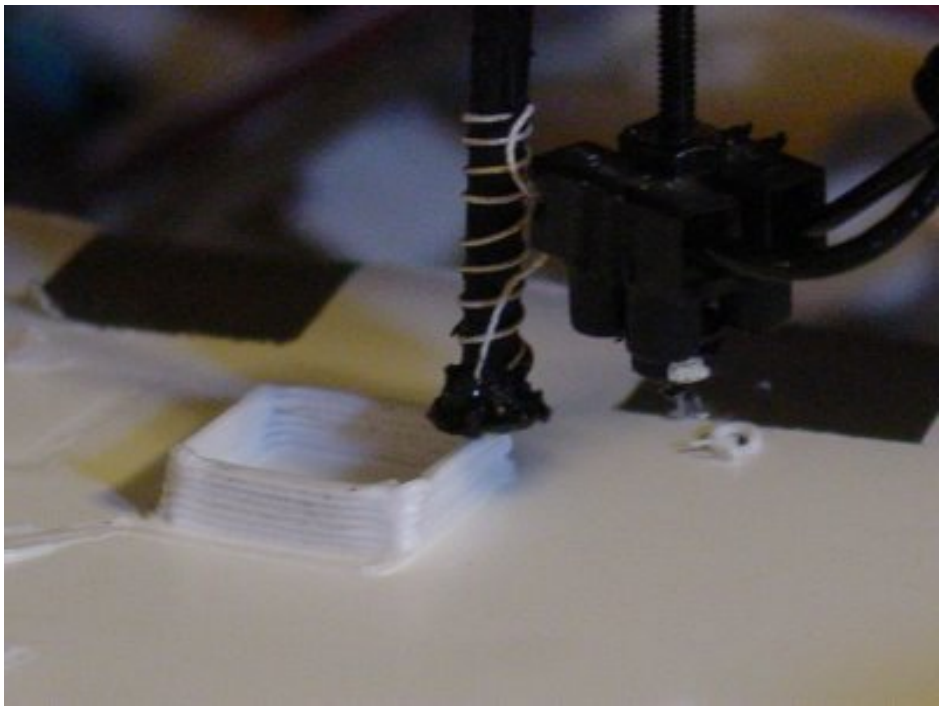
One thing on the reprap site that really impressed thethe people who are using commercial 3D printers was when both Vik andAdrian and Ed made shot glasses. You apparently can't make waterproofglasses on a stratasys. I heard about the tendency of stratasys printsto delaminate on the z-axis again, too.
Anyhow, the talk was well-received and O'Reilly was happy with it so we'll probably be asked again.
I noticed that several blogs had picked up on my talk. We got pretty good reviews.

The world class retard finally "gets it"

Saturday, 21st April 2007 by Forrest Higgs

It's seemed for the past few days that just about anything I did tohelp sort things out with Tommelise's x-axis problem just made thingsworse. Today, I even started having trouble with the y-axis.

Itwas the y-axis starting to have troubles that finallytipped me off a few minutes ago. I fixed the problem and both x and yaxes started performing beautifully, not perfectly, but beautifully asyou see here...

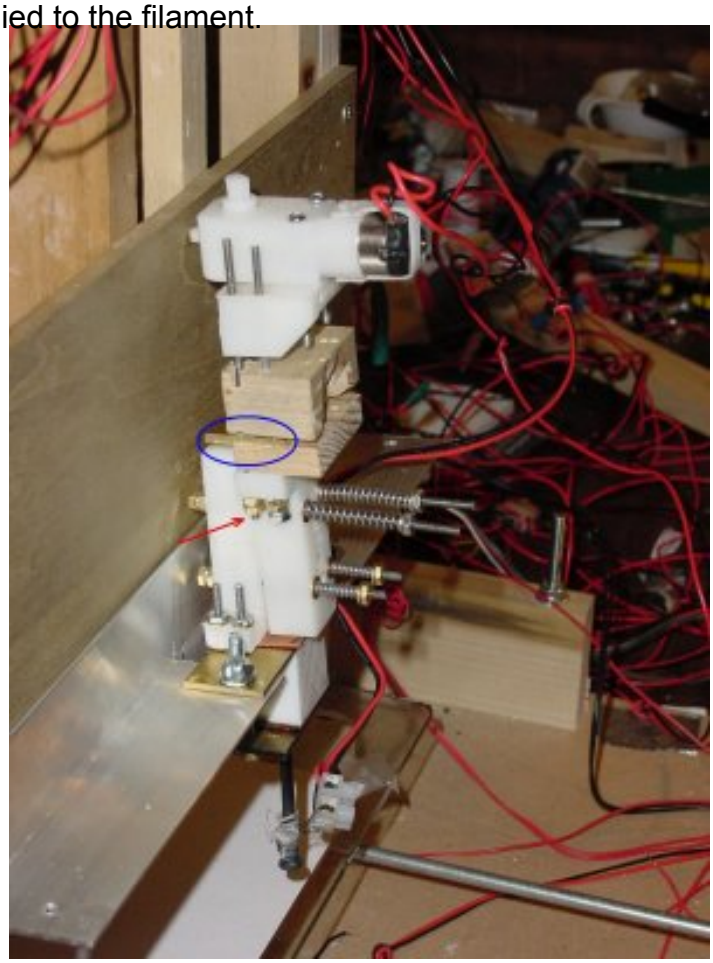


You'renot going to believe what it was. If I had any shame, I'd be embarassedto admit what the problem was. Fortunately for you, I have no shame.

Doing a maintenance breakdown on the Mk 1 AEM extruder

Saturday, 21st April 2007 by Forrest Higgs

Over the past few days I'd noticed that it was getting harder andharder to get the Mk 1 extruder to start pumping HDPE. I'd also noticedthat the gap (at the top of the polymer pump between it and thefilament guide was almost closed (gap indicated with red arrow). Thisis where most of the force is applied to the filament.



Tomy knowledge about the only way that that could happen was for thebrass bushing (circled in blue) that keeps the threaded rod that doesthe pumping in place to have been taking some wear. I've reportedbefore seeing a black dust that might have been brass powder in the topof the polymer pump. When I broke down the polymer pump and removed thebrass bushing I knew for sure then exactly where it was coming from.

More on the Mk 1 AEM extruder

Saturday, 21st April 2007 by Forrest Higgs

A few days ago someone asked me for more detail about how the Mk 1AEM extruder goes together. I'm a little hesitant to say a lot becauseit's not really finished in any meaningful sense. All the same, somemore pictures and description would probably be useful. As I'd saidbefore the Mk 1 AEM is basically a RepRap Mk 2 with some hacks andupgrades that I fancy are a bit easier to do and which enable it topump HDPE at 150-200 degrees Celsius rather than CAPA at 80-100. Oneof the problems that I faced in building up the Mk 2 kit that Adriansent along so long ago was that I had worn out the gearmotor that hewas kind enough to send along with it. While I was able to slot in aSolarbotics GM2 into the mounting the drive shaft of the GM2 was bothshorter and more robust that that of the gearmotor it replaced. I triedto alter the coupling to accomodate the GM2 and ruined it in theattempt.

As well, I did not have ready access to M5 studding andhad to replace it with 1/4-20 American threaded rod. Finally, I didn'treally have the skill to mill down the threaded rod to seat properly inthe bushings that I needed to make. In order to get around thesechallenges I adapted an early design that Vik Olivier had sketched out.

What to do next

Sunday, 22nd April 2007 by Forrest Higgs

Just figured out that the polymer pump contains about 24-25 cubiccentimeters of polymer and that it's going to take about 9 hours toprint it. Keep in mind that I'm running this thing about as slow as Ican while I learn how to run it. It's going to be going a lot faster asI get more confident with what I'm doing.

Right now a single axisspeed is running 1.7 mm/sec and I'm using single axis prints so that Idon't have to upgrade the control software to regulate polymer flowrates. Using diagonal infills will crank that speed up to 2.36 mm/sec.Shifting from 3/8-24 to the much more common 3/8-16 threaded rod willkick that speed up to 3.5 mm/sec.

Punch list

Sunday, 22nd April 2007 by Forrest Higgs

First, I need for the control panel to pull the extruder head away from the print when it's finished printing.

Done.

Second, I need forthe extruder head to lift a few millimeters when it finishes one linesegment before proceeding to the next one, proceed there and then setback down to extrude. This should be fairly easy.

Done.

Third, I need tosee if I can get away without pausing when I get to the end of a lineof extrusion. This is a firmware issue. I think that I will try thisfirst in the morning.

Firmware reprogrammed. Testing.

Fourth, I'venoticed that features in a print job that consist of a bunch of shortextrusion lines tend to melt down when they're printed simply becausethe hot extruder head perforce stays over the same area much longerthan would otherwise be the case. I need to see if I can confect analgorithm which will pull the extruder head away for a particularlength of time so that.

This is for later.

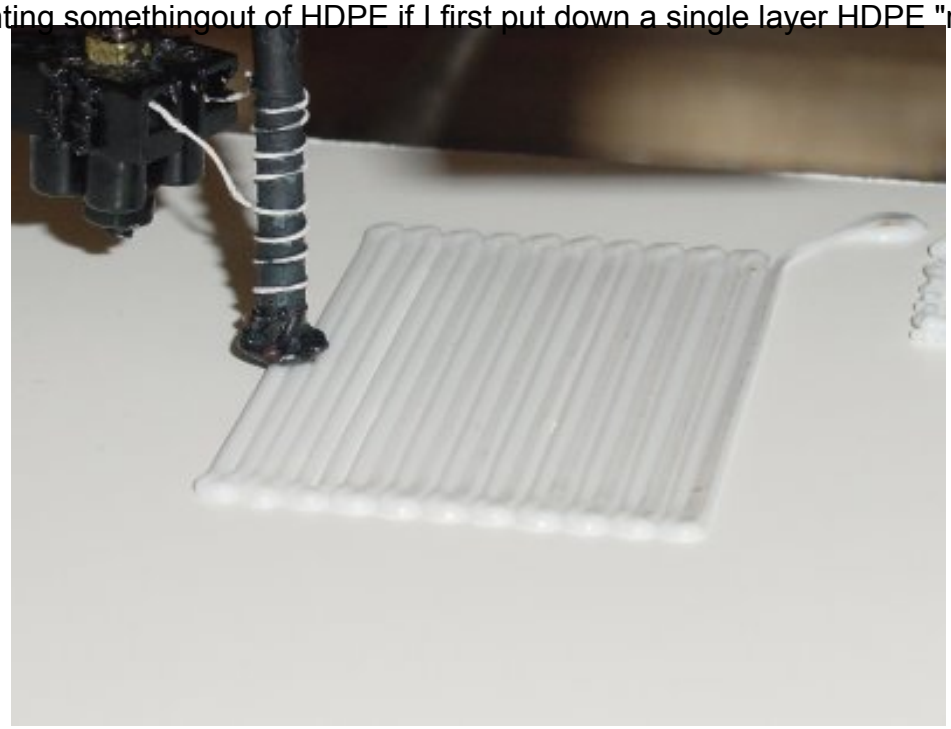
Fifth, I need tofigure out a way to interrupt a print job, recover the absolutecoordinates where it was interrupted and then save back the remainingprint instructions so that the job can be completed at a later date.That would allow me to shut down the printer at night and start itagain the next morning. I'm presuming that it's going to be some timebefore I'm comfortable leaving Tommelise to its own devices unattendedfor long periods of time.

After doing the first and second tasks I have an idea of how to do this that may not require a lot of reprogramming.

Tweaking the x-axis

Wednesday, 25th April 2007 by Forrest Higgs

You'll recall some time back that I discovered that I could get ridof a lot of the curling and distortion when I was printing somethingout of HDPE if I first put down a single layer HDPE "raft" like this.



Whatyou may not have known about this raft is that the extrusions run inthe +/- y direction. Notice how even the x boundary is. Heretofore,I've been noticing x-drift when I tried to build something vertically.This last weekend when I was upgrading the control panel I wroteanother button on the control panel to generate a raft where theextrusions ran in the +/- x direction.
This morning I got around to using it.

Serious printing

Thursday, 26th April 2007 by Forrest Higgs

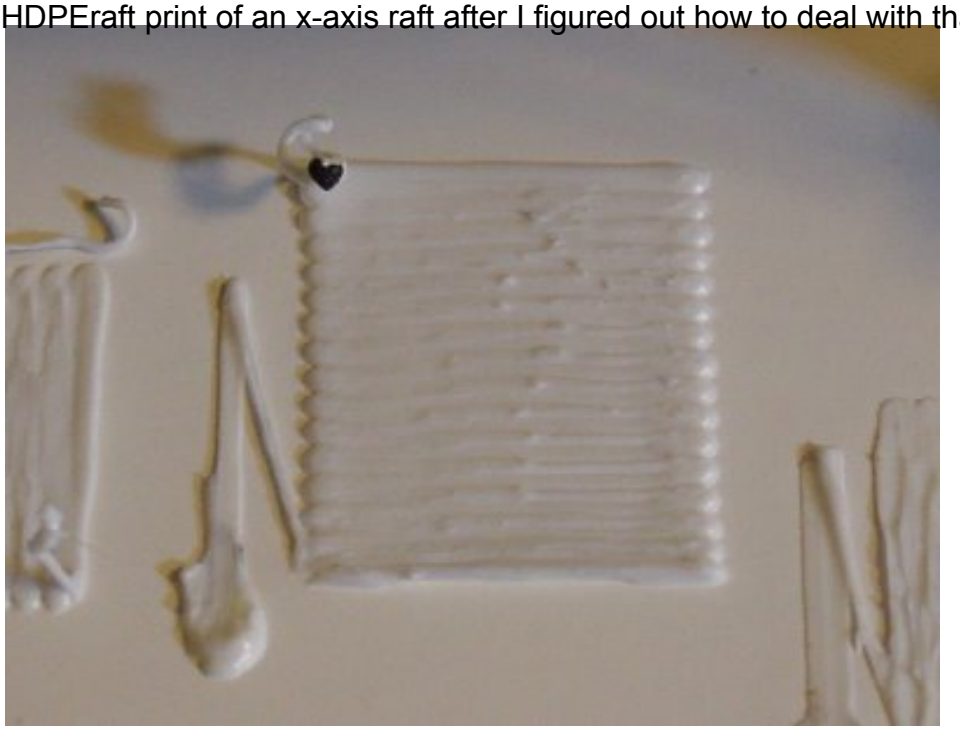
My first large order of 3 mm HDPE arrived today, 15 lbs worth @$4/lb material + $2/lb shipping.

This was good, because I have just about half a pound of HDPEcontinuous filament left. I also dropped by the art store and got twomore sheets (12 ft^2) of foamboard at $0.75/ft^2). I shouldn't have torestock for several months.
Now I can do some serious debugging and printing.

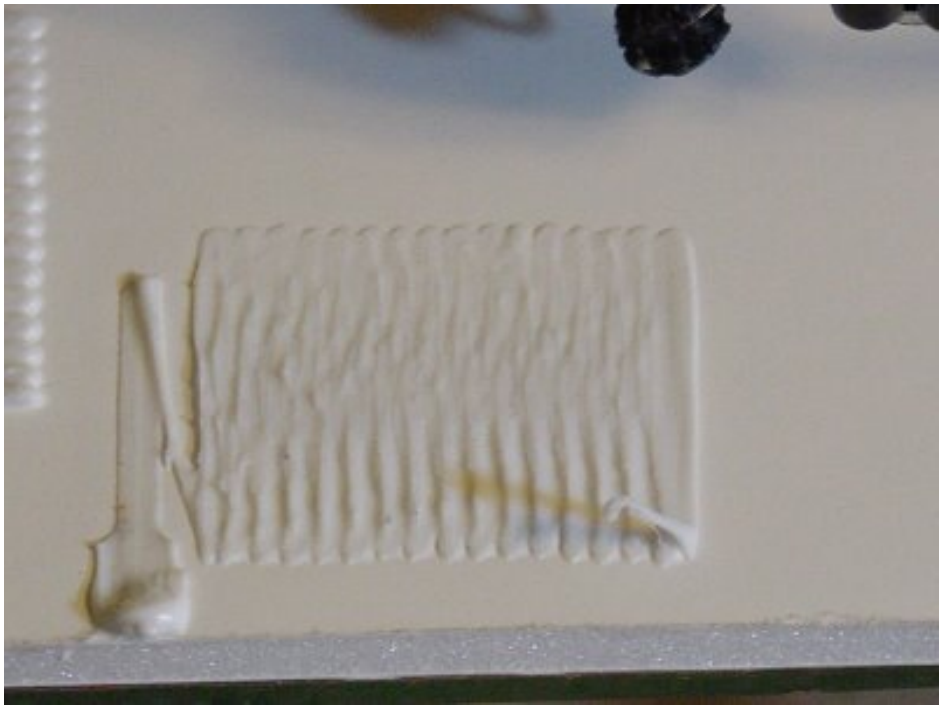
Ready to print

Friday, 27th April 2007 by Forrest Higgs

It's a long story and a story for another time, perhaps. I finally gotto the bottom of the problem with the x-axis drift. It was a firmwareissue with how I structured the interrupt routine. Here you see a HDPEraft print of an x-axis raft after I figured out how to deal with thatproblem.



As you can see the x boundaries of the raft are both straight and parallel to each other. As well, you can see a y-axis raft.

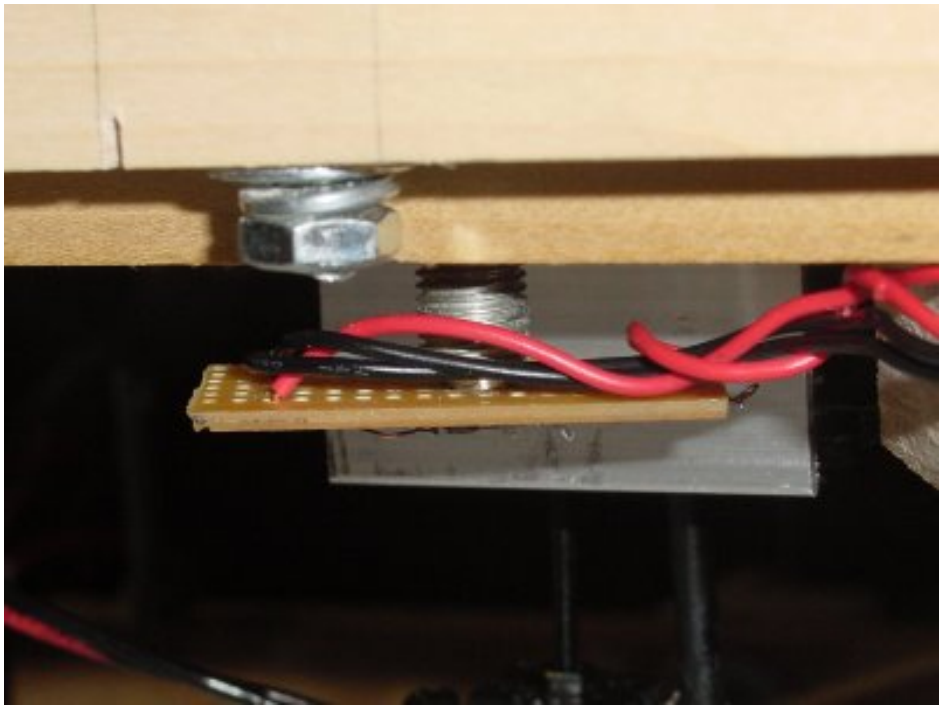


Again,no problems. There are no fudges or correction factors in the code anylonger. The firmware/hardware interface is sound.
We are ready to do some serious printing.

Fixing the z-axis encoder

Saturday, 28th April 2007 by Forrest Higgs

The AS50305 chip that does shaft encoding for the z-axis is ina very awkward location. The little board that holds it comes out ofalignment quite easily and when it does I lose z-axis control ofTommelise's positioning system. When I was tapping the z-axiscontroller once per layer this was no big deal. Now, however, I have touse the z-axis every time I jump from one line segment to another.

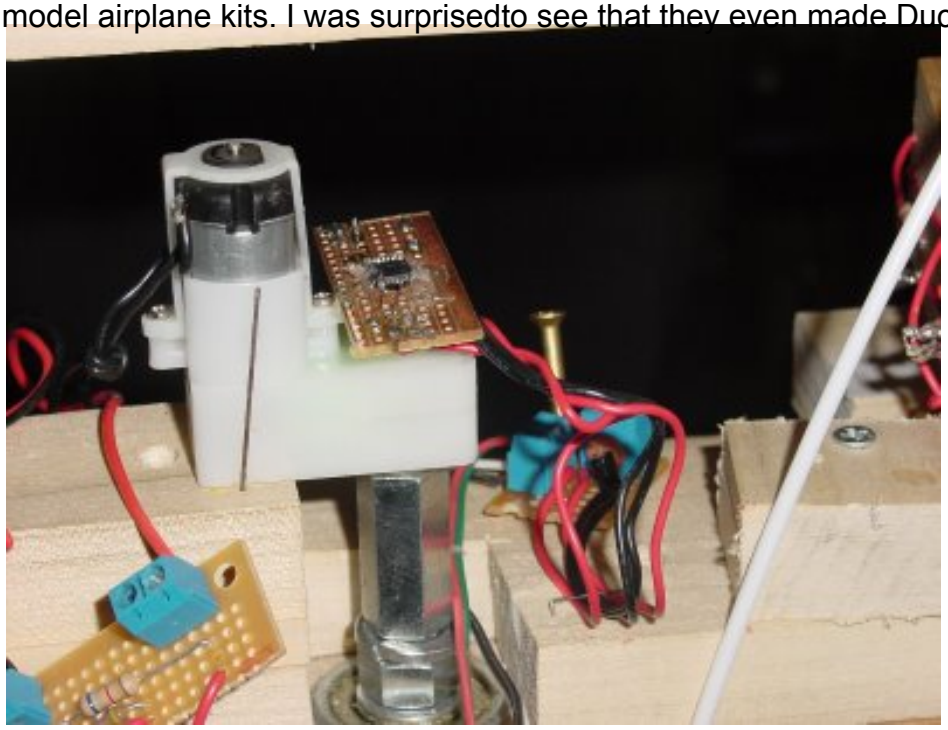


As you can see it is under the x-axis gantry and behind the xzpositioning plane. That means that if it gets out of alignment you'reworking with a flashlight and your fingertips and praying that youdon't pull one of those fine little filaments that connects the AS5035to the stripboard fragment that it's mounted on.

Memories of Duco Cement

Saturday, 28th April 2007 by Forrest Higgs

I finally got down to the hardware store and bought some JBWeldlike Zach suggested for tacking down nichrome. I looked at how long ittakes JBWeld to set, however, and also bought some Duco Cement. Ihadn't seen that since my childhood when I used Duco and Testor'scements to glue together plastic model airplane kits. I was surprisedto see that they even made Duco any more.



I set the screw terminal board down with a brass screw and set thecomms and power wires down with a paper stapler. The fit on theencoder magnet is perfect. I tested it and it pulses smoothly. Theredoesn't seem to be any problem with interference from the gearmotor.
Now gravity works to keep the encoder chip on the magnet instead of pulling it away.

New z-axis shaft encoder built and tested

Thursday, 10th May 2007 by Forrest Higgs

I finally got the time to complete the new z-axis shaft encoder board. I had one hiccough on the wiring and then it started working beautifully. This chip is in fine shape and I should be able to include the z-axis pulses in the interrupt scheme rather than using an A/D converter.

Interrupt-driven shaft encoding on all three axes

Friday, 11th May 2007 by Forrest Higgs

I've upgraded the z-axis control firmware to do interrupt-driven shaft encoding. It looks really, really good and very repeatable.

Lost weekend

Monday, 14th May 2007 by Forrest Higgs

I've been shifting over the firmware and control panel software to dealwith a positioning strategy that includes using the limits detectorsfor periodic resetting of the axes' start points. I got that prettymuch running in all three axes, but a bug crept in. One of theabsolute position variables (for the y axis) began to show up with aweird number in it at what first appeared to be random intervals.

I did the usual procedure of putting lots and lots of serial writestatements trying to track down the chunk of code that the variable wasgetting set to the random value in. That pretty much took up Saturdayand Sunday.

Sunday night I noticed that for the XML file I was trying to print thatthe number was always the same. That tended to cast doubt on thisbeing a random occurance.

At that point I got to wondering if perhaps I had a dimensionedvariable that was getting an out-of-range setting. Indeed, that turnedout to be what was happening.

The dimensioned variable into which I write command lines imported viathe serial link from the PC control panel was dimensioned to 10. WhenI upgraded the call for extruding lines in the xy plane I extended thelength of that command line from 10 to 12.

Thus, what was getting put into the y-axis absolute position variablewas the last byte of the y coordinate and a plus (+) sign to turn theexturder on. Once I extended that dimensioned variable things startedbehaving pretty much properly.

I'm presently print a raft and the first section of the polymer pump. Wish me luck.

Printing the polymer pump

Friday, 18th May 2007 by Forrest Higgs

I've been pretty quiet these last few days what with getting thefirmware/control panel interface working for the new axes resetauto-calibration scheme that Adrian suggested that I try. It's allpretty much working now and appears to be incredibly more reliable thanthe mass of patches and fudge factors that I was using before.

Withthe positioning problems largely solved I took on the problems ofmaking solid objects. The shot glass didn't really count since it was ashell, not unlike the flask that Vik down in Auckland just made of PLA.I've taken on the making of a polymer pump as the first solid objectprint job for Tommelise.

Image not found.

Over the past few weeks I've discovered that there is a lot more involved in printing solid objects than I'd suspected before.

Five layers

Friday, 18th May 2007 by Forrest Higgs

I printed a raft plus two layers of the polymer pump before I went tobed last night. This time the raft was very solid (95%/30%). Thefirst two layers went smoothly. At the third layer it became apparentthat I had the layer thickness a bit low for a (95%/27%) flow rate, sol jacked it up to 0.67 from 0.5 mm. I kept fiddling with the layerthickness for layers four and five and finally abandoned the printafter I managed to do layer five successfully but too messily to suitme.

I am going to let the print that I just did cool for a few hours to seeif I get any warping or curling. So far there isn't any corner curlingat all, but you never know. After that I will try a new print withfixed settings this time of 95%/30% for the raft and 95%/27% for thepump with a layer thickness of 0.62. The pump will have 21 layers at that thickness.

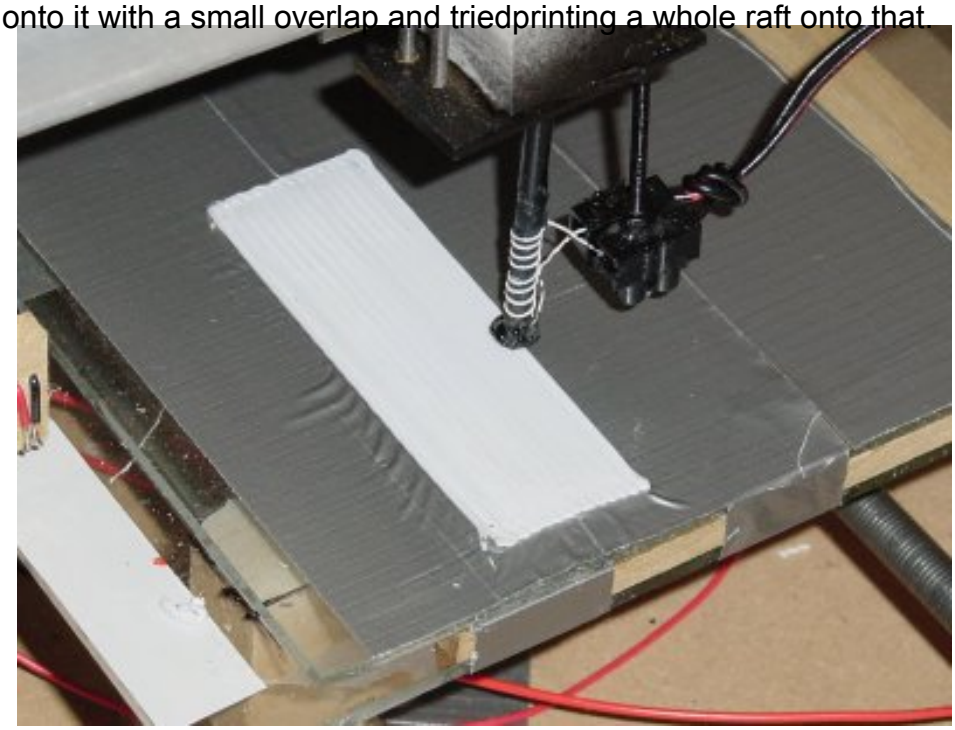
Dropping foamboard

Saturday, 19th May 2007 by Forrest Higgs

After printing the first five layers of the polymer pump thismorning I finally had to admit to myself that foamboard was not goingto work as a surface for printing HDPE objects on. While you can printa raft and a layer or two on it the repeated heating of the foamsubstrata warps the board to where you simply don't have a flat surfaceworth the name any more.

This was not a happy denouement for me.While I haven't exactly cornered the market on foamboard I have boughtseveral pieces and it hurts my Scots-Irish soul to know that I'm notgoing to be able to use it. It was also not a very convenient time forthis sort of development to manifest. Duringspare moments during the day I tried poplar sheet again. HDPE failed toadhere to it again, of course, so I roughened the surface with coarsesandpaper and HDPE *still* wouldn't stick to it. Ithen tried extruding directly onto glass with very similar results.Extruding onto float glass, even heavily sanded float glass only yieldsa very nice HDPE tape which wasn't exactly what I was attempting tocreate. This was not something that I needed to happen right now.

Scouringmy memory for a tricky approach I remembered that several times I hadaccidently printed HDPE onto the duct tape that I used to securefoamboard onto my xy working plane. I already had a piece of floatglass attached to the working plane so I simply put several strips of1-7/8ths inch wide duct tape onto it with a small overlap and triedprinting a whole raft onto that.



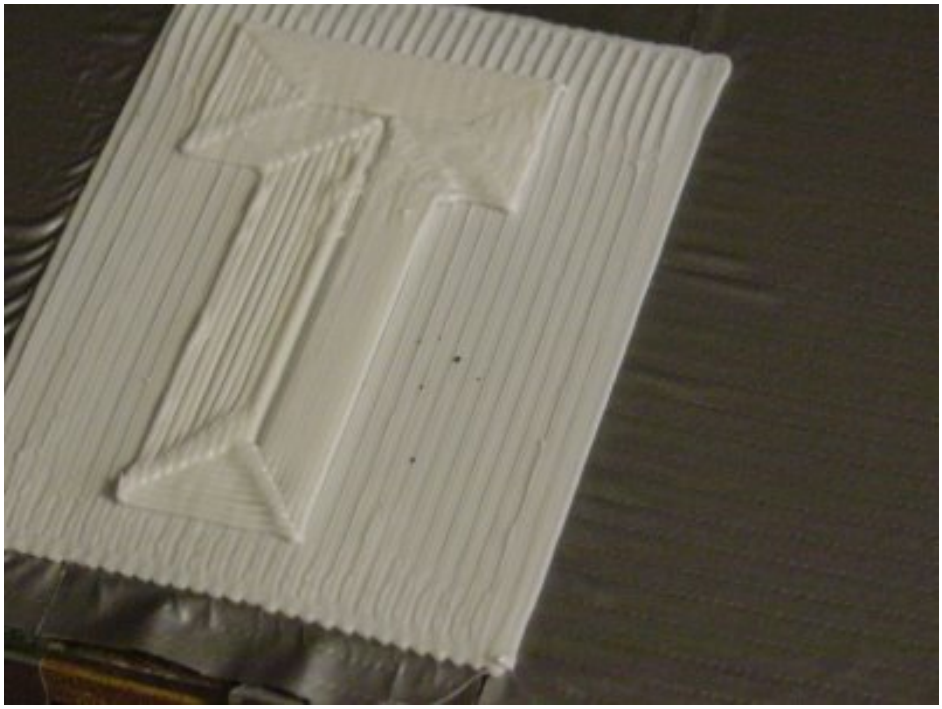
That worked. On the left-hand side of the duct tape layer you can seesome distortion patterns where the raft has shrunk a bit on cooling. Iam going to print a whole raft and then leave it over night to see ifit stays flat and then try printing the polymer pump onto it in themorning.

Another support material miracle?

Saturday, 19th May 2007 by Forrest Higgs

WhileI was doing some practice prints last night to get used to printing onduct tape something very unexpected happened. Recently, I've beenworking to get the polymer flow rate coordinated very closely with theextruder head velocity. Last evening I accidently managed to do thatwith a heretofore unencountered precision in a peculiar situation thatproduced some very interesting results.

I had laid down an HDPEraft and was trying to print a layer of the Mk 1 polymer pump on top ofthat. I had had trouble getting the layering interval right and managedto get it about 0.75 mm higher than it should have been. Ordinarily,that would have just made a mess of squiggled extrusions instead of aprint.



Allthe same it made for a rather messy print as you can clearly see. Ithad been a long day and I had demonstrated that I could successfullyprint on HDPE, so I nearly just packed it in and went to bed. As I wasshutting Tommelise down, however, I reached over to turn off theworktable light and in doing so noticed a very consistent shadow underthe right hand side.

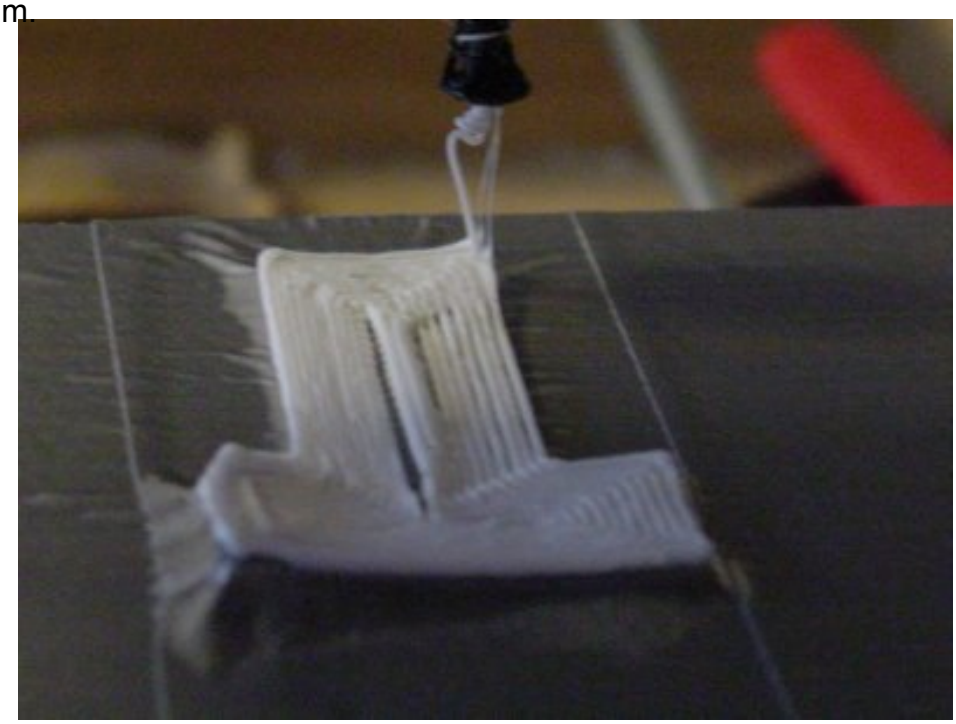
I'm very used to seeing that sort of thingwhen the extrusion doesn't adhere to the raft. In this case, however,the shadow was long and even rather than short and humped. I hadn'tseen something like that before, so I used a small metal rule to see ifI could determine the extent of this gap.

If I was intrigued before I was astonished when the rule easily slid fully 7 mm under the extrusion.

Duct tape fails

Saturday, 19th May 2007 by Forrest Higgs

The dream of being able to print objects directly onto duct tape without a HDPE raft remains a dream.



Thisis the result of printing two layers of the polymer pump directly ontoduct tape. The first layer worked fine. The second caused thesituation you see here.

Blog comments down for a little while...

Wednesday, 23rd May 2007 by Forrest Higgs

Hi all! A few months ago, the little picture of letters and numbers that you have to read and type in that blogs use to keep spambots out stopped working. I checked with the Serendipity people and it was working everywhere but on my ISP. I shifted over to moderating comments and that worked fairly well for a while. Unfortunately, spam bots can make canned comments and point you at their websites that have nothing to do with 3D printing. You haven't been seeing them because I moderate all the comments and delete all the spam.

In about the last two months, however, more and more spambots have found this blog and the time I've had to spend deleting spam in moderation mode has been getting bigger and bigger. Today, I counted up and discovered that I am getting anywhere from 20-50 spam messages per day. Enough was enough.

I contacted my ISP and explained the situation to them. They support Serendipity and didn't realise that this was happening, so they are trying to sort out the problem. With a little luck in a day or two I'll have my little numbers and letters back and can drop comment moderation. In the meantime, I have to leave the captchas (what they call the little numbers and letters) on so that the ISP people can try to solve the problem. That effectively keeps anybody from making comments.

I'm hoping that this won't take too long and apologise for the inconvenience.

Back to slicing and dicing...

Tuesday, 29th May 2007 by Forrest Higgs

I had to stop work on Tommelise about two weeks ago. A consultingcontract that had been sputtering along for the past three monthssuddenly caught fire when it was noticed that the deadline for contractcompletion wasn't getting any further away. Life has been interestingsince then.

Yesterday, after I realised that I had nailed down the loose ends downtight enough that I would be able to deliver on time and budget I wasable to get back to devoting a few hours to Tommelise again.

I had been working on 3D printing and dealing with shrinkage and warping issues. To do that I had been making alterations to the XML transfer file that is generated by Slice and Dice to serve as input for the Tommelise Control Panel. While that is a handy feature it is not a very efficient way to generate print files from scratch. Because of that I decided that I had been putting off several upgrades that I needed to put into Slice and Dice so that I could design parts in the Art of Illusion 3D modelling software again.

The reason that I had got away from using Slice and Dice was because the STL files that Art of Illusion generates are at such a fine resolution (STL files basically describe a solid as an envelope of triangular patches) that when I took a slice of the solid the perimeters of that slice tended to be described by line segments only 0.1-0.2 mm long. While Tommelise can extrude paths at that resolution it is not really good at extruding line segments that short. Basically, what I needed to do was connect all those tiny segments into bigger line segments.

When I wrote Slice and Dice I tried to collect tiny line segments during the slicing process. This was not notably successful. Currently, I am writing code to collect them after the XML transfer file is created. I am hoping that that will be a more fruitful approach. I will be reporting on progress in this effort iover the next few days.

Rafts sticking again

Friday, 22nd June 2007 by Forrest Higgs

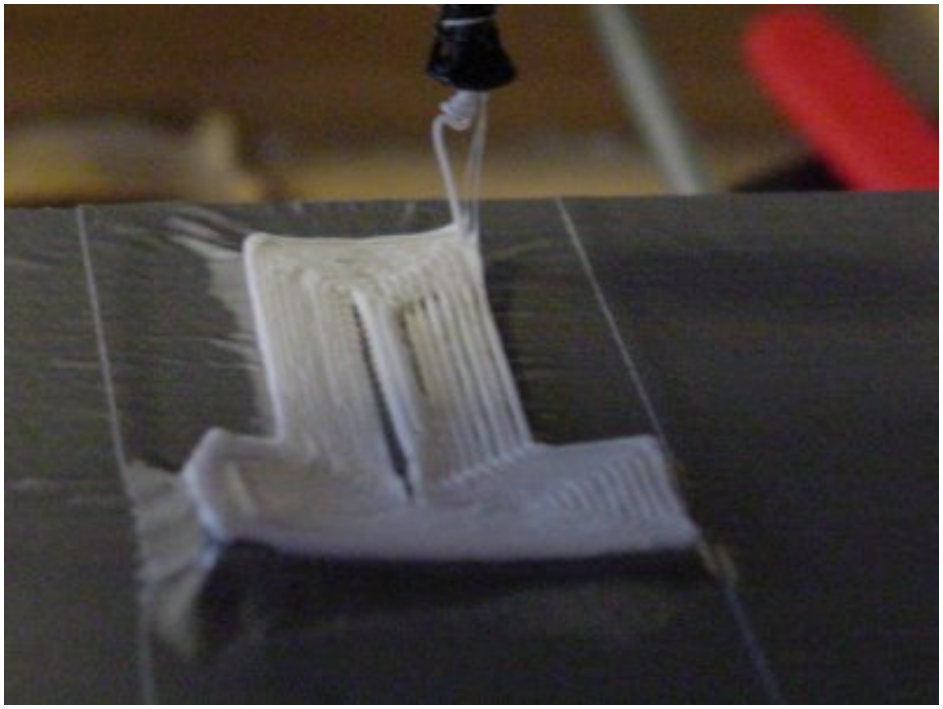
My suspicion was right. I was printing just a shade high. I had been doing test prints with infilled slices where the starting height of the print is close but not nearly as critical as it is when you are putting down a raft to print on. Once I started taking good care to determine the exact level of the top of the foamboard they stuck nicely at a amperage/flow rate of 85% (1.7 amps)/35 (gearmotor setting).

I had a little trouble with kinking of the filament since I refurbished the extruder barrel yesterday. I finally unkinked the filament and things seem to be back to being reliable again.

Printing HDPE at room temperature

Friday, 22nd June 2007 by Forrest Higgs

I've printed several pounds of HDPE know and feel that I can sharea few things about my experiences with printing this plastic out of amodified Mk II.
Cutting to the chase, it seems to be a lot likeCAPA only more so. It's an engineering plastic and melts at a lothigher temperature than CAPA. It also has a thermal expansioncoefficient that is significantly higher than CAPA. All the curlingproblems that you've seen Vik have with CAPA I've experienced with HDPEexcept that I can modestly say that HDPE curls with considerably morelan. This picture gives you an idea of what I'm talking about.



Whenyou create differential cooling by printing hot plastic on top of coldyou encounter some awesome internal stresses. I've printed objects inthe high dozens now and this behaviour was causing me considerabledistress. I came close several times to deciding that you just couldn'tprint anything meaningful with HDPE at room temperatures.

It may be

Saturday, 23rd June 2007 by Forrest Higgs

It may be that I don't have so much a problem with figuring the layer thickness as I do with deciding how far to lower the extruder in my software. Ah well, another bug to chase. At least this one looks like an easy one for a change.

Z-axis "bug" "fixed"

Saturday, 23rd June 2007 by Forrest Higgs

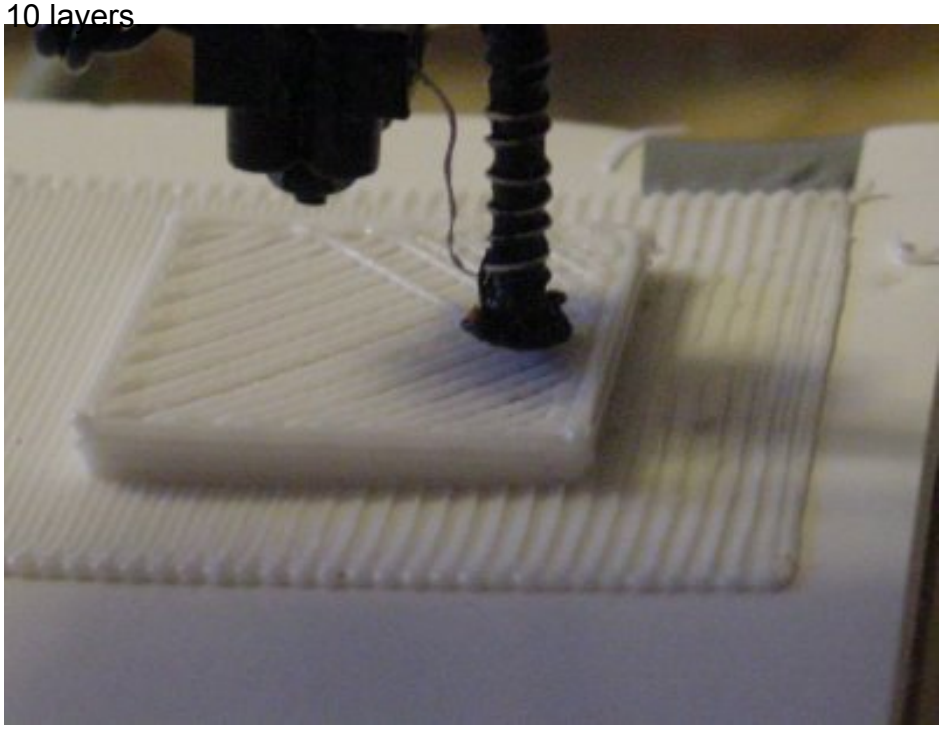
The z-axis "bug" turned out to be an improvement to the code that I made over a month ago that I forgot I'd made. Ordinarily, in the XML print file that the Tommelise Control Panel uses the declaration of a new layer automatically advances the z-axis one layer upward. I had changed that to read the layer number directly from the XML file instead of automatically advancing. That enables Tommelise to print several different print jobs concurrently at different levels of completion.

Once I understood that a simple numbering of the layers in my test XML files put things right.

Ten layers with no curling

Sunday, 24th June 2007 by Forrest Higgs

What a lovely present for my 60th birthday! Tommelise just printed 10layers of HDPE on an HDPE raft with no curling or warping. From thelooks of it there is nothing to stop me from printing another 10 layers.



Justa few statistics. The object printed was a 40x40 mm square on a 70x70mm raft. Each layer consumed about 800 mm^3 of HDPE and took just at 12minutes to print. The average translation speed of the extruder headwas 2.2 mm/sec. The infill density was set to 50%.

The sides of the square prism are vertical. There is a little blobbing at the corners but nothing outrageous.

Ithink I've got the knowhow now to print HDPE successfully. The HDPEraft seems to provide a stable platform that the print can adhere towhile Adrian's suggestion some time ago to reset the origin after everylayer appears to have got rid of the drift problem that plagued me forso long. No fudge factors were used.

Late last night I was stillhaving a little trouble with the layer spacing. I finally realised thatthe MZ command that positions the z-axis was using 16 bit directinteger transfer, which caused the same problems I'd had before withthe xy axes "moveto" command, viz, when I encountered a value of 125 inone of the two integer bytes it was interpreted as a "}". I upgradedthe MZ command this morning and the problem disappeared.

Afterwards, I did the ten layer print. No problems.

The50% infill basically creates an openwork rick of HDPE threads allowsair to flow into the solid and lets it cool quickly. It is also lesssensitive to the hot extruder head being around.

I didconsiderable software fiddling to stagger the laying out of thediagonal infill threads so that they had more chance to cool while theextruder was trying to print smaller details in a layer. Thisstaggering also had the unanticipated benefit of sticking the infillmore firmly to the outside perimeter of the layer.

Anotherunanticipated benefit is that doing a 50% infill effectively doublesthe printing speed of

Tommelise when you look at it on a volumetricbasis.
My next tasks will be to get a firmer grip on theturning on and off of the extruder head so that I will have an evenstartup to a layer print and not get threading when I skip areas whereholes are supposed to be in a layer while infilling.

Something curious

Sunday, 24th June 2007 by Forrest Higgs

I let the 10 layer print sit for most of the day to see if it would warp or curl. It didn't. Just for fun I decided a little while ago to print another ten layerson top of what I did this morning. When I tried that I discovered thatthe 10 layer print had got a bit taller. It was enough taller that itcaused trouble when I tried to print at the next level up there wasn'tany room to speak of for the layer to be put down. That would suggestthat the 5.5 mm thick print had picked up possibly another 0.55 mm inheight or so.

I'm not quite sure what to make of that just yet.

An update on "no curling"

Monday, 25th June 2007 by Forrest Higgs

Late last night I cut the test print out of the foamboard that it was printed on and trimmed off the excess HDPE raft.

While the top surface stayed flat there was a downward bulging on thebottom surface to an extent of about 1-1.5 mm over the 40 mm span ofthe square prism. This bulging pushed into the foamboard and was notapparent until the print was removed.

The print was structurally rigid to a point where no deflection couldbe induced manually. Floating the print in a dish of water confirmedthat its density was running about 0.45-0.5 grammes/cubic centimeter.

A casual attempt was made to see if the bulge could be relaxed byreheating the print in a toaster oven. While this experiment wasbroadly successful it must be noted that the infill structure collapsedas a result within the print as a result of the heating.

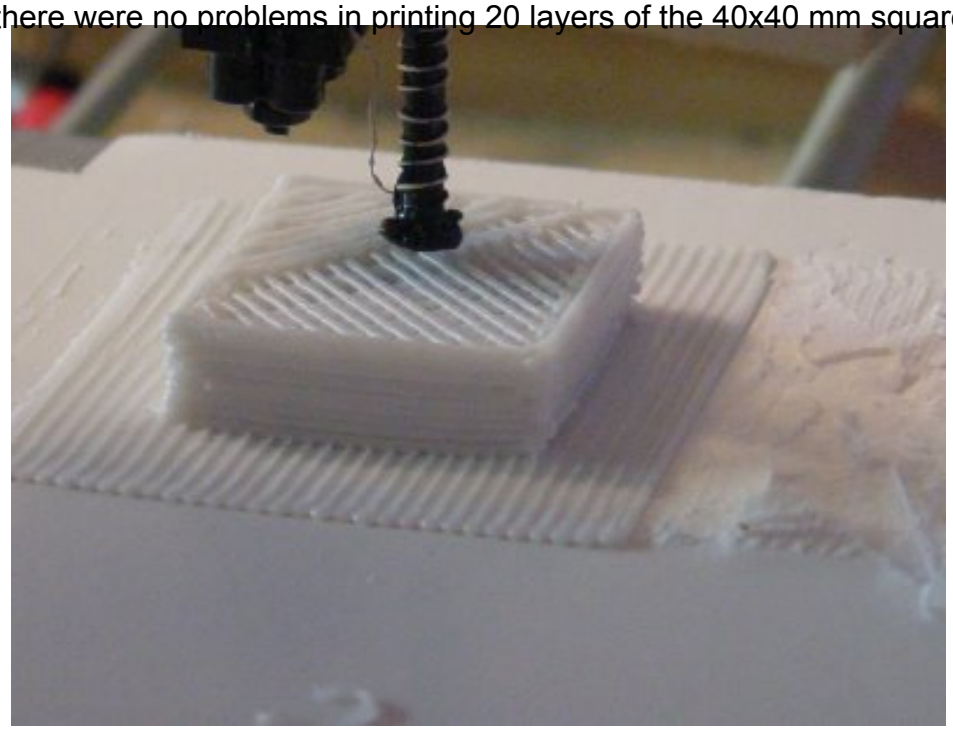
The extent of the bulge observed with a 50% infill is considerably lessthan the curling observed with previous prints using a 100% infill. The extent of curling in those often exceeded 5 mm over similar spans.

Given the encouraging results using a 50% infill, I plan to do severalprints with lighter infill percentages in the range of 25-33%.

Twenty layers

Monday, 25th June 2007 by Forrest Higgs

As I expected, there were no problems in printing 20 layers of the 40x40 mm square prism.

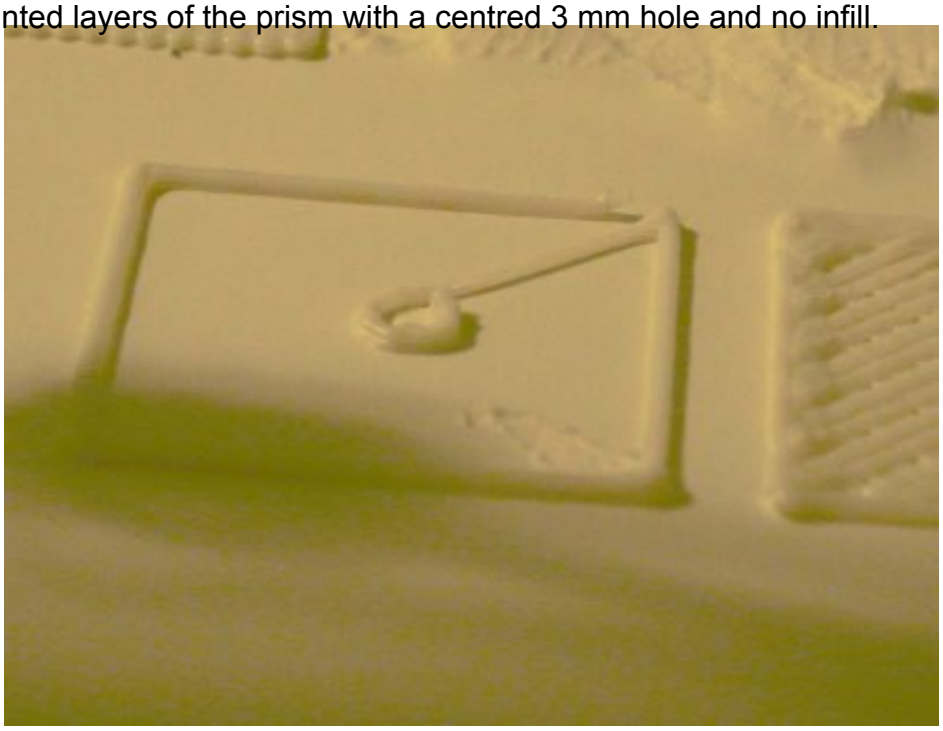


Brief statistics; 40x40x11 mm prism, 50% infill, 3.25 hours print timeincluding 70x70 mm raft, 11.5 cm^3 of HDPE printed with an averageextruder head transition speed of 2.2 mm/sec.
I think that I can go on to printing useful things now.

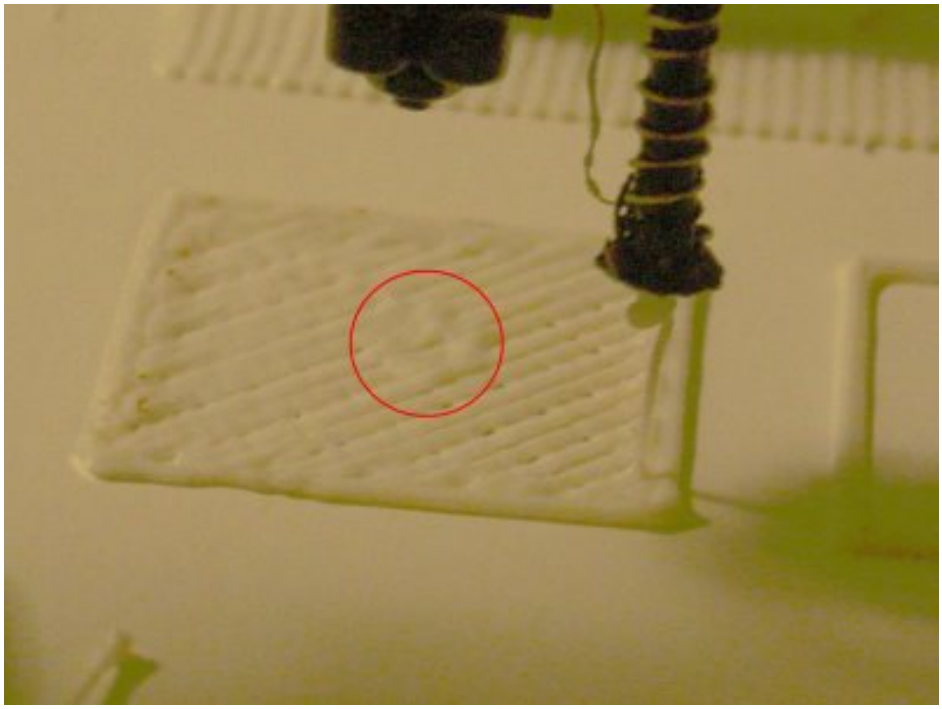
Making a hole in it

Wednesday, 27th June 2007 by Forrest Higgs

Now that I know that I can print thick objects I can startaddressing other challenges in 3D printing using Fused DepositionModelling with HDPE. Dealing with a 0.8 mm HDPE extrusion and apositioning system which can position at about 0.1 mm but which needs arunning start to get rolling makes the printing of small features a bitmore of a challenge than it might otherwise be.
I knew fromprevious work that I'd eventually confront these issues, so I'm notparticularly upset or despondent. They are, however, hard work.Mercifully, I have the XML formatted transfer file between the Sliceand Dice software and Tommelise's Control Panel. That lets me tweakobjects without having to rewrite the Slice and Dice code every time Iwant to try something. That has been a godsend.
Ultimately, Iwant to design and print a better polymer pump for my Mk 1 extruder.One of the things that it would be nice to have in the pump is aligned3 mm holes for the built-up bolts that are spring-loaded to keep thepump firmly connected to the HDPE feed filament. I decided to see if Icould set a 3 mm hole in my test square prism.
Here are two printed layers of the prism with a centred 3 mm hole and no infill.



Thehole models as an octagon. You can see that the beginning of the hole'sboundary suffers from the lag time between when you turn on the polymerpump and start tracking the perimeter and when you get HDPE coming outof the extruder orifice. Obviously, I need to make my extrusion startupfirmware code a bit more sophisticated.
Where things get nasty, however, is when you start doing infill.



You can see that I get good welding between the hole's perimeter andthe infill tracks. You can also see that moving the extruder head overthe hole tends to blur the hole with residual HDPE dribbling out of theextruder. In truth, there isn't much dribble with HDPE. It doesn't,however, take a lot of dribble to nearly completely obscure a featurelike a 3 mm hole.
While it is easy enough to drill out the holes with my Dremel handtool, it would be nicer if I could get a better grip on the flow out ofthe extruder barrel. That looks to be my next firmware project.

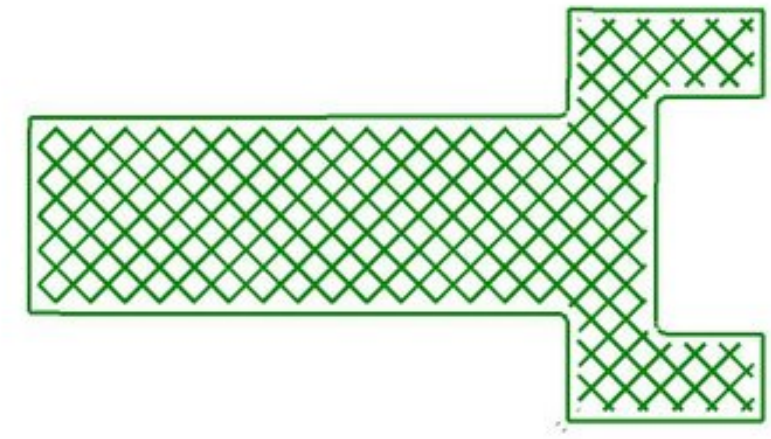
Printing the night away

Saturday, 30th June 2007 by Forrest Higgs

After demonstrating that I could print 20+ layers with infill inHDPE I decided to take a crack at printing a polymer pump again, thistime starting from Art of Illusion rather than building an XML transferfile from scratch.

Heretofore, Slice and Dice has been aonce-through routine that simply takes a STL file of an object that youwant to print and slices it at a given thickness and creates a STLprint file that can be used in the Tommelise Control Panel. While itgives a graphical display of what is going on while it is processing Itypically haven't got the patience to sit down while it grinds through30-40 slices.

Since I haven't been trying to do anything toocomplicated printing out a layer or two to check to see if anything isamiss has been no big deal. With more complicated object, however, youare talking about 15-20 minutes/layer. That's slow. What I needed is asimple graphical method of displaying a print layer-by-layer. I'm nottrying for anything as grad as what Adrian is doing in Java, just a 2Ddisplay of a slice or several slices so that I can see how the infillis fitting together.



Hereare two overlain slices of zone two of a polymer pump that I designedin Art of Illusion. This cheap little bit of code is letting me spotproblems in a layer without having to spend half an hour watchingTommelise print. Once I can see where they are it is a relativelysimple matter to edit them out of the XML transfer file.

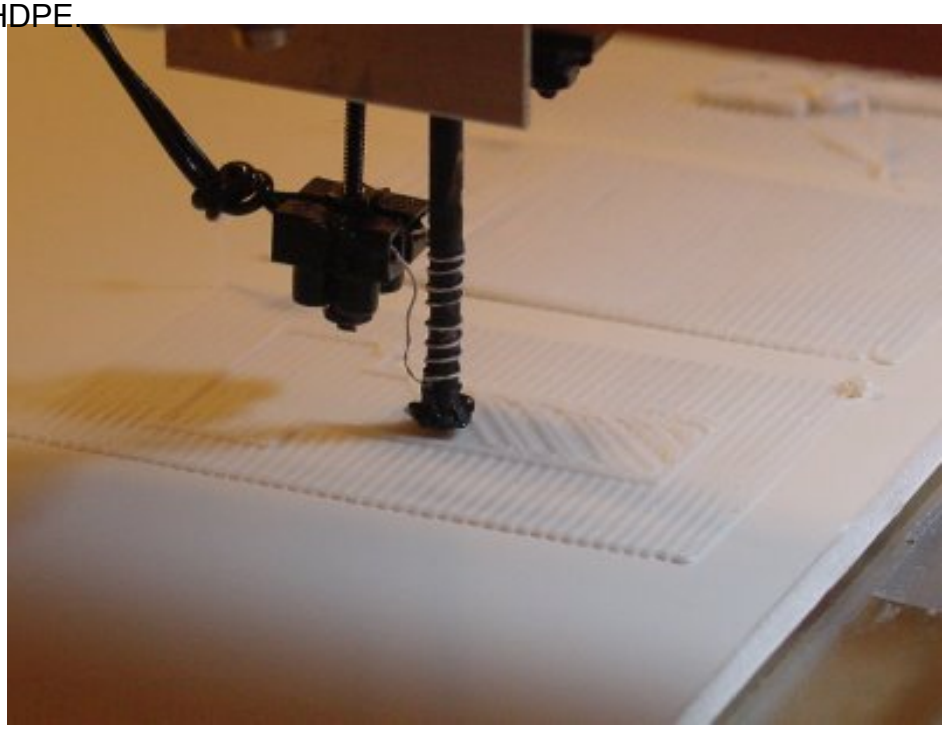
Can you see the problems in this view?

Printing a polymer pump

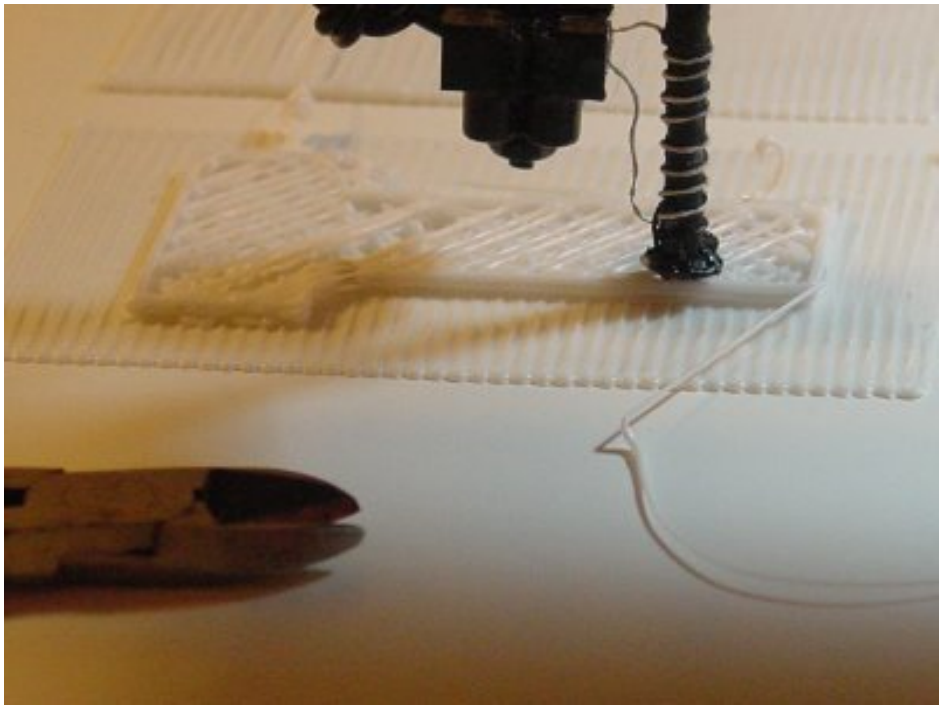
Tuesday, 10th July 2007 by Forrest Higgs

I finally got Tommelise to the point where it was worthwhile attemptingto print something useful. I'm trying to print a polymer pump for theMk 1 AEM extruder as a first project. It took several days to get theinfill right after designing the pump in Art of Illusion. I'm using aloose infill of about 33% and will not be treating the top surface toclose the volume.

If this works it is going to be very crude. That's all right, though. This is all about developing knowhow with HDPE.



So far I've printed the raft and am just about through printing the base level of the pump. Printingat 33% infill presents some interesting problems. The extruded threadtends to cool rather quickly. This makes welding to the rest of theprint matrix a bit dodgy at times and causes unanticipated troubles.The biggest one is that the thread will not weld just where you want itto and gets dragged out of position giving you a sort of "drunkspider's web" infil as you see here.



Overall, however, even with this sort of infill nastiness you get a remarkably strong printed object.

Aside from the "hairy" appearance of the print that Vik has reportedearlier from the tendency of the extruder to dribble at times when itisn't extruding, there is another behaviour that I hadn't seen untilthe last few days as I approach the end of the two pound roll of HDPEfilament that I am currently using.
Towards the end of a roll the filament tends to be rather tightly woundand carries a memory of its shape within the roll as it goes towardsand into the extruder. This can cause feed problems if you don'trotate the remaining roll regularly. Even then the Mk II and the Mk IAEM that I am using has a difficult time eating kinky filament.

Definitely a McAllan's 12 year old single malt evening...

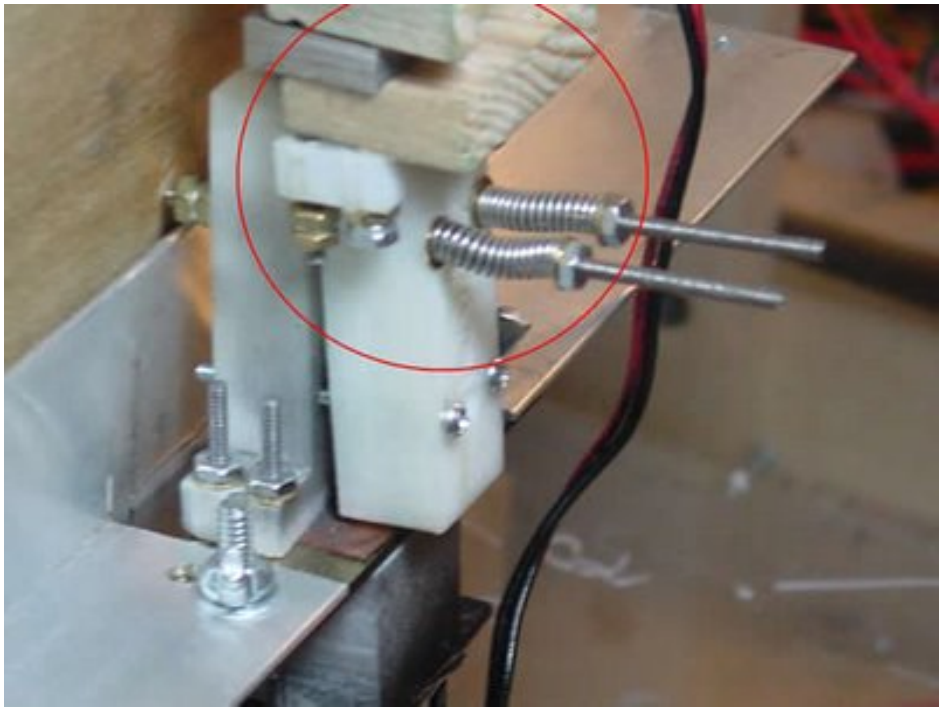
Tuesday, 10th July 2007 by Forrest Higgs

Back in March I pretty much had Tommelise running. It had taken meabout three and a half months to design and build the thing. Almostimmediately I was able to print a 150 ml shot glass of sorts and toastmy success.

Little did I know at the time that it was going totake me another three and a half months to print anything useful withit. Neither did I have the slightest idea how much pain and sufferingwas going to be a part of those months, either.

Well, it's done now.

Beinga congenital idiot, I wanted to go Vik one better and do a part of mypolymer pump first off. You can see here what I had in mind. I did somemodifications to Adrian's excellent Mk II design to let it work withHDPE and my own perversities. I had to use several chunks of wood tomake the Mk II into the Mk I AEM extruder.



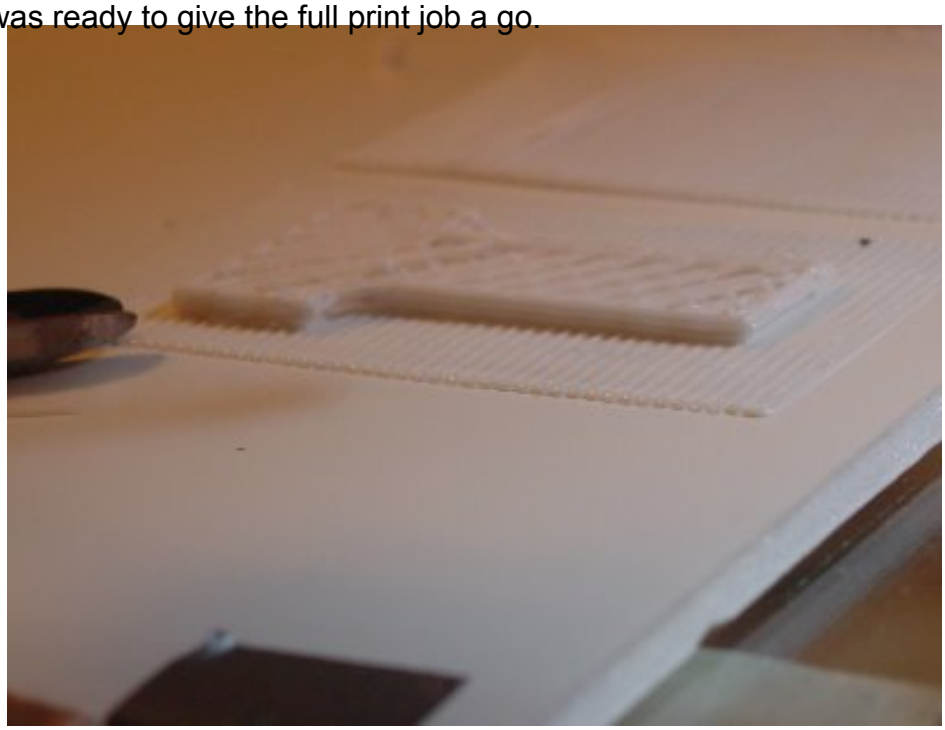
Idecided rather than simply duplicate a Mk II that I'd redesign it a bitand get rid of that chunk of wood that you can see circled in red justabove.

I designed the thing in Art of Illusion and made astraightforward rush to just print it way back in March. Biiiiigggmistake. I guess I made half a dozen rushes to print that darned thingcoming at the problem from half a dozen directions.

Finally inJune I decided that I had to solve the corner curling problem with HDPEor I was never going to print anything worthwhile. On 24 June on my60th birthday I cracked that problem. I had to use a loose diagonalinfill to get it to work, but work it did.

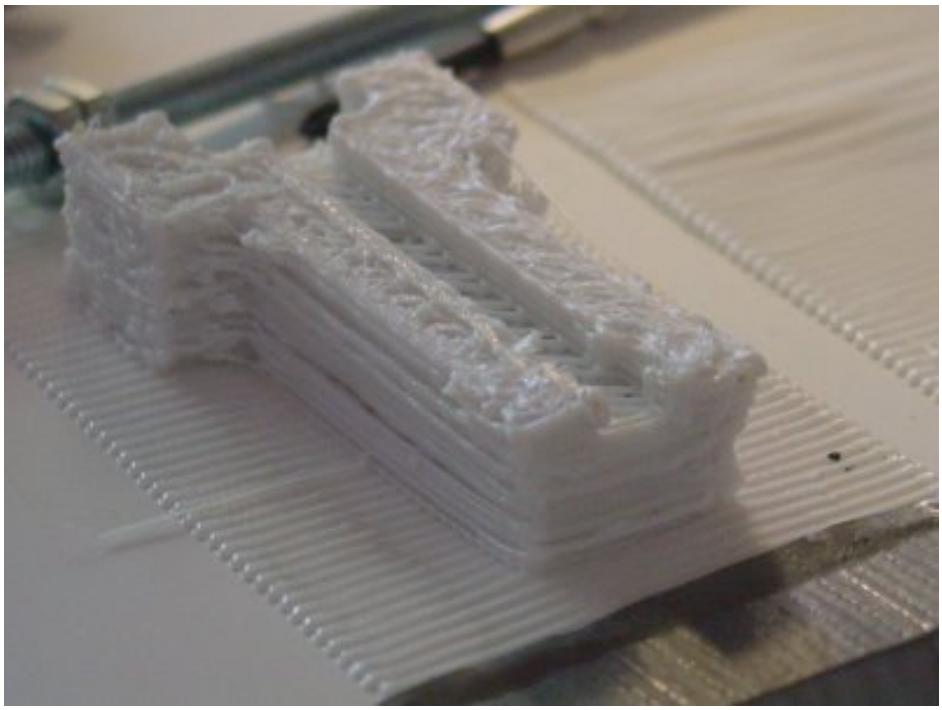
Since then I've nothad a lot of time to work with Tommelise, what with contract deadlinesand the

like. Last weekend, however, I'd got a little bit ahead anddecided to give it a go one more time. Mercifully,the Tommelise Control Panel runs pretty well in background on myworkstation, so I was able to print things while doing my consultingcontracts without the two getting in each other's way. Monday night Iwas ready to give the full print job a go.



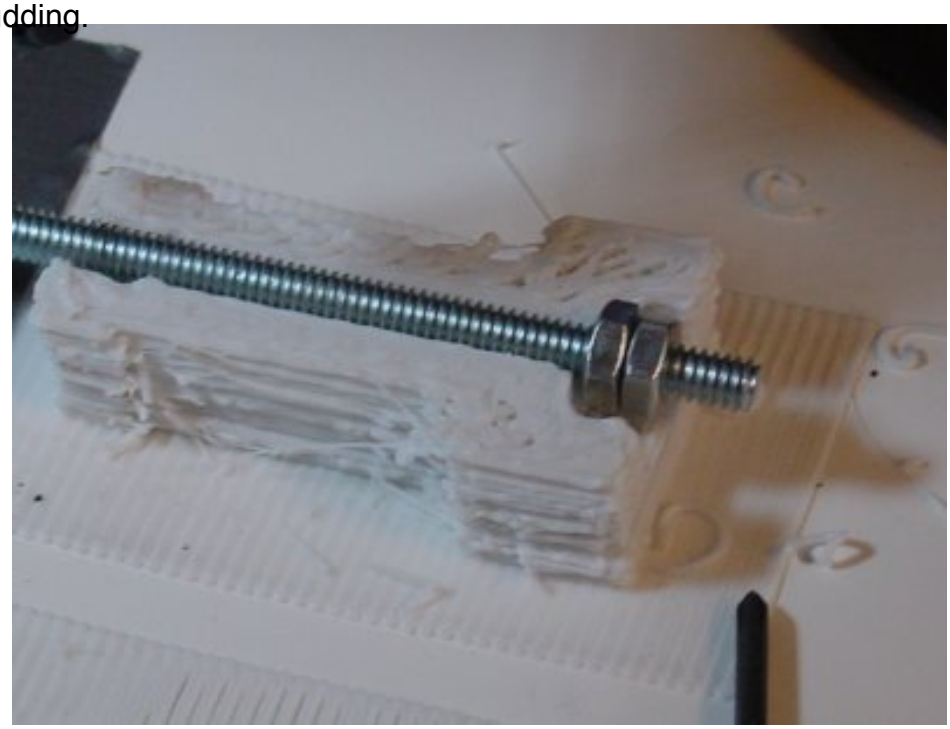
Things went fairly smoothly Monday evening. I shut down the system and went to bed at about 2300.

Igot up about 0430 as usual and started the thing up again. There was abit of drama in that the print had cooled during the evening and thefoamboard that I was print on had warped a bit as a result.

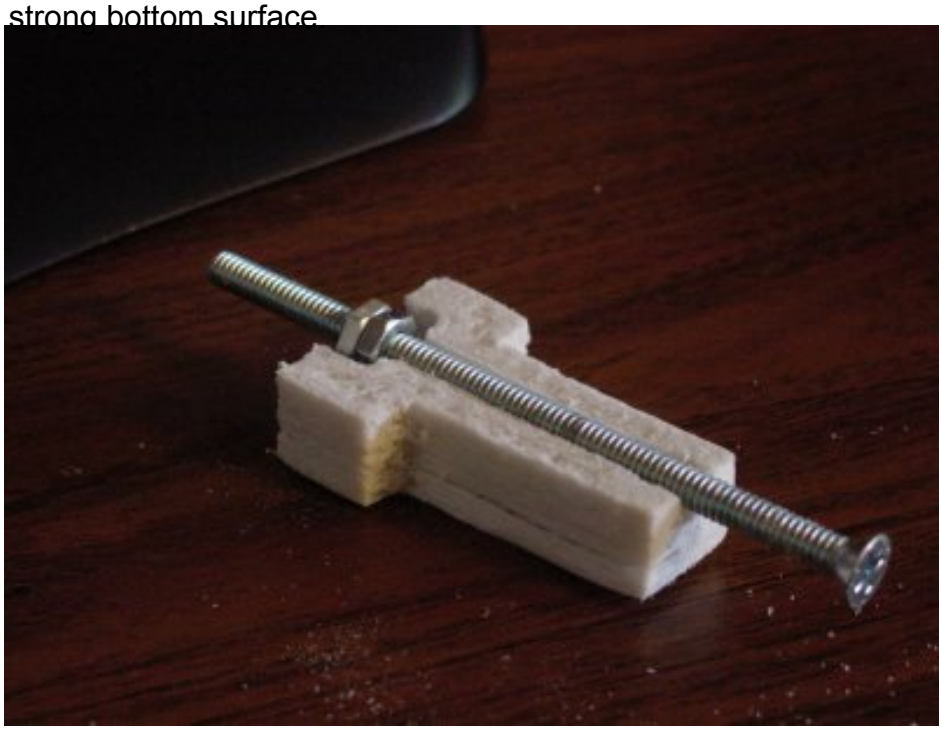


Youcan see the gap between Monday night's printing and Tuesday's at thelower left hand side of the pump. The gap is nasty looking but iscosmetic and has no apparent structural implications. Tommelise waswell into the third horizontal cross-section of the polymer pump whenthis picture

was taken.
I sized the polymer pump to accommodate a 1/4-20 threaded rod (6.35 mm) instead of an M5 (5 mm) metric studding.



Hereyou can see me checking the partially completed pump for fit. Note the"hairiness" of the print at this point. Once the print was finished Icut it out of the foam board and trimmed the excess raft away from thepump. I didn't try to pry off the raft on the bottom face of the pumpbut rather let that serve as quite a strong bottom surface.



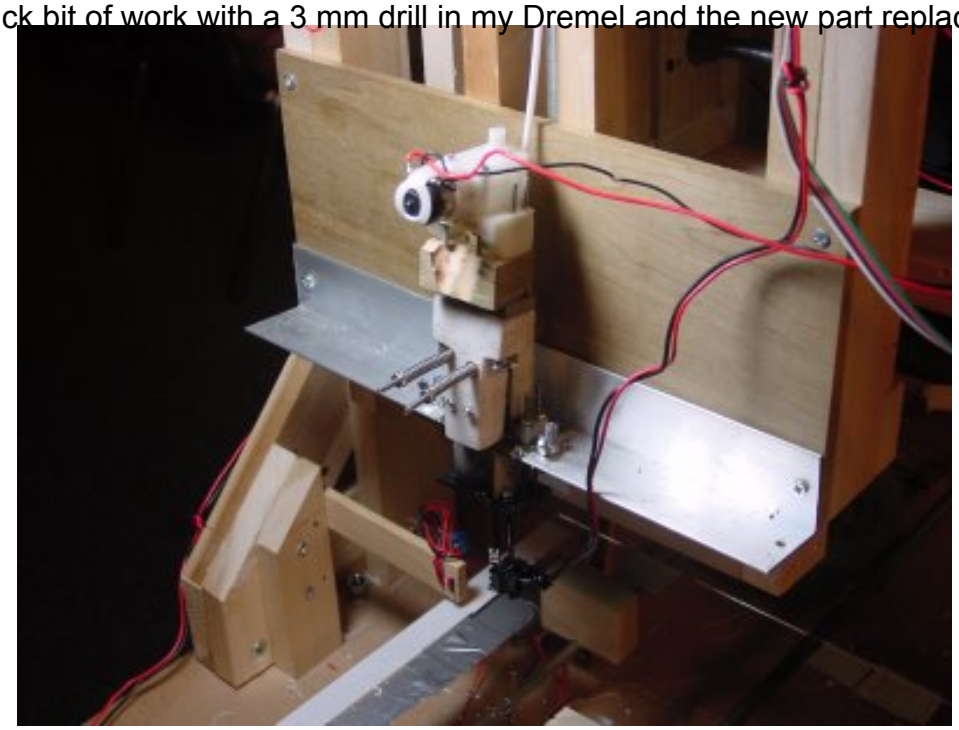
Here you can see the pump after trimming and a bit of a clean up undertaken with a bread knife and a piece of sandpaper.

Here is another view of the pump taken a bit closer. I use two nuts anda lock washer rotating against a steel thrust collar to keep thethreaded rod in the pump rather than milling away some of the threads. I designed a pocket into the top of the pump houseing to accomodatethose nuts more cleanly and snugly than was possible with my wood andplastic lash-up.
Just as an aside, the base of the pump was run with a 33% infill whilethe flanges enclosing the sides of the quarter inch threaded rod wererun at 75% infill for more strength.
Now all that remains is to drill the bolt holes in the pump and install it. It cleaned up rather well, I thought.
It's definitely time for some McAllan's, so I'll bid you good evening.

Installed

Tuesday, 10th July 2007 by Forrest Higgs

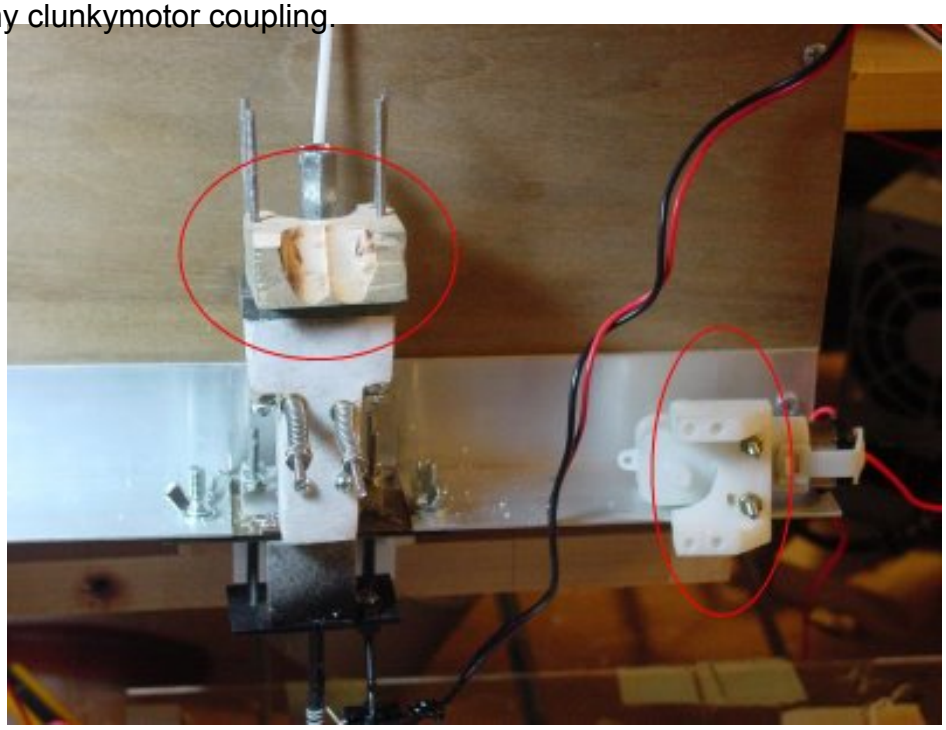
A quick bit of work with a 3 mm drill in my Dremel and the new part replaces the old.



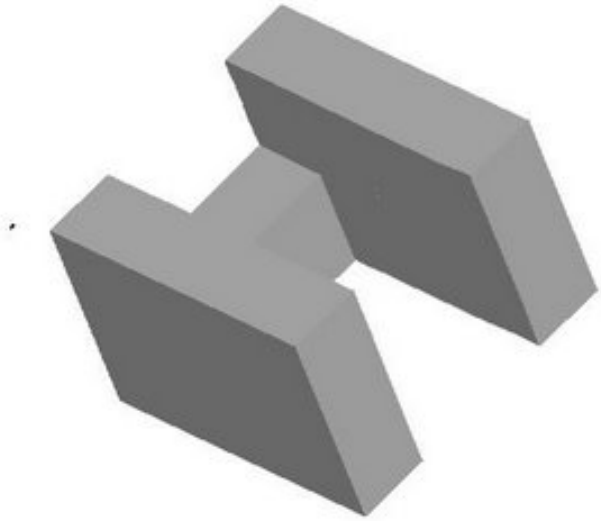
Moving on

Thursday, 12th July 2007 by Forrest Higgs

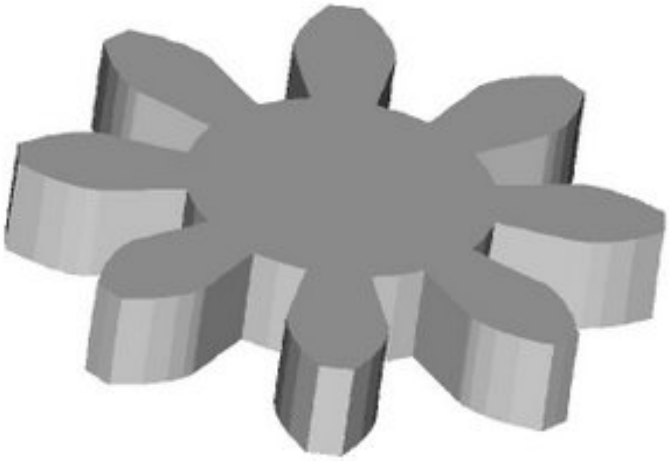
When I set out to print a polymer pump for the Mk I back in March Ialso resolved to replace the other chunk of wood in the Mk I thatraised the gearmotor mount another 20 mm to make way for my clunkymotor coupling.



Here you can see the two bits that I want to replace with one new printed one.
I am hoping that it will be an easierexercise than printing the pump was, but I will just have to see, Iguess. I designed the clunky new mount in Art of Illusion (AoI) andSlice and Diced it into its two printing zones this evening inbackground while I was doing consulting work in foreground.



Here'sa screen grab from AoI to give you an idea of what it will look like.It's nothing special, but it will get rid of that other unsightly pieceof poplar out of the Mk I.
What I intend to do after that and getting the whole updated Mk Irolling is a bit trickier. Quite some time ago I wrote a script forAoI to design involute profile gears. A few months ago Vik had a go atmaking some small ones on Zaphod that were a little less thanwonderful. I've got 15 lbs of HDPE filament and am not so constrainedas Vik was at the time with CAPA, so I'm going to do a 50 mm gear firstoff to see what I'm going to have to do to get all of that workingright.
I took a few moments off this evening and designed the test gear in AoI. Here is what it will look like.

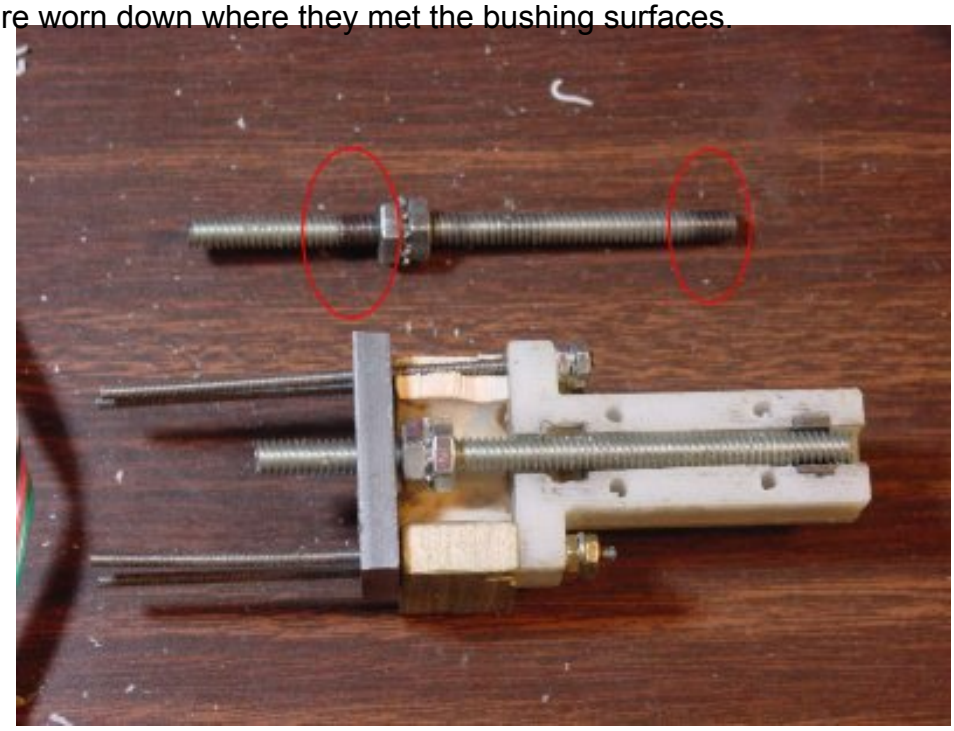


To paraphrase the unfortunate Robert F. Scott, I'm going to try this but I may taking some time at it.

Bushing wear on the Mk I AEM

Sunday, 15th July 2007 by Forrest Higgs

Fitting the new HDPE polymer pump gave me the chance to do a completebreakdown of the Mk I AEM extruder and do an assessment of how it isholding up. It had been working fine and everything inside looked allright. I did notice that the threads of the studding that actuallypumps the filament were worn down where they met the bushing surfaces.



You can see the wear circled in red in this pic. The steel bushingsseemed to have suffered little more than the odd scratch on the contactsurface, a result that leads me to believe that the bushing steel isconsiderably harder than that of the studding.

I replaced the brass upper bushing with quarter a quarter inch thicksteel one on March 25. Since then I have run about 1.2 kg of HDPEfilament through the extruder during about 300 hours of continuous operation.

Although the existing 4 inches of studding had been working fine, Idecided to replace the 1/4-20 studding that is the pump in the Mk I AEMjust for luck and to see what issues that such a replacement wouldentail.

The biggest issue is that a fresh piece of studding puts more frictioninto the Mk I than a worn one does. Running the Mk I with HDPE takesjust about every bit of torque that the Solarbotics GM3 gearmotor thatpowers the pump has. The safety clutch on the GM3 slips at 60oz-inches and makes a loud click when it does. I heard that a lotwhen I started running in the new studding. At first I had to useneedle nosed pliers to get it running again. After about 15 minutes Icould clear it with my fingers and after 30 minutes it began to clearitself.

I ran the pump for several hours till I got a full hour of continuous operation without hearing the clutch slip.

Just as an aside, I buy three feet of 1/4-20 studding at the localhardware store for about $1.80. That is enough to make 9 replacement sections of studding for the Mk I at about $0.20/replacement.

Designing the next generation Mk 2 AEM

Saturday, 21st July 2007 by Forrest Higgs

I've got, right at the moment, something like 350-400 hours ofoperation time on the Mk 1 AEM 2 amp extruder. Overall, it's been areally good little design deriving as it does from Adrian's original MkII. It's got me from nowhere to having been able to successfully printa replacement part in HDPE.

That said, I've been thinking aboutwhat the next generation extruder ought to look like. Adrian's Mk IIwas designed to extrude polycaprolactone (CAPA). While CAPA is a toughplastic you can cut into it with the metal thread of 1/4 inch studdingrelatively easily. That means that the springs that clamp the polymerpump to the filament guide can be lightweight and the torquerequirement on the studding pump aren't huge.

HDPE is a tougherplastic than CAPA. It didn't take me long to realise that the #4springs that worked well with CAPA were not quite adequate for HDPE. Isized those up to the toughest #6 springs that I could find and havesince been able to do production extrusion in HDPE. Mounting #6 springson #4 studding isn't a happy sort of thing to do, but it can be made towork.

The Solarbotics GM3 gearmotor handles both CAPA and HDPE fairly well.

It'ssafety clutch is set to 60 oz-in of torque. This is way more than youneed for CAPA and maybe about 10-15% more than you need for pumpingHDPE at the rates that I currently use. I've been able to barely pumpHPP (homopolypropylene) at production rates with the GM3. ABS, however,is out of the question.

Next Generation?

Tuesday, 24th July 2007 by Forrest Higgs

The new motors from Solarbotics just arrived.

I've included a GM9 at the right of the pic. It is very like the oneDarwin uses on the Mk II extruder though with a different gear ratio. Above and to the left of it is the GM-17 which I plan to use on the Mk2 AEM extruder that I am presently designing.
I took a few moments and ran the GM-17 at 5 and 9 volts. That thinghas HUGE torque. I couldn't stall the drive shaft with my fingersrunning it at 1.5 volts. I think that it is going to work just finefor pumping HPP and ABS.
The tiny little pair of motors with brass gearboxes are GM-11a. Thoseare expensive little mites at $16.80 each. The trick is that theydeliver 7.5 oz-in of torque and, I think, will drive the Tommelise xypositioning stage on a diagonal at about 6 mm/sec.

Next Generation?

Tuesday, 24th July 2007 by Forrest Higgs

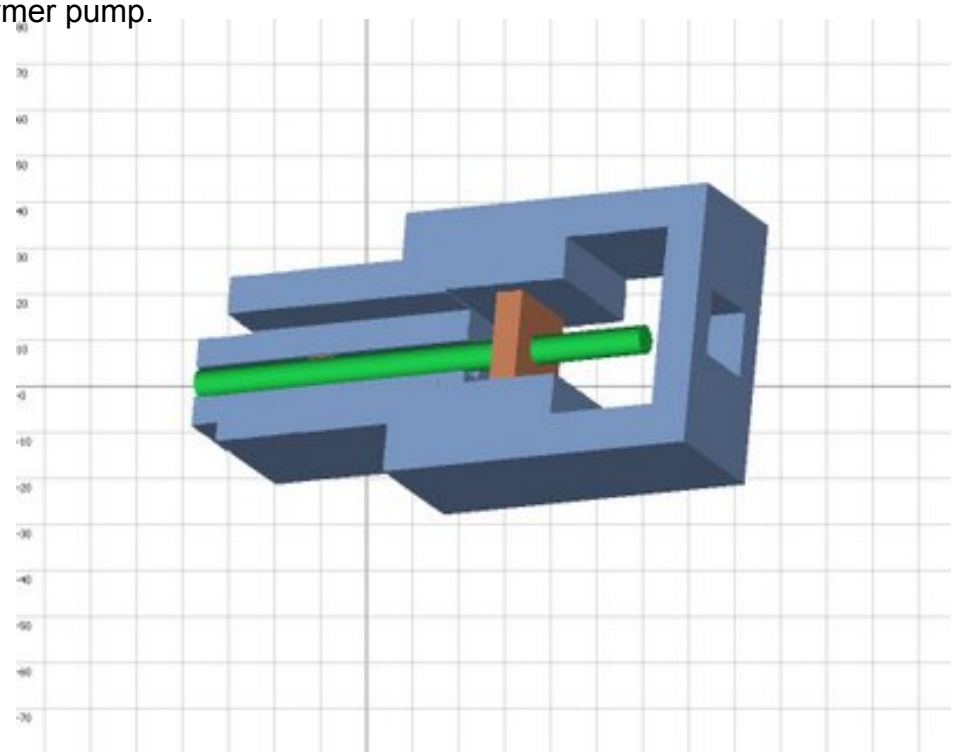
The new motors from Solarbotics just arrived.

I've included a GM9 at the right of the pic. It is very like the oneDarwin uses on the Mk II extruder though with a different gear ratio. Above and to the left of it is the GM-17 which I plan to use on the Mk2 AEM extruder that I am presently designing.
I took a few moments and ran the GM-17 at 5 and 9 volts. That thinghas HUGE torque. I couldn't stall the drive shaft with my fingersrunning it at 1.5 volts. I think that it is going to work just finefor pumping HPP and ABS. Dan was right. The mounting holes for the GM-17 are both further apart and at the other end of the gearmotor from the GM-3.
The tiny little pair of motors with brass gearboxes are GM-11a. Thoseare expensive little mites at $16.80 each. The trick is that theydeliver 7.5 oz-in of torque and, I think, will drive the Tommelise xypositioning stage on a diagonal at about 6 mm/sec. As well, these tiny little motors, unlike the ones I am using now, arealmost impossible to hear from a few feet away. This is a great ideaif a kid decides to print something overnight in his bedroom.

Second pass at the Mk 2 AEM polymer pump

Tuesday, 24th July 2007 by Forrest Higgs

Now that I have a GM17 motor in hand I was able to make a more exacting design iteration on the Mk 2 AEM polymer pump.



There are some tricks to getting Art of Illusion to do what you wantwith Boolean operations to make an object like this. I'll write themup sometime when I have a moment.

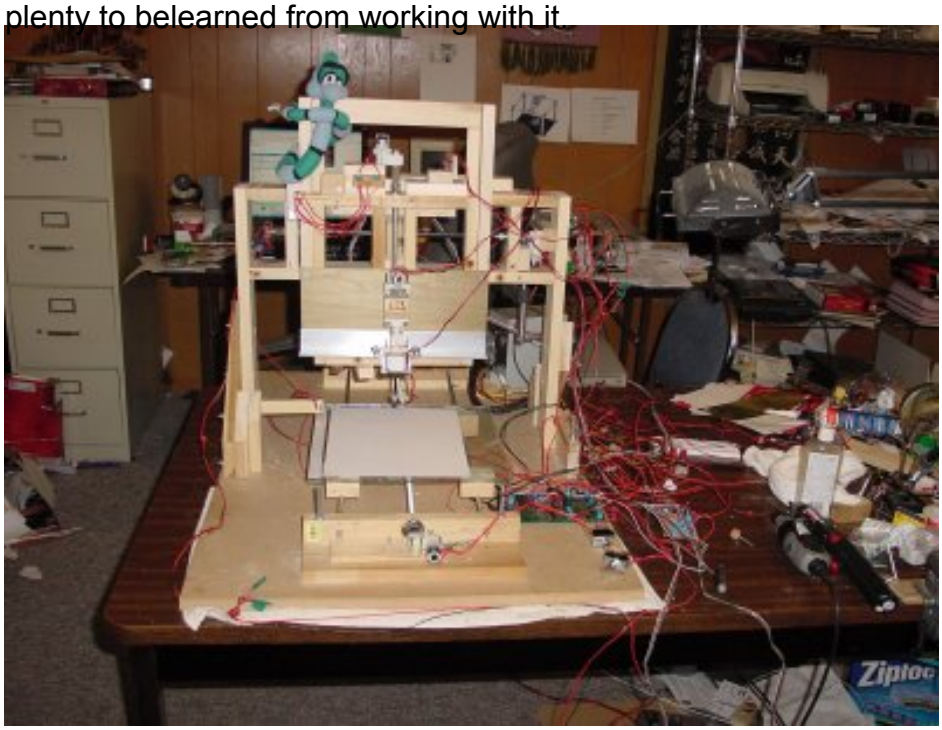
Tommelise 2.0

Thursday, 2nd August 2007 by Forrest Higgs

I haven't been posting much in the past few months. Sadly formy 3D printer work and happily for my overdraft, I'm been doingconsulting work in computational linguistics pretty much around theclock. With a little luck I might even manage to get my overdraft undercontrol if this goes on as long as my client says it will.

Yesterday,however, I had a major turn-in of contract deliverables and was able toget a hot meal, a bath and six hours of glorious uninterrupted sleep.My client is busily trying to put what I gave them on their serverfarm, so I haven't heard from them today and I don't expect to hear toomuch for several days. This lets me work on the next deliverables at amore civilised pace.

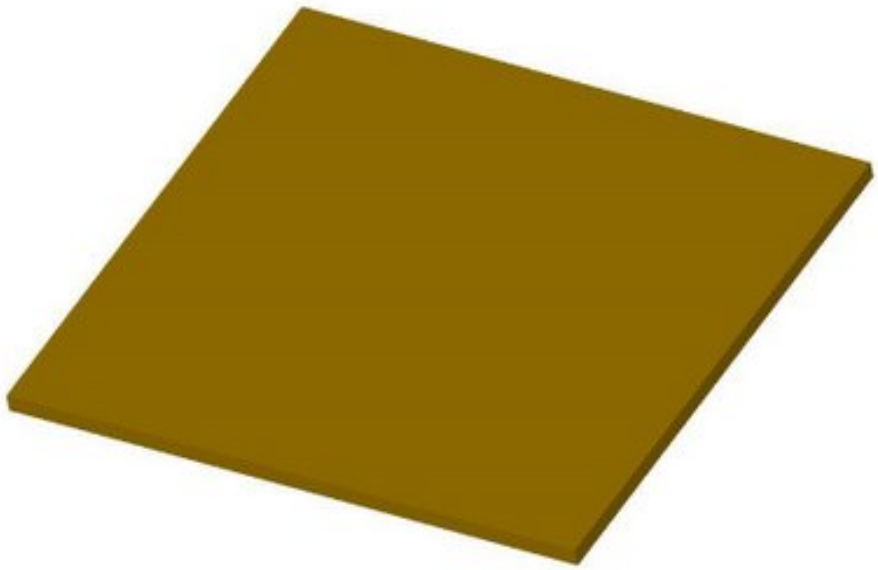
As you recall, some days ago I posted sometentative ideas for a new extruder design. This evening I decided thatit was time to start thinking about Tommelise 2.0. Mind, Tommelise 1.0works fairly well. I've been able to print usuable parts on it withdifficulty for about a month now and there is still plenty to belearned from working with it.



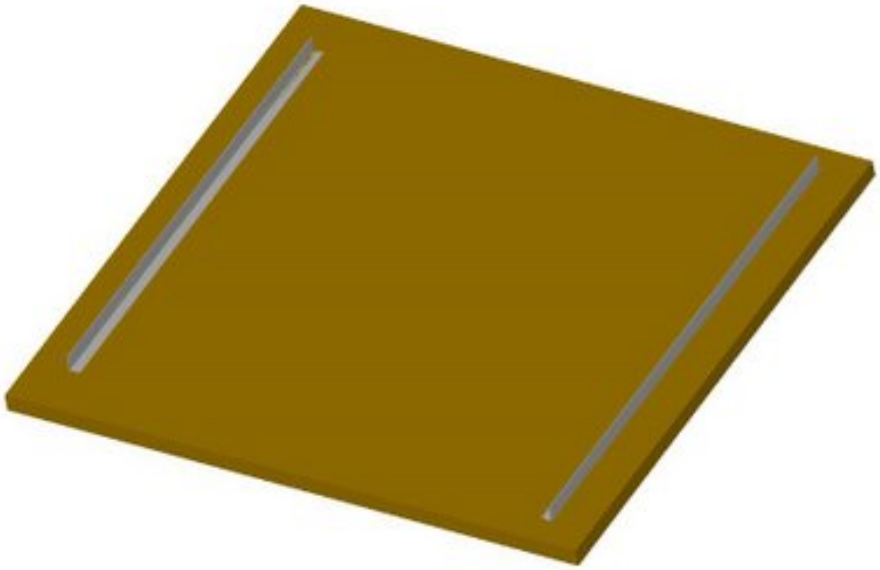
Infact, I've already learned a considerable amount from it. Mostly, I'velearned that it is a complete lash-up, not that I didn't know thatalready. While I love the old thing, esthetically it deeply offends theold architect in me. There are so many things about it that could beimproved.

After I got the Mk 2 AEMextruder designed and the parts for it bought, I turned to thecartesian positioning system of Tommelise. Another shopping trip got mesome parts that I think will form the basis of 2.0. This evening I dida little design charette with Art of Illusion and here is what I cameup with.

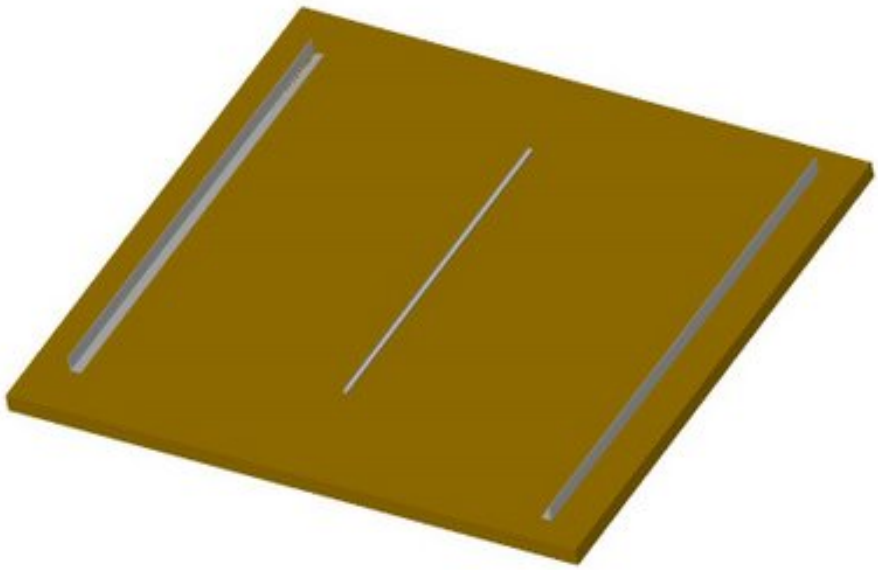
First, I started with a 24x26 inch piece of 3/4 inch chipboard.



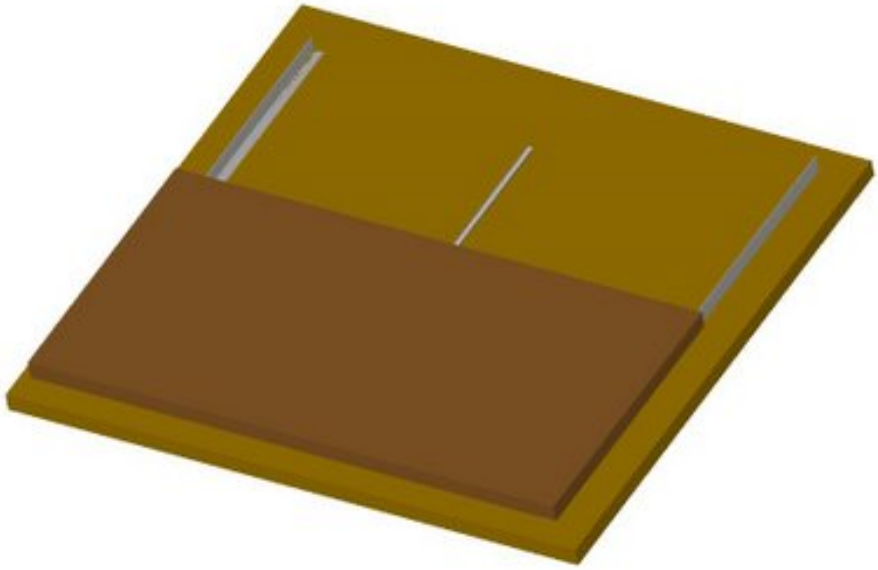
Ihaven't been very happy with the 1/4 steel rod that I've used for they-axis guides on Tommelise 1.0, so I bought some 1/16x1/2x3/4 inchaluminum "L" profile. for guides instead. It's cheap stuff and I canscrew it directly onto the base plate.



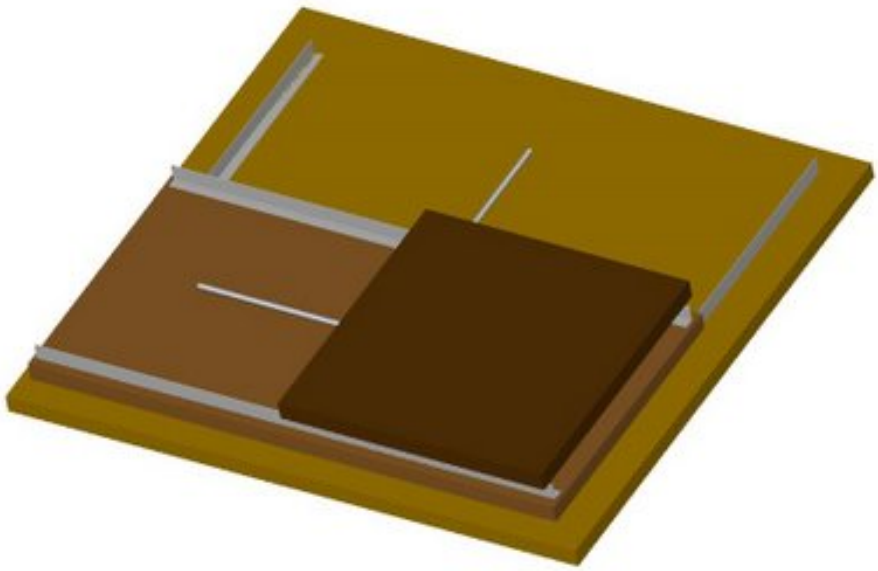
I'vespaced them about 22.5 inches apart for reasons that will becomeapparent in a few moments. I then locate the threaded drive rod for they-axis between the two guides.



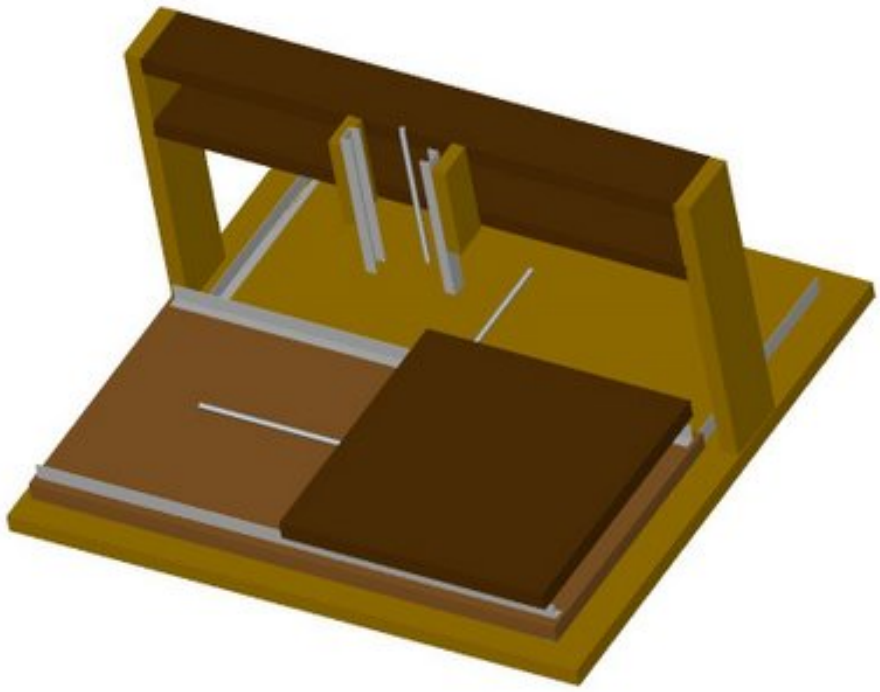
Ifyou are willing to locate the drive motors for the axis under theworking plane, the threaded drive rod need be no longer than thedistance you want the y-axis to traverse. I'm looking for a 12 inchtraverse, so I made the rod 12 inches long plus a few extra inches atthe ends for accommodating the mounting blocks. This shorter drive rodlets me size down from 3/8-24 inch threaded rod to either 1/4 -20 or1/4-24 inch rod, which is both cheaper and, more importantly, has only44% as much mass as the 3/8 inch rod. This means that if I use thegearmotors to drive the rod, it will be much less apt to overshoot asTommelise 1.0's y-axis has been.
I'm not showing the motor ormounting blocks, since this is a simple design walk-through and not adetailed design study. To this arrangement I apply the y-axis workingplane.



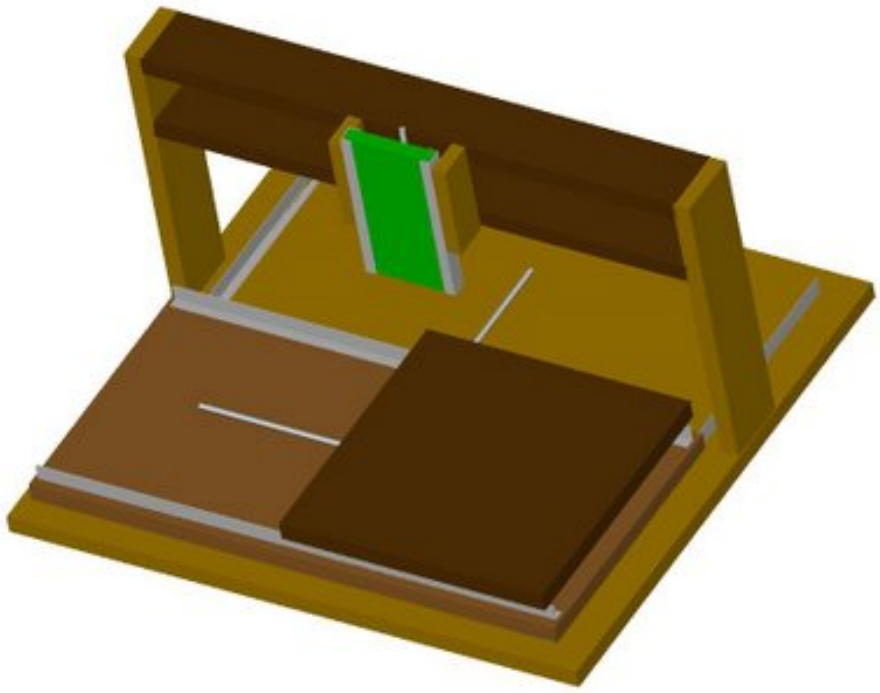
WhatI have now is a 12x24 inch plate, probably more chipboard that cantraverse 12 inches back and forth on the y-axis. In a major departurefrom Tommelise 1.0, to this I add x-axis rails and drive rod.



And on top of that I place a 12x12 inch xy working surface. All of the xy motion is thus attended to.
Thus, the gantry in Tommelise 2.0 doesn't have to cope with anything except raising and lowering the extruder(s).



Herel've used a 1/16x1/2x3/4 inch "U" profile in aluminum. Into that Islice another piece of chipboard onto which I can mount a pair ofback-to-back extruders, one for polymer and one for support material.



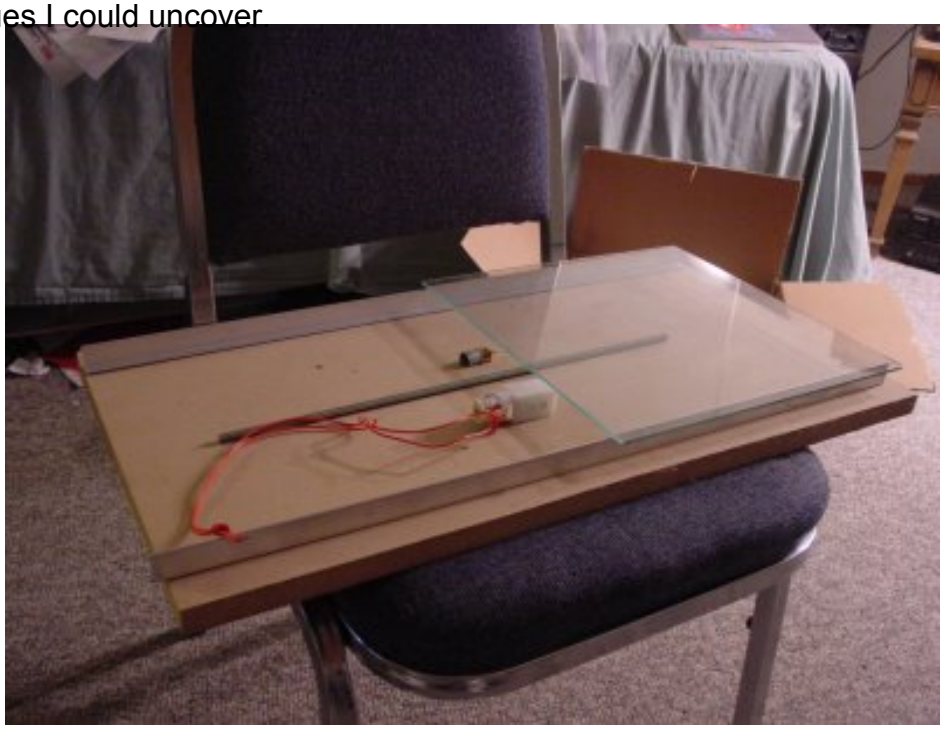
And that is, effectively, all there is to it.
As you can see, Tommelise 2.0 is a much cleaner design than Tommelise1.0. Tommelise 1.0 has a 24x36 inch footprint that gives you a 8x12inch working surface. Tommelise 2.0 has a 24x26

inch footprint thatgives you a 12x12 inch working surface.
If you demount the much simpler gantry assembly, you can get Tommelise2.0 into a standard 62 inch xyz measure airline suitcase with lots ofpolyurethane foam around it. That's important for taking it toconferences.
As well, Tommelise has many fewer parts and much more commonalitybetween axis assemblies than 1.0 does. It is much closer to what myhypothetical 12 year old might be able to manage to build with basicwoodworking tools.
Comments?

Mocking up the x-axis

Friday, 3rd August 2007 by Forrest Higgs

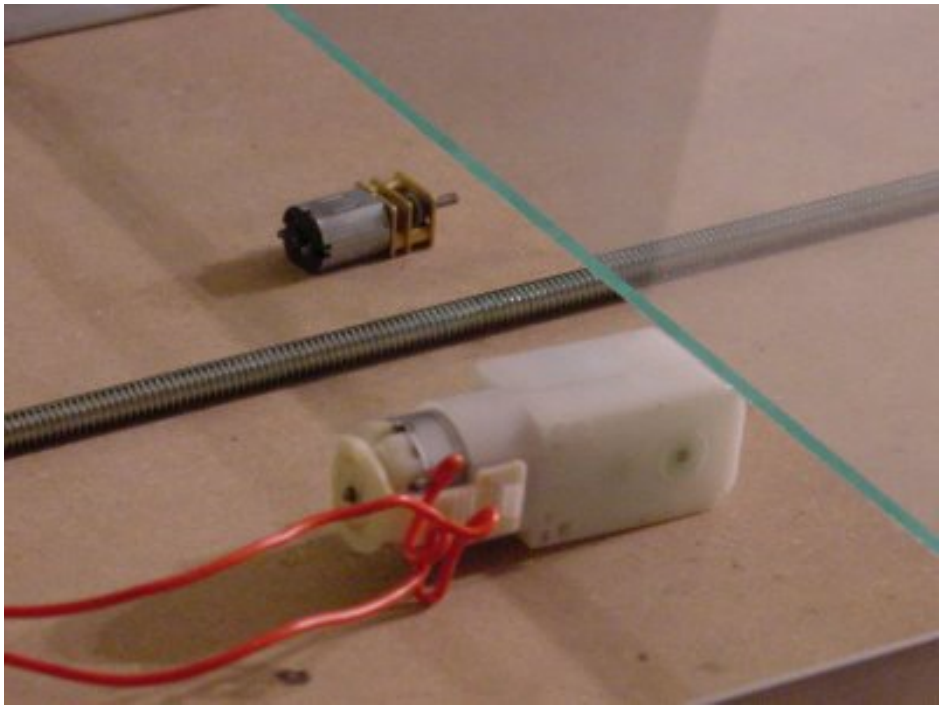
I'm waiting on a project manager to repair some errors in a datafile he sent over a few days ago this morning, so I used the time tomock up the x-axis stage of the proposed Tommelise 2.0 to see whatdesign issues I could uncover.



Whiledesign charettes using a 3D modeling system like Art of Illusion arevery useful, there is always additional information to be acquired byassembling a full-sized mockup of the thing you are designing.

The big issue that I am trying to confront at this moment is what kind of linear motion system to employ.

Tommelise1.0 uses a threaded thrust rod attached to a shaft-encoded gearmotor.The momentum built up by the thrust rod while the gearmotor runs makesfor control issues. I've shifted down to 1/4-20 inch threaded rod from3/8-24 to improve on that situation.



I'm looking at two design options now, turning the threaded rod orturning the thrust collar. Turning the rod means that you needbearings, or at least bushings and some fairly tricky lock washerassemblies. Turning the thrust collar means that you need some gearsand a bit of tricky cutting to seat the gears onto the 1/4-20 inch nutthat you use for the thrust collar.
Mostly what I am doing now is seeing what kind of clearances that Ihave under the xy work surface (the float glass plate in the secondpic). It appears that I can accomodate either type of gearmotor andthe small gearmotors with little more effort that jacking up the guidebushings on the work surface.

Buying the bits

Saturday, 4th August 2007 by Forrest Higgs

I spent the morning at the hardware store, Radio Shack and Potters'electronics. I'm beginning to think that the parts for Tommelise 2.0can be kept under $150 without a lot of strain.

The biggest, nastiest expense are the magnetic shaft encoder chipswhich, thanks to the fact that they're made in Europe, are up to$12-14/chip. I'm told that Austria Microsystems has a cut down shaftencoder chip that I can get for under $3, but I'll believe that when Ihave some in my hand. I'm going to press them on that next week.

The Euroboard stripboards have gone up in price as well, something like50% to $15 since the last one I bought in April. I'm thinking thatthat is more Potters' than a real bit of inflation, though.

My son returns on the 15th and asked if I'd have a prototype running bythen. I'm going to see if I can manage that. My client's web servicesserver went south last night so I can't do consulting work till Monday,so I'm going to have a good run at it between now and then. :-D

Trying out gearmotors

Sunday, 5th August 2007 by Forrest Higgs

I did some experimentation with my gearmotor options for Tommelise 2.0.

For now the tiny GM-17's are out of the running. I tested them to seeif they were able to turn the 1/4-20 inch threaded rod and got them tooperate at as little as 1.5v.

As a rule of thumb I've discovered that for running the positioningsystem you can generally expect that you will achieve about half of thelisted free-running rpm for your motor. Free-running rpm for the GM-17is about 341, which gives me a translation speed of about 6 mm/secunder no load. Practically I get about half of that, which means thatI can do a 45 degree diagonal with x and y axes run by GM-17's of about4 mm/sec. Those I will have to run at 6 volts.

Running at 6 volts will mean that I have to use an adjustable voltageregulator like the LM317. I bought one of those. It's rated at about1.5+ amps, but dissipates 15 watts, which is just a shade wasteful.

The big problem with the GM-17 is attaching it to the threaded rodshaft. Its drive shaft is 2 mm in diameter which is a bit finicky towork with in an attachment to 1/4 inch threaded rod. If I save theGM-17's for a fixed threaded rod/rotating thrust collar configuration Iwill be able to make a more facile connection.

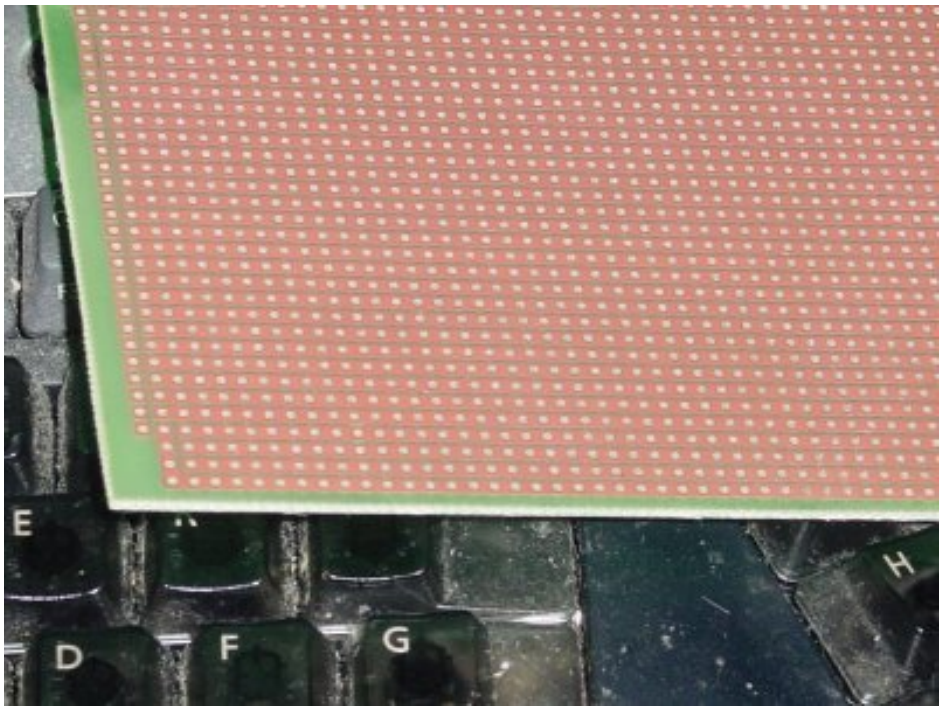
For turning the 1/4-20 threaded rod, for now I am going to use 224:1GM-2's hacked to a 14:1 ratio. I tested those and they have enoughtorque to move Tommelise 1.0's y-axis positioning stage just under 6mm/sec. Shifting from 24 to 20 threads/inch in going to the 1/4-20threaded rod kicks that up to right at 7 mm/sec or 9.8 mm/sec for usingx and y axes for a 45 degree traverse. Using the GM-2's will let me use the couplings that I've developed forTommelise 1.0. For now I am just going to use a poplar bushing. I maygo ahead and order some 1/4 inch bearings or use 8 mm skateboardbearings plus some tape.

I am going to build a separate controller board and comms card forTommelise 2.0. The last strip board controller that I built forTommelise 1.0 had virtually no mistakes in it that required jumpers. I'm going to see if I can clean those few errors up for 2.0. As well,I was able to find some 16 pin sockets to house the half-H controllerchips (754410) instead of kludging 18 pin sockets and filling theadditional two pin holes with glue. I've made a practice of puttingthe 754410 chips in sockets. I got tired of wasting them when I builta board and soldered them into the board only to have to throw theboard and chips both away. I don't mind throwing a socket away eventhough it costs nearly as much as the 754410. Something aboutunnecessarily throwing an IC away seems to just offend me. I don'tknow what that is about.

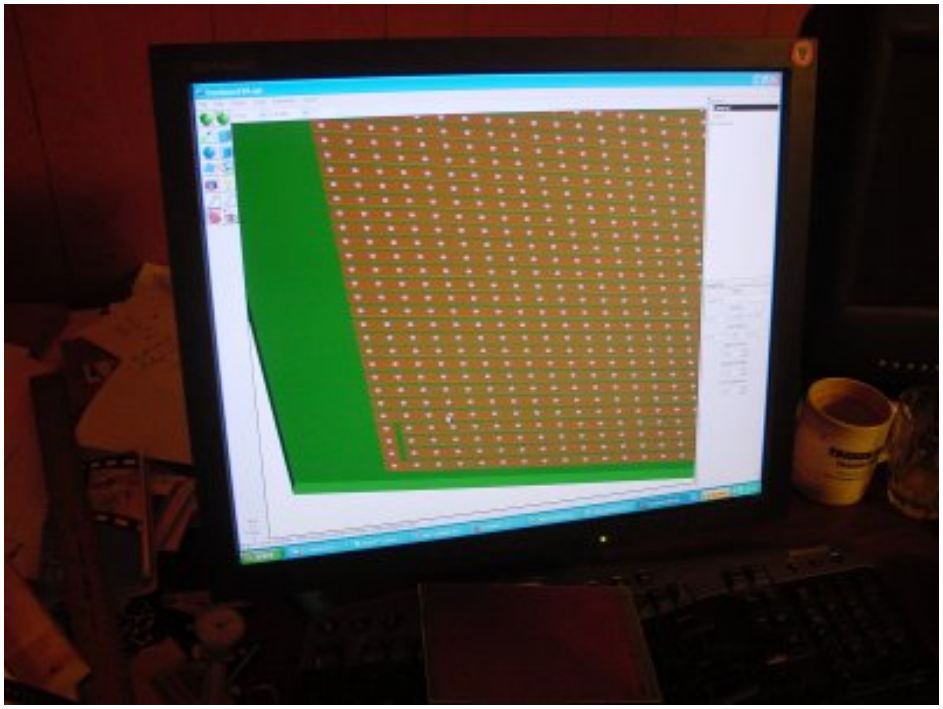
Virtual Circuitboards

Monday, 6th August 2007 by Forrest Higgs

While there haven't been any substantive changes to Tommelise'scontroller board for 5 months and there appears to be only one or twominor ones needed to make Tommelise 2.0, I'm still building controllerboards on stripboard, something pre-Darwin RepRappers were doing a yearago. Even though Ihave pretty stable board designs, I still don't feel comfortabledesigning special purpose boards for Tommelise. I just don't want tobecome a prisoner to set designs and inventories.



Iuse, essentially, a full-line (stripboard) Eurocard and a full-linehalf-Eurocard. They're made in Ft. Worth and can be had from...
http://www.the-hdb-store.com/servlet/the-VELLEMAN/Categories
...for $6.90 and $3.50 respectively.
SinceI don't intend to invest in purpose-made boards like RepRap has anytime soon, I've got a bit of an added burden on me of making thebuilding of circuit boards for Tommelise more accessible to people whoare innocent of prior electronics fabrication experience.
About a year ago I took a crack at creating a Eurocard board in Art ofIllusion. The big problem with that approach was that when you starteddoing boolean operations to punch about 2,000 holes in the board theEurocard got very nasty very quickly.
This afternoon, I revisited the question and decided that it might befruitful to just make a Eurocard in AoI out of a large set of partsrather than trying to make it of many fewer parts via booleanoperations. Happily, it turns out that AoI has no trouble whatsoeverworking with thousands of simple objects.
I reduced the Eurocard to its essentials and left off things like the single hole plated islands that you see on the left of the actual board. In practice, those are of little use.



The features on the Eurocard are rather small for AoI to make a nicerendering of the whole board. The limitations of both ones videocardand Java tend to get in the way.
From here, I will attempt to model the various components that go intoTommelise's controller board in living, loving colour and then placethem on the virtual Eurocard. Tommelise builders will be able to bringup the card in AoI and compare what they are doing while building it ona one-to-one basis.

Changing directions (again)

Monday, 13th August 2007 by Forrest Higgs

This is a very painful blog entry to write. I feel like such a weathercock doing this, but here goes.

I've collected everything that I need to build Tommelise 2.0. I havebeen getting on with the design and proof of concept exercises duringodd moments for the last week. I've been torn between buildingTommelise 2.0 by hand or using Tommelise 1.0 to make big parts of thenew Tommelise.

It all came to a head on Friday after I did a major deliverableshand-in for my consulting firm.

Some weeks ago I'd designed a newpolymer pump that would have a lot more torque and in general be a loteasier to run and maintain than the one I'm using now, which is a lashup of the old Mk II that Adrian designed more than a year ago.

I had designed the new polymer pump to mate with the old Mk II filamentguide. On Friday evening it became obvious that I was still stuck withmade up #4-40 long bolts and nuts. I needed to increase those in sizeto at least 1/8 inch or better yet #10 machine bolts. If I did that,though, I was going to have to redesign the filament guide as well. IfI redesigned the filament guide I would have to redesign the polymerpump yet again.

The upgrade that I did a while ago for the polymer pump took over 12hours to print. The new one was going to take longer still, possibly18 hours. If I redesigned the filament guide it would get bigger stilland probably take a full day to print. This snowballing effect wasgetting ridiculous.

I then had the bright idea of just making the thing out of poplarwood. By Saturday morning I knew what I had to do to get that tohappen and had made a few tentative cuts on parts for the pump.

At that point the house of cards of careful rationales that I had built up began to come apart.

I was designing a new extruder so that I could extrude ABS and homopolypropylene AND print HDPE faster. That's a great idea.

The problem with it is that I would be doing all that on Tommelise 1which really hasn't been performing all that well in terms of doingrepeatable and trouble-free positioning. Mind, it's doing a LOT betterthan it was, say, six months ago but I still can't get it to do finework like gear teeth at any scale. I tried that a few weeks ago withdisasterous results.

Obviously, I need better positioning performance and I'm just not getting it.

After reading some comments by "nophead" at the RepRap forums I got thedistinct impression that I ought to be getting a lot better performanceout of my shaft-encoded than I am.

All today (Sunday) I've been cobbling together a series of exercises for the positioningsystem that ought to result in a rewrite of the positioning routines inmy firmware that will give me much improved performance.

This afternoon, I began setting up some test routines to carry out theexercises. I get depressed some times because this is all so damneddifficult and seems to come with such painful slowness for me. I don'tseem to have much in terms of gifts beyond a rude sort of Scots-Irishbloodyminded stubbornness. It gets me places, but it sometimes seemsto take me forever.

Wish me luck. I have no idea how long this is going to take.

A matter of timing

Monday, 13th August 2007 by Forrest Higgs

I took a few hours and began to explore the firmware programming issues associated with shaft encoding gearmotors.

My first exercise was to find out exactly how much time the variousparts of the existing firmware was taking. I'm running an 18F4610 at20 MHz. Here are a few of the things that I've discovered.

Solving a linear equation - ~100 uSec

Solving a quadratic equation - ~150 uSec

Reading a byte of data at 9600 bpi - ~1 mSec

Tommelise runs wonderfully when I am extruding straight lines of morethan a few mm in length. Where I get in big trouble is when I have abunch of little line segments that are ~0.1 mm in length. This doesn'tsound like any big deal until you start thinking about tracing theperimeter of a small circle, for example, a hole for a #4 built up boltor a tooth of a involute profile gear.

The instruction that I use to talk to Tommelise when I extrude a lineis 15 bytes long. Roughly speaking, just to transmit that takes 0.150Sec. What that means is that if you are looking at 1 mm of 0.1 mm linesegments you require .15 seconds just to transmit the information,never mind actually processing it into something that will drivegearmotors and extruder heaters.

A major firmware code bottleneck that I've discovered so far is that I'm trying to solve a linear equation of the form...

$$y = mx + b$$

many, many more times than are necessary. I've got to get that equation out of the innermost loop of the move_to command.

The other major exercise that I undertook today is initiate interruptswhen the 18F4610 powers up instead of during each move_to command. That gets me away from missing shaft encoder pulses with theinaccuracies that such missing implies. It also allows me to keeptrack of absolute position in Tommelise.

Thanks for giving me the confidence to try that, nophead.

Modeling firmware behaviour

Wednesday, 22nd August 2007 by Forrest Higgs

I've been running single axis calibrations for about a week now andhave a fair idea of the dynamics of the gearmotor/rotary encodercombination now. My old approach of interrupts on and off only when Iwas trying to print a line of polymer was definitely a bad idea. If youleave them on all of the time you get a repeatable absolute positionwithout any problems as nophead implied in some of his forum postings.For me, that has been a wonderful step forward.

What hasn't beenfun, however, is trying to get the extruder head to follow a pathproperly from one point to another. Now that I have an absolutepositioning readout it becomes obvious that my firmware routine forrunning the x and y axes is just a little short of nice.

Armedwith all of my new understanding, on Thursday I started trying to writea new xy movement routine. I have an IDE to check my code against whichlets me trap a lot of problems. I even have an API that will let mebuild forcing functions and quite complicated models of devicebehaviour to test my firmware.

The IDE, however, simulates every clock cycle of the CPU, which means that if you have an instruction like...

```
WaitMS 5000
```

...inyour firmware code, you can cook and eat supper without having toseriously worry about missing anything when you are simulating an18F4610 running at 20 MHz. The net result of all that is that while theIDE is useful for preliminary debugging of firmware it is of limiteduse for detailed testing of complex firmware.

What I had beendoing was to write my firmware code and use the serial link to outputdata about system operation on the fly. The problem with this is thatthe time that it takes the 18F4610 to transmit serial data about systemperformance is large with respect to the operations that I amattempting to do. That makes error trapping of real-time operationsoperating in the millisecond range more than a bit difficult.

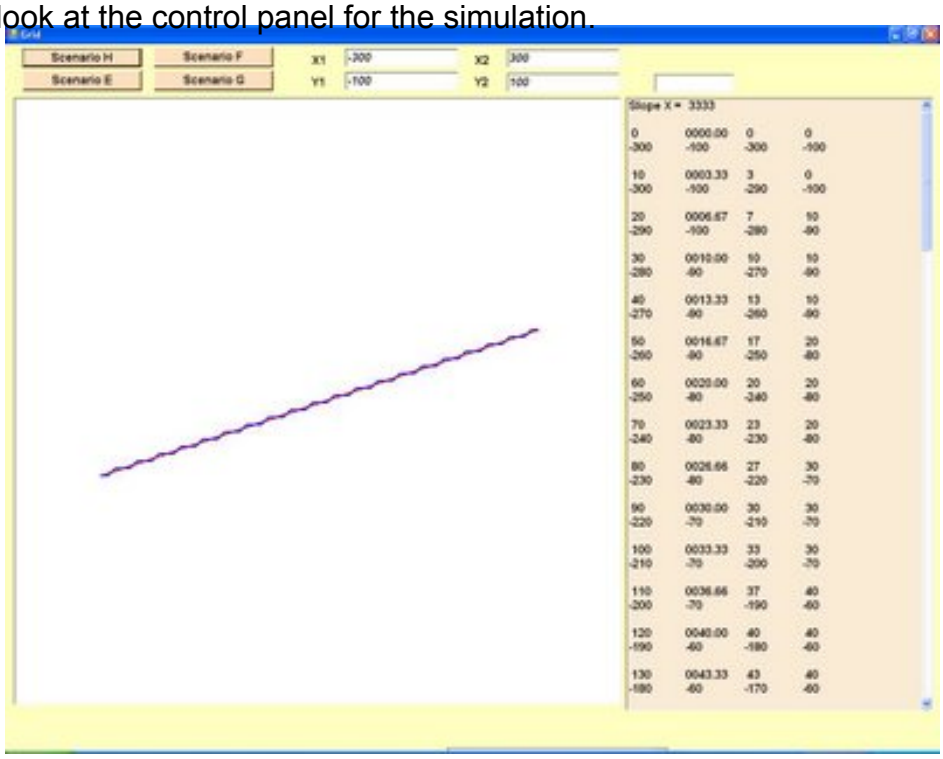
I'vebeen remarkably lucky with my 18F4610's so far. I've ruined one andchanged the program on the second one literally hundreds of timeswithout ruining any of the pins. With what I am trying to do now,however, the development cycle time is beginning to really get in theway of progress.

Setting aside the time it takes to actually make afirmware code change it takes about 5 minutes to extract the 18F4610from the controller board, put it in the PIC programmer, compile theupgraded firmware, program the chip, put it back in the controllerboard and fire up Tommelise to see if what I did helped matters.

Toimprove on this development cycle time I built a model of the firmwarein Visual Basic .NET 2005. I used all integer maths with a few tricksto mimic real number operations, just as I would have had to have donein Oshonsoft PIC Basic. VB.NET's graphics let me keep track of what washappening in a way that would have been impossible in firmwaredevelopment. I was also able to build models that introduced bothrandom and biased errors into the performance of the gearmotor/encoderensembles that mimic the behaviour of the real equipment.

In theVB.NET simulation I was able to evaluate eight different approaches tocontrolling xy motion in Tommelise and make dozens of improvements inthe code, a process that would have taken me

months had I taken theconventional development route.
Here is a quick look at the control panel for the simulation.



Thescale that you are looking at is one of a single pulse of theencoder. The red line represents the geometric path between X1/Y1 andX2/Y2. The blue path represents what the gearmotor/encoder drives forthe x and y axis would actually produce. While the blue path looksgnarly that is because of the resolution of the display. It isactually operating well within a tolerance of +/- 0.1 mm.
Once I had all that working I confronted the task of correctingovershoot and undershoot errors in the positioning system on the fly, as it were. Previously, I had been putting in correction factors forover and undershoot conditions and praying that they were generalenough to give me decent results. I got a surprisingly long way doingthat, especially after I took Adrian's advice and started resetting thexy after each layer of printing.
The natural tendency in a control scheme is, if you overshoot, toreverse the affected gearmotor direction and have another go at hittingthe target position. While that works it introduces a number ofsources of potential error into the mix that are best avoided. What Itried instead is, since I am typically printing paths considerablylonger than 0.1 mm, to correct the overshoot/undershoot condition inthe next 0.1 mm print increment.
My gearmotor/encoder ensemble code turns on the gear motor for a numberof milliseconds and then turns it off. Next it clocks a number ofmilliseconds delay to allow for the gearmotor to stop. In practice Idon't wait for the gearmotor to stop completely. When I map theensemble that powers an axis I get a performance table that looks likethis.

X-axis Gearmotor/Encoder Performance
(pulses per power/delay cycle)

Power (msec) \ Delay (msec)	1	5	10
9	9.33	10.18	11.23
10	10.37	11.21	12.16
11	11.33	12.21	13.2
12	12.33	13.26	14.26
13	13.32	14.3	15.22
14	14.33	15.28	16.3
15	15.32	16.31	17.29

Youcan do a little quick math and get a feel for what is going on here. First notice that the sweet spot for this chart, that is, the placewhere you get the fastest translation speed for a resolution ofapproximately 0.1 mm is at a power setting of 12 msec and a delay of 1msec. You get 12.33 encoder pulses for that. To get 0.1 mm accuracyyou need 12.1 pulses for Tommelise, so you are getting slightly betterthat 0.1 mm accuracy.
If you put in more delay, you slow down the axis. If you throw in more power, you reduce the resolution.
You can back into the proportions of time that are required by thissweet spot. You have 12 msec for power and 1 msec for delay, so therest of the time is spent cycling the PIC chip in the control loop. Ata translation rate of 1.6 mm/sec you need about 62.5 msec to traverseevery 0.1 mm. That means that there is approximately 49.5 of that 62.5msec cycle that the PIC firmware gets to look for the proper number ofencoder pulses. That's pretty good, but it tells me that I may want toclock the 18F4610 up to 40 MHz when I build the faster printingTommelise 2.0.

What this performance graph lets me do is basically slow down or speed up the axis translation speed slightly to correct for undershoots and overshoots detected by comparing where the extruder head is supposed to be with where the encoders say it actually is at the end of each 0.1 mm transition.

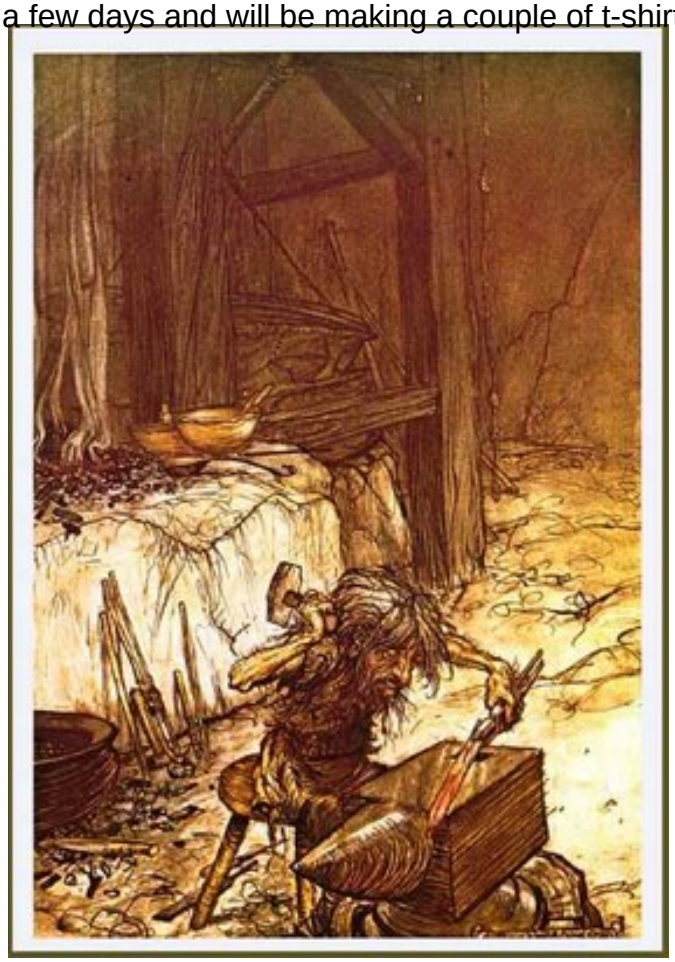
I think that I am ready to start programming the firmware on the PIC, now.

Mime gets a t-shirt

Saturday, 25th August 2007 by Forrest Higgs

Arthur Rackham prints are few and far between.The drawing that he did that is the theme for the Tommelise project wasthe dwarf *Mime at his forge*, was drawn by Arthur Rackham in 1911 as an illustration for Wagner's *Siegfried.*Mid-week I finally ran down a print in high enough resolution to justify printing a t-shirt.

I should be receiving it in a few days and will be making a couple of t-shirts for myself.



If anybody else is interested let me know. There is a t-shirt shop inmy little town. I am going to see if they are interested in fillingorders at their cost.

Some notes about gearmotor/rotary encoder ensembles

Sunday, 26th August 2007 by Forrest Higgs

I've put in pretty much every spare hour that I've had for the pasttwo weeks trying to get a grip on the issues involved in runninggearmotor/rotary encoder ensembles at high accuracy. For a long timenow I've been able to accurately extrude relatively long, straightlines of polymer at good accuracy and repeatability. Currently, withthe gearmotors that I am using I can get a top diagonal extrusion speedin the xy working plane of about 2.4 mm/sec.

What I haven't beenable to do well is to print a sequence of very short line segments(~0.1 mm) rendered as a series of print instructions on Tommelise. Forthis reason, I can make things like polymer pumps, but am totally pantsat making features like holes in said pumps.

Twoweeks ago I decided that I would see how far I could go withgearmotor/rotary encoder ensembles on Tommelise 1.0 before trying tobuild Tommelise 2.0. Here is what I've discovered. The chiefproblem with the ensemble seems to be the inability of the gearmotor tostop promptly after the power is switched off. The controller board forTommelise 1.0 runs the gearmotor at about 11v. I've discovered thatafter shutdown momentum of the gears within the gearbox coupled withthe momentum of the turning threaded rods will carry on the rotation ofthe drive shaft of the gearmotor for ~15-25 degrees after the power isshut down .

In practice, what this means is that using 3/8-24threaded rod I get a minimum repeatable step of about 0.2 mm +/-0.0125.I've been able to push the step size down to about 0.11 mm but when Ido that the error goes up to about +/-0.025 or so.

What I havediscovered, however, is that accuracy for very small single print linesis very much a function of the speed with which you attempt to printthe line. When you want to do this sort of many line segments, fineaccuracy printing you have to dial down the transition speed into thevicinity of 1 mm/sec or a bit below.

I wrote in a bit ofdiagnostic code into the firmware routine so that you could see how thegearmotor/rotary encoder ensemble is behaving.

Step Timing Exercise

Pulse	MSec/pulse
1	29
2	6
3	9
4	10
5	9
6	11
7	10
8	14
9	15
10	19
11	29
Total MSec/step	161

Thisparticular test did a single axis transition of 0.09 mm (11 pulses onthe rotary encoder) You can see how many milliseconds it took toachieve each pulse of rotation. When the gearmotor is powered up youcan see that that first pulse took 29 msec to achieve. This is nothard to understand in that this is the time that it takes the geartrain to go tight (overcome backlash) and overcome the frictionintrinsic in the axis movement.

In this case the power was switched off after the second pulse. Pulses 3-11 represent the time that it took for the gearmotor to wind down(126 msec). The total 0.09 mm transition took 161 msec. The extruderwas also moving at an average speed of 0.56 mm/sec.

While this very slow performance for printing short line segments mightseem to be a tremendous disadvantage it wants remembering that the vastmajority of the print time of a layer is spent doing infill. OnTommelise 1.0 infill proceeds at a print rate of 2.4 mm/sec. Theexisting Mk 1.0 AEM extruder will operate quite happily over a range of0.5 - 12 mm/sec.

Printing very short line segments on Tommelise

Monday, 3rd September 2007 by Forrest Higgs

For the past several weeks I have been doing a major rebuild of theTommelise 1.0 firmware. Heretofore, I was able to achieve fairly goodaccuracy (~1 mm) for printing as long as the line segments that I wastrying to print exceeded about 5-10 mm in length. That enabled me toprint, for example, a polymer pump as long as I didn't attempt to printthe bolt holes through it.

I had designed Tommelise 2.0 before it struck me that it was pointlessdesigning an improved model of Tommelise without achieving betterpositioning accuracy for the sorts of very short line segments oneneeds to delineate such things as bolt holes.

The first thing that I resolved to do is get rid of theon-again-off-again nature of interrupt monitoring that I was doingbefore. That done, finding out how much slew was intrinsic to thegearmotor/threaded thrust rod/rotary encoder ensemble.

Tommelise needs a shade over 12 encoder pulses to move one of its axes1 mm. To get that to happen fairly consistently I needed to power thegearmotor up for two pulses, shut it off and then pause for 100 msec.

Using that regimen, which pushes the positioning stage along one axisat about 0.8 mm/sec, I can print segments of a length of 0.1 mm longwith an accuracy of ~+/- 0.05. That allows me to carefully traceboundaries of a layer. Once that is done I can do diagonal infill atan accuracy of about ~+/- 0.1 mm at a print rate of right at 2.4mm/sec.

Given that I now have an absolute positioning capability I don't haveto reset the positioning system very often. I'm guessing that I canprint at a rate of about 4 cm^3/hr using this firmware scheme. That will let me print gears.

Four down, four to go

Sunday, 9th September 2007 by Forrest Higgs

I've had my youngest daughter here for the past week or so on holiday,so I haven't been devoting as much time to Tommelise 1.0 firmwaredevelopment as I might. All the same, I have been able to make someprogress. Here are a few highlights.

The first big advance is keeping two separate but overlappingcoordinate system storage units. The first is what I call "ideal"because it keeps track of where the extruder head oughtto be. The second I call "absolute" because it keeps track of wherethe extruder head actually is. Mind, both are in absolute units asdictated by the rotary encoder. The units are pulses from the rotaryencoder. Given that I am using 3/8-20 threaded rod, you get...

(128 pulses/turn)(24 turns/inch)/(25.4 mm/inch) = 120.9 pulses/mm-turn

This works out to just under 12.1 pulses/0.1 mm of linear travel alongan axis. I'm averaging about 5 pulses slewing error or about 0.04 mmslewing error. The 95% error rate is closer to 0.09 - 0.10 mm.

The development of the firmware routine to carry this off has become amajor project. Basically, I have to guide the extruder head from onepoint on the xy plane to another along a path determined by the 2Dlinear equation...

y = mx + b

As a practical matter I use that equation for values where (X2-x1)>= (Y2-Y1) where m <= 0.5. Where (Y2-Y1) > (X2-x1), I usethe alternative linear equation...

x = ny + c

This is a pretty tricky thing to accomplish with a compiler, viz,Oshonsoft BASIC, which has neither true integers nor real numberoperations. At first I tried to do all of the real number operationswithin the PC control panel. While that worked it meant that I had ahuge amount of information going back and forth between the PC and thecontroller board.

For me the situation was esthetically unacceptable. The PIC 18F4610 isquite a powerful controller with a lot of flash memory available.

I got around the problem with cardinals rather than true integers bykeeping track of when numbers went negative. To handle real numbers Iresorted to the ancient Greek trick of expressing real numbers as apair of integers in the expression.

N(real) = A(integer)/B(integer)

That left me with 4 conditions to keep track of depending on the signsof the x and y values of the coordinates of the point that the extruderhead was being moved to and twice that for a total of 8 conditions whenthe state of the slope was taken into account. So far the routinedoing all that takes 10 Kbytes of flash memory on the PIC. A lot ofthat will go away when I pull the serial output that monitors what ishappening within the processes.

Right now, I've got everything working for the four cases where (X2-x1)>= (Y2-Y1) and am working on debugging the code dealing with thecondition where (Y2-Y1) > (X2-x1).

All eight done

Monday, 10th September 2007 by Forrest Higgs

My grown "kids", both on holiday, slept in this morning leaving me tomy own devices so I got the four paths for the condition (Y2-Y1) >(X2-x1) done and working.

Now I am going to see if I can print the outline of a proper involute profile gear properly.

First perimeters traced

Wednesday, 12th September 2007 by Forrest Higgs

I got the new xy_move routine working with the rest of the ControlPanel and firmware ensemble this morning. There were a few false startsuntil I tracked down some code in the z_move routine that turns off theinterrupts that was left over from an earlier version of the firmware.

Asidefrom that I hadn't run the z-axis in some time and it was a littlesticky from a slight, humidity driven expansion of the wooden slideassembly. Once I ran it a half-dozen times, however, I had no furtherproblems.

From the looks of it I can keep 95% slewing errorswell below +/- 0.1 mm for line segment lengths and closer to +/- 0.1 mmfor 0.1 mm line segments.

Now, I've got to hook up the code that controls the Mk 1 extruder and I should be good to go.

Refurbishing the Mk 1 Extruder (again)

Friday, 14th September 2007 by Forrest Higgs

I haven't run the Mk 1 in over a month and before that I'd run itpretty much 24x7 for several months. When I started testing the newxy_move routine I noticed that it was having a bit of trouble keeping asteady extrusion rate and seemed to require a much higher operatingtemperature than before.

It sounded like time to schedule amaintenance workup. Usually, rising demand for power in the Mk 1indicates that the nichrome heating coil has begun to separate from thebarrel. Oddly, though, this time the coil was tightly bound to thebarrel.

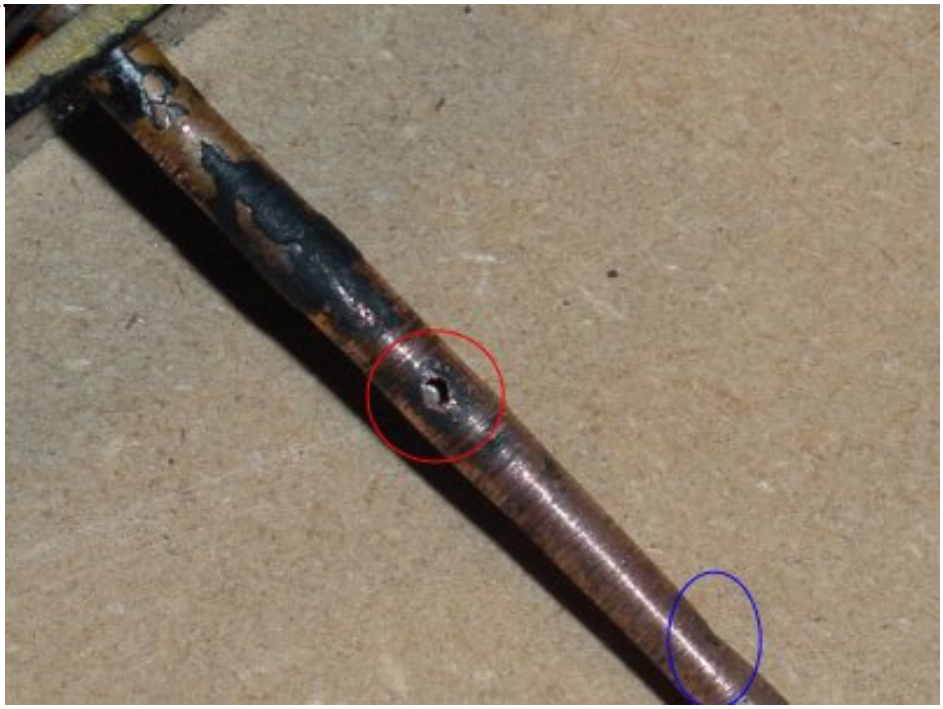
I then checked the wear on the top bearing and the threaded polymerpump shaft. There was virtually no wear on the polymer pump shaft andthe top bearing, which is under the most stress, had worn less than amillimeter, a situation that the spring-loaded pump should have easilyaccomodated for.

I then took a look at the output voltage of the power transistorchecking to see if I'd got some sort of firmware problem into the mixduring the rebuild. Again, no problems there.

Finally I took a very close look at the extruder barrel. It seemed tobe a good time to refurbish the nichrome wire the insulation of whichwas little more than ash. Mind, there was no shorting as a check ofthe ohm rating for the coil quickly revealed.

After stripping the nichrome I sanded off the many layers of bbq paintand discovered something that I'd worried about for some months.

The 5/32 inch thin-walled copper tube used for the extruder barrel isrelatively easy to bend, especially when it is heated. I've hadvarious mishaps with the extruder barrel over the months and found thatit can easily straightened by hand. Ordinarily these little bendsmatter very little, but occasionally you get a slight kink in thebarrel which restricts the cross-section available to the polymerfilament.



When I cleaned the barrel I discovered that I had two of kinks thatneeded to be worked out. You can see the two kink sites circled on thepic. While I was able to straighten the blue circled one acceptablythe red circled kink shattered rather than straightened.
Time for a new extruder barrel.
I am not at all happy about the service life that I am getting from thecopper heated extruder barrel. I guess that I need to go ahead and geta MAP torch and braise one out of brass and see how that goes. A brassextruder barrel would certainly be simpler to make. Unfortunately, Ihaven't been able to get good braised joints on brass with my propanetorch.

Building a "new" heated extruder barrel for the Mk 1

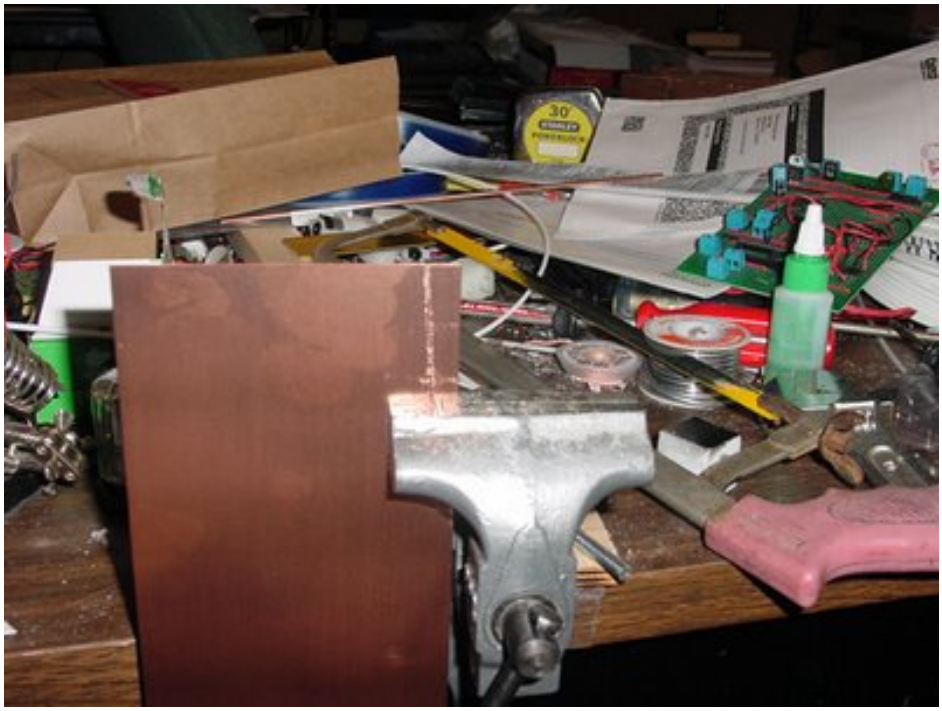
Sunday, 16th September 2007 by Forrest Higgs

I woke up on Saturday at 0100 with a gonzo case of foodpoisoning thanks to my letting a few New Zealand mussels get past theirbest before date when I was making an Udon for supper on Friday night.After a day, a couple of litres of IV fluid, some Vikadin and musclerelaxants I was ready to do on Sunday morning what I had intended tostart on Saturday.

Given my "warmed over death" state of healthI decided to stick with with making another copper heated extruderbarrel rather than attempting to make one out of brass with a MAPPtorch.

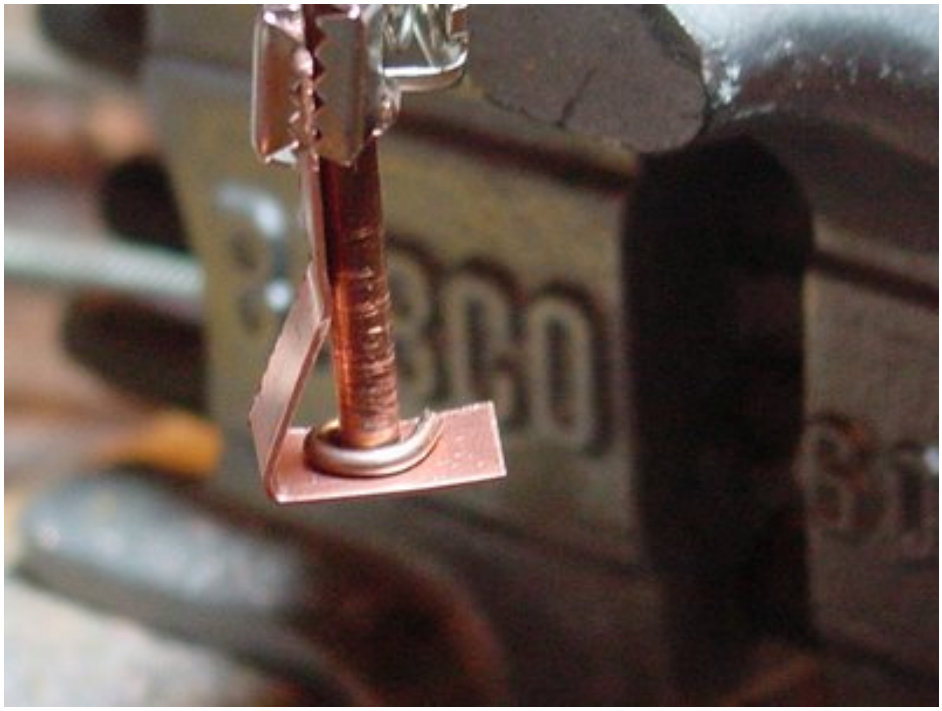
I used measurements off the old one and reused the bottommounting plate. Since you will be clipping and grinding everything thatisn't right on the bottom of the extruder barrel you need take no greatcare with measurements.

Recoveringthe plate was a simple matter of simply cutting the old extruder barreland sliding the mounting plate off with a pair of pliers and my vise.After that I had to cut a .005 inch blank for the extruder orifice.



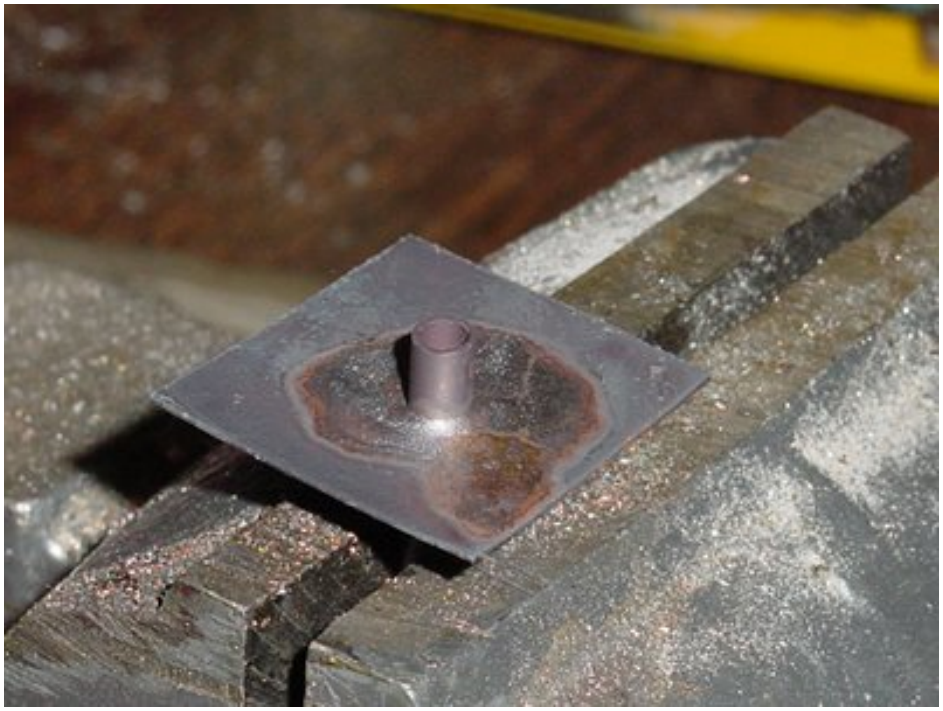
As you can see 0.005 inch copper plate is pretty easy to work with.

Afterthat it was a simple matter of pounding a small dimple into the 0.005inch copper blank so that the orifice would be a little below theextruder barrel. Mind, the barrel is only 5/32 of an inch in diameterso there is not much that overhang in any case.
Once that wasdone I used needle nose pliers to bend the copper blank so that thedimple would seat over the bottom of the 5/32 inch copper tube.



Then there was the matter of bending a piece of phosphor copper braising alloy around the copper pipe and seating it as shown. *Youneed to braise phosphor copper in a well-ventilated place. A cheap 10amp clamp will hold the assembly together quite nicely.* I attach the vise to the clip rather than the 5/32 inch tube in that the copper is quite soft.

Thebraising was done with about 10 seconds of the propane torch. Keep inmind that the hottest part of the torch flame is just a few millimetresbeyond the tip of the combustion cone.
I cut a new PTFE flangeout of copper before I realised that I would able to drill out the oldone quite successfully. All the same, I used the new one, if only forthe look of the thing.



Atthis point I made a mistake, the same one I made last time I buit oneof these. I let the flange rest on the vise rather than using thealligator clip to space it well above the vise. Copper is conductiveenough that I had a hard time getting the 5/32 inch tube to a highenough temperature for the braising melt to spread out properly. Once Irealised what I'd done I let the whole thing cool off and then used thealligator clip to keep the flange well above the vise. There was littletrouble getting the braised joint to work properly after that.

Please note that the 5/32 inch copper pipe achieves a pressure sealwith the PTFE block by sliding into a 5/32 inch seating in the blockrather than depending on threads. I use this strategy on either end ofthe block. In this way I depend on neither the strength of the PTFEthreads nor its resistance to hoop stresses over small amounts of PTFEto achieve the extruder barrel operating pressure.

While this all looks rather intimidating, I did everything that I show pics of in a little over 30 minutes. Documenting it in the blog, however, took about two hours.

If I have time in the morning I will grind off the flash around theextruder nozzle and apply three coats of high temperature BBQ paint. Applying the 6 inches of insulated #32 nichrome wire and sealing itwith three more coats of BBQ paint will probably have to wait tillTuesday in order to let the paint cure properly.

Was it Vikodin or was it just being 60?

Monday, 17th September 2007 by Forrest Higgs

This morning just after midnight I woke up and had a "left thefaucet running" moment, viz, had I put the support plate on my newextruder barrel before I braised the copper PTFE flange onto it. Idecided that I really didn't want to know right then and if I got upand looked at it I'd be tempted to try to fix it. I'm not thatwonderful at midnight work sessions.
This morning, I got up and sure enough I'd left the brass support plate off.

Iused the recycled copper PTFE plate this time to cut down on the amountof sawing I'd have to do. Having had a trial run at it last night Iavoided some of the mistakes I'd made then and got the whole thingdone, with the 0.5 mm hole drilled too, in about 5 minutes. I wasn'thurrying, either.

Here it is.

Finishing the extruder barrel

Monday, 17th September 2007 by Forrest Higgs

I finished up the painting of the extruder barrel about dark yesterdayand let the six coats cure most of today. Late this afternoon Idesigned a 2 amp heating coil of #32 glass fibre and Teflon insulatedNichrome 80 wire. It is rated at 0.32 ohms/foot. Using Ohm's law...

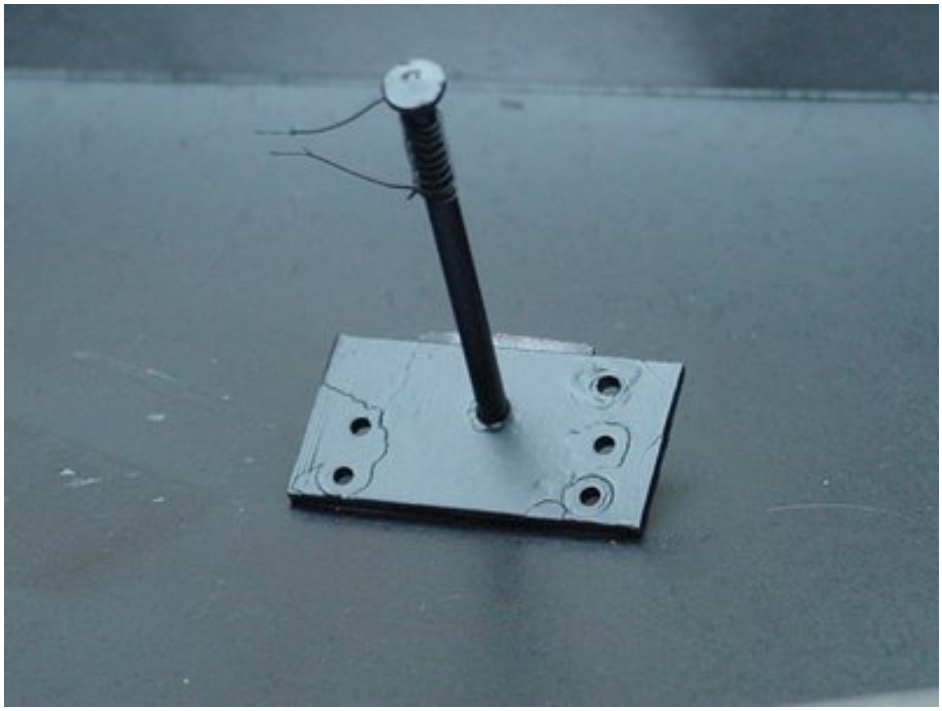
R = V/I = 11.5v (measured off of the power transistor)/2 amps = 5.75 ohms

LENGTH (inches) = (5.75 ohms/10.32 ohms/ft)(12 inches/ft) = 6.686 inches

(Read the full story)

I extended this to 6.75 inches to account for contact losses with the connector.

Iwrapped the wire on the last inch of the extruder barrel as before andtied it down with very thin copper filaments taken from a piece of #20stranded connection wire.



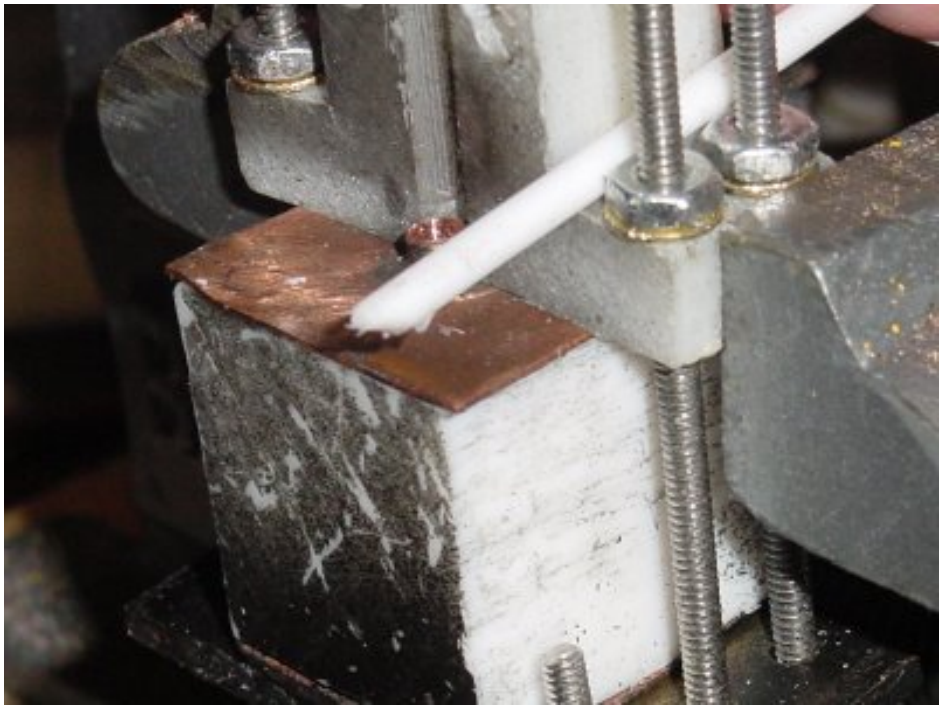
At that point I began to apply six more coats of BBQ paint to secure the heater coil to the extruder barrel.
One word of warning. Braising the copper tube makes it quite soft. Very little force is required to bend or kink it after braising. It'salso the Devil's own job to straighten it out without kinking itafterwards.
I checked and had free travel all the way down to the extruder orifice with 3 mm HDPE filament. No kinking or binding.

Run in

Tuesday, 18th September 2007 by Forrest Higgs

I got the extruder barrel finished this morning and ran it in for a fewhours. After the last coats of paint were done I rebuilt the extruderbarrel/PTFE block/filament guide assembly.
When I rebuilt thebarrel correctly on the second try I used the recycled PTFE blockflange from the old barrel. That was warped, of course, and when Itried to reassemble the pressure system that holds the guide, block andbarrel together, nothing wanted to fit right.

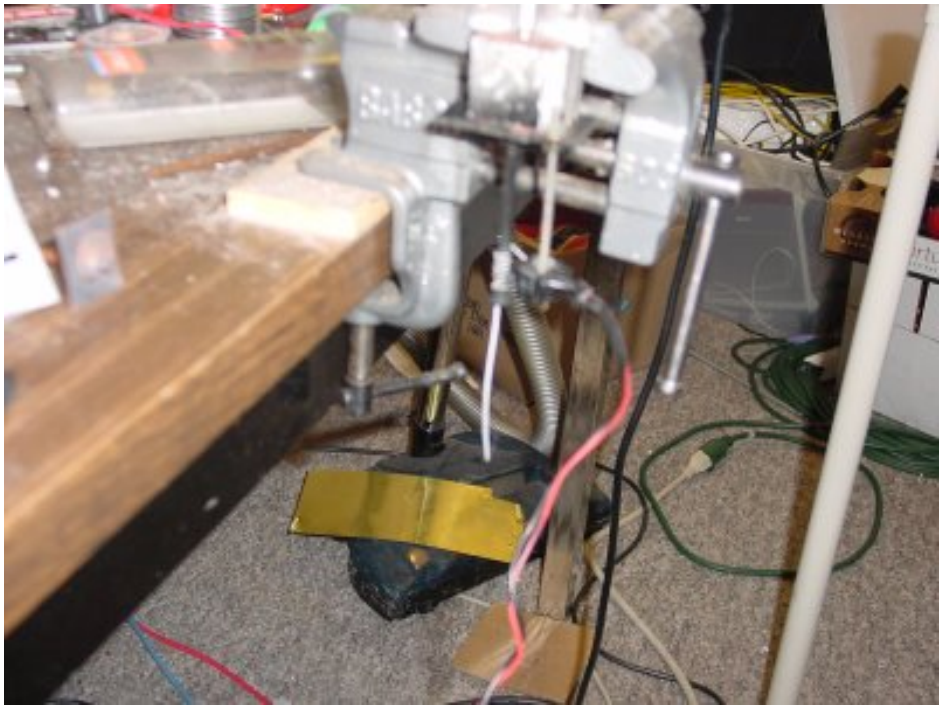
Isolved this by putting the whole assembly in the vise and giving it asqueeze. While I was at it I tightened the nuts and bolts. That washandy.
I had a little troublegetting the alignment of the assembly right so that the filament wentthrough the whole assembly as easily as it did just through theextruder barrel. Using a piece of #4 threaded rod let me get by thatone fairly easily though. Some more thought is needed in this part ofthe design.



Onething that I haven't mentioned which wants saying is that you shouldnever try to thread a bent piece of filament into the pump. As well,you should taper the filament as shown with sidecutters so that itfinds the filament path easily and doesn't snag on the transitionsbetween plastic and metal. **That is important!**

I reassembled the whole Mk 1 and mounted it in the vise for a test run of an hour or so.

Itlit right up at a setting of 90% for the heater and 25% for the drivemotor. I ran it a bit hot and let the filament fall free to the floorto see what I got. Here is the extrusion exercise starting up.



The Mk 1 did a very smooth extrusion that fell about 20" to the floorand coiled nicely, as you can see. It looks like nothing so much ashigh quality fishing monofilament.

My son Adriaan tried to do a Tarzan on the coil to see if there wereany faults in the extrusion. There weren't. It measured out at adiameter of 0.69 mm. This is less than one usually encounters because,I presume, of the stretching the extrusion did on the way to thefloor. I was running it a bit hot, too.

Gears are going to be possible

Wednesday, 19th September 2007 by Forrest Higgs

Just got through with my first 7-toothed gear perimeter. The printlayer has bad settings for height and flow rate, but it's prettyobvious that the work I put in on improving the positioning system,never mind making a new extruder barrel is paying off. The teeth onthe gear all look the same regardless of their location on the gearperimeter. I made them really big so that I could be sure and not foolmyself.

A lot of the problems are coming from the routine that I use togenerate the perimeter. It does the teeth twice at the same level foreach layer. I've got to fix that.

BTW, I've discovered that you don't really need an LED to tell youthat your extruder is operating. The length of #32 Nichrome 80 fromthe heating coil back to the connector glows bright orange whenever theheater is active, so why duplicate functions? :-D

Thoughts on printing at ambient temperatures

Sunday, 11th November 2007 by Forrest Higgs

I hit a technical problem back in September and you haven't heardmuch from me since then. I've been able to print in HDPE and am well onmy way towards sorting out the problems with using brushed gearmotorsto run both the extruder and the Cartesian positioning system ofTommelise. I'm publishing in the main blog rather than the builder'sblog this time in that the issues that I am addressing impact on bothDarwin and the various repstrap machines that are abuilding.

As you are all aware by now, I am sure, HDPE and to a lesser extent CAPA tend to curl when extruded at ambient temperatures.

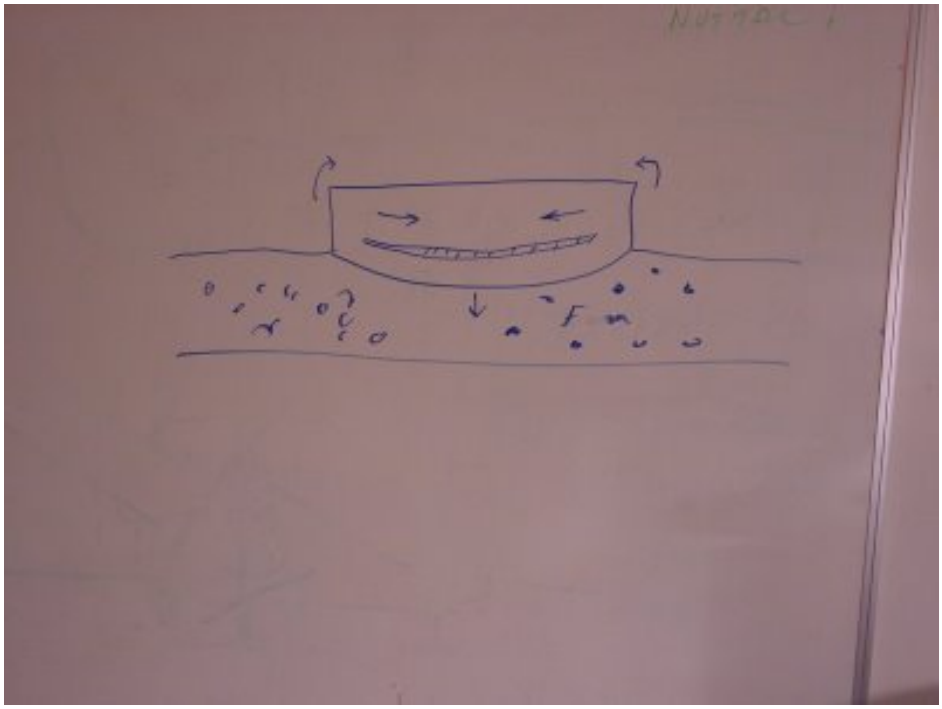
This curling tends to happen at corners when the aspect ratio of object being printed approaches 1. Whenyou have long objects you tend to get curling in the plane of thelongest dimension. Nophead published some really compelling photos of this effect over in the builders' blog last month.

I got very discouraged about the warpage issue and resolved not toshare my misery with the rest of you if and until I came up with aviable solution to the problem.

Somemonths ago, I printed an updated version of the polymer pump forTommelise. I was able to defeat, to an extent, the tendency for thepump to warp during printing by using a lattice infill rather than asolid one.

Itwasn't until I separated the pump from its HDPE printing raft andcleaned it up a bit that I noticed that the back side of the pump hadwarped all the same.

Youcan see that the curling is in the plane of the longest dimension inthis case. You can also easily see a delamination occurring betweenlayers roughly a quarter of the way from the bottom of the pump to thetop. I will talk about this in more detail a bit later
Also notice that the top surface of the pump, the critical one, is perfectly flat.



Here's what I think has happened.

- I've printed the pump onto a HDPE raft much larger in area than the pump itself.

- The pump can't peel off of the foamboard simply because the contact area is much larger than it would have otherwise been.
- Printing each successive layer creates a situation wherecontracting plastic is creating a moment arm which tends to bow theaccreting pump down into the foamboard. The foamboard is notmechanically strong enough to resist this downward thrust.

All that doesn't explain why the top surface is flat and the bottom onebowed, however. In that regard, here is what I think is happening. I've noticed that if I position the extruder nozzle too close to asurface the rate at which plastic can extrude is constricted.

Now whether or not the pump is trying to bow down into the foamboard,the extruder still tries to extrude in a flat plane. When the extruderhead reaches the left and right extremes of the pump it is not able toput down as much plastic as it is in the centre simply because thedistance between the extruder head and the last layer is smaller at theextremes and bigger in the middle.

If I were trying to print a solid infill this situation would have probably caused more distortions in the print than it did.

The top and bottom ends of the pump are square with the top simplybecause the cartesian positioning system keeps returning to the sameperimeter with each layer regardless of the fact that the previouslayer tries to shrink inwards a bit as it cools.

The delamination mentioned earlier happened at exactly the layer wherel stopped printing one evening and picked up again the next morning. Ihad a bit of difficulty repositioning the extruder to start up again. This likely weakened the weld between the new layers and the old ones.

Proposed Solution

It would appear that we might well be able to cancel out the downwardbowing of the print by printing on a solid piece of HDPE instead offoamboard. Solid HDPE has a high thermal resistance

which shouldpreclude extruded plastic from cooling off too quickly and not weldingproperly. Keep in mind that this delamination developed over severaldays and was not apparent when I originally separated the pump from thefoamboard and cleaned it.
Such HDPE planks are easily acquired in the form of cutting boards atany hardware store. They are usually at least 3/8 - 1/2 inch thick andshould have more than enough resistance to bending, especially if I usea loose infil in my print.
Of course, getting the printed object off the HDPE plank will beconsiderably more difficult than getting it off of foamboard. All thesame, using a heated wire to separate the printed object or, failingthat, a hand jigsaw should do the trick. As well, printing the pieceon top of a cross-hatched rick strong enough to resist downward thrust,but less solid than solid HDPE should make separation a bit easier.
The HDPE baseboard should be relatively easy to resurface with a small belt sander. I checked and those are quite inexpensive.
Keep in mind that constraining the printed object in HDPE in thismanner does not mean that it will not warp after it is separated fromthe HDPE baseboard. I'm hoping that it won't to any great extent,however.

Saturday morning ironies

Saturday, 26th January 2008 by Forrest Higgs

As you know, I began to slow down on development work on Reprap last September and then stopped altogether in November. This was not a matter of lack ofdesire, but more one of having had my day job get the better of me. Since September the hours required by my day job begun to run 12-14daily and for the past month or two began to hit sixteen with arequirement of waking up 2-3 times during the night to feed yet moredata into my dual CPU Xeon workstation for analysis.

Recently, I started hitting some deadlines. I've made them and got afew days breathing space afterwards before something else came up. There was one just before Christmas that gave me a chance to think backover what I'd done with Tommelise. The major departure that Tommelisetakes from the mainline Reprap effort lies in its use of shaft encodedDC motors. I've learned a lot about these systems, much of itfrustrating.

The major lesson that I've learned is that shaft encoded DC motors aregreat as long as you have relatively long runs (several mm or cm) thatyou can make. Once you start trying to make rapid changes of directionin the XY plane you have to slow the system down dramatically in orderto get desirable print resolutions. When I was printing large, simpleobjects shaft encoded DC motors worked just fine. I was able to hitextruder head speeds of up to 2.5 mm/sec without a lot of trouble. Once I wanted to do things like print holes and curves,however, I found that my extruder head speeds were dropping down below1 mm/sec. The reason that that happened was that for rapid changes intrajectory the inertia intrinsic to the positioning system became thecontrolling factor. As well, the fact that the extruder itself had a significant time delaybetween when the controller told it to do something and when thatsomething actually began to happen. Probably for some of the brightpeople in Reprap there will be solutions to this problem. I franklydon't feel bright enough, sadly.

Anyway, some months ago Zach Smith put together a sweet little steppermotor controller built around the L297/8 chip set. I've used the L298for about a year now for motor control and quite respect itscapabilities. Having a ready-made controller board design aimed atgeneral stepper motor control seemed to be a good place to startexploring stepper motors again after an 18 month hiatus on my part withthese devices.

I already had several L298's on hand and bought some L297's last week. Those last arrived yesterday.

Now comes the irony.

My little consulting company had a pair of identical HP printers thatwere getting long in the tooth and unreliable. Over Christmas, Ireplaced the both of them with a lovely new high capacity color printerthat is a lot cheaper to operate.

I had several bipolar NEMA 17's left over from year before last, but Iwanted to start working with just a little stepper to get used toZach's board. As a result, I was looking through the Jameco websitefor something cheap that would suit my purpose. For some reason,however, I had avoided making a purchase.

Yesterday afternoon I finally put it together that I ought to salvagethe steppers out of my newly

retired HP printers, so I started takingone apart. *When I got the chassis off and located the motors Idiscovered amids gales of my own laughter that both of the motors inthe printers were shaft encoded DC motors*.

Getting on with Steppers

Tuesday, 29th January 2008 by Forrest Higgs

I've been sorting through my parts inventory to see if I haveeverything that I need to make Zach's stepper board. Now that theL297's have arrived, I pretty much do. About the only thing that isoutstanding is the little trim pot and the connectors. I'm going out toPotter's Electronics and Radio Shack this morning to see if I can scoreone of those. Otherwise it's back to Mouser. Zach's board layout is lovely. Here's what I've got done so far.

I'venoticed that Zach and others wanting to use threaded drive rods withsteppers keep running into the inevitable problem of not being able toget those nice NEMA 17 and 23 steppers to turn fast enough with enoughtorque to give them the printing speed they'd like. I've taken aslightly different tack. Most of the big steppers that are gettingordered are 1.8 and 0.9 degree step beasts. What I ordered is a nicelittle 15 degree stepper.

Insteadof the massive 3/8-24 inch threaded rods I've used before I amdownsizing to a 1/4-20 or 1/4-28 inch threaded rod. That gives me about0.04-5 mm linear displacement per step. It also takes a lot less torqueto turn that sort of rod. The much larger step should let me gethigher linear displacements than we can with the higher resolutionsteppers.

I have no great hope that I've got the first stepper motor first time out. In that I only spent about $6 on it, it's no big loss if it isn't and at least it gives me a motor to test the control board on that doesn't exceed the capacity of the L297/8 chipset.

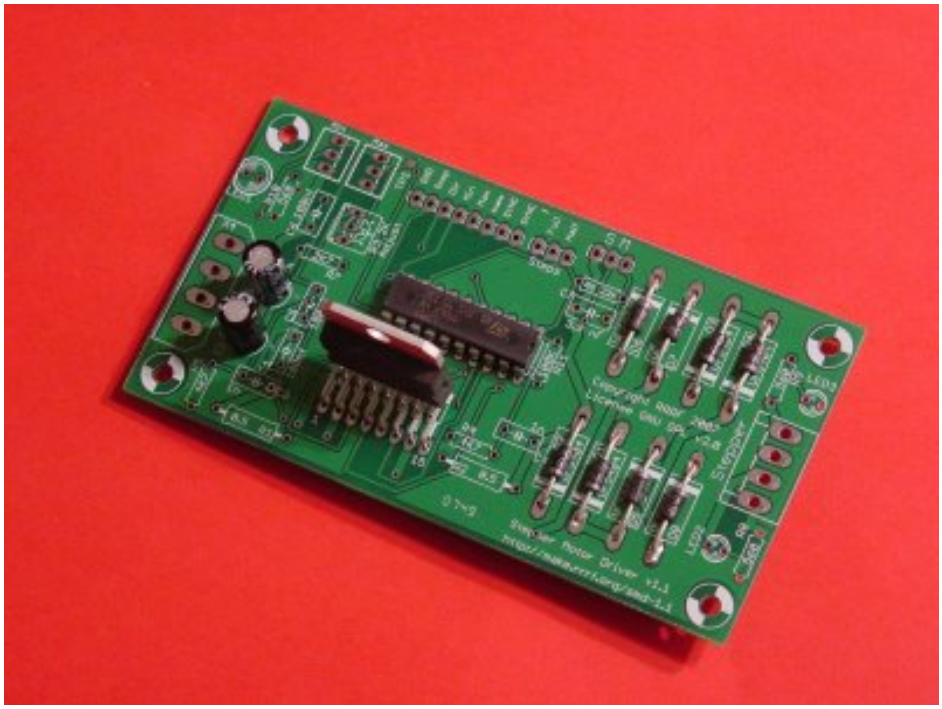
In shopping around for steppers I found a number of potential candidates that might be usuable with Darwin at Jameco. I also found some very interesting steppers in NEMA 23 size at great prices at an outfit just up the road from me called Kysan Electronics. About the only downside to Kysan is that they have a $100 minimum order. That might be great for the RepRap Foundation store, though.

Wielding or wrecking the soldering iron

Tuesday, 29th January 2008 by Forrest Higgs

I took a break this evening from programming a new variablesselector routine and decided to put some components onto one of Zach'sstepper boards. My old Radio Shack soldering iron is pretty much usedup, but I cleaned the nasty old tip and fired it up.

I decidedto do the Schottky diodes first and immediately ran into trouble. Theones that I have in stock are 1N5819's. They're rated at 1 amp and 40volts and are ultra fast. When I got a look at Zach's board, however, Inoticed that the board markings specified gargantuan UF5404's, whichcosts about the same but is rated at 3 amps and 400 volts. I guess thatZach is expecting some heavy duty back EMF off of those NEMA 23's thatDarwin uses. Oddly, though, when I looked at the board's Master Bill of Materialsthe Schottky diodes specified there were SB360's, a Fairchild productalso rated at 3 amps. I expect the BOM got written up after the boardwas designed and reflects a more readily available component orsomething of the sort. It's a little confusing at first glance, though.



My little 1N5819's look really dinky on that board.

I was also able to place the pair of 100 µF electrolytic capacitor cansbefore I made the mistake of finding the replacement soldering iron tipthat I bought ages ago for my cheap soldering iron. Of course, Iimmediately had to replace the cruddy old tip with the shiny new one.

Sadly, I'd left the old one in for far too long, apparently, in thatthe old tip torqued off in the mounting socket of the iron. Thateffectively ruined it and put paid to my soldering efforts for theevening.

It wasn't any big loss, though. I sat down and started working my waythrough what I previously though was a fairly nice stock of resistorsand discovered that I had exactly none that were usuable in Zach'sboard design. Oh well, it's off to Potter's in the morning if therains let up a little bit.

Getting my ducks in a row

Thursday, 31st January 2008 by Forrest Higgs

I finally made it down to Potter's Electronics and got everything Ineeded for making up Zach's stepper controller board except for the 2Ktrim pot and the 2 watt, 0.5 resistor that, I think, is probably not needed in any case.I also picked up an entry level Weller soldering workstation to replacemy dead and gone Radio Shack soldering iron and a set of American-madeside cutters and needle-nosed pliers instead of the Swiss-made RadioShack junk that I had.

I'venoticed that Radio Shack equipment lasts about 6-18 months underregular use. That's great for the sometime hobbyist but a waste of timefor the average committed Reprapper.

It's obviously going to take a few extra weeks, what with ordering fromMouser and all, to get the parts to finish Zach's board and a littledriver board that I will need to tell it what to do. That delay got meto thinking about the little bipolar, 15 degree step, stepper motorthat should arrive today or tomorrow.

That little item only draws 0.44 amps at 12 volts while Zach's controlboard will supply up to 2 amps per phase. What I really want to knowis whether such steppers

- provide enough torque to push the xy positioning table on Tommelise 2.0
- run fast enough to improve the printing speed significantly beyond what the Solarbotics DC gearmotors will provide.

Last night it finally hit me that I could answer those questions withmy existing Tommelise 1.0 controller board via reprogramming and usingtwo of the SN754410 Quadruple Half-H Driver chips that it already hasinstalled. Those are good for 1 amp/phase, much more than what thelittle stepper motor requires.

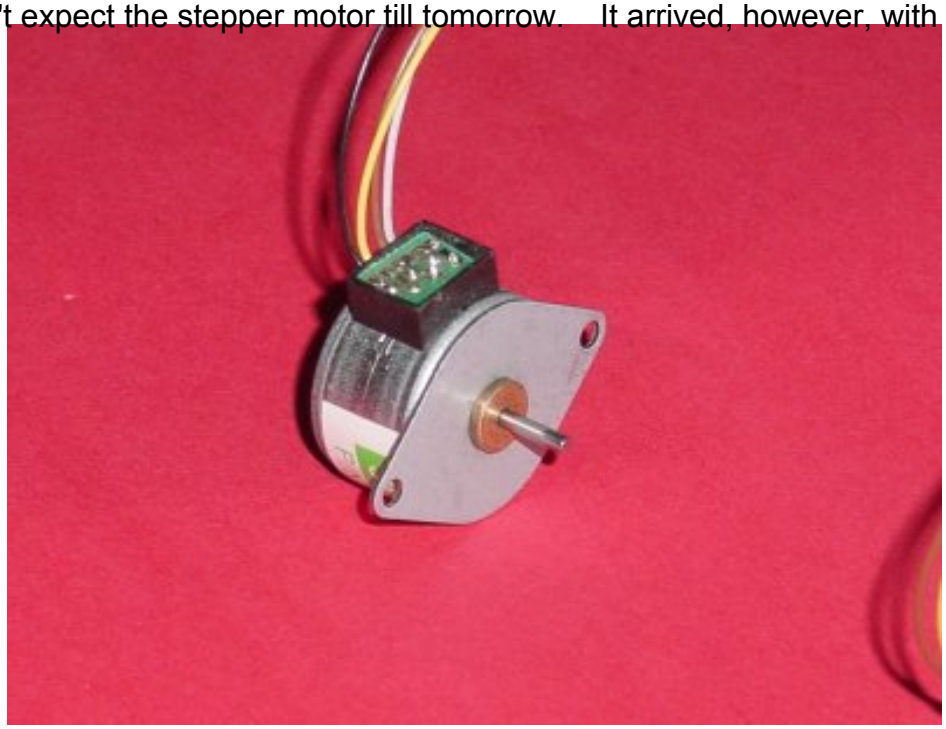
The nasty thought then arrived that should that be the case it might bemuch more cost effective to design a stepper friendly Tommelise boardusing cheap SN754410 chips and some of the old 16F628A chips that Ialready have on-hand than to use specify Zach's board which is designedto handle much more robust stepper motors.

Of course, all of this depends on the small stepper motors being ableto supply enough torque and high enough speeds. That remains to beseen. The upside is, however, that I will be able to answer thatquestion this weekend rather than sometime next month.

Stepper motor arrives

Thursday, 31st January 2008 by Forrest Higgs

I didn't expect the stepper motor till tomorrow. It arrived, however, with the noon post.



Turnsout that they went from the old style system which was a geared stepperto the new style which is lead and cadmium free but which also isgearless. They are pretty sure that it has the same torque reading, butthe can measurements are pretty much the same, so I'd take leave tosuspect that that's not so.

I pulled out the digital multimeter and measured the phase resistance and got 27.3 which is just about what it said in the catalog. It will be interestingto see what sort of torque and speed that this thing delivers. Staytuned.

Checking out the stepper

Thursday, 31st January 2008 by Forrest Higgs

I was able to establish the stepping sequence for the 15 degree, bipolar stepper with a 1.5 v battery. It is

+	-
Black	Brown
White	Yellow
Brown	Black
Yellow	White

More on skateboard bearings

Monday, 18th February 2008 by Forrest Higgs

I wandered down to a local surf and skateboard shop on Saturday and bought a handfull of skateboard bearings. They're all grease packed and, according to the shop's owner wear out after a few months on a board. That's not particularly good news given how many hours of constant service you can expect to be putting a home-built 3D printer to.

3D Scanning for stupid people...

Monday, 18th February 2008 by Forrest Higgs

...like me. :-D

I've been mulling over ways to get my little Cyclops 3D scanner projectgoing again with a minimum of effort. I decided to leverage what Ialready know how to do rather than get into all the drama of patchingmultiple scans that other open source 3D scanning systems like David do.

Daviduses a line projecting hand laser and a little PC web cam and a LOT ofPC processing power to get you a 3D surface description of an object.It occurred to me that I already know how to make both encoders,gearmotors and steppers to work with PIC's. If I were to buy a littleSharp IR distance sensor like one of these.

Icould shortcut much of the software sophistication that the Germanshave gone through with David. You point one of these little chips atyour object and read off a voltage and then translate the voltage intoa distance. Mount one of these to swivel up and down and keep track ofwhere it is at and put your object on a turntable with the same kind oftracking and you've got a very simple IR radar set that can easilydevelop a surface description of an object.

The only wild card was how much beam dispersal you got. I talked withMatt at Acroname this morning and discovered that you can see the beamthrough an IR camera and that the dispersal that you can see off of thecentreline is running no more than 0.8 degrees. What that means isthat most of the power in that spot is running in about 0.2-0.3 degreesoff of centreline.

I bought a couple and look forward to seeing how they work.

I can also leverage this tech into some robotic 3D imaging to let amobile robot reconstruct its surroundings. I have a project that hasbeen on the back burner where that would be very handy.

Bingo! The PC-side Visual Basic 6 problem solved!

Tuesday, 19th February 2008 by Forrest Higgs

As I reported earlier, I got the PIC18F4550 prototype board working andwas able to talk to it from my PC using the compiled Visual Basic 6 appthat Vladimir Soso of Oshonsoft thoughtfully supplied along with hisUSB extensions to his PIC 18F family BASIC compiler. Things clouded up and rained, so to speak, when I tried to translatehis Visual Basic 6 code into Visual Basic .NET. I kept hammering at itand talking to friends who had also done a lot more transition of codefrom Visual Basic 6 to Visual Basic .NET 2003 than I had. I'd, afterall, only transferred about 150K lines of my own coding. I shudder tothink how many lines poor Tyson transferred about 4 years back. Afterchatting with Tyson and about 10-12 hours of playing chimpanzees onkeyboards in my spare moments while I was doing production runs, Ifinally came to the conclusion that I was wasting my time trying to runVladimir's HIDTerm.dll in any of the .NET IDE's.

It was at that moment that the Google Angel dropped a nice little page hit on me...

http://www.lvr.com/hidpage.htm#MyExampleCode

Jan Axelson writes books about developing code for talking toperipherals via serial and USB ports. On the above link, he gives setsof sample code with which you can test Vladimir's board and from whichyou can begin development of your PC-side software to drive your USBapp using such languages as...

Visual Basic .NET (separate examples for multi-threaded and single thread)

Visual C# .NET

Visual Basic 6

Visual C++ 6

I downloaded his uncompiled sample code for Visual Basic .NET and ranit in debug with no hiccoughs whatsoever. It immediately identifiedVladimir's board and was able to write and read back 2 bytes of datafrom the board. Upgrading the code to make it handle the full 8 bytesthat Vladimir's Oshonsoft IDE makes possible should be no big deal.

I like Jan's code because you just have code to play with, no black box.ocx's or .dll's to cope with. I'm definitely going to buy his twobooks on USB comms and building USB peripherals from scratch. For youguys still herpling along with serial he's got a few good books onserial comms, too!

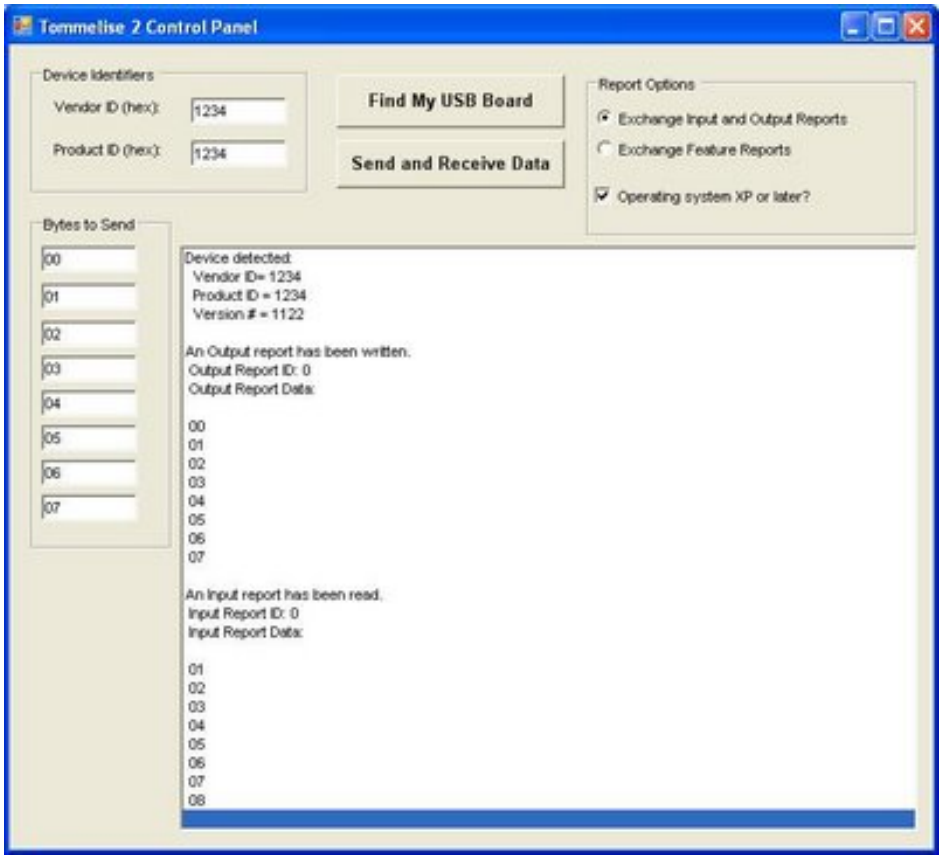
So... People we are good to go. Vladimir's firmware support for USBseems to work very sweetly on the PIC18F4550-side. He also supplies aVisual Basic 6 example in compiled and code form using either an ocx ordll that works fine (I've tried the compiled forms, mind.) Jan Axelsonprovides good code (I've checked the VB.NET multithread example and itworks nicely.)

I'm ready to rock, which is good because my stepper motors for theproject just arrived in the post. I've got boards to build! :-)

Good to Go!

Wednesday, 20th February 2008 by Forrest Higgs

I threw away about 90% of the features that Jan Axelsonwas kind enough to include in his very comprehensive VB.NET sample codeand kept the bits that I needed to keep a nice, fast data link goingbetween Tommelise 2.0's PIC microcontroller and the PC.



Inow have the equivalent of Vladimir's test code, except now it is .NETand it is multi-threaded as well. The next step is to slap an SN754410onto the prototype board and get a write code on both sides to startdoing stepper motor control directly from the PC. This is REALLY nice!

Extending the USB prototype board

Thursday, 21st February 2008 by Forrest Higgs

My eyes finally gave out this afternoon with staring at thepainfully slow unfolding of optimal variable selection for neural netswhile I was doing quality assurance for a new piece of productionsoftware. Fortunately, I was able to launch a test run of several hoursand got a few minutes to rest my eyes and do something different for achange.

The "different for a change" was actually not all thatdifferent. I cobbled an SN754410 quadruple half-H driver chip onto myworking prototype USB board.

It's not exactly an elegant design, but it works. I'm really startingto hate these last few spools of solid core wire I bought fromPotter's. The plastic insulation on the wire shrinks outrageously whenI use it to connect strips. I didn't have nearly that kind of troublewith the wire from Radio Shack. Unfortunately, Radio Shack wire tendsto come in two colours, orange and black, which is nice for AllHallow's Eve decorations, but not very useful for many otherapplications.

Anyway, I just finished a continuity and voltage check under power. Ihad one missing wire and one dry joint, both of which I fixed. I'll dothe check over again in the morning after I've been awake a few hoursto double check my work.

I've found that if I do wiring up one day followed by a first boardcheck a few hours later with my multimeter and then a second check thenext day before I start plugging in chips I save myself a lot of dramawith chips crackling like distant gunfire and overheating voltageregulators. It only took me about 25 years to figure that out. :-s

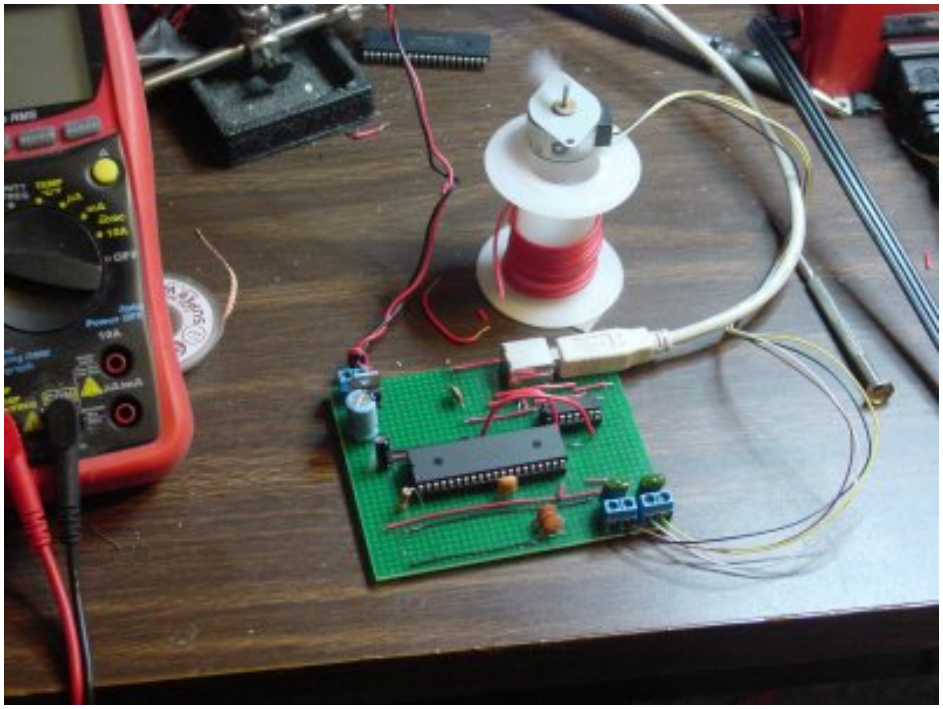
I should be able to try to crank up one of the new stepper motors either tomorrow or Friday at this rate.

Stepping out with USB

Saturday, 23rd February 2008 by Forrest Higgs

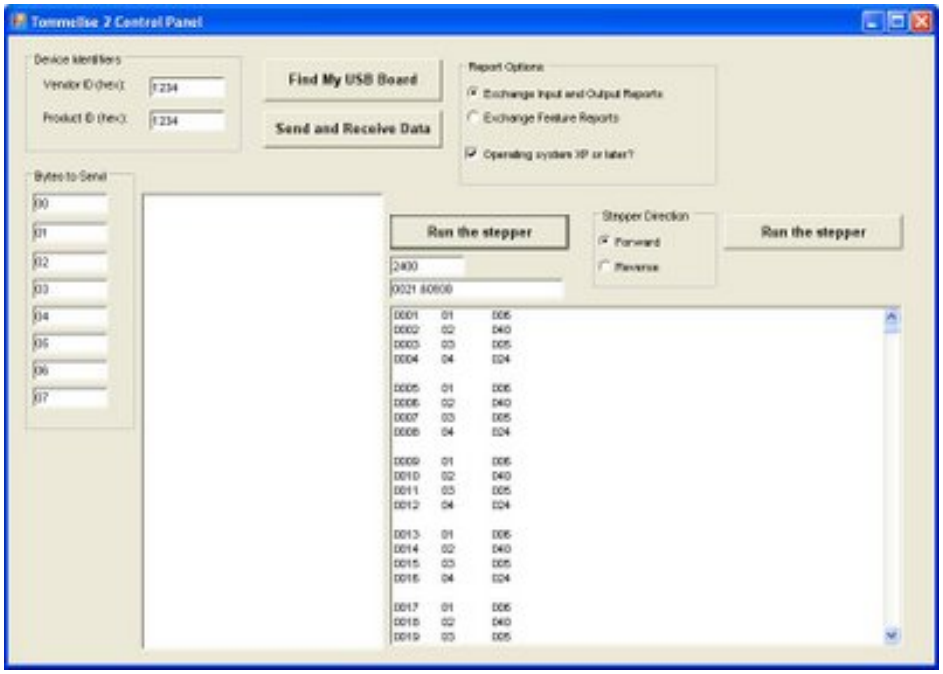
Got it! Eat your hearts out! :-D

Running a stepper motor from the USB prototype board was a little trickier thanI'd hoped, but a lot simpler in other ways. The tricky part had to dowith the tricky little ways of the 18F PIC chips. I set up the stepperto run on 6 pins of Port D. For the longest time I could write code onthe IDE and simulate it and it would do what I wanted. When Iprogrammed the actual chip, however, it wouldn't.



Finally,I remembered some trouble I'd had with the 18F4610 that I used withserial comms with Tommelise 1.0. If Microchip had a pin for everyfunction on the bigger 18F chips you'd have about 100-125 pins whichsimply isn't manageable for a DIP configuration, so they combinefunctions share them on pins and let you select which functions youwant by setting bits on system configuration bytes.
The problemis that sometimes how the datasheet tells you to select a particularfunction isn't quite how it's really done. It turns out that Port Dshares pins with a streaming output port. The sharing isn't exactly avery happy one. As I'd found with the 18F4610 sometimes you not onlyhave to select pins with a Tris command but also do a Lat command toget the darned port to select over some other feature. The 18F4610 hasthat problem with Port C. The 18F4550, which I am using now for it'sUSB compatability, does as well.
When I did a LatD command half of the pins I needed started working. Pins 0 and 1, however, didn't work whatever I did.
Finally,after a lot of reading on the datasheet, I discovered that the USB canpush data straight to the streaming port and that a particular bit wasset to make that happen. The LatD got me back 6

of my 8 pins.Unfortunately, I was using pins 1 and 2 which the LatD command couldn'tget back for me.
I tried setting the bit, but something inVladimir's implementation of USB kept that bit fixed, so PortD.0 andPortD.1 simply wouldn't work under any circumstance, or at least anycircumstance that I could confect. Once I accepted that, I simply movedthe lines that PortD.0 and PortD.1 were controlling on the SN754410 toPortD.6 and PortD.7 . After that, things worked perfectly.



Rightnow I can drive the stepper at about 6.4 mm/sec on one axis or 9 mm/secat a diagonal assuming a 20 threads/inch pitch. I am sure that as I getto know USB better I am going to be capable of a lot more.
Thereally nice bit is that I can generate what the port settings ought tobe on my PC and then blast them over to the 18F4550 via the USB link.The firmware needs only to set the bytes it receives to the appropriateports and keep track of the limits detectors to make sure thatTommelise doesn't run off the rails. This makes for very, very simplefirmware programming.

To every thing, there is a season...

Sunday, 24th February 2008 by Forrest Higgs

Well, I've been putting it off, but with the success of the USBprototype controller board, it is finally time to say goodbye toTommelise 1.0.

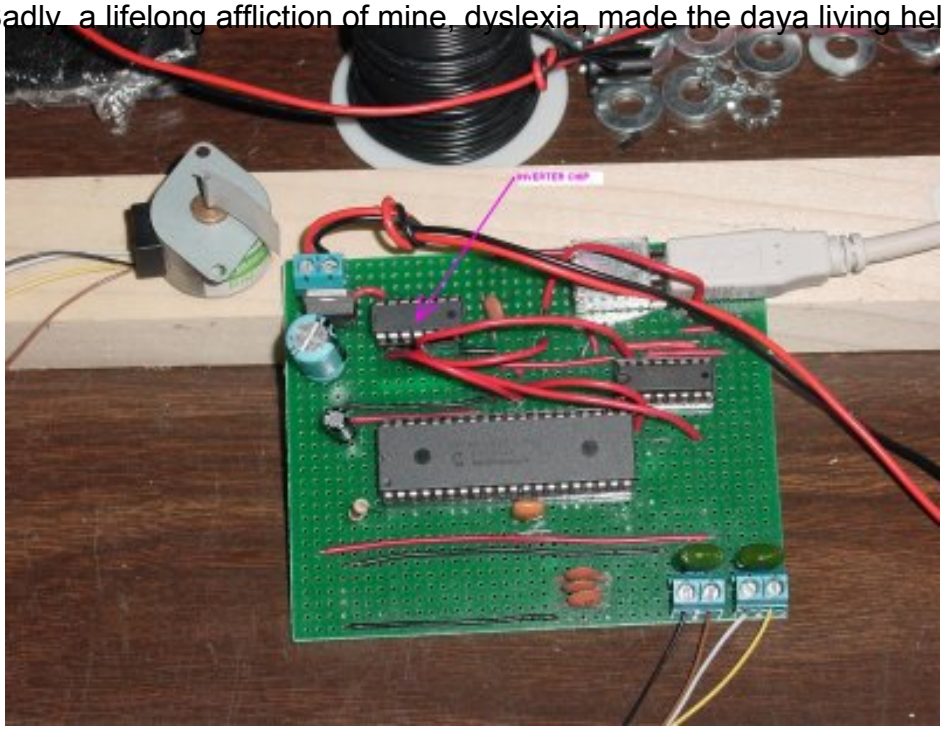
I'm clearing a place under another work table for it. Snake themascot will be moving to Tommelise 2.0 as soon as there's a place forhim to coil around. :-)

And so, we begin again.

Dyslexia and the MC14069UB chip

Sunday, 24th February 2008 by Forrest Higgs

I had planned to get a lot done on Tommelise 2.0 today. The good Lordknows that I worked hard enough to have got a long way. I had plannedto graft the MC14069UB inverter chip onto the prototype board and seehow big a problem it was going to be to save a few pins on the PIC18F4550. Sadly, a lifelong affliction of mine, dyslexia, made the daya living hell.



Things started nicely enough. I grafted a 14 pin socket onto theprototype board and isolated it from the 18F4550. Then I added anotherconnection to the output pin that is the first of two pins going to theSN754410 quadruple half-H driver chip that controls the first phase ofthe stepper and connected it to one of the six inputs for the MC14069UBinverter chip.

When you put a 5 volt signal into an inverter, you get 0 volts output. Likewise, when you put 0 volts into an inverter you get 5 volts. Thisworks beautifully for direction for the SN754410 in that if you wantit to go one way you put in 5 v and 0 v respectively into that phase'sdirection pins. Similarly, if you want it to go the other, you put 0 vand 5 v into the pins. Ordinarily, it takes 3 pins per phase for abipolar stepper, one to tell the SN754410 whether the phase is on oroff and two to tell it which direction to go in (polarity for youpurists). Using an inverter means that it takes two microcontrolleroutput pins to run a phase of a stepper instead of three. This doesn'tsound like much till you realise that we have to drive three bipolarsteppers and a DC gearmotor on Tommelise 2.0. That is a savings ofseven pins. When you consider that the PIC 18F4550 only has anabsolute maximum of 32 output pins, that's quite a savings.

Anyway, I got the first phase of the stepper wired up and tested andthings looked fine. Right about then my dyslexia struck and everythingwent south. Connector wires seemed to dance over the board and I wasconstantly swapping wire locations for the next 5-6 hours.

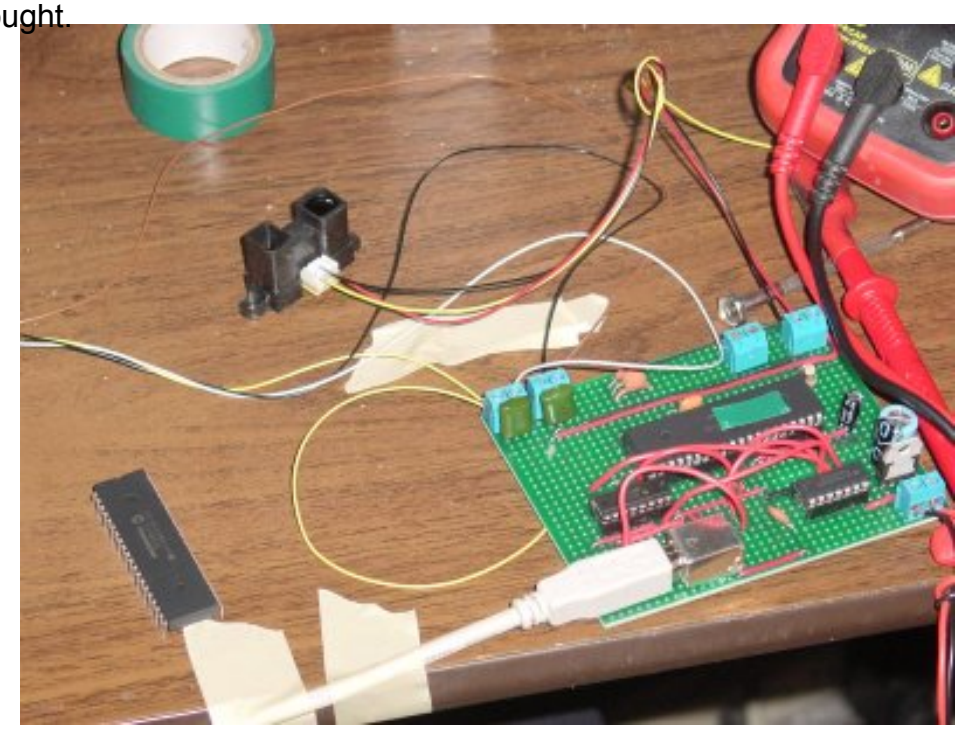
I aggravated the problem with a spot of creativity. I suddenlyrealised that I could control a whole stepper motor with 3 pins insteadof 4 if I gave up the possibility of half-stepping. Trying out thatoption led me to fry my inverter chip with 12 v power from the steppermotor when I got a spot of solder where it didn't need to be.

After I sort of got that fixed and a new inverter chip in the socket, Idiscovered (I think. I'm not certain) that one output port of theinverter chip does not have enough milliamps to drive two pins on theSN754410, which the 3 pin scheme needed to happen. After a few morehours sorting out my dyslexia-scrambled wiring, I dropped back to theoriginal 4 pin scheme.
That works now.

Preliminary measurements on the Sharp IR distance measurement chip

Monday, 25th February 2008 by Forrest Higgs

I did a quick lashup for doing some rough measurements with the Shart IR distance measurement instrument I bought.



Hereyou can see the instrument itself to the left and above the prototypeUSB controller board. I slapped on a few extra screw terminals tohandle the three lines off of the instrument.

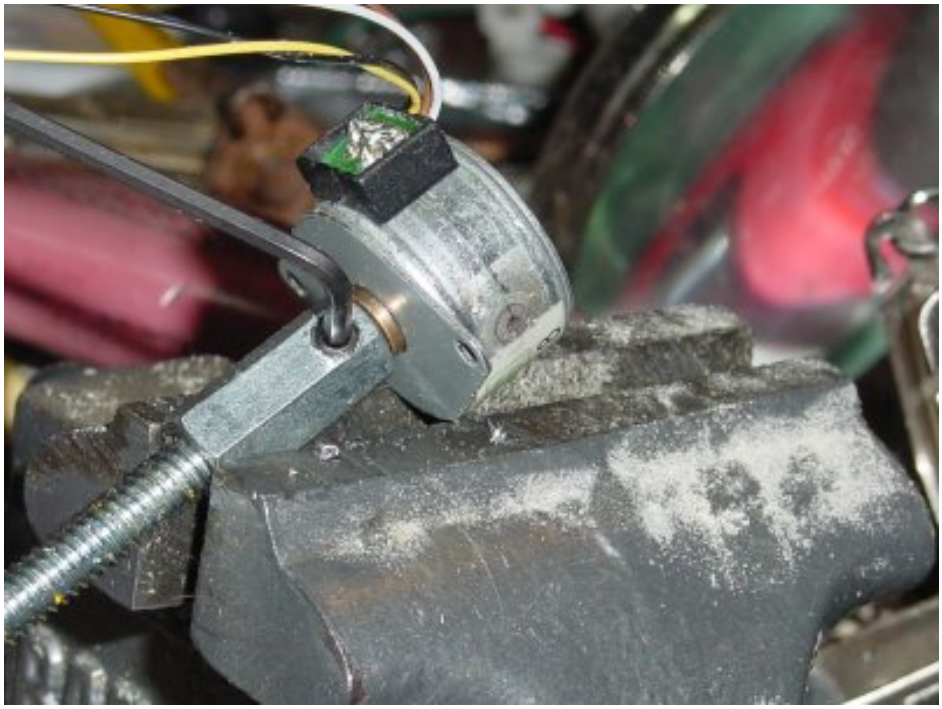
I was able to get it to spot a piece of 3 mm HDPE feedstock at 500 mmaway. This one has a

range of about 1500 mm and a voltage range of0-2.75v. Given that it costs right at $12.50, that's not badperformance at all. Sharp has another model which is focussed in to300 mm. That might a nice one to mount on a Reprap to scan an ongoingprint job.

Steppers and couplings

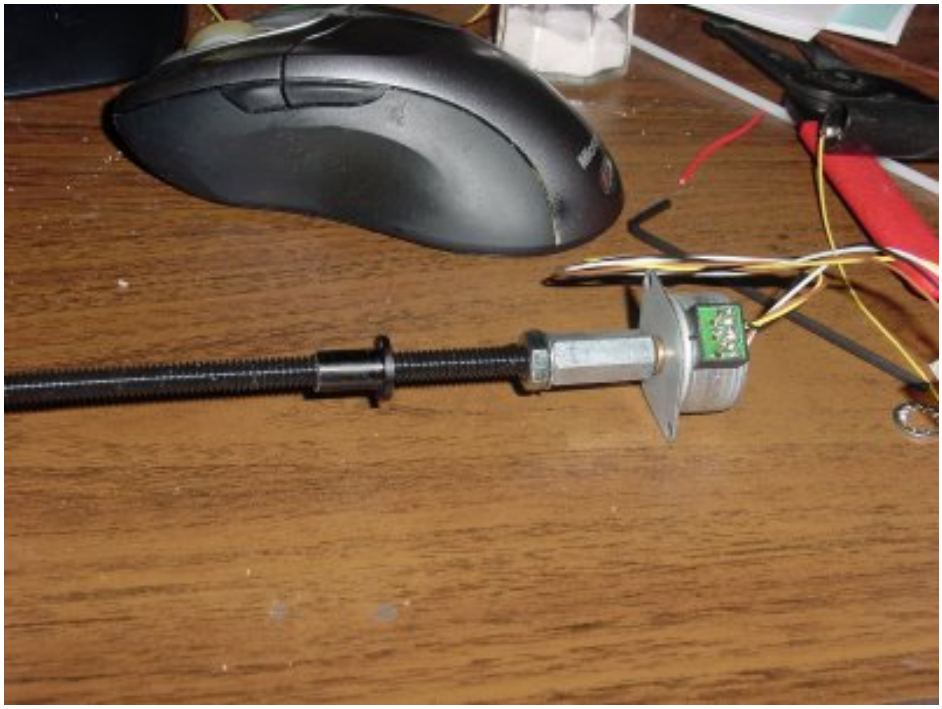
Wednesday, 27th February 2008 by Forrest Higgs

It's been quiet for the past few days. My day and night job have really been eating into my Reprap time.
MyUSB work has pretty much hit a plateau until my Jan Axelson book on USBprogramming arrives. That will be towards the end of next week. In themean time I've spent a few hours hammering at the stepper instructionstill they're all working right. I'm also getting used to all of thethings that can come unstuck on my stepper board and how to diagnosethem from how the stepper is behaving.
I slapped my multimeteron in amperage mode and discovered that one phase of the motor consumesa max of 220 milliamps with no load. That drops to about 190-195 as Icrank the speed up. When I accidentally leave one of the phases chargedit jumps to about 340-350. When that happens both the motor and theSN754410 heat up quite quickly. As a result of that I reprogrammed thefirmware to turn off both phases 10 millisecond after the end of a run.That keeps the motor and the SN754410 quite cool. That's good, becausethe motor tends to skip when it heats up.
Havinghit a dead end there for the nonce, I started preparing a prototypeaxis. First thing I needed was a coupling. Since I found the #10-24nylon rod too floppy for horizontal axes I shifted over to 1/4-20. Idecided, therefore, to make a coupling from a piece of steel 1/4-20 boltand a coupling bolt.



It'spretty easy to do. You just screw in a short 1/4-20 machine bolt intoone end of the coupling nut and then drill it with a 5/64ths bit, whichis just a shade larger than the 0.08 inch stepper drive shaft. Since Ido this with a hand Dremel tool, if I get ithe alignment wrong I justthrow away the bolt and get another one. When I have one good enough Iscrew it back into the coupling nut leaving the head of the bolt abouta hacksaw blade's width short of flush with the coupling nut. I thenscrew in a longer bolt from the other side of the coupling nut to lockthat one into place.

Then I cut the head of the short bolt offleaving the shaft of the bolt screwed into the coupling nut. At thatpoint I use my 5/64ths bit to drill a guide hole into the top side ofthe coupling nut and then drill it out halfway through with a #24 bit.
Atthat point I can easily tap the hole through the side of the couplingnut and halfway through the bold shaft with a #10 tap set. Once that'sdone I can seat a headless Allen bolt in the threaded hole and use thatto lock the shaft of the stepper to the makeshift coupling.



It's then easy to seat the 1/4-20 nylon threaded rod and lock it onto the coupling with a 1/4 inch nut and lock washer.
You can see a 1/4 inch nylon bushing sleeve on the threaded rod. Although I have M8 bearings which seat on this sleeve quite nicely Iam, for now, going to see if just the bushing alone with be adequate.

Tommelise 2.0 goes cross-platform

Friday, 7th March 2008 by Forrest Higgs

One of the big complaints from the core Reprap folks about Tommelise isthat it is a Wintel phenomena. Some things I'm doing in my day job maywell allow me to deal with that criticism.

As I'd mentioned some time ago, Microsoft is offering Visual Basic .NET 2008 Express for free. While it's not open source, neither is Java.

The big news, however, is a little piece of open source software thatenables you to write things in Visual Basic .NET and a dozen otherlanguages and run them on Linux machines as real apps rather thanemulating Windows within Linux.

Check out Mono.

Also get a load of the languages it supports.

Lordy, it even supports Oberon!

A bit further with USB

Saturday, 8th March 2008 by Forrest Higgs

I managed to get some time to work on Tommelise USB. Some days ago I finally unearthed my copy of VB6. I disconnected the stepper motor code and just tried to do USB I/O.

I did it first in VB.NET 2005 and then in VB6. Both were doing I/O at a rate of about 800 bytes/sec, which definitely the low speed setting. I disconnected the write/reads in the VB codes in both VB compilers and it is definitely the I/O, not the VB that is running slow.

I'll have another go at resetting the configuration values again in the morning and see if I can break it out of this 800 bytes/sec thing.

Edging into full speed USB

Sunday, 9th March 2008 by Forrest Higgs

After twiddling with the configuration I was able to establish that I was moving bytes faster than the 800 bytes/second that Jan Axelson says you get with USB Low Speed, so I guess that I'm edging into Full Speed. Right now I'm getting about 2000 bytes/sec. Full speed is supposed to be giving me as much as 1.2 megabytes/second, so I'm not doing all that well. Still, it's progress.

Following that, I shifted the USB port polling to an interrupt-driven scheme. That allowed me to separate the USB comms from the operation of the stepper motor. That didn't seem to affect the performance of the USB link while I was able to run the stepper motor at quite high speeds of a sort that I haven't seen since my initial tests on the 18F4610 board from Tommelise 1.0

That is real progress! :-D

Digging into the 18F4550

Monday, 10th March 2008 by Forrest Higgs

I appear to have got this USB-enabled 18F4550 chip to to what I want itto do. It has a rather complicated pre and post scaler scheme for theclock crystal which enables you to run it at its maximum speed of 48MHz regardless of what kind of clock crystal you use. That's VERYhandy.

You can also run the USB module and the CPU at different speeds, whichlets you do high speed USB comms with a low speed CPU. I guess that'shandy, too, though how escapes me. I have a natural tendency to runelectronics and most machinery as fast as I can make it go. That'sjust a personality trait of mine, I guess.

Anyhow, I was able to get the servicing of the USB comms done withinterrupts which left the vast majority of CPU cycles free to run thestepper. Once I did that I was able to get some respectable stepperspeeds out of it.

About the most speed that I could reliably get out of my little Jamecostepper was about 460 steps/second. That doesn't sound like a lot tillyou remember that it has a 15 degree step. Driving a threaded rod witha 1 mm pitch means that you can hit about 19 mm/sec transition speed onone axis. There's not a lot of torque available at that speed. I planon running Tommelise 2.0's axis at about 8 mm/sec which means that Ineed a very leisurely 192 steps/sec. There is quite a bit of availabletorque at those speeds.

I've noticed a little thing about the 18F IDE that I want to pass alongfor all of you who are using USB-enabled PIC 18F's and Oshonsoft'sBASIC firmware compiler. It may apply to more chips, but I can onlysay that I've only seen it with the 18F4550. I've also used the18F4610 which isn't USB-enabled and I haven't seen this "problem" withit.

Before all of you C language jocks yawn and ignore the rest of thepresentation as just more of Forrest's antediluvian maunderings aboutBASIC please follow what happened. I suspect that Vladimir atOshonsoft's "mistake" with his BASIC compiler is very likely duplicatedwith other firmware compilers including C language ones simply becauseof the peculiar nature of the USB-enabled PIC chips.

Here is what I've noticed.

I'm using the 18F4550 to do USB coms with my PC and drive a littlestepper motor. I've been having some trouble getting Full Speed out ofmy USB connection, but that's a topic for another posting.

I am using WaitMs to set the timing for steps with my stepper motor. Iset the step interval for 5 msec. When I measured the speed of thestepper, however, I was able to do 250 revolutions in 13 seconds. I'musing a 15 degree stepper motor so what that means is that in those 13seconds I made...

(250 revolutions)(24 steps/revolution) = 6000 steps

I did that in 13 seconds which means...

(13 seconds)/(6000 steps) ~ 2.17 msec/step

which is interesting because I programmed...

WaitMs 5

for each step.
I dug through the datasheet for the 18F4550 and think that I may have figured out what happened. The 18F4550 has a peculiar handling for clock speeds. It's very nicebut a bit complicated. Microchip wanted you to be able to run highspeed USB comms regardless of how fast you want to run your chip andthey didn't want you to have to install two crystals, one for the CPUand one for the USB.
Here is what they did. The chip has an internal clock which runs at 4 MHz or you can install a faster crystal. That's nothing unusual. Whathappens after that is unusual. The 18F4550 has a prescaler whichcranks whatever crystal speed you utilize up to 96 MHz and then onepostscaler for the CPU and another for the USB. If you postscale forthe CPU you can divide that 96 MHz for a maximum CPU speed of 48 MHz,which is what I did.
What I noticed is that when I divided the wait that I specified by what I observed...
(5 msec)/(2.17 msec) ~ 2.3
Then I noticed that when I divided the postscaled CPU clock speed by the crystal speed
(48 MHz)/(20 MHz) ~ 2.4
That's pretty close seeing as I was measuring the time my stepper wasrunning with my digital watch, which has a stopwatch feature.
What I am guessing is that Vladimir uses the statement...
Define CLOCK_FREQUENCY = 20
to calibrate the WaitMs and probably the WaitUs functions. For theoverwhelming majority of PIC chips that should work perfectly. Forthese peculiar USB-enabled PIC chips with their pre and post scaling ofclock speeds, however, it might not be.
When I did the post "Curiouser and curiouser" a few days ago, Chris (nophead) asked...
So is the static torque the same then?Are you saying torque falls of faster with speed?Am I correct in thinking you are using the same driver chip, just with an inverter in h/w instead of s/w?
Whatwas happening is that the 18F4550 was trying to run the stepper about2.3 times faster than the 18F4610 was due to the pre and post scalingof the clock frequencies.
I was estimating stepper speeds using the WaitMs statements. It's turned out that...
WaitMs 5
in the 18F4610 is really about...
WaitMs 2
in the 18F4550.
With the 18F4550 I was running the stepper faster and was thus furtherdown the torque curve than I had been with the 18F4610. That seemsto explain the deterioration of performance that I experienced when Imoved from the old 18F4610 board from Tommelise 1.0 to the 18F4550prototype board for Tommelise 2.0.
My guess at the time that...
It definitely seems to be something to do with the CPU. I'm guessing that there is something about that dual timer scheme that I'm not understanding.
turned out to be right. It was sure painful turning that guess into a solution, though.

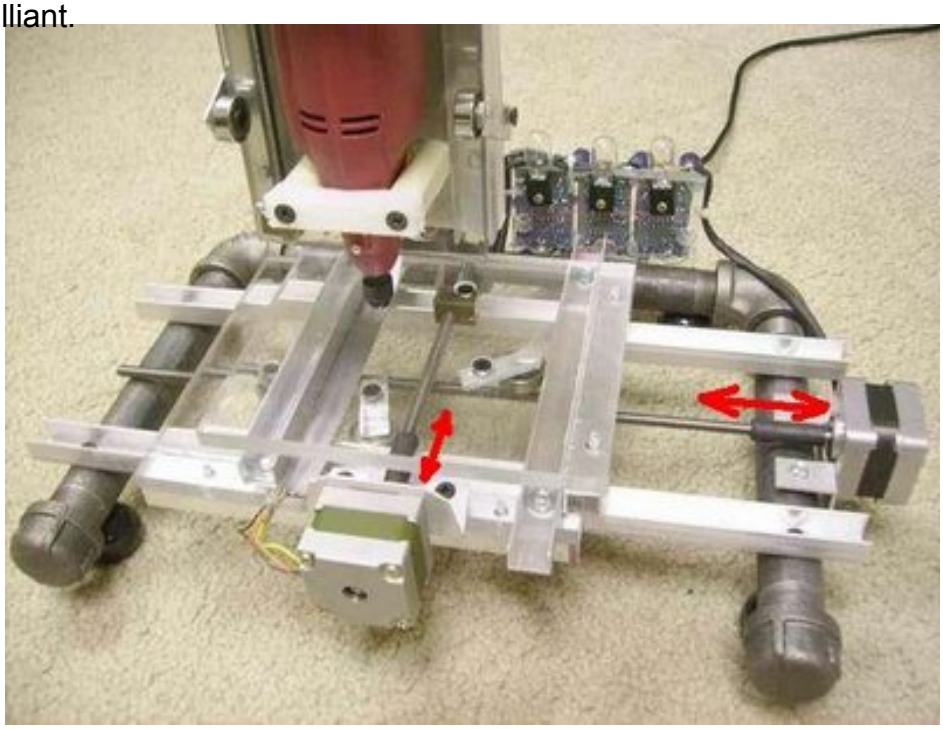
A note about the McWire design

Monday, 10th March 2008 by Forrest Higgs

I know that many of you are quite taken with the McWire cartesianpositioning system design and several of you are making one. I know Iquite like it and am stealing several sweet design tricks off of it foruse in Tommelise 2.0.

I've spotted a problem with the design that I discovered, however, which I will pass along to you. One of the really cool things about the McWire design is the way hehandles the connection between the steppers and the threaded driverods. He has a fixed stepper mount and a flexible connection betweenthe stepper and the drive rods. He also makes no attempt to secure theother end of the drive rod.

What that means is that his design is remarkably insensitive to misalignments. It self-corrects. That's utterly brilliant.



It is that very portion of the design that gives me trouble, not that Idisagree with it. What the design implies is that any thrust generatedby friction with the guide rails is going to go straight into thestepper motors. That's not really a problem with the beefy NEMA's thathe uses. They can handle a substantial amount of axial thrust and havelittle, if any play in the drive axis.

Now here comes the bad part. Those NEMA steppers virtually always comewith a 1.8 degree or smaller step. Assuming a 1 mm pitch on yourthreaded rod that means you have to make 200 steps to move the axis 1mm. That's good in one way, especially for a CNC router design likeMcWire. The threaded drive rod multiplies your already substantialstepper torque and you have lots of force on hand to counter whateveryour router throws into the object you are forming. You also get someawesome positioning accuracy.

The bad part is that if you are using the design for FDM, that torqueisn't really needed and you don't get a lot of speed from the axesbecause you're having to take so many steps to move anywhere. You canpump of the speed, sort of, but when you do that substantially youravailable torque drops like a stone.

Darwin gets around this by using a belt drive, which enables it to geta lot more axis speed out of

the NEMA steppers than you can with athreaded rod.
I've taken the approach of using tin can steppers which give me up to a15 degree step. Even though my steppers have much lower availabletorque, I don't have to run them nearly as fast to get the same amountof torque and axis movement as a much more powerful NEMA gets at itsmuch higher speeds. I don't get the accuracy, but then theNEMA/threaded rod combination gets over a magnitude more than I need inany case.
The problem with the tin can steppers, however, is that they haveconsiderable play in their drive shafts, about 1 mm in the ones thatI'm using. That means that I am going to have to include some means ofstopping that play if I am going to use the McWire design approach.

An interesting thing about USB

Monday, 10th March 2008 by Forrest Higgs

One of my clients was supposed to supply a quad cpu/quad core workstation for a big chunk of work I'm to be doing for them. While I am sure that it eventually will get here the delivery date for the new system keeps getting pushed out into the future.

As a result, Saturday I decided to get an external disk to back up a bunch of inactive projects that I have on my in-house system to give me some elbow room to do the new work whether or not the new system gets here. Initially, I bought a MyBook 1 TB external disk which is USB 2.0 compatable. It was supposed to be be plug and play. NOT!

Two MyBooks and one Maxtor OneTouch 4 Plus later, the people at Dell and I narrowed down the problem to a motherboard fault in the USB circuitry. This is odd, because I have a variety of USB peripherals hooked to the system and have never had any trouble. When I put the external disk on, however, it wouldn't talk to the Dual Xeon. Very odd.

I wonder if this board fault has anything to do with my transmission speeds problem on the USB prototype board that I've been noticing.

Anyway, I've attached the OneTouch to my old server and am pumping my critical files over the network onto the external disk. I only have USB 1.1 on the old server, so it is like watching paint dry.

Buffered USB running

Sunday, 16th March 2008 by Forrest Higgs

Wow! When I set about to shift USB comms on the 18F4550 prototypeboard from direct input to buffered I had no idea what I was gettinginto.

A lot of the drama had to do with needing to handle large amounts ofdata in real time in a PIC. The Oshonsoft BASIC manual, whileactually quite well written with many helpful example snippets of code,apparently never contemplated what I was attempting when it was written.

First, it turned out that I couldn't dimension an array larger than 8bits (one bank of RAM). Then, it developed that while I coulddimension an array to 8 bits, I could only index it to 7 bits. Isuspect that I would have been able to work through these problems,such as they were, had I had a USB input module on the IDE so that Icould simulate it. Unfortunately, Vladimir fell ill shortly after hereleased the USB extensions last year and hasn't yet come right withhis health, so, no USB input module just yet.

I'd got spoiled being able to simulate my code before I actuallystarted programming the actual chips. I hadn't realised just howspoiled till I had to do all my debugging on the chip instead of in theIDE. What a pain!

Since I didn't have a lot of buffer space, I decided that what Itransmitted from the PC ought to be as concise as possible. I wantedto keep the information I needed to do a step down to one byte. Sincea stepper can move one step in either direction or not at all I hadthree possible states for each stepper. That takes 2 bits for eachstepper. I also wanted an plastic extruder and a support materialextruder. That takes 2 bits as well.

Here is how I set it up.

Bit 0 - X stepper
Bit 1 - X stepper
Bit 2 - Y stepper
Bit 3 - Y stepper
Bit 4 - Z stepper
Bit 5 - Z stepper
Bit 6 - plastics extruder
Bit 7 - support material extruder

The biggest number I could get with this arrangement is.

10101010 = 170

This means that I can I can run all three axes of Tommelise 2.0 simultantously.

It also means that it becomes very simple to tell whether a byte in thebuffer was a step instruction or something else. My commandinstructions simply had to be bigger than 170 and less than 255. Thatleft me room for a considerable number of operational commands.

Anyhow, I'm able now to fill the buffer on the PIC from the PC and thentell the PIC to run the stepper motor(s) using the instructions I putinto the buffer. The (s) part awaits my building a full prototypeboard that can handle all three stepper motors.

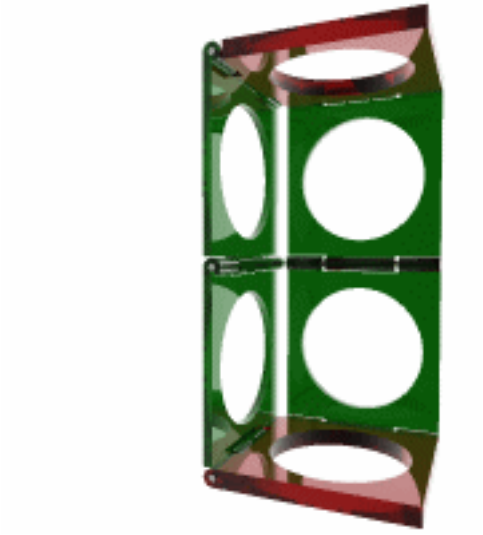
Using the buffer gets me past the problems implicit in trying to runthe steppers pretending that the PC is a realtime device, which itisn't. I can get a good turn of speed out of the stepper(s) now

byrunning it and the USB using a 0.4 msec interrupt cycle.
My next step will be to create several buffers so that I can be loadingone from the USB port while at the same time running the stepper fromanother buffer.

Thinking about guide rails...

Monday, 17th March 2008 by Forrest Higgs

There is something about guide rails that just feels *wrong*. Given how plastic prints warp I just can't ever see us printing a HDPE or PCL rail that would be worth the effort. That led me back to the Sarrus Linkage.



It's pretty easy to make and I could see how I could print one of these on Tommelise.

Hmmm...

New IR thermometer arrives

Wednesday, 19th March 2008 by Forrest Higgs

The cheap IR thermometer that I bought arrived. It is a Raytek MT6. I got mine from Arizona Tools for $67.99. Testing it, it does pretty much everything my old Triplett ProTemp did.

Aboutthe only thing to keep in mind about it is it's accuracy. Its sensoris a thermopile with a detection range of 6.5-18 microns. That's verygood in way and not so good in another. Rather than having acompensator circuit with a reference temperature pot it simply assumesthat the ambient temperature is 23 degrees C.
If the ambient temperature is way off of that you've got big troubles. For me 23 is going to be fine because that's about the averagetemperature I keep my working environment.
As to accuracy, for measuring the temperature of your extruder barrelor prints you're looking at an accuracy of +/- 1.5 C. with arepeatability of +/- 0.5 C.
I was a bit upset with this and went back to check the literature of myold Triplett ProTemp thinking that it was much more accurate than theRaytek. It turns out that while Triplett publishes their temperaturerange and other technical details in their literature, they verycarefully avoid talking about accuracy and repeatability.
Anyhow, it looks like the Raytek MT6 is good value for money.

Buffering USB comms for Tommelise 2.0

Thursday, 27th March 2008 by Forrest Higgs

I think that I've finally cracked the problem I was having withbuffering USB data transfer between the PC and Tommelise 2.0's PIC18F4550 microcontroller.

Recapping briefly, while a 3D printer needs to operate in real-time thePC that nominally manages it doesn't as a practical manner. WithTommelise 1.0, the data flow between PC and printer was quite sparse. The PC would tell the printer to start extruding at a particular xyzcoordinate and to stop extruding at another. The printer would thenconvert that information into a huge number of instructions drivinggearmotors and checking their progress with shaft encoders.

That control approach worked fairly well as long as what I was printingconsisted of a set of long, straight extrusion paths. Once I wanted toprint a large number of very short extrusion segments I started runninginto all sorts of problems.

The barriers to doing such short segments were many. Probably the mostdaunting, which led me to abandon gearmotors in favour of my currentwork with tin can steppers had to do with the inertia and momentumintrinsic to gear motors. It is very difficult to get one to start andstop on a dime, so to speak, without having to drop the speed of themotor down to something quite low. I found that in order to do shortsegments I was operating in the vicinity of ~1 mm/sec or less. Withlonger segments I could run the print speeds up to several times that. Stopping at a specific place is much easier when you have a few momentsto put the brakes on.

As you might imagine, my firmware control schemes got more and more complicated as a result of all this.

While microcontrollers are very handy little devices they are notparticularly good at doing floating point calculations. The lack of ahardware floating point processor, while understandable in such small,cheap devices, means that if you want to feed xyz start and stoppositions into your microcontroller and then let the firmware do therest you have more than a few calculations to do. Again, with longline segments this isn't really a problem. With very short ones,however, you find that you are working your little CPU to death andcausing troubles, as a result, with your interrupt routines.

Shifting to faster comms and moving the difficult calculations over tothe PC seemed to me to be a good course of action. I started lookingat the PIC 18F4550, which is a native USB microcontroller and prettymuch pin compatable with the PIC 18F4610 that I had successfully usedwith Tommelise 1.0.

While the model numbers of the two PIC's are very similar and whilethey look very similar in many other respects they turned out to bevery different beasts. The 4450 has a complicated two-part timingscheme to accomodate the USB comms. The good news, however, is that itcan also use readily available 20 MHz crystals to achieve 48 MHz CPUclock speeds. That is proving VERY handy.

The bad news was that the the USB transfer speed that I've been able toachieve with the USB extensions to my Oshonsoft Basic compiler arequite low (~1,200 bytes/sec). A bit of work on my comms protocol,however, got me down to one byte per stepper step for the whole of theTommelise positioning and extrusion system. That meant that I couldrun Tommelise 2.0 at a

print speed of 8 mm/sec on one axis (~11 mm/secon a diagonal) with a data transfer rate of about 250 bytes/second andrun it at the maximum that I could get out of my tin can steppers atless than twice that.

The last big obstacle, which I just got past, had to do with accessingdimensioned RAM memory in the PIC 18F4550. Oshonsoft Basic has a hardtime dealing with dimensioned arrays larger than 127 or at most 256. Iwent on with a multi-banked memory access scheme leveraging thoselimitations into something quite a bit bigger. Unfortunately, the codethat was needed to impliment and manage such a memory band seemed to begetting bigger and nastier by the hour.

Finally, some kind soul in the Oshonsoft IDE user group...

http://tech.groups.yahoo.com/group/pic_sim_ide/

...explicated what had previously been some obscure text in theOshonsoft Basic manual dealing with a Pointer scheme. After a fewhours getting used to using Oshonsoft's pointer capability I was ableto vastly simplify my RAM usage code.

I now have a 1,008 byte buffer for handling USB data. That gives meroughly 5 seconds of continual printing operation at full printingspeed. It takes me about a second to completely fill the buffer. Thatshould allow me to deal with ordinary Wintel delays and to recover fromPC timeouts without a lot of drama.

Taming the tin can stepper

Wednesday, 16th April 2008 by Forrest Higgs

I finally got everything to do with Tommelise moved over to and runningonmy new XPS 720 H2O so I can start doing some useful work again.

Some weeks ago I hit a little bit of a wall trying to buffer USB datacoming into my 18F4550 microcontroller in RAM. While the memory bankswere arranged in 256 byte banks, dimensioned variables couldn't be anybigger than 128.

Eventually, I discovered that Oshonsoft BASIC has pointer operatorswhich allow you to handle larger chunks of RAM and used those insteadof dimensioned arrays. Once I had that going all right I ran into yetanother headache.

Once I had loaded a bunch of step information into my buffer and toldthe microcontroller to use it, I discovered that the stepper would runfor a while and then hesitate or stutter for a moment and then go on. There was nothing that I could see in the firmware coding that ought tobe causing that. All the same, it looked like a coding error since thehesitation always happened at the same place.

After tracking the code I discovered that it always happened in the128th byte of code that I read out of the buffer. That was the onlyplace that it happened. I thought for a while that it might be a faultin the chip, so I traded chips and still got the same problem. I thenthought that it might have something to do with where I'd located thestart address of the buffer, so I moved that around in the RAM. Samesame.

I'm not too proud of the fix that I finally resorted to. I discoveredthat if I artificially moved the start of the buffer, say, 130 bytesback from the point where I actually started using RAM for the bufferand then started the count ffrom 130 instead of 0, everything ran justfind. It's nasty, but it works.

For the life of me I still don't know if this is a compiler problem,some damned coding problem that I'm too blind to see or somethingpeculiar to this particular PIC chip. Be warned, though. If you aretrying to configure a buffer in the 18F4550, you may be in for someunwelcome excitement. All the same, it's "fixed" now.

These little 15 degree tin can steppers are fascinating little beasts. It seemed for a long time while I was chasing down this hesitationproblem that tin can steppers were finicky things. They seemed toerratically decide to just sit there and vibrate instead of step fromtime to time. I could never know exactly when it would work and whenit wouldn't.

I got to the bottom of that bizarre behaviour a few days ago when infrustration I picked up the stepper when I started it up and noticedthat the damned thing always worked when I was holding it. The simpleanswer to that was that it just liked being held. In fact, it was alittle more complicated than that.

I have a 1/4-20 coupling nut that I've fashioned into a connectormounted on the one that I test. The stepper just lies on a chair nextto my PC and isn't in a firm mount. What I hadn't realised is that thecoupling nut weighs about as much as the little stepper.

What was happening was a real-time demonstration of Newton's Third Law,viz, for every action there is an equal and opposite reaction coupledwith inertia. Lying free on the chair when the

stepper tried to stepthe stepper motor would move clockwise and the coupling nutanticlockwise. The step wasn't getting quite finished because of that,so the next three steps were out of phase. It would just sort of sitthere and tremble rather than doing anything useful, like turning thecoupling nut. When I held the stepper motor, my hand had a LOT moremass than the coupling nut so it turned quite handily.

There is a little coding issue that has also come up. I have to keepvery good track of where in the four steps for a bipolar stepper that Iam while I am running the stepper, even if I've shut down that steppermotor for a while. If I don't I can try to start up out of phase andlose a step or two till the motor gets back into phase. That's notreally hard to do. I just have to set up variables that store thenumber of the last step, 1-4, that the motor took.

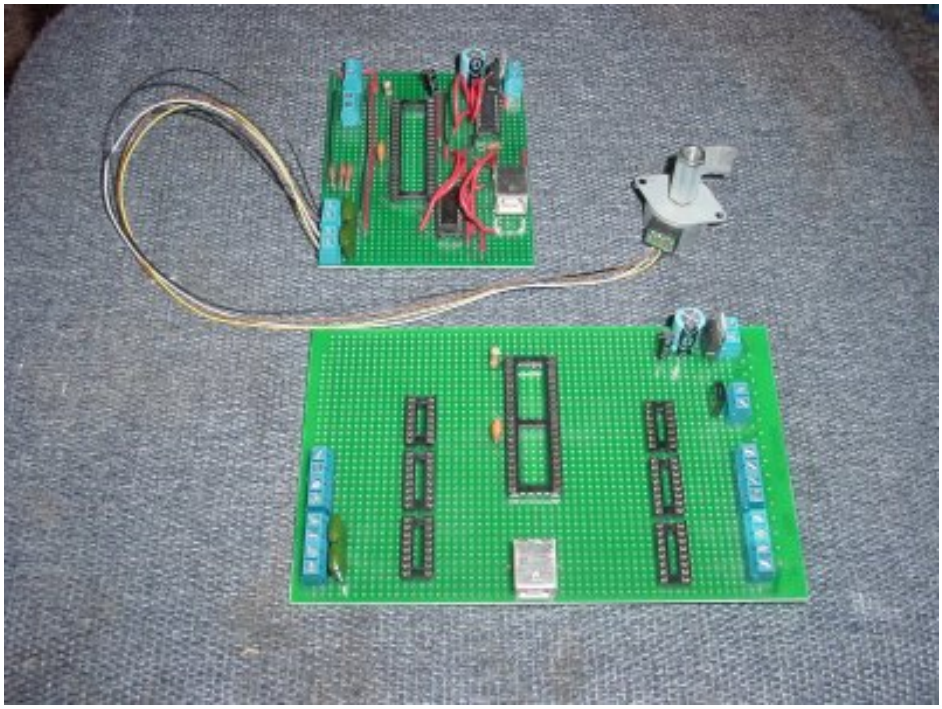
That's pretty much everything that needed doing before I go ahead andbuild up a full prototype board that runs all three steppers, theextruder and the barrel heater.

Wish me luck on that. :-D

Building the big board for Tommelise 2.0

Thursday, 17th April 2008 by Forrest Higgs

I had my youngest daughter and son at my place for Easter a few weeksago. As usual I cleaned up rather quickly the day that day before mydaughter was to arrive.
"Cleaning up" for me means filling whatever box comes to hand with allthe loose bits that lie around my place and setting them aside. Thatis quick and quite efficient. Unfortunately, it can make finding whatyou "cleaned up" quite difficult afterwards.
In spite of that, I managed to find all of the parts that I need to build the full controller board for Tommelise 2.0.



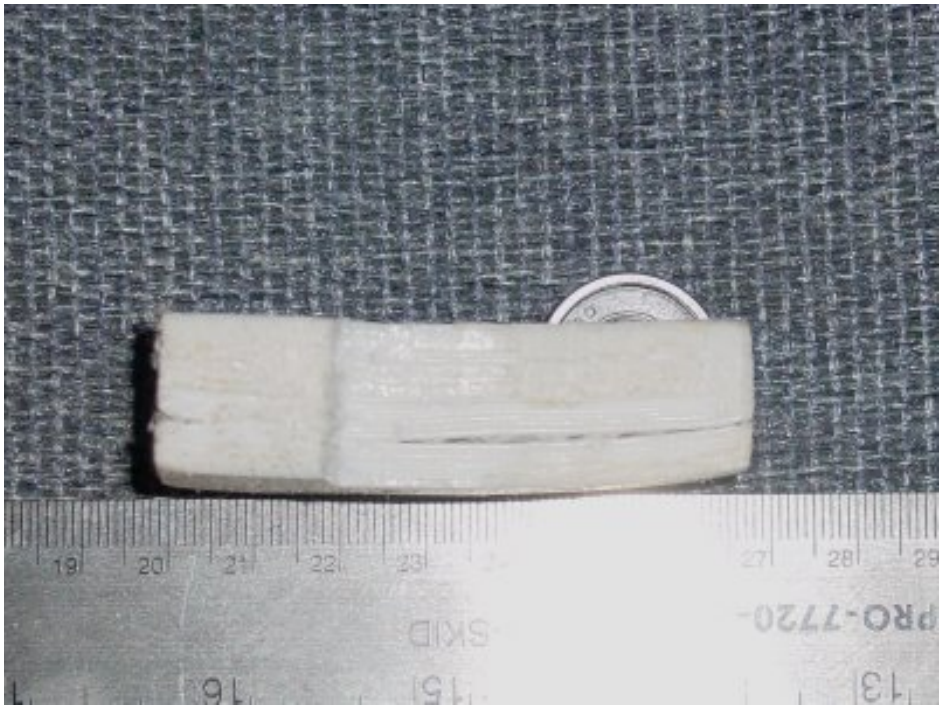
You can see the prototype board above the full controller board. Ihaven't exactly decided how I want to lay out the hex inverter andstepper driver chips yet. It looks like there is going to be plenty ofroom, though.
While I've got all of the connectors to drive the three steppers, theextruder motor and the extruder heater in this photo, I haven't put inthe connectors for the limits detectors in this picture. They aregoing to be in the upper, left-hand side of the board.

Revisiting the warped HDPE polymer pump

Thursday, 17th April 2008 by Forrest Higgs

While I was digging around yesterday for the parts I needed to build upmy full-sized controller board for Tommelise 2.0 I unearthed theextruder polymer pump that I printed with Tommelise 1.0 last year.

From Chris' (nophead) work subsequently, I know now that mydelamination problem came from not extruding the HDPE hot enough. Evenso, looking at it I decided that it would be interesting to see how thewarping that I encountered compared with what Chris experienced withsimilar (50%) fills.



Asyou can see, the span of the piece is right at 68 mm as opposed toChris' 40x10x10 standard. The cross-section of the main span is20x20.

Assuming spherical warping with 2 mm warping at the extremes of the 68 mm span one gets a radius of curvature of

R = (x^2 - y^2)/2y

R = ((68/2)^2 - (2^2))/(2*2) = 288 mm

Correcting for Chris' 40 mm standard width you get warping of

x^2 - 2Ry + y^2 = 0

Which is quadratic. Solving for y...

a = 1

b = -2R

c = x^2

y = (-b +/-(b^2 - 4ac))/2a

y = (2R +/- ((2R)^2 -4(x^2))^0.5)/2

y = (576 +/-(576^2 - 1600)^0.5/2

y = ~0.7 mm

Measuring the warping of the "t" across the top of the pump, which isabout 40 mm wide, I noticed

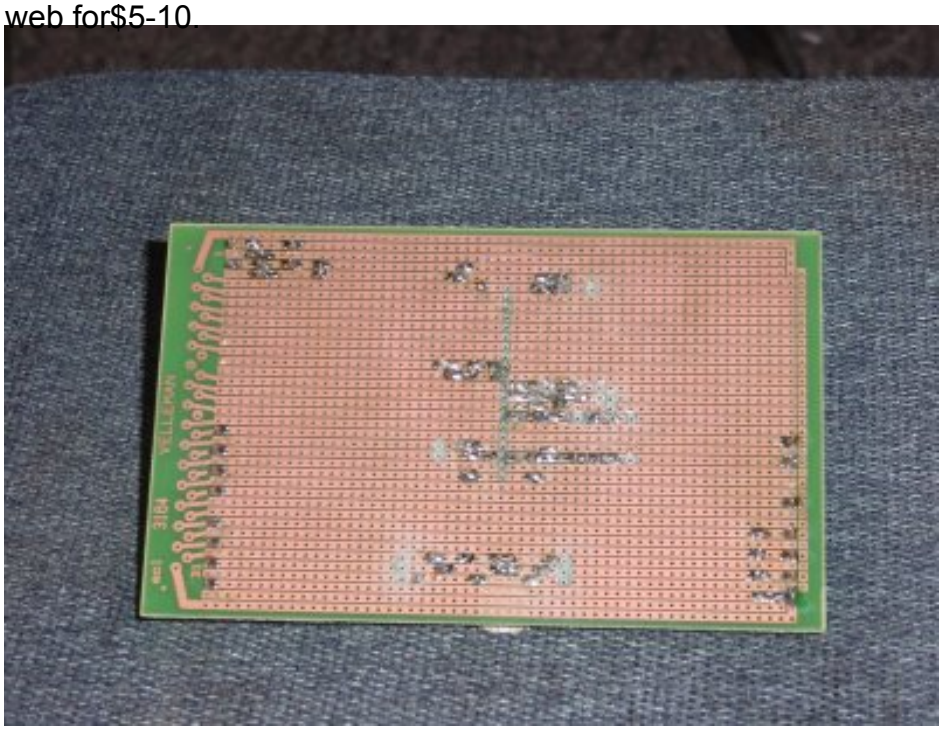
that the warping was just a shade less thanthis.
Figuring for 20 mm, the width of the pump

y = (576 +/-(576^2 - 400)^0.5/2
y = ~0.2 mm
Mind, I am just eyeballing, but the numbers look fairly reasonable in terms of the degree of warping that I am measuring.

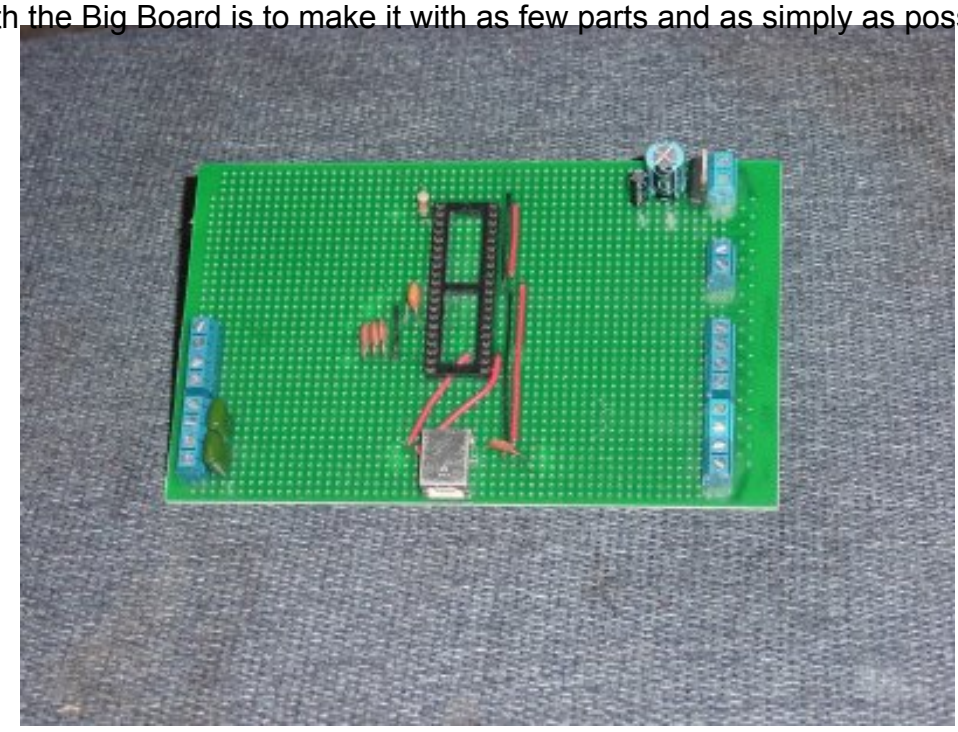
Testing USB on the Big Board for Tommelise 2.0

Sunday, 20th April 2008 by Forrest Higgs

I had a few hours to continue building up the Big Board for Tommelise 2.0, today.
SinceI am by no means settled on a final design to the point where I ambrave enough to undertake printed circuit board designs, use a Eurboardstripboard for building my boards. You can get them on the web for$5-10.

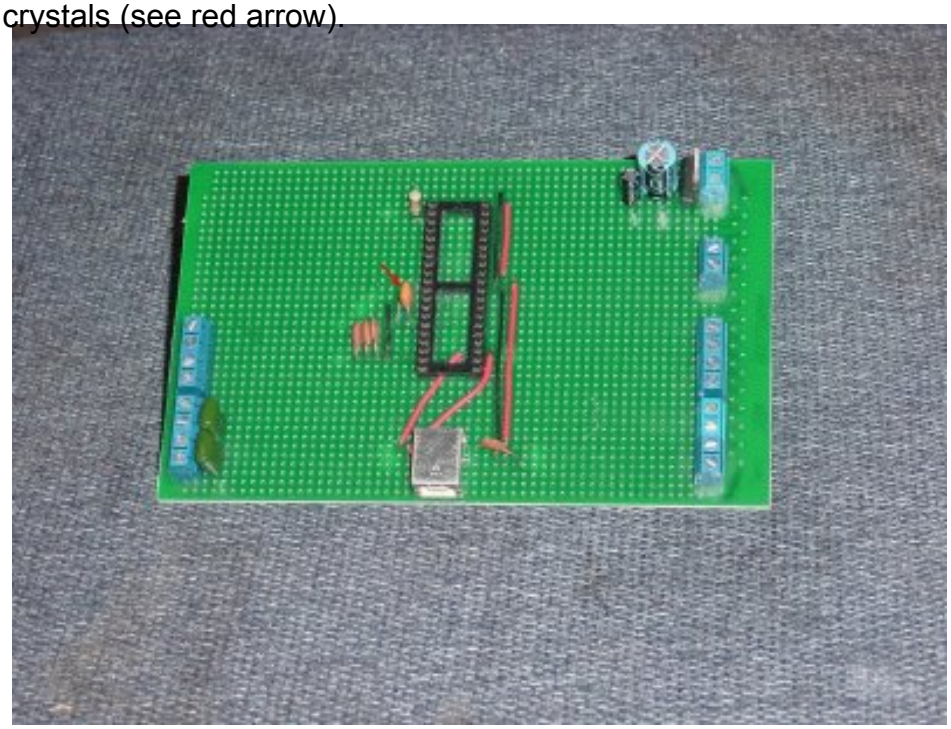


My concept with the Big Board is to make it with as few parts and as simply as possible

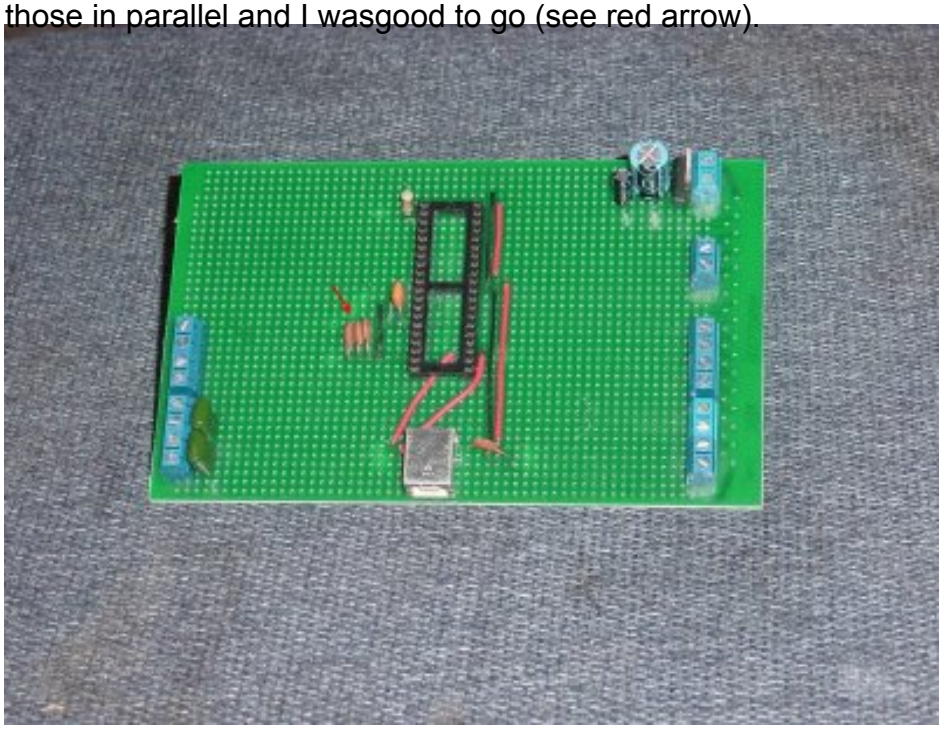


ThePIC 18F4550 does that quite nicely. That chip lets you go directly toUSB connectivity with your PC without the drama of MAX232 chips toconvert what PIC understands to be serial comms to standard RS232protocols. Similarly, the PIC 18F4550 doesn't require a conversionchipset to get

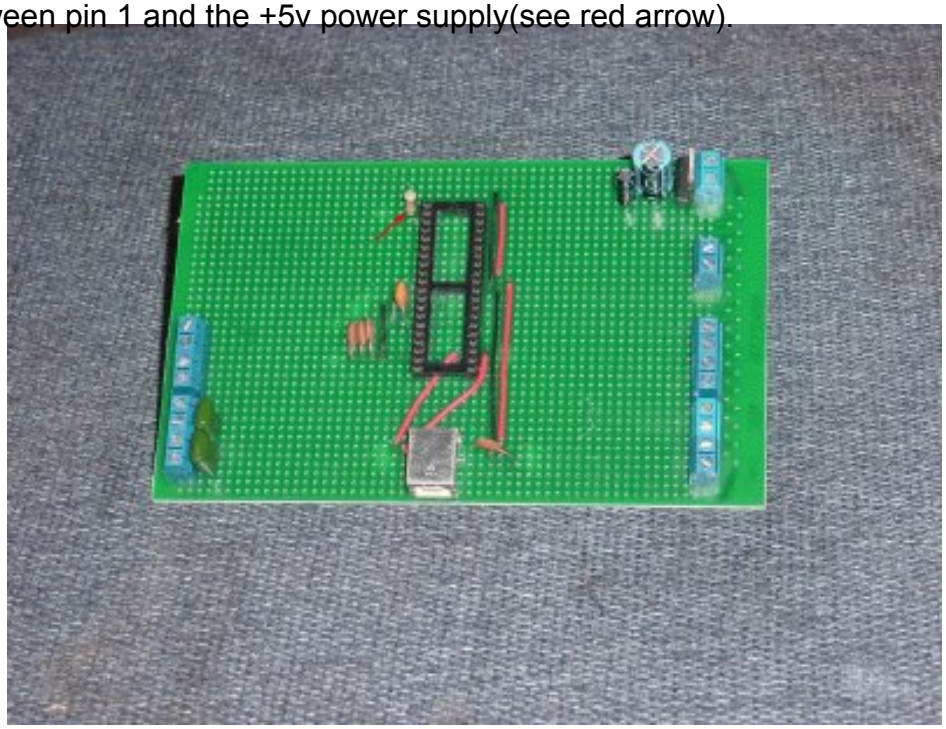
USB signals like the Arduino boards do either.
Clocking the PIC 18F4550 can be very simple if you use three pin ceramic resonators instead of the usual clock crystals (see red arrow).



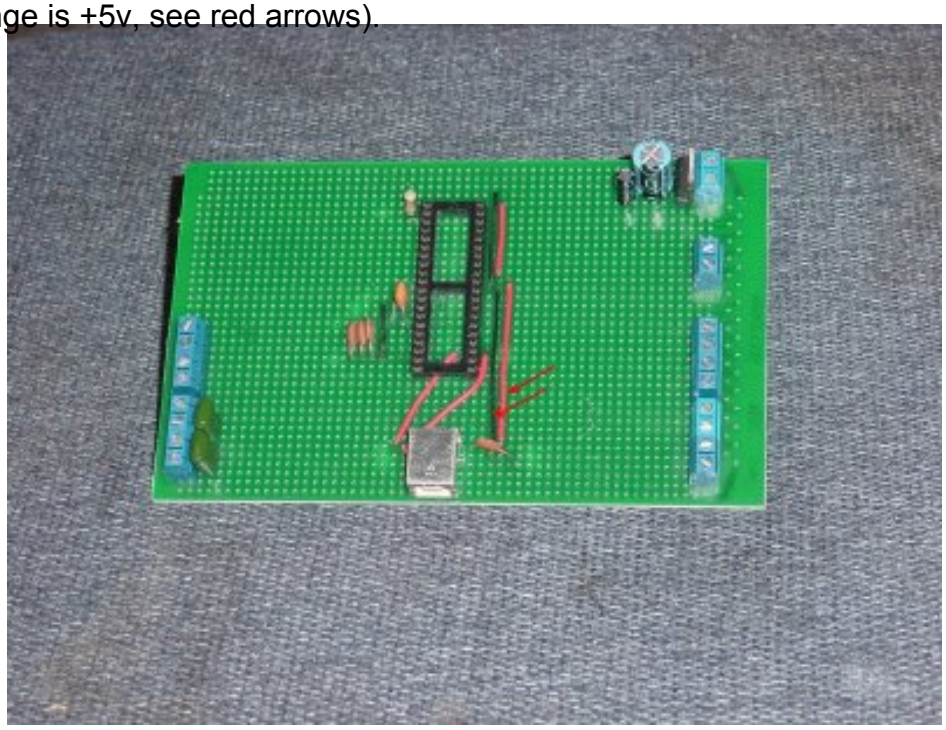
Thenice part about these ZTT ceramic resonators is that they have thecapacitors that you ordinarily have to install with clock crystalsinside the resonator rather than separate. That eases assembly and cutsdown on your board space.
One little bit of naughtiness of thePIC 18F4550 is the need to put a capacitor between pin 18 (Vusb - 3.3vUSB power supply) and ground. The data sheet says that you are to use aminimum of a 220 nF capacitor. The schematics that I have found forthis PIC on the web all use 470 nF capacitors, though. Locally, I foundthat 220 nF capacitors were as rare as the proberbial dead monkey,white crow and straight palm tree. Ditto, 470's. What I had a lot ofwere 104 nF capacitors, so I put three of those in parallel and I wasgood to go (see red arrow).



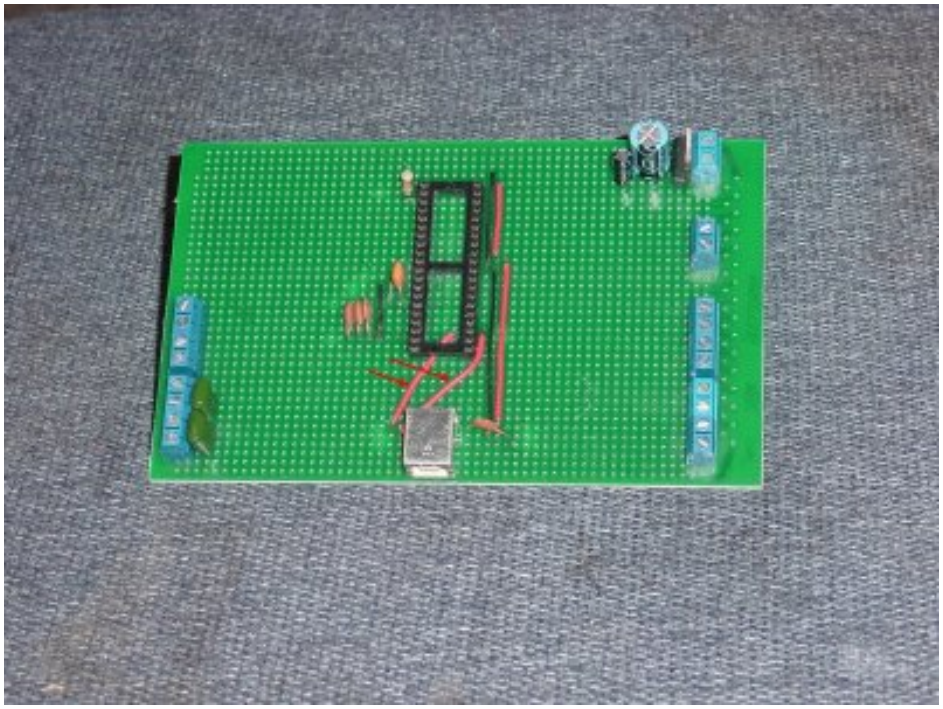
Oneother thing that the PIC 18F4550 requires in common with many other PICchips is a 10K pull-up resistor between pin 1 and the +5v power supply(see red arrow).



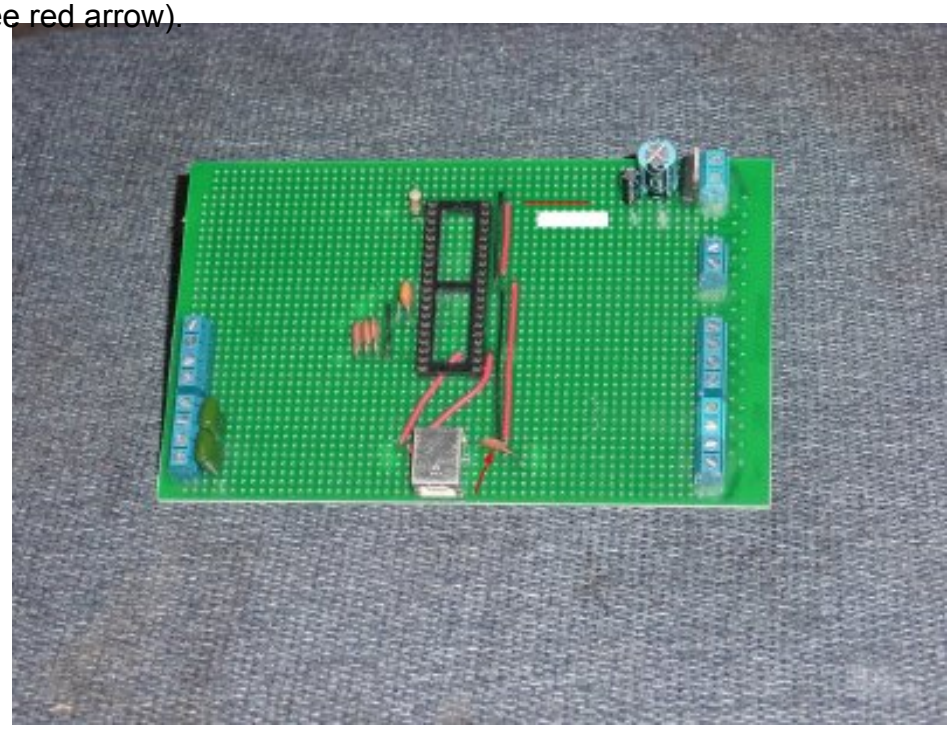
Ordinarilythis pin has a pushbutton on it which lets you manually reset yourboard. I've never had the desire for this sort of thing and simply letthe board reset on powerup.
Hooking up the USB port is trivial.The +5v power and ground lines go on the right-hand (black is groundand orange is +5v, see red arrows).



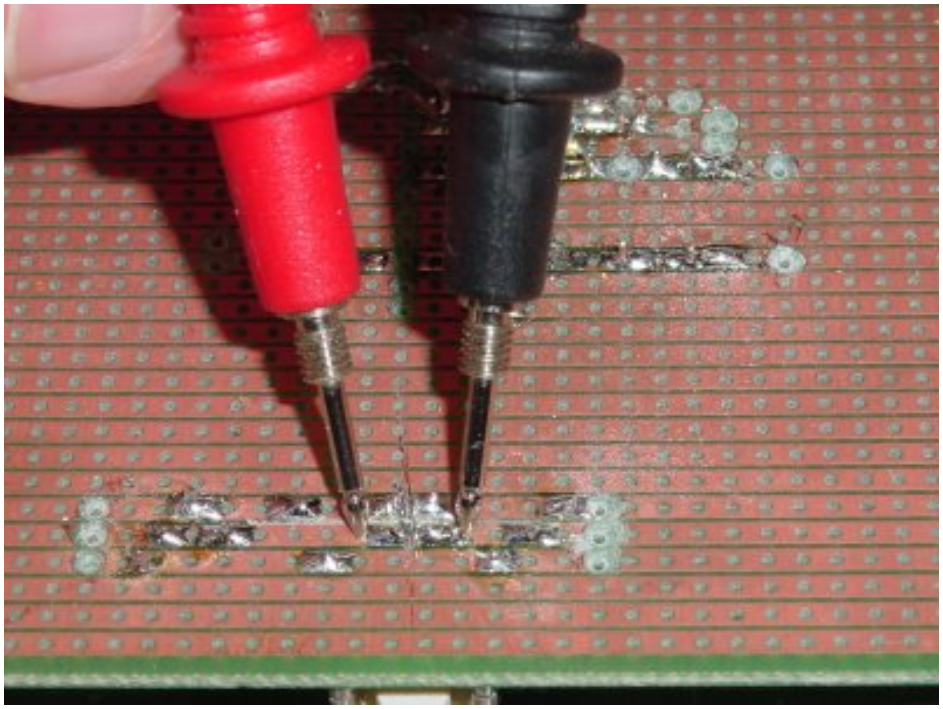
Comms lines from pins 23 and 24 go on the left (see red arrows).



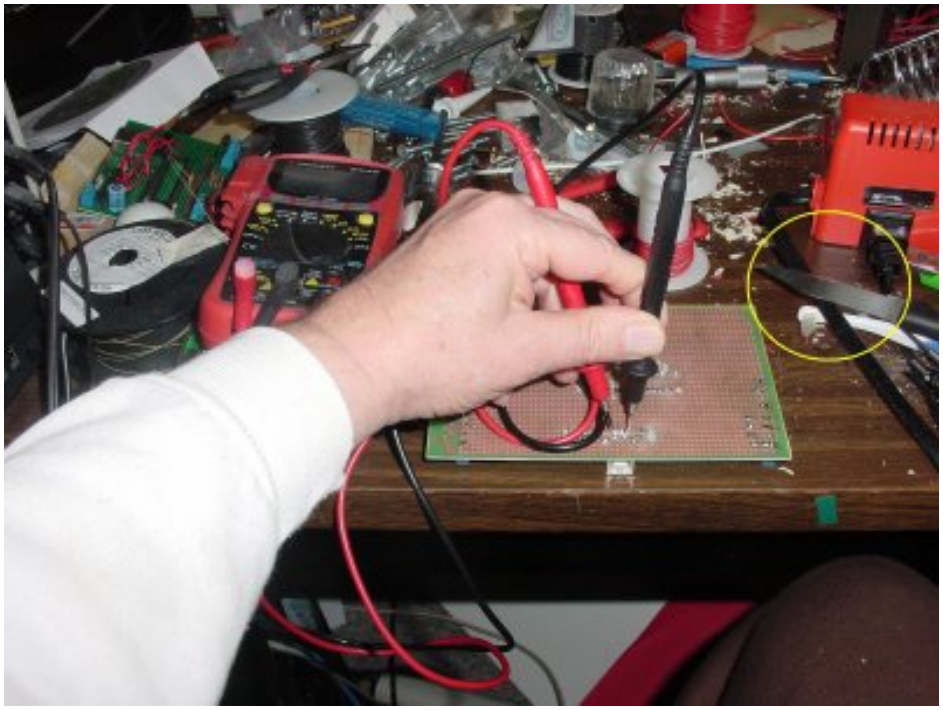
Onelittle detail is that you need to include a 104 nF disk capacitor tofilter noise out of the power supply lines (see red arrow).



Interestingly,you can power your PIC 18F4550 very nicely with the 5v power that comesfrom your USB port connector. I've even been able to run one of the tincan stepper motors using only this power source. That's not somethingthat I would recommend that you try to do, however.
The biggestpain in making up this board is in securing and connection the USB TypeB connector box. The live pins are in a cluster of 4 only one holeapart.

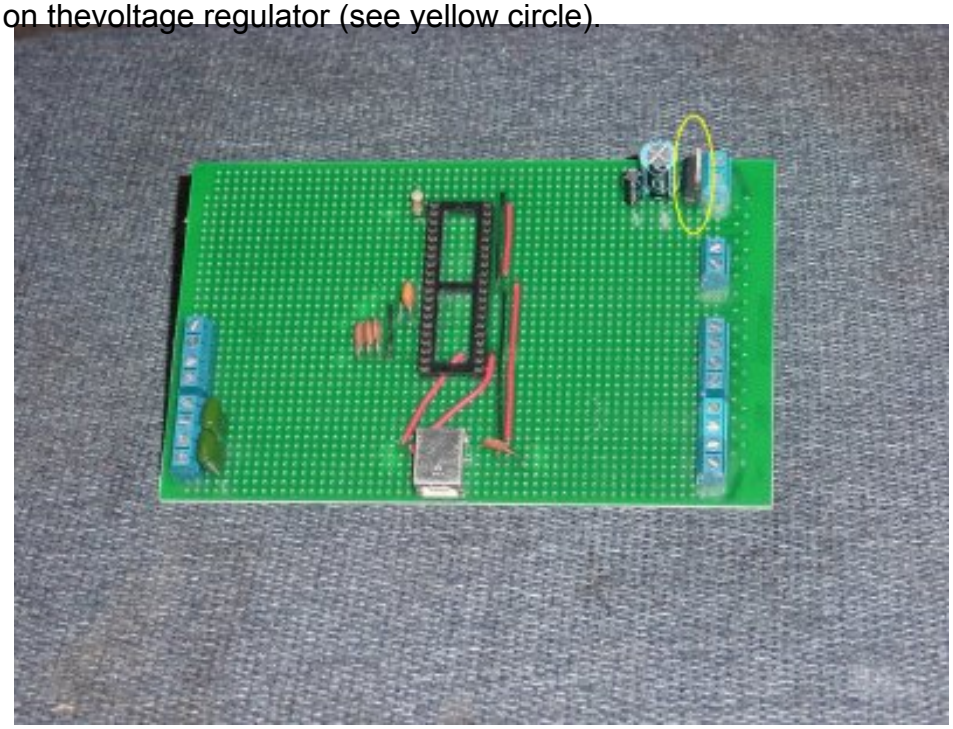


Withstrip board this means that you have to very carefully cut the stripbetween two adjacent holes rather than drilling out a pin hole betweentwo live pins, which is the usual practice. You can see in the abovepic that I've sliced the two strips vertically between the two pins. Todo this I've used a serrated paring knife and cleaned the solder dustout with a round bristle toothbrush (see yellow circle).



Becausel am completely pants at soldering I always carefully check every pairof adjacent strips and every break between parts of a strip to makeabsolutely sure that I haven't either crossed the break with a bead ofsolder or got some solder dust into the break which shorts it. I foundone bridge checking even this little bit of soldering.
I'd recommend the practice to anybody, no matter how good at soldering you think you are. Afterall that I fired up the board and checked to see that I was getting thevoltage that I was expecting where I was expecting it on the board.Mind, I did my preliminary powering up without

the PIC 18F4550 chipinserted. If you don't do that, you can easily burn out a PIC chipwithout meaning to. At ~$8/chip, that's not a habit I'd suggest thatyou get into.
One thing that I always do when I power up a boardlike this for the after a bout of soldering is to keep my finger on thevoltage regulator (see yellow circle).

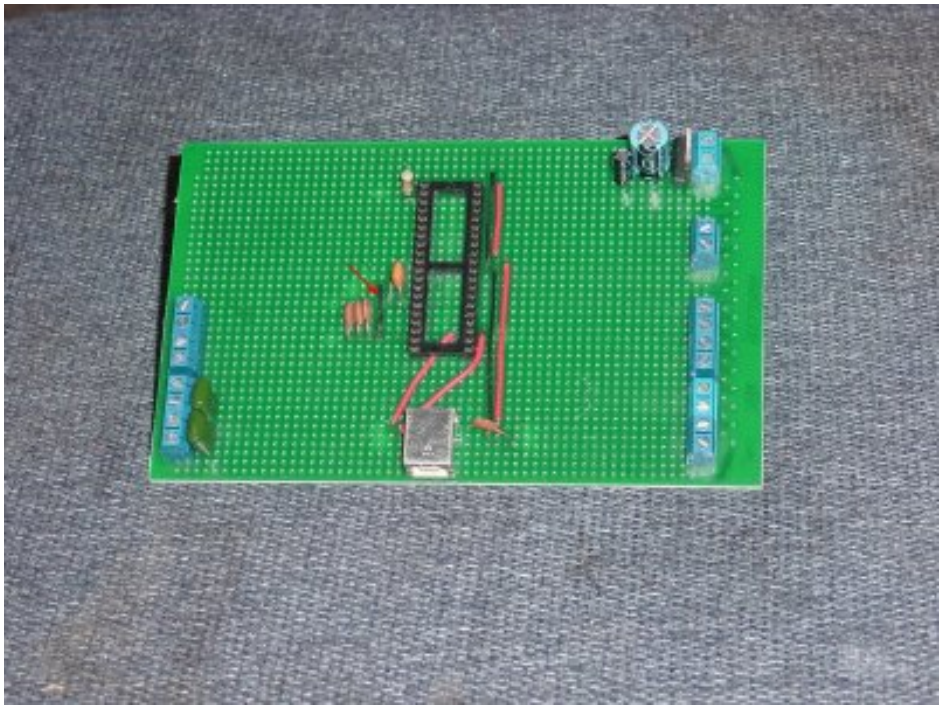


If you have a short in the soldering you've done the regulator willheat up quite quickly and you will know to shut the power off beforethings get out of hand. Mind, these regulators are quite tough, muchtougher than the chips they power.
What you SHOUDN'T do is to power up your board and then decide to putyour finger on the regulator some time later. You can easily get asecond degree burn that way if it hasn't already fried one of thetracks on your strip board. I'm not telling you this as something I'veheard about. I am ashamed to say that I've blistered fingers and friedstrips both more than a few times. This safety tip was a hard learnedlesson for me.
A blister on your finger means a few days for things to heal andfeeling like twit. Frying a strip on your board can mean $10+ inbuying another board and more than a few hours building it back upagain if you can patch it.

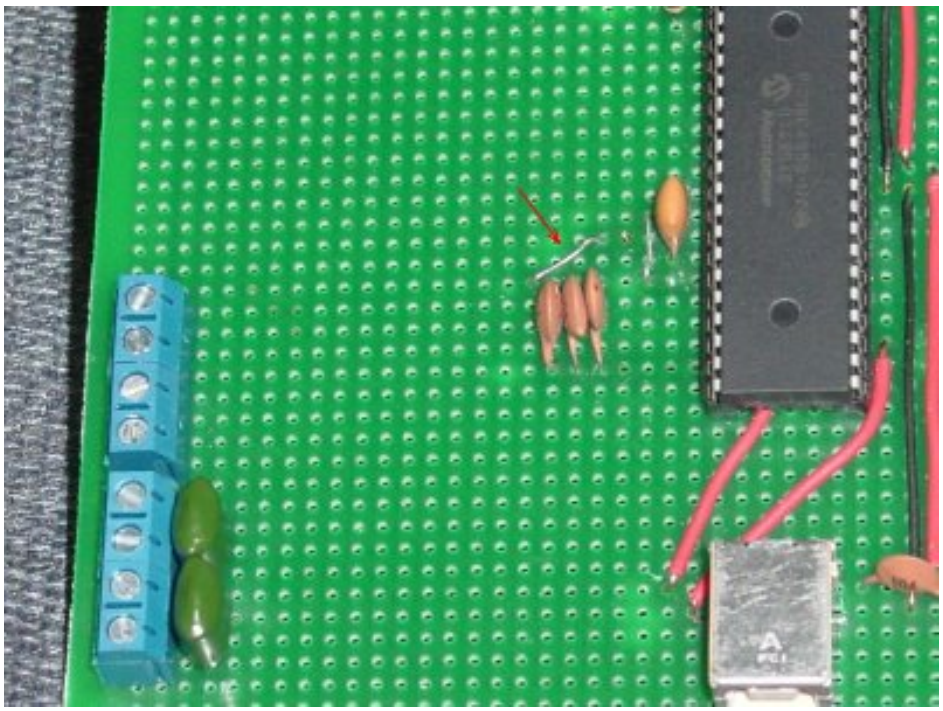
USB comms established on Big Board

Sunday, 20th April 2008 by Forrest Higgs

I hadaccidentally taken the three 104 nF capacitors that were supposed to bewired in parallel between pin 18 and ground were connected wrong. The3.3v supply (pin 18) had been accidentally connected directly to groundbypassing the capacitors (see red arrow).



OnceI fixed that my PC immediately recognised the board when I plugged itinto a USB port. Using the Tommelise 2.0 control panel I was also ableto see that tests of USB and buffer code is working like it does on theprototype board (see red arrow).



Now I must start wiring up a hex inverter chip and one of the stepper driver chips (754410).

A note on the MC14069UB hex inverter chip

Monday, 21st April 2008 by Forrest Higgs

Some months ago I tried to use the MC14069UB hex inverter chip to reduce the number of pins that were required on the 18F4550 microprocessor to drive the SN754410 quadruple half-H driver chip from six to three.

The effort at that time was not successful.

Now that I am working on the Big Board for Tommelise 2.0, I went over all of the data sheets for the various chips that will be required and unearthed this little jewel on the MC14069UB data sheet.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either VSS or VDD).

I wasn't doing that before. It will be interesting to see if that was what was causing the wayward performance in the driver chip that I encountered before.

Three pin stepper control

Friday, 25th April 2008 by Forrest Higgs

When I built the small test board for the 18F4550 I attempted toreduce the number of i/o port pins that were required to drive astepper using the SN754410 quad half-H driver chip. If you just assignan i/o port pin to each of the equivalent pins on the 754410 you needsix; two pins to control the two phases in the tin can stepper and apair of pins to control the direction of current through each phase.

Tin can steppers don't cope well with half-stepping or full stepping. Wave stepping is about the most they can deal with. That simplifiesmatters in another way, though.

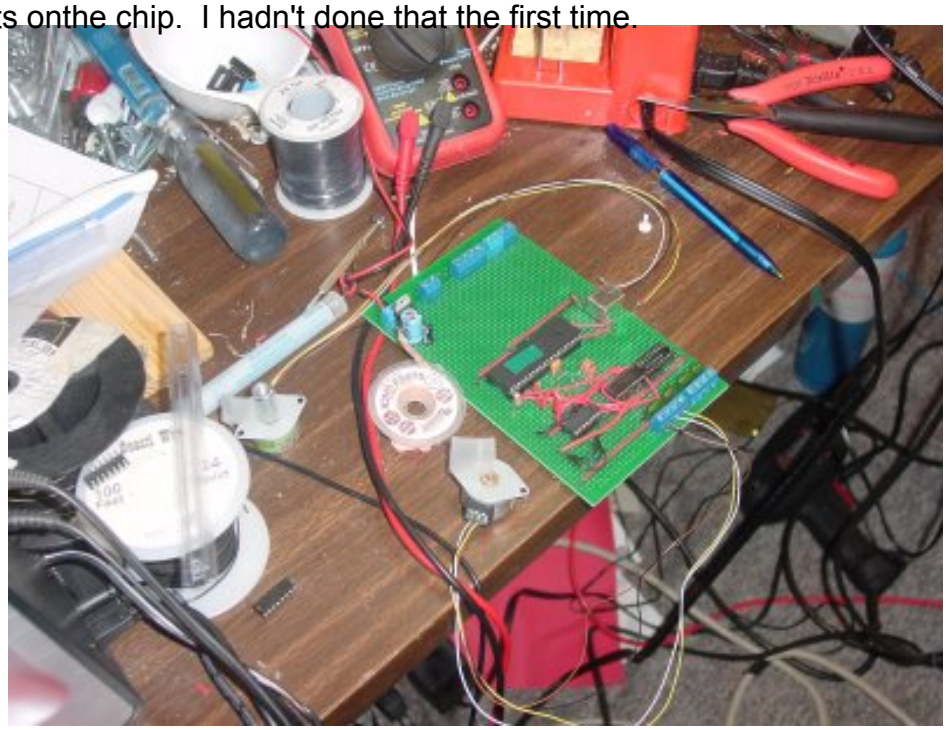
With wave stepping you only charge one phase at a time. When you areonly using one phase at a time you really only need to control thedirection of the current through the activated phase. The point ofthat is that it doesn't matter if you use the same two pins to controlcurrent directions for both phases. That means that you really onlyneed four i/o pins to control your tin can stepper.

What I realised is that if you use an inverter circuit on a hexinverter chip (it converts a +5v signal to a 0v signal and vice versa)you really only need one pin to control current direction. That getsyou down to a total of three i/o pins to run a stepper.

The implication of that is that you can run your whole xyz cartesianrobot with a limits switch for each axis with a total of 12 i/o pins or1 and one-half ports on your Pic chip. That's not bad work.

On the small prototype board I tried that and failed miserably. Somehow the hex inverter chip just didn't seem to work. I assumed atthe time that a single hex port somehow couldn't supply two input pinson the 754410. Chris (nophead) assured me, however, that it ought tobe possible. In order to get on with my life and work, I just backedup and used six i/o pins for the time being.

When I set about to build the big prototype board, I decided to givethe hex inverter scheme another try. Reading the data sheet on the hexinverter chip I discovered that you MUST ground all unused inputs onthe chip. I hadn't done that the first time.



Well, that was what the problem was. You see the working result of theeffort, mare's nest of wiring that it is. The little tin cap stepperis very happy to work with three pins.

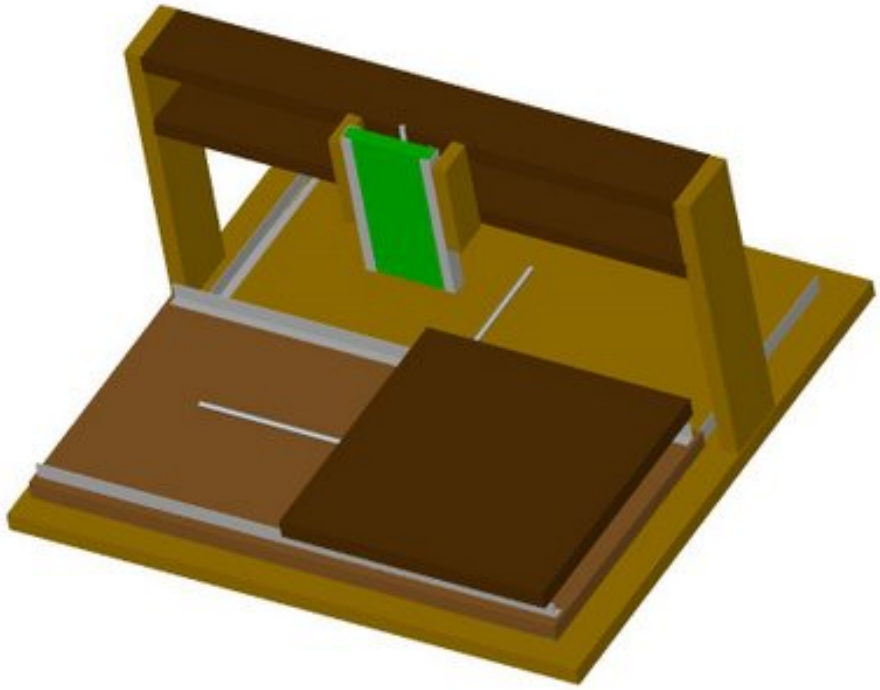
The mare's nest of wiring will give you a hint that this didn't just happen.
There were a number of false starts. Basically, what I did wrong isthat I didn't drill out a strip that connected one of the ground pinson the 754410 with one of the phase coils. This meant that when Ienergised that coil I put 12 volts directly into the 754410's groundpin.
If the 754410 chip that I was using had failed spectacularly with acrack and smoke I'd have found the problem much sooner. As it was itjust quietly gave up the ghost. When I checked the pinouts that halfof the chip simply wasn't sending an output signal. Eventually, Ipulled the chip and checked the voltages of the signals going in anddiscovered that while I was putting the right stuff into the chip theright stuff wasn't coming out.
Having decided that the chip had failed, it was an old one, I put a newone in the socket and THAT one failed with a loud crack and a cloud ofsmoke. Being tired and a bit stupid, I decided that the chip wasfaulty and put another 754410 chip in the socket and blew it up, too.
That was when I found the undrilled strip.
Once I drilled that out and installed my last 754410 chip the stepperstarted working nicely on four i/o pins. Once I knew that it wouldwork on four pins I wired the hex inverter chip back into the circuitand got the pin count down to three.
That's why the wiring looks so nasty.
Mercifully, I recently discovered that Jameco carries 754410 chips at a much better price than Mouser does. As well, they are just up the road from me in the Bay Area, so when I order I sometimes get delivery by post in as little as one or two days. I ordered a dozen.

Engaging the windmill

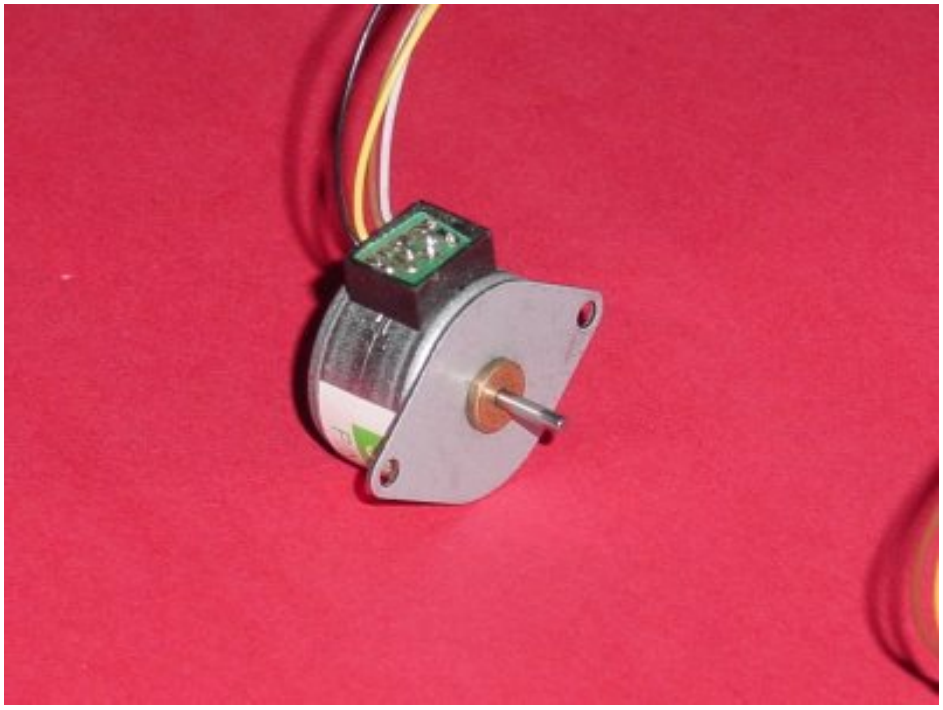
Friday, 25th April 2008 by Forrest Higgs

Working on Reprap has had a distinct picaresque feel to it. We rideoff armed with our soldering irons and side cutters determined to takeon and vanquish the whole factory/manufacturing establishment as weknow it.

Iwonder at times if this isn't the likely outcome of our efforts. Attimes, though, I find myself frightened that we might just manage toaccomplish what we have set out to do. What we are doing has thepotential to change the world every bit as much as the internet has.
LastAugust I had got as far as I could with the Tommelise 1.0 repstrap andset about to design the next generation. I wanted it much simpler andmore elegant.



Moreimportantly, I want it to be small enough to fit in a suitcase orstored under a child's bed with the z-axis folded down. I stoleshamelessly from the McWire CNC router project to make that happen. Mycredo was that if a bright twelve-year-old couldn't make one it wasn'tgoing to really get a lot of market penetration as an open sourcetechnology.
Keeping that credo in mind, I reluctantly put aside my little Solarbotics gearmotors and began to work with tin can steppers.



Thesesolved a lot of problem, but not all. The big problem I've been havingwith the design process is that I've been trying to make all the partscost individually as little as possible.

While that has workedwell with a lot of things, like the controller board, that kind ofthinking has got in the way at times. Now that I've actually beentrying to integrate tin can steppers with lead screws and thrustcollars I've begun to see that thinking of the cartesian robot part ofTommelise 2 as a sack of parts to be assembled has not been the bestway to proceed.
I've admired the way that Chris (nophead) hasgone about doing things. Rather than trying to build a cartesian robotfrom the ground up, he simply went out and bought one, more or less.The point was that Chris has a reliable positioning system and has, asa result, been able to undertake signal work in working out theproblems intrinsic to actually printing objects while the rest of ushave been struggling to get our positioning systems to actuallyposition reliably.
I'm not saying that we all should have beendoing what Chris has been doing, mind. I'm merely saying that how hehas gone about doing things has some useful lessons for me at least.
WhatI've learned is that while making a McWire-like positioning robot ispossible, getting the axes right is a fiddly exercise and not somethingthat your average bright 12 year old is going to be likely to succeedat too often.
I think, however, that I've found a way. Keepingin mind that my tin can steppers are nice to work with, I've revisiteda variation on that theme that I've been looking at, on and off, forover a year now.

Whenyou really look at this linear actuator stepper motor, it's brilliant.It is basically a tin can stepper with the drive shaft drilled out andthreaded onto an acme lead screw.
To make it work you basicallybolt down both ends of the lead screw, bolt the motor onto theunderside of your positioning table, connect the electrical leads via acoil cable and you are good to go. Since the lead screw doesn't have torotate the overall inertia of your actuator is simply that of therotor. That means that you can get a substantial amount of thrust outof this system.
As well, your thrust collar is built in andanti-backlash. There is no questi0n about your getting the positioningaccuracy you want because it is intrinsic to the actuator.
Three of those and you have a positioning robot. No questions asked.
Aboutthe only thing that kept me from moving in this direction a long timeago was cost. When you

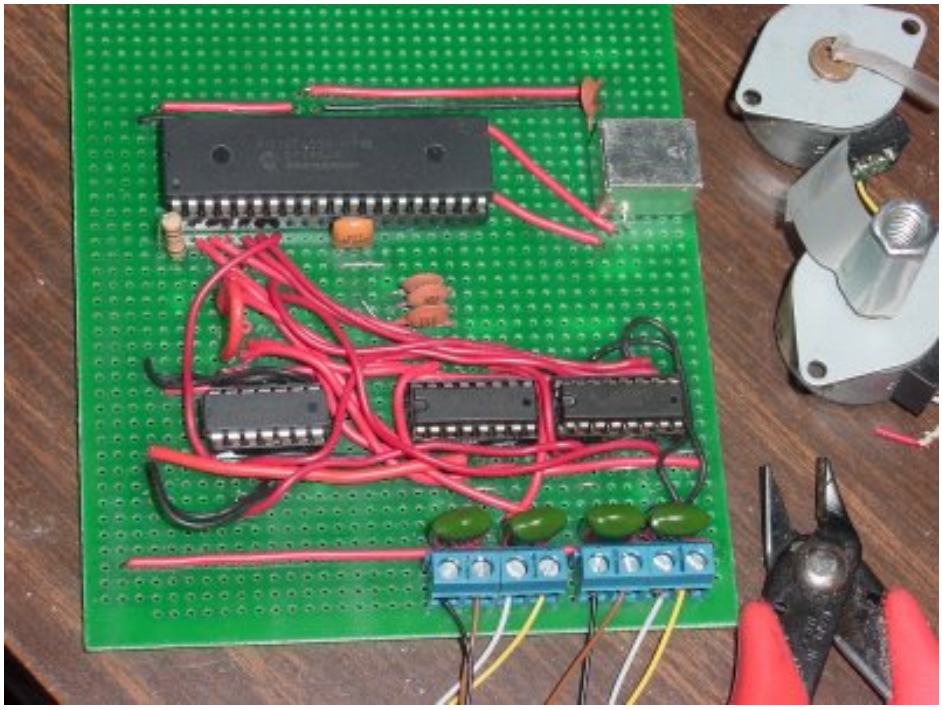
figure the cost for a one-off purchase of anaxis that can do a 12 inch transition you are looking at about $110.Doing a 12x12x6 inch build volume costs you $324 retail.
Oddly,talking with Annie Fan at CW-Motor in Shanghai made me think againabout this situation. She was able to show me that the $75 LinEngineering stepper that works perfectly with Darwin can actually behad for about $15 if you order direct from China. I got a feel for howmuch the price could come down when I talked with Joe Rossi at Haydon,a major vendor of linear actuators. The price reduction from 1 to 100sees a drop of 2/3rds. That means that the $324 set of linear actuatorsare actually going to cost you about $100.
I expect that theShanghai cost of these will be about half of that. There is nothingintrinsically expensive about these things.
Joe Rossi has been agood source of information about this technology. Chris has been veryinterested in having a really fast transition speed for his reprap. Joenoted that...
With the proper drive and source voltage(48v and a chopper drive), we have got these guys Double stack versionsup to 20"/second with very light loads and fast lead screws.
Twenty inches/second is about Warp 9. I'll be happy if I can get one inch/second.
I've acquired a used Haydon linear actuator to fiddle with.

It's unipolar, but I'm not much troubled about that. It will let me find out how well my controller board works with it.

It's really very simple

Saturday, 26th April 2008 by Forrest Higgs

Once I got the x-axis circuitry wired and going it was a matter of about ten minutes to wire up the y-axis.



Mind, I'm just meatballing the wiring, so it looks really nasty. Works, though. I guess that I'll have to run this through Eagle and do a PCB of this when I get it to working prettily.

First tests of the Haydon linear actuator stepper motor

Monday, 28th April 2008 by Forrest Higgs

The linear actuator stepper motor that I acquired arrived today. Islapped it on to the Tommelise 2 Big Board and it ran immediatelywithout complaint.

Whatyou see here is the board cycling it backwards and forwards for 240steps with a 7 msec interval. It transitions at a rate of roughly 14.5mm/sec at that setting.
In that it is like the other tin cansteppers I've been working with. It can be made to run as fast as 17mm/sec, but not reliably. A 7 msec interval is the fastest setting thatit can operate at and deliver substantial force with no skipping.
The really amazing thing about this tin can stepper, however, is thatit is dead silent. As you can recall from other videoclips I've postedthe microphone on my camera is incredibly sensitive and made previoussteppers and gearmotors I've taped sound very loud. This one yousimply don't hear at all.
Mind, that is double what Darwin iscurrently achieving and means that it can print a maximum of roughly0.8 kg of plastic in 24 hours.

Nophead spots a major boo boo...

Sunday, 4th May 2008 by Forrest Higgs

I began with a kidney stone on Thursday morning and have been onpain meds the whole weekend. The smooth muscle pain meds make me feellike my head is in a sack of unwashed wool.

Chris (nophead)noticed that although I was trying to measure the HDPE/aluminumfriction coefficient, I'd neglected to take the HDPE cutting board outof the polythene bag it was in. He said...

Iread somewhere that the coefficient for friction for HDPE is not farbehind PTFE but it looks like your chopping board was still in its bagwhen you did the first test. Mind you I just got a new chopping boardwhich is black like yours and it is labeled PE LLD which I assume isLLDPE.

After a few moments of disbelief that I could havedone something so stupid I finally accepted that he was right and redidthe experiment.

To do the remeasurement I simply sat the PTFE experiment on top of the now unbagged HDPE cutting board.

The HDPE weighed 24 oz while the PTFE weighed 8 oz and I placed a 12 lblive load on it. It took 1 lb 4 oz to get it started moving and 1 lb 2oz to move it steadily.

μ = 18 oz/(24 oz + 8 oz + 192 oz) = 0.08

For the whole x/y positioning assembly the required thrust would be...

total thrust (xy) = 0.08(48 oz + 24 oz + 32 oz) = 8.32 oz

While for just the x-axis positioning table it would be...

total thrust (x) = 0.08(24 oz + 32 oz) = 4.48 oz

Thatmeans that I can easily use a 36000 series linear actuator for the xypositioning assembly and a 26000 series for the x positioning assembly.Who needs PTFE? Thanks, Chris!

Good to go

Tuesday, 6th May 2008 by Forrest Higgs

My hand full of used 26000 series linear actuators arrivedyesterday. While they had no lead screws, I was able to wire one up andassure myself that one, at least, wasn't broken.
Here is a look at the 26000's alongside of the 36000 that I already own.

Being native bipolar rather than unipolar as my 36000 series unit isthe 26000's are near as powerful as it is. Having the 26000's in-handI was able to verify that they require a 0.002"/step lead screw ratherthan the 0.004"/step lead screw that the 36000 does.
This morning I called in an order for six feet uncut of each of the twotypes of lead screw. When it arrives I should be able to buildTommelise 2 quite quickly.

Breaking the PC/Microcontroller comms bottleneck

Sunday, 11th May 2008 by Forrest Higgs

I2C EEPROMS offer the possibility of cheap, huge byte buckets toeven out comms flows from the PC to Tommelise 2.0 (T2)...

Sometime ago I shifted from the Pic 18F4610 to the 18F4550 so that I couldshift from very awkward RS-232 serial comms electronic to the much moreelegant internal USB 2.0 standard. That shift proved to be a runawaysuccess in terms of board simplification. I was able to completely getrid of the MAX232 board that shifts PC serial comms to a serial commsthat a Pic chip can understand. As well, the shift came just in time,because the new graphics work station I took delivery on last monthsimply doesn't have a serial port. Sadly, though, the shift ran away ina few unforseen ways as well with less happy results.

Copingwith USB data flow into a Pic is rather like putting a fire hose nozzlein your mouth and then signalling someone to turn it on. Data comes inat an enormous speed, even at the lowest setting (800 bytes/sec) andyour problem quickly shifts from one of getting data fast enough fromdeciding what to do with it once you have it.

I had noticedearly on with Tommelise 1.0 that when I was printing roads more than afew mm long I had little trouble. I would transmit the starting andending point of the print road from my PC to the 18F4610 controllerwhich would then translate it via the Bresenham line method intoinstructions for the positioning system's gearmotors. While veryefficient, the Bresenham line method still requires some divisions thatare, for a simple CPU like the Pic's, cycle intensive. For lines ofsignificant length, this isn't a problem. For lines making up a curve,however, which may be as little as a tenth of a millimeter long, thePic's cpu found itself rather overwhelmed. This led to the positioningsystem falling out of sync with the extrusion rate and blobbing theprint. An answer to this was to slow down the overall print speed. Bythe time I was printing at less than 1 mm/sec, however, I decided thata new approach was required.

Subsequently, this problem began tobe encountered with more mainstream Reprap prototypes, though, sincethey were using stepper motors rather than the much more controllerintensive gearmotors that I was using, at a lower level.

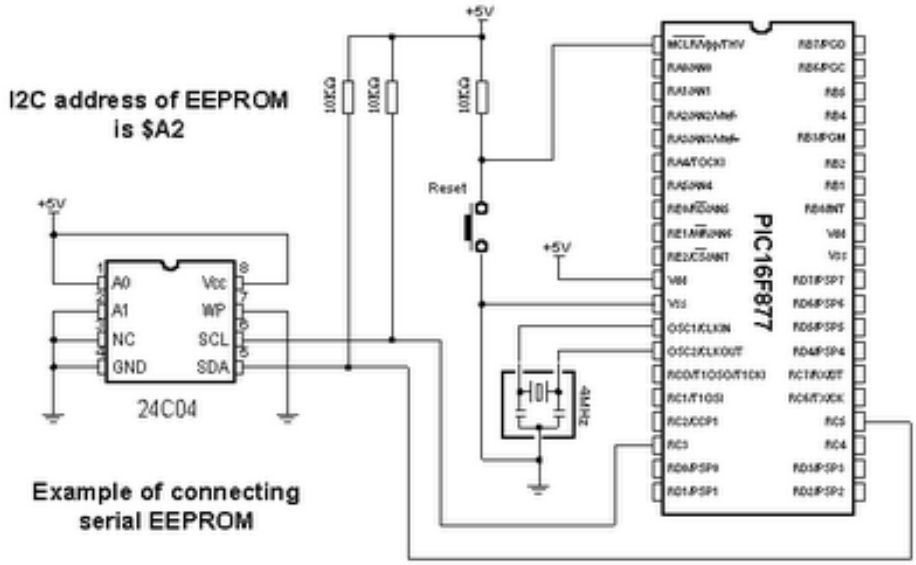
Iaddressed the problem of too much calculations being done in the Pic bydoing all of that in the PC and then shipping step by step controlinstructions over the USB link to the T2 controller. I also smoothedout the USB data flow with a 2 Kbyte RAM buffer within the 18F4550.

Istill wasn't satisfied. My Pic, fast as it was and compared with mostit was blazingly fast running at 48 MHz, still had to look after threestepper motors, at least one extruder motor and a handfull of limitsdetectors and the like. While I had no doubt that it can handle all ofthat, I still worried about pressing it close enough to its limits topreclude my making major additions to the control strategy. That hadbeen a problem over and over with the token ring Pic boards on Darwin.

Mainlywhat I was worried about, however, was getting operations statusreports from T2 back to the PC that I have operating it. USB comms,while much faster, aren't nearly as flexible as serial comms. Askingfor and getting back status reports from the PC seemed to be overlycomplicated given the steady flux of print instructions that the Picalso had to cope with.

The obvious way to go about that was toto use an SD card. Indeed, my new workstation came

with an SD slotbuilt in. The idea of just writing all the print data onto an SD cardand then reading it from the Pic has tremendous appeal. That approachleaves your entire USB line open for monitoring of the print's progressas opposed to transfering data to make the print happen at all. Theonly problem with that approach is that the SD card, in common with avariety of mass storage devices, behaves exactly like a mass storagedevice, viz, it has a file access table (FAT). Your Pic has to get theFAT entry for the file you've transferred and then go picking throughthe SD card itself to get the bits of data that make up the file. Onceagain, the overhead of doing all that was more of a load than I wantedto put on my 18F4550 given all the other things it already had to do.
WhatI really needed was something I hadn't seen in over 25 years, viz, a"byte bucket". These were little devices that were briefly in vogue atthe beginning of the PC era back in about 1980-1985. In those days, thePC's were slow, but the peripherals, especially the printers, werepositively glacial in speed. Microcomputer owners using the old CPM andearly versions of DOS noticed that they got fairly nice screen responsetill they decided to send something to their printer. At that pointthey might as well go out to lunch unless it was only a page or twolong.
Enter the "byte bucket". It was a device embedded into theserial cord that basically just had a bunch (very little in those daysgiven the astronomical cost of RAM) of static RAM memory that acted asa buffer between the primitive PC and the primitive printer. This was,more or less, the situation that I have been trying to cope with.
makinga USB line into a "byte bucket" is pretty much beyond my technicalcapabilities besides probably being a good idea in the first place.Remembering the "byte bucket", however, brought to mind another datastorage scheme that I tried out in about 1980. I was working for theSwedish government then and I had to build a datalogger for ascientific site in a romote part of the Algerian Sahara. It would bevisited several times per year and the data collected. While Ieventually used an old Siemens 4.25 inch floppy disk drive sealed in aplastic bag to keep dust out for the job, I had had a much more elegantsolution that I worked with that eventually proved impractical at thetime. Electrically erasable read only memories (EEPROMS) had just comeon the market. I had in mind to simply slot in an EEPROM burner boardinto my data logger and put the data on EEPROMS. The person visitingthe datalogger would simply take another EEPROM populated burner boardwith him and return with the data-filled one from the site.
Ihad to give up the idea, because the burner boards tended to blow outabout 10% of the EEPROMS that you put in them at the time. Theprogramming circuitry was simply too crude for what the EEPROM'srequired in terms of care and handling.
I revisited the idea and discovered this intriguing little schematic.

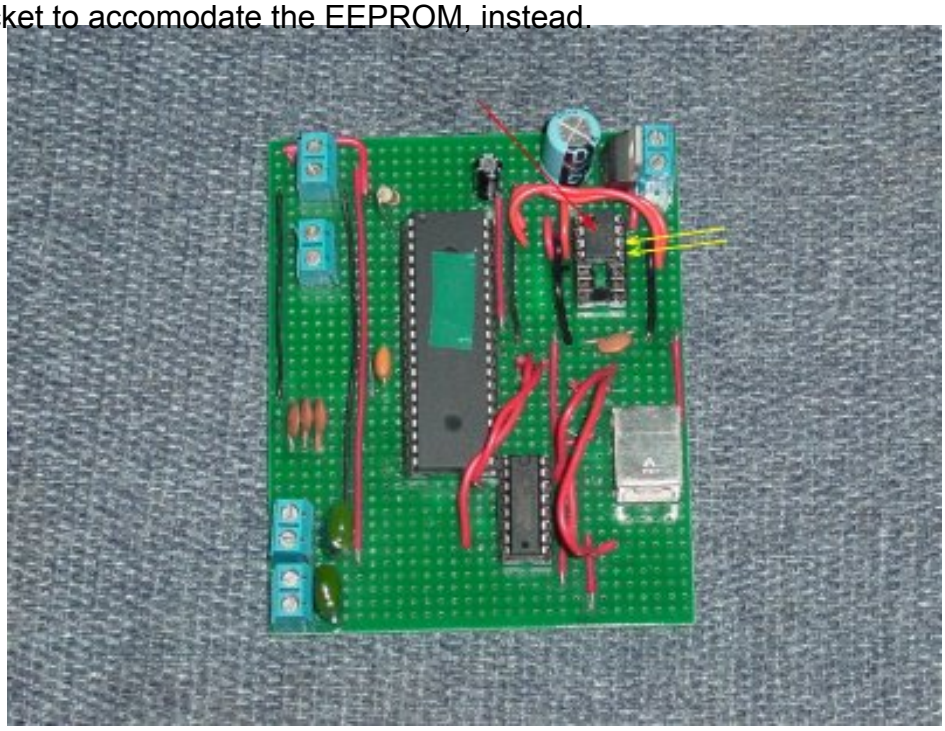


Thatis trivial! I could connect EEPROM's to my Pic via simple I2C commsprotocols. Initially, the examples I found were only putting a fewextra kilobytes onto the Pic. My ideas were much grander. I wanted toburn a huge amount of EEPROM memory and print for hours.
Googlingaround I discovered that none other than Microchip, maker of my Pics,had a solution. The 24FC1025 has a built-in I2C comms buffer, holds 128 Kbytes of dataand can be programmed in 3 seconds with low voltage from my Pic. Further it can be reprogrammed a million times before it wears out andcosts about $3.80 retail. Better still, you can address four of thesefor 524 Kbytes of storage from the same I2C port.
One of these cheap little chips can hold about an hour and a half'sprint instructions for T2 operating at a print speed of 7.5 mm/sec anddoing a 50% fill. Ever so often, and much more often than every 90minutes, one needs to reset one's axes so that you minimize cumulativeerrors from skipped steps by your stepper motors. Using four, which iseasy to do, gets you about 6 hours of uninterrupted printing.
The cpu overhead of reading off of one of these I2C-ready EEPROMS isminiscule While it takes 3 seconds to program one of these chips, youcan read it in a fraction of that time and the number of instructionsrequired to do a read is minimal.
Reprappers have rightly made a big to do about reducing the energyconsumption of their open-source 3D printers. T2 should be drawingunder 20 watts and printing as fast as Darwin is presently with thepossibility of going twice that fast. Sadly, what we are discoveringis that the PC is using most of the power, not the printer. Myworkstation, for example, draws 400 watts just idling and up to 800watts doing heavy processing.
Here's a breakdown. T2 will be able to print roughly 1 lb/day on fullduty. If I'm printing ABS that means that the ABS to keep the printergoing will cost about $7/day. Electricity to drive the printer toprint that pound of ABS will cost roughly 11 cents while theelectricity to drive my work station to look after T2 will cost roughly$2.10.
Not to put too fine a point on it, running T2 for a month would burn$210 of ABS, $3.30 worth of electricity to run the printer and $63worth of electricity just to keep the PC going. Fully 23% of the costof running T2 is taken up with electricity for the PC.
At first, I will be using the PC to monitor print progress in T2. Eventually, however, it seems reasonable to think about fixing themicrocontroller for T2 to run in stand-alone mode with a few LED's and,perhaps, a buzzer to sound alerts when things go awry. These EEPROMSshould give me a start on that.

Wiring up the 24FC1025 EEPROM

Friday, 16th May 2008 by Forrest Higgs

Integrating the 24FC1025 EEPROM on an existing board proves much easier than expected...
I decided that my first foray into the use of the 24FC1025 EEPROM wouldbe on my old 18F4450 prototype board. I already had a Dip socketinstalled left over from experiments with the inverter chip. With thatalready there it was a simple matter of cleaning out the old connectingwires and rewiring the socket to accomodate the EEPROM, instead.



You can see the red arrow pointing to the24FC1025 EEPROM. The twoyellow arrows point to the two pins that form the I2C connection withthe 18F4550 microcontroller. I am using, for now, RB6 (pin 39) and RB7(pin 40) for the I2C comms.
All of the rest of the wiring that you see to the other pins on theEEPROM are connections to +5v and GND to configure the chip's addressand to tell it to let the 18F4550 write on to it.
I've noticed in the threads that there is a big push for more buffermemory than four of these chips provide. Right now a suite of four ofthem will hold about six hours of print instructions for Tommelise 2. The big barrier that I have right now is space on the T2 Big board. IfI want, I can create an accessory board that requires only +5v and GNDplus a pair of leads for each bank of four EEPROM chips that I use. Each bank of four costs about $15 so it could well be that Nophead'snotion of using a SD card without a FAT directory is more costeffective.
Checking, I see that I can get a 3 GByte SD card from Tiger Direct for$10 and a connector for the card costs about $5 from Mouser,essentially the same price as my EEPROM approach with 2,000 times thememory. If I can read and write the flash memory in an SD card asquickly as I can EEPROM it will be a very good deal. The only bitmissing is how to write to an SD card without a FAT. Nophead seems toknow how to do that.
For now, however, I'm going with the EEPROMS. They solve theimmediate, if not the long-term problem of large scale buffering ofdata from the USB port.

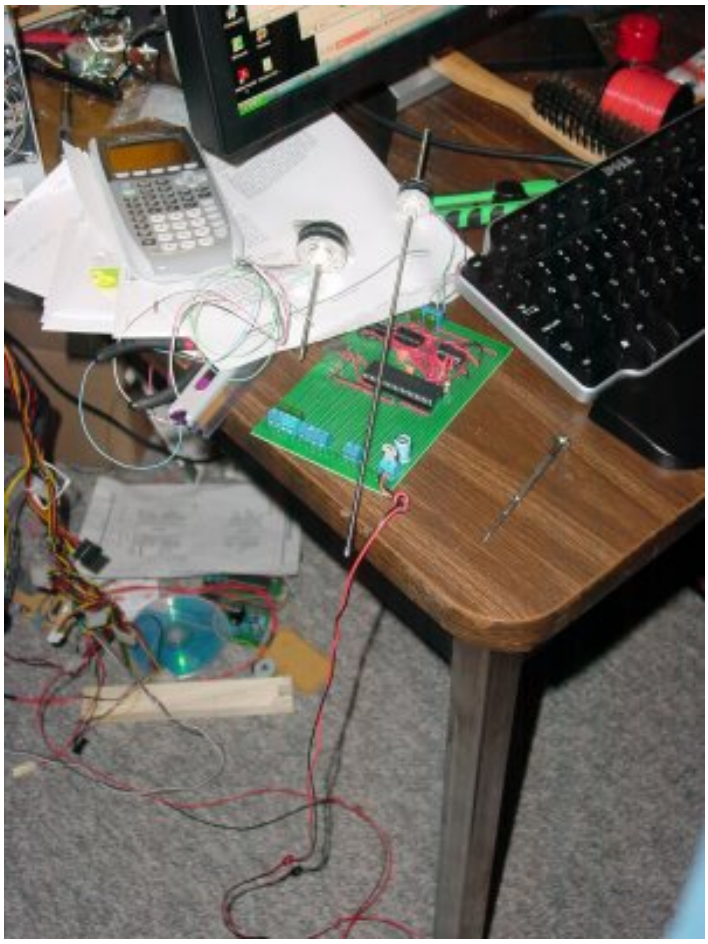
Testing the Haydon 26000 series linear stepper motor

Sunday, 18th May 2008 by Forrest Higgs

The performance of the used 26000 series linear stepper motors proves much better than expected...

Iacquired a handful of used Haydon 26000 series linear stepper motorsfor a song last month. Friday, I took delivery on a 6 foot long pieceof the 0.02"/step lead screw that goes with this model and this weekendI was able to cut a 390 mm lead screw for one of them and tested itwith my Pic 18F44550/SN754410 quadruple half-H driver chip board.

Here you can see the 26000 linear stepper motor with the 390 mm lead screw attached to the T2 Big Board.



I've placed the more powerful 36000 linear stepper motor with its short lead screw beside it for comparison.

Itested a Haydon 36000 series linear stepper a few weeks ago and managedto get a peak speed with significant torque of about 15 mm/sec. Withits shorter step and smaller size I didn't expect as good performancefrom the 26000.

Imagine my surprise when the 26000 happily ran at 650 Hz and managed 34mm/sec with significant thrust. I tuned it down to 400 Hz and got abit over 20 mm/sec out of it with substantial thrust and no complaints.

Oddly, these steppers, which are ordinarily designed for either 5v or12v operation, had been specially wound for 7v. I ran them with about11v input at a reduced duty cycle to avoid

overheating them. It is arather simple matter to use a L7805CV regulator to convert 12v power tothe approximately 8v input into the SN754410 that I need to power thesesteppers.

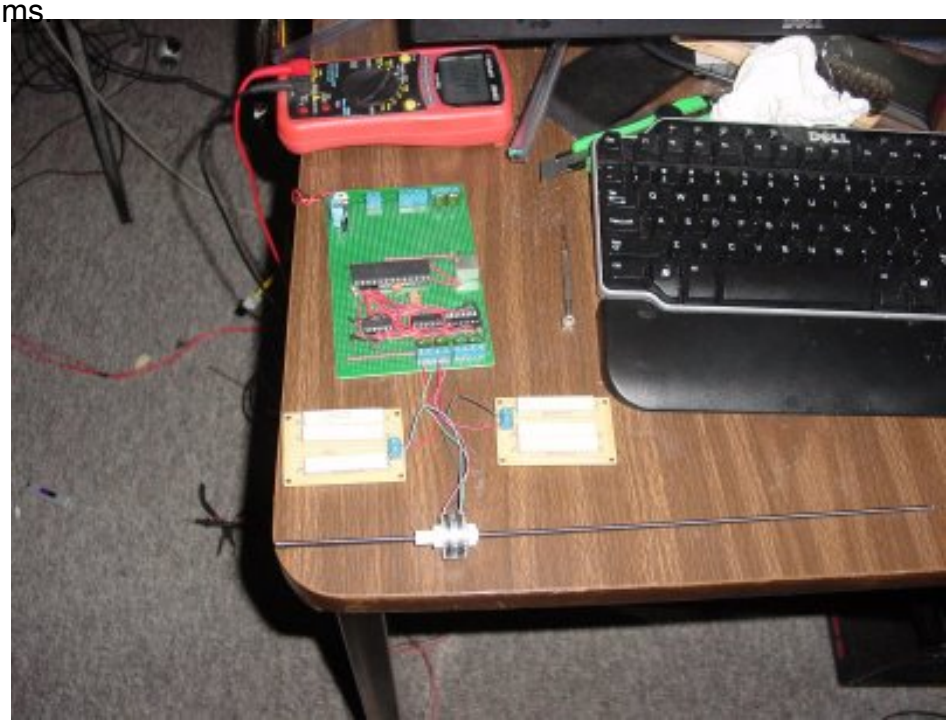
Tuning the voltage on the Haydon Series 26000 linear stepper

Monday, 19th May 2008 by Forrest Higgs

Results from running the Haydon series 26000 linear stepper at its design voltage...

Theset of used Haydon series 26000 linear steppers that I acquiredrecently had windings designed for 7.05v. I took a bit of time offduring lunch and bought a set of wirewound resistors to put into serieswith the motors to bring the voltage down from 11v to something closerto 7v.

I needed about 15 ohms and about 5 watts powerdissipation. Radio Shack had 10 ohm, 10 watt wirewound resistors. I puttwo of them in parallel to get 5 ohms and then attached the third inseries to get my 15 ohms.



One resistor assembly was required for each phase of the bipolarstepper. From a comments chat I had with Nophead there was some hopethat I would be able to achieve somewhat higher speeds with thestepper. That turned out not to be the case. I was, however, able torun the stepper continuously without its overheating.

The 26000 ran somewhat slower than it did with higher voltage. Whereasbefore I had been able to achieve 650 Hz (~33 mm/sec) reliably I wasnow only able to achieve 400 Hz (~23 mm/sec) with usuable thrust and333 Hz (~17 mm/sec) with enough thrust, I think, to drive the whole x/ypositioning table. That is more than double what I'd hoped for.

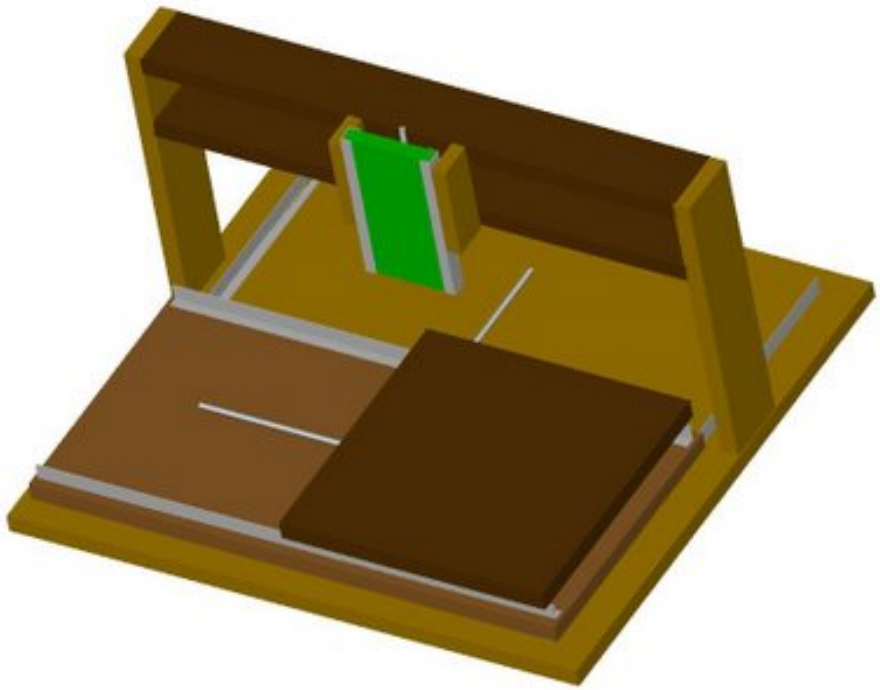
The wirewound resistors that I bought today were quite pricey. Each ofthose little resistor boards cost about US$7. Mouser, however, has apurpose made 15 ohm wirewound resistor rated at 10w for US$0.68.

T2 takes shape

Wednesday, 21st May 2008 by Forrest Higgs

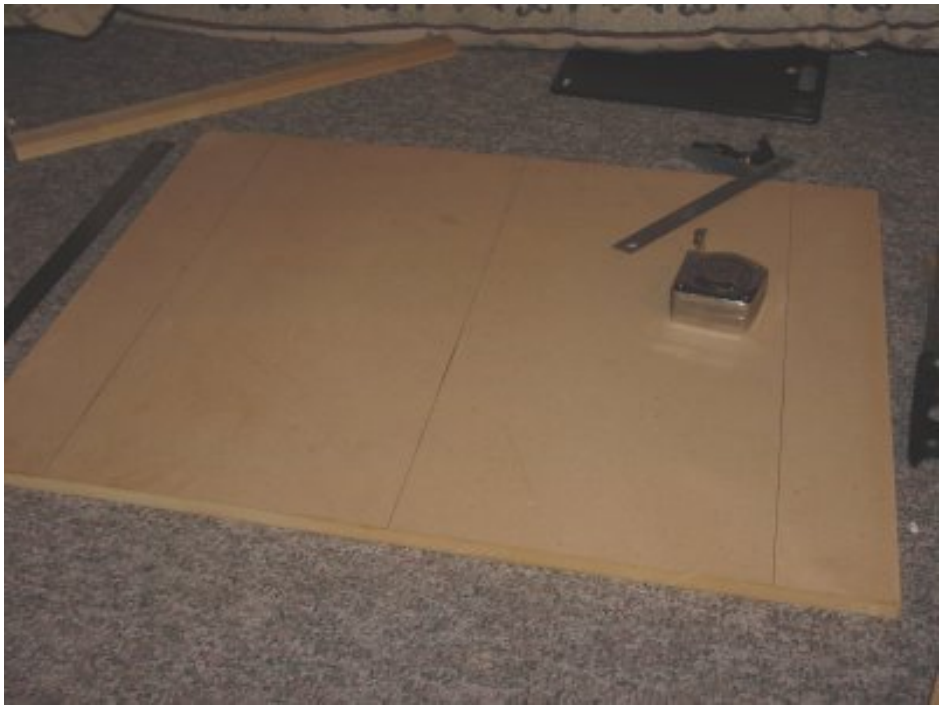
Work begins on Tommelise 2.0's y-axis...

Tommelise 2.0 uses a fairly standard CNC gantry geometry.



Youhave a working area of 12" x 12" which can be positioned anywhere inthe xy plane which is carried by a 12" x 24" x-axis. The x-axis iscarried in turn by a 24" x 24" y-axis. The 6" z-axis is mounted on agantry which straddles the xy working area.

While the footprint for T2 is four times larger than Darwin's, it is a remarkably simple design to build.



I made the base plate of particleboard which I cut to 24" x 32". Iallowed 4 inches on either side of the 24"x24" working area toaccommodate the risers for the gantry.
In order to provideclearance for the linear stepper motors which move the axes I mountedthe aluminum "L" guide rails on top of two, 24" long 1-1/2" x 3/4"poplar boards.

Isecured these to the base board with 3", 1/4"-20 bolts. I locked theboards onto the plate with c-clamps and drilled the bolt holes at onego through both the poplar and the base board.

Afterbolting the poplar board down it was a simple matter to tap thealuminum "L" extrusion and slide it over the exposed bolt ends for afirm friction fit.
Repeating the process for the other rail was trivial.

Drilling and installing the second rail took a matter of minutes.
It took a total of about an hour to do all of the cutting, drilling and bolting of the base plate and x-axis. I used...

- small hand saw
- 24" straight edge
- standard metal hack saw

- 4 " jaws vise (for holding the aluminum "L" extrusion while I cut and drilled it)
- carpenter's adjustable square
- 2 x 4" c-clamps
- cheap mitre box
- cheap power drill
- 1/4" metal/wood drill bit
- 3/8" prick punch for setting the holes to be drilled in the aluminum

Please pardon the obsessive list. I'm trying to keep track of what tools are required to build T2.

Building the y-axis table for T2

Thursday, 22nd May 2008 by Forrest Higgs

Stealing ideas from the McWire CNC makes the work go quickly...

Ibought a few pieces of perspex, lucite to you Yanks, to use instead ofthe monotonous particle board. It's a bit lighter and a whole lot morephotogenic. To create the y-axis support table for the x-axis, I used a24" x 6" inch piece of perspex to bridge the gap between the 1-1/2" x3/4" x 24" poplar supports for the aluminum guide rails that willeventually support the x-axis stepper and the xy working surface for T2.

Hereyou can see the poplar supports and perspex (still in its blueprotective wrapping) jigged with c-clamps and ready for drilling.

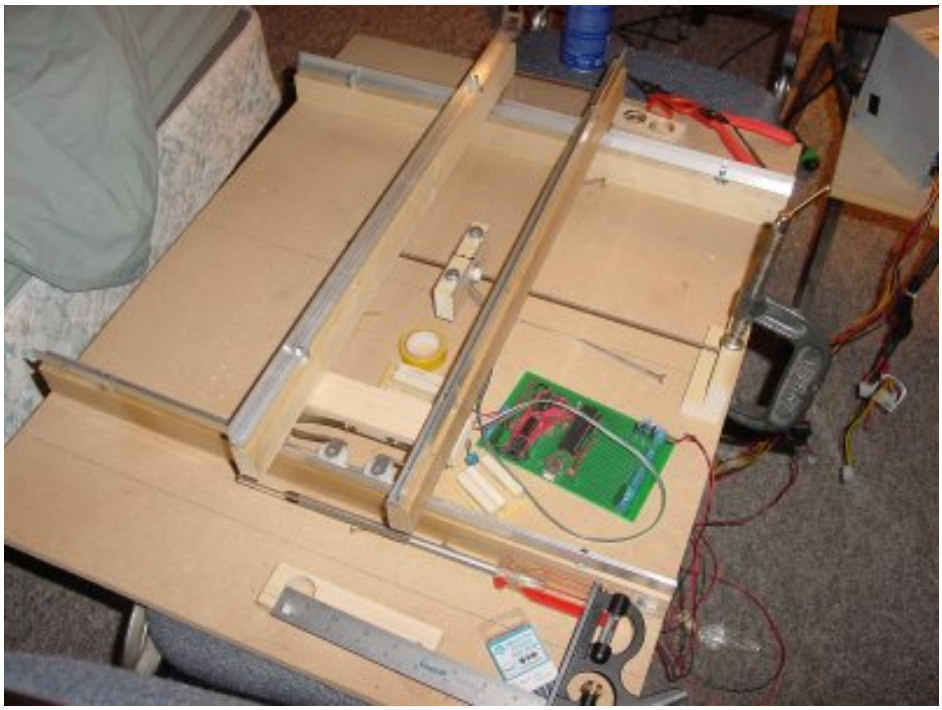
Drilling and setting the structural bolts is a matter of a minute or two.

Aquick trip to the hardware shop got me some drawer glides to serve moreor less the same purpose for T2 as they do for drawers.

Running the Tommelise 2.0 y-axis

Saturday, 24th May 2008 by Forrest Higgs

It appears that the Haydon 26000 series linear stepper is more than adequate to run the y-axis... This is going to be a short report in that I am exhausted after a long day at my day job and this getting this drive lashup rolling. Mind, that second bit wasn't all that hard. I was just nervous that the 26000 series wasn't going to deliver enough torque to do the job. I needn't have worried.
I finally settled on mounting the stepper on the moving y-axis table and keeping the lead screw fixed.



The installation is simple and neat. The stepper is slipped into a 1" diameter hole in a 4" x 1-1/2" x 3/4" poplar. The lead screw is screwed into an identical piece in which a 3 mm hole has been drilled at the proper height.

I was able to run the y-axis at 18.4 mm/sec reliably with a 32 ounce load. I've had the axis cranking back and forth for over an hour now. The action is smooth as silk. I do, however, have to do a little work on the acceleration/deceleration coding in the firmware. I accelerate all right, but the deceleration isn't what it needs to be.
At no point on the stepper can or the SN754410 quadruple half-H driver chip exceeded 25 degrees C.
Saturday morning, I sorted out a few things with the guide rails and ran the axis under load. I dialed back the transition speed to a touch over 12 mm/sec.

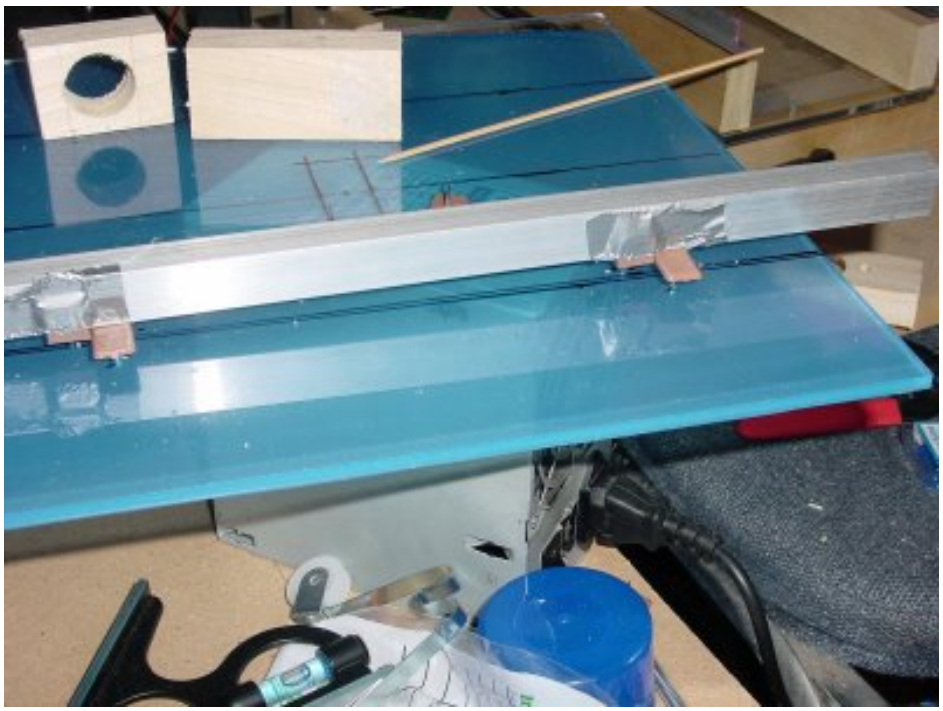
At that setting it handled 5 lbs of dead load without complaint. The stepper can and driver chip warmed up to just at 40 degrees at that loading as I expected but there were no problems.

Building and running the T2 x-axis

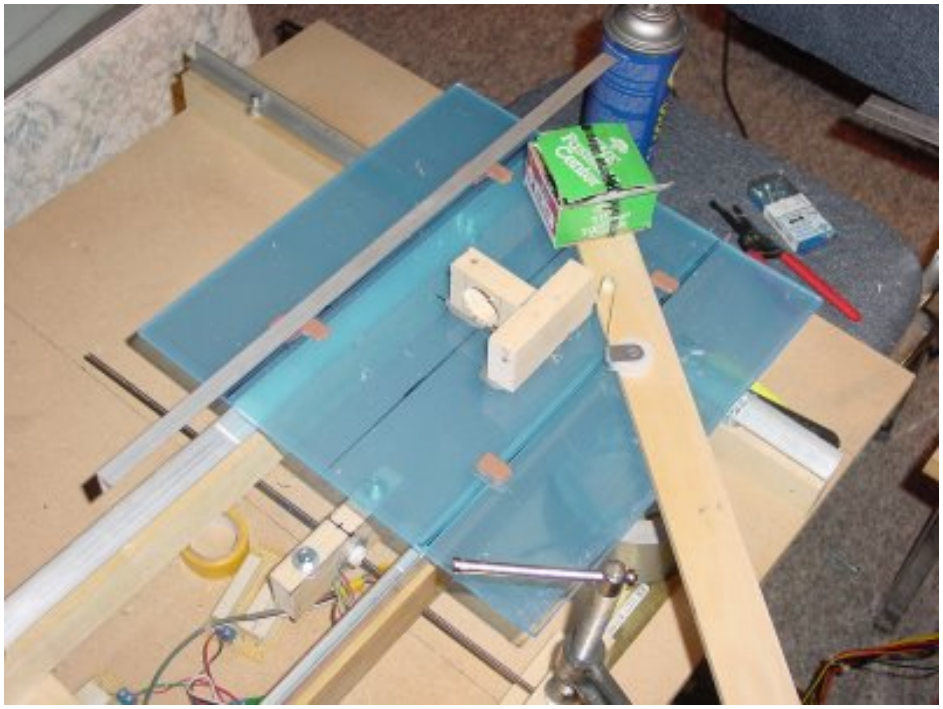
Sunday, 25th May 2008 by Forrest Higgs

After getting the y-axis running, building and testing the x-axis was *almost* a snap...

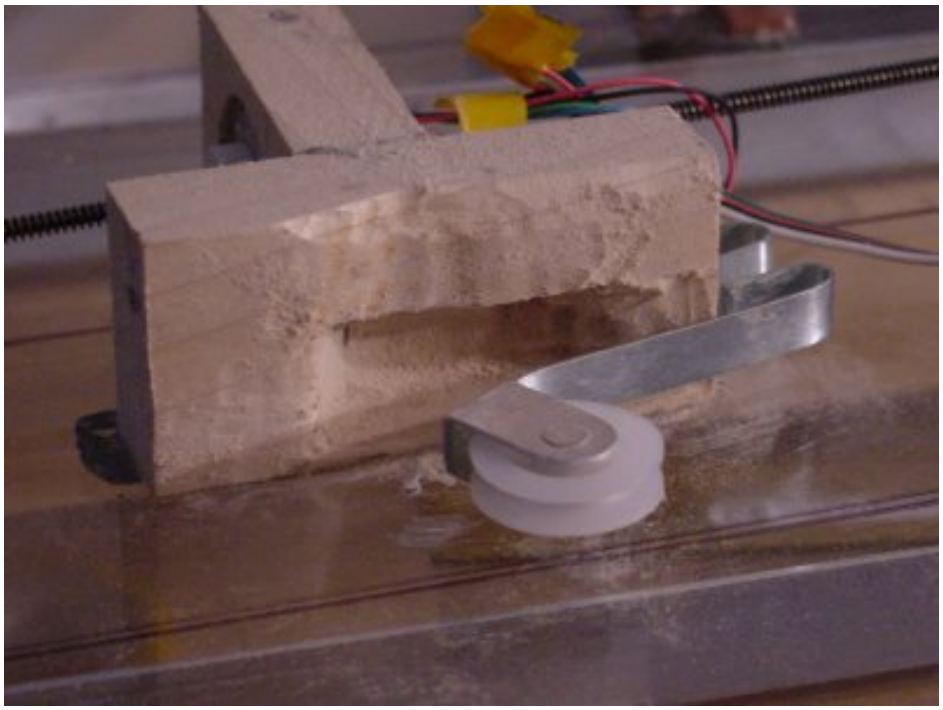
I learned a few things from building the x-axis. The most importantlesson was to go about placing the drawer glides on the axis table in amore reasonable manner. The "more reasonable manner" was to duct tapethe glides onto a piece of aluminum guide rail and then mark where thenail ends ought to be going onto the axis table.



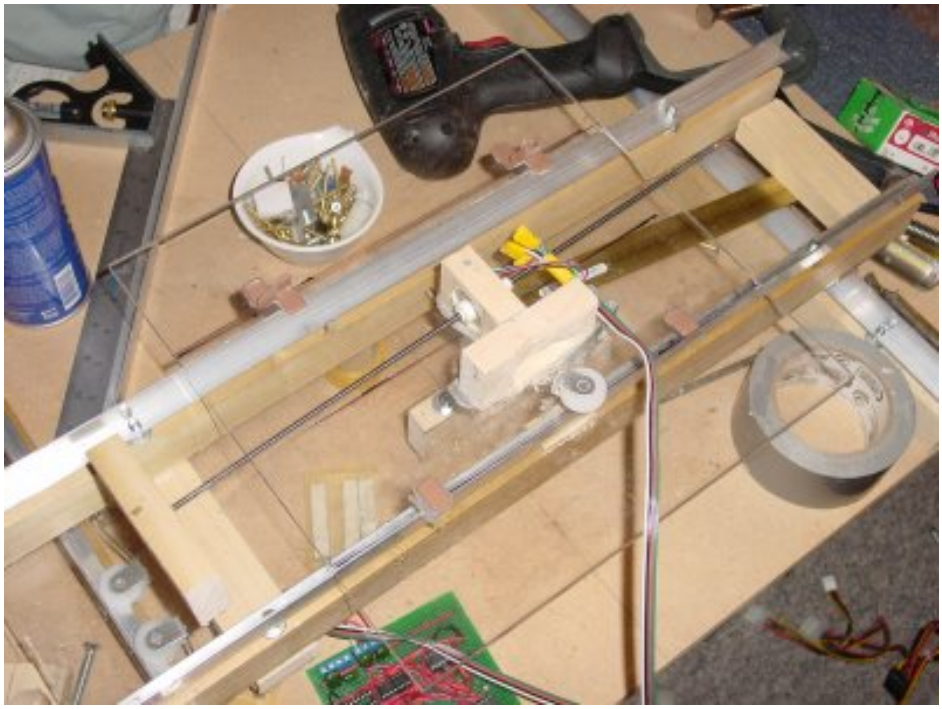
This worked relatively well, which is to say that it was a lot easier to dothan the catch-as-catch-can approach I used on the y-axis.
Iwanted a flat surface for the x-axis table so that I could attachwhichever printing surface I needed. For this reason I avoided usingbolts as I had with the y-axis and used epoxy instead to attach mountsand glides.



Because of the width of the piece of perspex I had I set the x-axis rails about6 inches apart. Unfortunately, this made a problem with seating thebogey for this axis. There was simply not enough room between the bogeyand it's mounting block. Because of this most of the time I spentbuilding this axis was expended on excavating a recess in the bogeyblock to allow the bogey room to properly seat between the bogey blockand the rail. This took about half an hour with my Dremel tool and agrinder.



I'll know better than to get into this kind of fix in the next prototype Ibuild. Here you can see the whole x-axis ensemble prior to my seatingit and connecting up the lead screw.



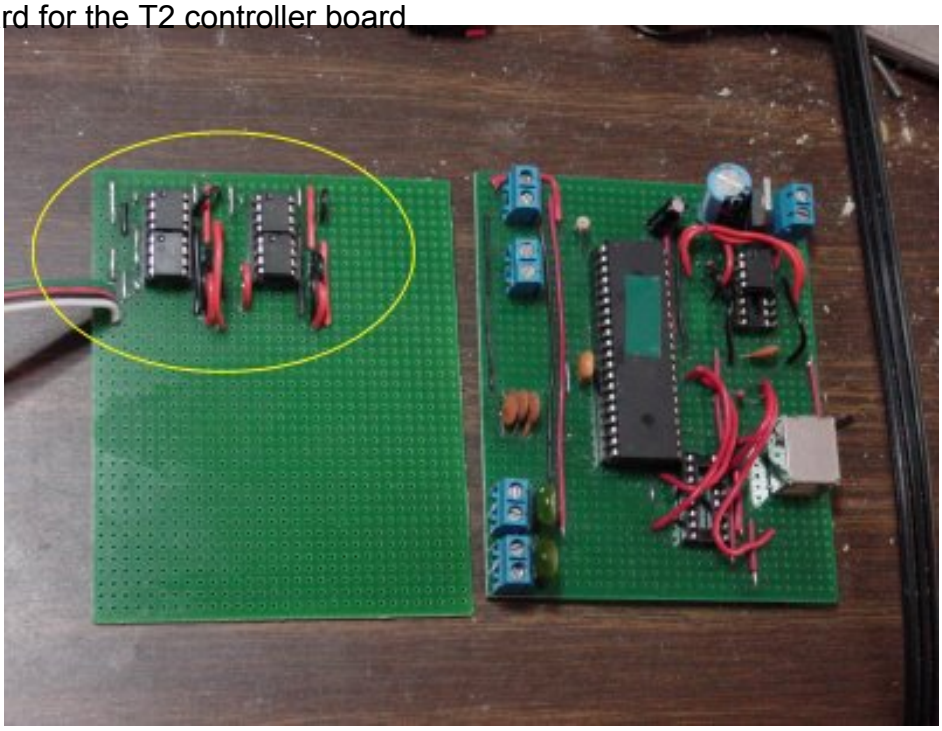
Once I tacked the lead screw down with some wood shims and a c-clamp, we were good to go.

A 512 KByte data buffer for T2

Monday, 2nd June 2008 by Forrest Higgs

Having proved the practicality of using EEPROM for a print instruction buffer a full-sized one is built for T2...

AfterI'd got used to how to write to and read from the 24FC1025 EEPROM chipI put together a little add-on board for the T2 controller board



You can see it circled in yellow on the upper left part of half-sizedEuroboard format stripboard that I use for development. I'm attachingit to the T2 controller board with the flat cable carrying the SDA,SCL, 5v and GND lines. The T2 controller board is already crowded withthe stepper driver chips so I decided that it was good sense to justappend a bit more board space rather than crowding the chips on themain board.

I finished my usual messy job of soldering and wiring early thismorning. Now I will put it away for a day or two so that I can havefresh eyes and no preconceptions when I test it for short circuits andwiring faults. I've found that not getting in a hurry when you areprototyping new circuits gets the job done a lot faster and with a lotless drama, smoke and crackling.

T2 USB transfer speed takes a big jump

Tuesday, 3rd June 2008 by Forrest Higgs

The comms rate for the USB connection between the PC and the 18F4550 took an 8-fold jump... read more

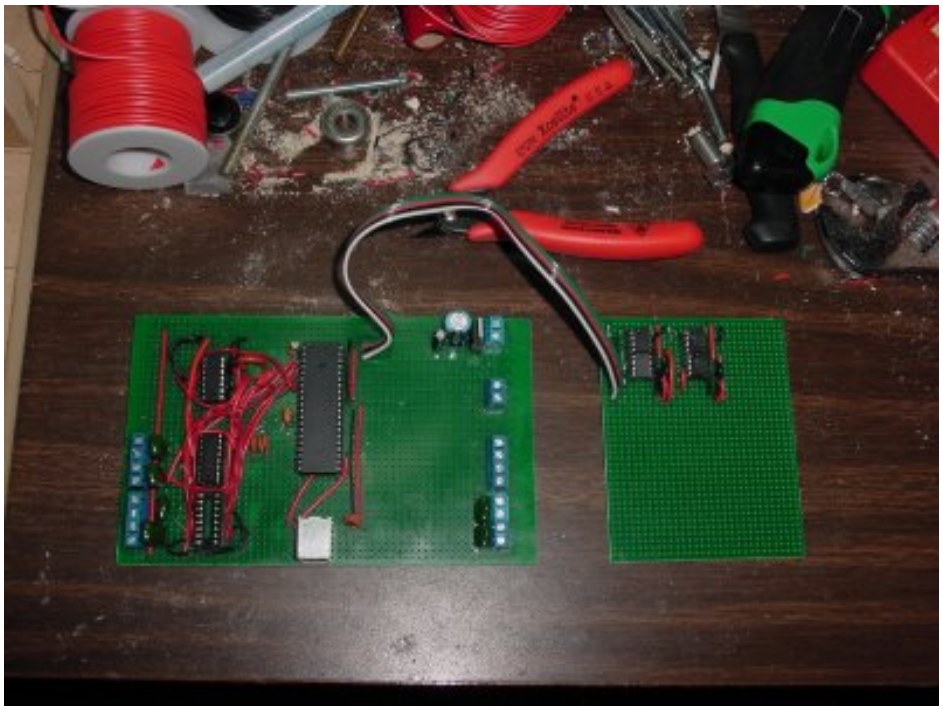
Vladimirat Oshonsoft wrote the USB module for his Pic 18F BASIC compiler abouta year ago. When I started using it some months ago I noticed thatregardless of what I did it was pretty much impossible to get it tomove data at much more than 800 bytes/sec. After establishing that Icould use the USB module, albeit at a rather low speed, I went on toother things. I messaged Vladimir about this issue, but didn't receiveda response. Mind, Vladimir has had health issues for the past year so Ididn't press the matter.

A few days ago when I was working on the EEPROM buffer problem I began linking the work that I was doing to the Oshonsoft IDE user communitythat chats on Yahoo groups. I mentioned that I couldn't drive the USBlink beyond about the low speed setting of 800 bytes/second and Chris Bate there said that he'd been getting somewhere between 5-7,000 bytes/sec out of it.

Igot to thinking about that and realised that Vladimir had made a newoffer on his compilers and IDE's that I had taken up and that I had anewer edition of his USB module than I had originally worked with. Thisevening I tested the new USB module and discovered that it veryconsistently delivered 6481 bytes/sec with compiled VB.NET code on myPC.

That means that instead of 11-12 minutes to load my 512Kbyte EEPROM data buffer, it now takes about 90 seconds including theprogramming of the four EEPROMS. That is ***NICE***!

I also wired the 512 Kbyte buffer add-on board to the partially built T2 big board.



The big Haydon 3600 series linear actuators are supposed to arrive tomorrow.

512 Kbyte EEPROM buffer operational on T2 big board

Wednesday, 4th June 2008 by Forrest Higgs

Circuit chasing and firmware testingof the 512 Kbyte EEPROM buffer from the Tommelise 2.0 controller boardis complete...

Late this afternoon I set down with my multimeter and checked all ofthe circuits in the 512 Kbyte data buffer. I'd forgot to drill one ofthe strips between one side of an EEPROM circuit. Other than that, thebuffer circuitry was fairly cut and dried.

Firmware testing, however, was a little more complicated. EEPROMs 1,3, and 4 tested out perfectly, but chip 2 wouldn't take data. Afterquite a bit of puzzled checking I discovered that I had configured the#2 chip to be a #3 chip in addition to the ordinary #3 chip that Ialready had. The buffer circuitry was writing the data that shouldhave gone on the #3 chip on both the #3 chip and the #2 chip withoutcomplaint. Once I sorted out the swapped leads the board tested outproperly.

There was one mishap, however. After doing the circuit chasing I putthe chips back on the T2 big board and got the mux inverter chip inbackwards. When I powered the board up it promptly overheated, ofcourse. I caught it fairly quickly, so I am not sure whether it wasruined. The buffer testing didn't require the inverter chip, so I justpulled it and got on with the testing. I've got plenty of inverterchips, so whether or not it is ruined is of little consequence.

With luck tomorrow I will be able to take a crack at writing thefirmware that will allow me to fill the buffer with print data andmaybe run a stepper motor or two. That should be fun, especially nowthat I have a fast USB connection.

PIC 18F Instruction Set

Saturday, 7th June 2008 by Forrest Higgs

d = 0 means destination is WREG. d = 1 means destination is FILE and this is the default.

Byte Oriented

ADDWF f,d,a	Add W to f where d=0->W, d=1->f, a is generally not specified (access bank stuff)
ADDWFC f,d,a	Add W and Carry bit to f
ANDWF f,d,a	And W with f
CLRF f,a	Clear f
COMF f,a	Complement f
CPFSEQ	Compare, skip if f==W
CPFSGT	Compare, skip if f > W
CPFSLT	Compare, skip if f < W
DECF f,d,a	Decrement f
DECFSZ f,d,a	Dec f, skip if 0
DCFSNZ f,d,a	Dec f, skip if not 0
INCF f,d,a	Increment f
INCFSZ f,d,a	Increment f, skip if zero
INFSNZ f,d,a	Increment f, skip if not zero
IORWF f	inclusive-OR W with f
MOVF f,d,a	Move f (usually to W)
MOVFF f,ff	Move f to ff
MOVWF f,a	Move W to f
MULWF f,a	W x f
NEGF f,a	Negate f
RLCF f,d,a	Rotate left f thru Carry (not-quite multiply by 2 with carry)
RLNCF f,d,a	Rotate left (no carry)
RRCF f,d,a	Rotate right through Carry
RRNCF f,d,a	Rotate right f (no carry)
SETF f,a	Set f = 0xff
SUBFWB f,d,a	Subtract f FROM w with Borrow
SUBWF f,d,a	Subtract W from f
SUBWFB f,d,a	Subtract W from f with Borrow
SWAPF f,d,a	Swap nibbles of f
XORWF f,d,a	W XOR f

Bit Oriented

BCF f,b,a	Bit clear, bit is indexed 0 to 7
BSF f,b,a	Bit set
BTFSC f,b,a	Bit test, skip if clear
BTFSS f,b,a	Bit test, skip if set
BTG f,b,a	Bit toggle

Control Operations

	BC n	Branch if Carry, n is either a relative or a direct address
	BN n	Branch if Negative
	BNC n	Branch if Not Carry
	BNN n	Branch if Not Negative
	BNOV n	Branch if Not Overflow
	BNZ n	Branch if Not Zero
	BOV n	Branch if Overflow
	BRA n	Branch Unconditionally
	BZ n	Branch if Zero
	CALL n, s	Call Subroutine
	CLRWDT	Clear Watchdog Timer
	DAW	Decimal Adjust W
	GOTO n	Go to address
	NOP	No operation
	POP	Pop top of return stack (TOS)
	PUSH	Push top of return stack (TOS)
	RCALL n	Relative Call
	RESET	Software device reset
	RETFIE	Return from Interrupt and Enable Interrupts
	RETURN s	Return from subroutine
	SLEEP	Enter SLEEP Mode
Literals		
	ADDLW kk	Add literal to W
	ANDLW kk	And literal with W
	IORLW kk	Incl-OR literal with W
	LFSR r,kk	Move literal (12 bit) 2nd word to FSRr 1st word
	MOVLB k	Move literal to BSR<3:0>
	MOVLW kk	Move literal to W
	MULLW kk	Multiply literal with W
	RETLW kk	Return with literal in W
	SUBLW kk	Subtract W from literal
	XORLW kk	XOR literal with W
Memory		
	TLBRD*	Table Read
	TLBRD	--etcetera; more table stuff--

Extended Instruction Set

Not used at this time

Talking to the EEPROMs through the USB link

Monday, 9th June 2008 by Forrest Higgs

After a great deal of difficulty, I got the PC-USB-18F4550-EEPROM link working smoothly...

I discovered a little something in the Oshonsoft IDE reference manualabout the USB module that is a bit misleading if you don't read veryclosely. I'm the sort of programmer that depends very heavily on codesamples rather than close readings of the manual. If you're like methat will get you into trouble with USB in the Oshonsoft manual.

Here's how it works. Looking at the relevant part of Vladimir's sample code for USB, which works very well, bye the way.

```
loop:
    Adcin 0, an0
    If an0 < 50 Then
       PORTB = 0
       UsbStop
       While an0 < 100
       Adcin 0, an0
    Wend
    PORTB = 0xff
   UsbStart
   PORTB = 0
   Endif
   UsbService
   Goto loop
End

usbonftout:
    Toggle PORTB.7
Return

usbonftin:
   UsbFtBuffer(0) = UsbFtBuffer(0) - 1
   UsbFtBuffer(1) = UsbFtBuffer(1) - 1
   UsbFtBuffer(2) = UsbFtBuffer(2) - 1
   UsbFtBuffer(3) = UsbFtBuffer(3) - 1
   UsbFtBuffer(4) = UsbFtBuffer(4) - 1
   UsbFtBuffer(5) = UsbFtBuffer(5) - 1
   UsbFtBuffer(6) = UsbFtBuffer(6) - 1
   UsbFtBuffer(7) = UsbFtBuffer(7) - 1
Return

usbonioout:
   Toggle PORTB.6
```

```
Return

usbonioin:
    UsbIoBuffer(0) = UsbIoBuffer(0) + 1
    UsbIoBuffer(1) = UsbIoBuffer(1) + 1
    UsbIoBuffer(2) = UsbIoBuffer(2) + 1
   UsbIoBuffer(3) = UsbIoBuffer(3) + 1
   UsbIoBuffer(4) = UsbIoBuffer(4) + 1
   UsbIoBuffer(5) = UsbIoBuffer(5) + 1
   UsbIoBuffer(6) = UsbIoBuffer(6) + 1
   UsbIoBuffer(7) = UsbIoBuffer(7) + 1
Return
```

Notice how he just toggles the LED's on PortB pins 6 and 7 in the "out" routines and does some quick calculations in the "in" routines.

The naming of the "in" and "out" routines are unfortunate in that Whenyou get data IN from the PC the "out" routine is the first one called.When you go to send it back to the PC the "in" routine is called.That's sort of backwards, to me at least.

Anyhow, silly person that I am I decided that the "in" routines were whereI needed to do my manipulation of the incoming USB data from the PC. Igot that idea looking at the sample code. That works up to a point.You can do a considerable amount of data manipulation in that routine.Unfortunately, if you do too much calculations you will find that thePIC 18F locks the dual port RAM committed to the USB link just when itit gets ready to send the data stream back to the PC. If you haven'tfinished executing your code in the "in" routine when that happensyou'll find that your routine hangs on the statement which tries toaccess the IoBuffer and what you wanted to send doesn't get sent.

If you shift all that coding to the "out" routine the problem goes awayin that that routine is executed just after the data arrives from yourPC and has to finish executing before anything else in the way of USBactivities can happen.

Anyhow, I spent two days figuring that out. I hope this post saves you all the cursing and confusion that this caused me.

After getting past that hurdle the rest of the link code firmware was rather trivial to develop.

I'm going to try to talk Vladimir into developing some low level USBcommands like the ones he did for the I2C commands that were so usefulin connecting up the 24FC1025 EEPROMS. I was reading in the datasheetfor the 18F4550 that you can specify a USB buffer up to 1024 bytes. Vladimir has it set to 8 bytes currently. If I could up that to 128bytes I could get rid of quite a bit of firmware code and write more orless directly from the USB port onto the 128 byte EEPROM buffer. Thatwould be quite a bit more elegant than what I have currently.

I know that I could do the same thing by inserting assemblerinstructions into my BASIC code. The firmware I have right now,however, works fine and I'd rather spend my time attacking the mountainof other tasks that I need to get done to get Tommelise 2.0 runninginstead. It's good enough for now.

I may be a bit quiet for the next week or so. I'm getting my gallbladder removed Thursday and the surgeon tells me that it will be lateSaturday or Sunday before I'm up and around again. Wish me luck. :-D

Mapping print data onto the EEPROM buffer bank

Monday, 23rd June 2008 by Forrest Higgs

<!-- /* Style Definitions */ p.MsoNormal, li.MsoNormal, div.MsoNormal{mso-style-parent:"";margin:0in;margin-bottom:.0001pt;mso-pagination:widow-orphan;font-size:12.0pt;font-family:"Times New Roman";mso-fareast-font-family:"Times New Roman";}@page Section1{size:8.5in 11.0in;margin:1.0in 1.25in 1.0in 1.25in;mso-header-margin:.5in;mso-footer-margin:.5in;mso-paper-source:0;}div.Section1{page:Section1;}-->

A few simple functions allow for mapping of print dataonto a bank of EEPROMS...

First, I must apologise for the long gap between this and my last post. On 12 June I went in for what was billed as a routine bit of gall bladdersurgery. It went rather dramatically wrong and I wound up a lot sickerand sorer than I'd any reasonable expectation of having been. This lastfew days, however, I've been on the mend to the point of hammering on theTommelise project again from time to time.

The 24FC1025 EEPROM chip is a lovely little device in someways and areally nasty one in anothers. While it's cheap and holds a lot of data onthe plus side it also has a vexing little addressing scheme. Onesector on one chip will hold 65536 bytes of data. To fill the chipcompletely you have to switch sectors. Further, if you want to use asmany as four of these chips you also have to give it a ship address (0-3).

What I wanted, otoh, was to use the maximum number of chips (4) and write orread any byte off of the entire bank of EEPROMS (524288 bytes total) withone LONG integer (4 bytes).

Figuring out where you were in a specific sector is easy. You simply usea single WORD (2 bytes) integer. What wanted addressing specifically wasthe sector and chip addresses. Here is how I did it in OshonsoftBASIC. Given the LONG variable, ***absolute_address...***

```
Function ctl_byte(absolute_address As Long) AsByte

  'this function takes a cross-chipaddress and generates
  'the appropriate control byte for itsans the
  'read/write bit

  Dim ctl_byte_1 As Byte
  Dim ctl_byte_word_1 As Word
  Dim dummy_byte As Word

  dummy_byte = 0xa0

  ctl_byte_word_1 = absolute_address.HW

  ctl_byte_1 = ctl_byte_word_1.LB

  If ctl_byte_1 < 0x01 Then
```

```
  'chip 1, sector 0
  dummy_byte.1 = 0
  dummy_byte.2 = 0
  dummy_byte.3 = 0
Else
  If ctl_byte_1 < 0x02 Then
    'chip 1, sector1
    dummy_byte.1 = 0
    dummy_byte.2 = 0
    dummy_byte.3 = 1
  Else
    If ctl_byte_1< 0x03 Then
     'chip 2, sector 0
     dummy_byte.1 = 1
     dummy_byte.2 = 0
     dummy_byte.3 = 0
    Else
      Ifctl_byte_1 < 0x04 Then
       'chip 2, sector 1
       dummy_byte.1 = 1
       dummy_byte.2 = 0
        dummy_byte.3= 1
      Else
       If ctl_byte_1 < 0x05 Then
         'chip 3, sector 0
         dummy_byte.1 = 0
         dummy_byte.2 = 1
         dummy_byte.3 = 0
       Else
         If ctl_byte_1 <0x06 Then
           'chip3, sector 1
           dummy_byte.1= 0
           dummy_byte.2= 1
           dummy_byte.3= 1
         Else
           Ifctl_byte_1 < 0x07 Then
             'chip4, sector 0
             dummy_byte.1= 1
             dummy_byte.2= 1
             dummy_byte.3= 0
           Else
```

```
            Ifctl_byte_1 < 0x08 Then
              'chip4, sector 1
              dummy_byte.1= 1
              dummy_byte.2= 1
              dummy_byte.3= 1
             Else
              'wehave an out of
               'range absolute
               'address
              dummy_byte= 0xff
            Endif
          Endif
        Endif
      Endif
    Endif
   Endif
  Endif

  ctl_byte = dummy_byte
End Function
```

You select byte 2 and which directly relates to the chip and sector on theEEPROMS and then convert that byte to its equivalent control byte which you usewith I2C addressing of the EEPROM bank. The structure looks a bit intimidating,but it actually executes in about 8 µsec on the PIC 18F4550. I will probably change what Ihave here for a SELECT CASE structure eventually. It doesn't make muchdifference in the resulting assembler, though, so it's a bit pointless.

To get the sector address out of the absolute addess takesanother, even smaller function.

```
Functioneeprom_chp_sect_addr(absolute_address As Long) As Word
  Dim dummy_word As Word
  'parse the absolute address
  dummy_word = absolute_address.LW
  'get the eeprom's sector address
  eeprom_chp_sect_addr = dummy_word
End Function
```

Measuring extruder temperature using Adrian's circuit

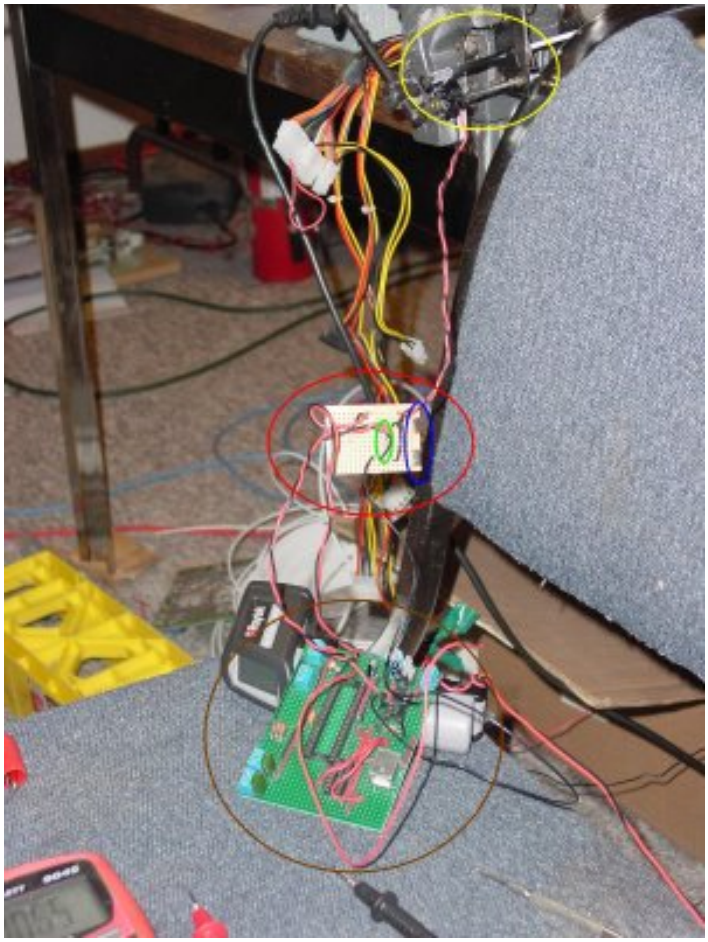
Wednesday, 25th June 2008 by Forrest Higgs

I prototyped and tested Adrian's proposed circuit **for using the resistance of the Nichrome 60 heater on Tommelise's extruder barrel to measure its temperature...**

Quitesome time ago, Vik Olivier suggested that it should be possible to usethe resistance of a Nichrome heater wire to also measure thetemperature of what it was heating if one were clever enough. A fewdays ago, Adrian Bowyer published a circuit design that could make thatpossible. I intended last weekend to do a lashup of the circuit to seeif it would be of use in my next generation extruder design forTommelise 2.0. Because of some other firmware issues that took longerthan I expected I didn't get around to cobbling the circuit togethertill yesterday. It seems to work quite nicely.

A little background, first. While, I've been quite addicted to thenotion of a single control board design for literally years now, Irecently decided for convenience to have a main controller board runthe positioning system and another, smaller board run the extruder(s). The reason for that change was more practical than philosophical. Iknew very well what was needed to run the positioning system. Theextruder, thanks to the growing number of people making and usingDarwins with their Mk II extruder, had entered a period of rapidevolution. What with dribble controls, encoding the pump shaftrotation and now this nice new proposal for measuring extrudertemperature, it seemed to me that it would be good to just freeze themain control board design and put another microprocessor to work justlooking after the exturder(s). In my design the two boards would talkto each other via a I2C link, a technology that I've recently had a lotof experience with in setting up my EEPROM buffer.

To that end I put an add-on board to handle Adrian's circuit onto my old 18F4550 prototype board. You can see the lashup here.



The yellow circle at the top of the pic encloses the actual barrelextruder barrel. The red circle in the middle marks the add-on boardand the brown circle at the bottom encloses the 18F4550 prototype board.

Looking more closely at the add-on board (yellow circle) I've markedthe two transistors that are TIP 122's in Adrian's design and BD681'sin mine. The green circle right beside them encloses a relatively highwattage 1.6 ohm resistor a handfull of which I bought to build upZach's stepper controlled board and then never did. I couldn'tunderstand why Adrian put the 220 ohm resistors between the transistorsand the microcontroller, so I left them out. Mind, I expect that thereis some very good reason and I'll find myself putting them back in atsome point. Mostly, I didn't have any 200 ohm resistors in my supplydrawers, so, since I hadn't used any such thing heretofore I omittedthat part of his design.

The Tommelise extruder heater is rated at 6 ohms. Figuring a nominalvoltage to the heater of 12v that means that I should read about 3.3vat the sense point when the system is cold. Resistance for Nichrome 60at 250C is about 1.15 times its resistance at ambient. What that meansis that the voltage measured should drop about 0.34v over a range ofabout 230C. Given that the 18F4550 has a 0-5V range, 10 bit A/Dcircuit on-board what you are looking at is about 3.3C resolution forthis kind of temperature measurement.

While I've not had time to design the firmware I did a cursory test ofthe circuit manually by tapping the extruder heater with a 5v supplysource and then quickly measuring the voltage on the other side of thecircuit with a digital meter after I'd disconnected the power input. The voltages are moving in the directions that Adrian suggested, so Isuspect that my circuit is, more or less, right.

I fiddled with a range of resistor values for the 2.6 ohm one that Iwas using and discovered that while I could move the cold voltagemeasurement up and down fairly easily, there was relatively littlechange in the range of voltage that temperature differences yielded. Adrian suggested that one might want to put an op amp chip into thecircuit to get better resolution. Another way to approach this wouldbe to use a higher resolution A/D chip instead. Microchip offers both a 12 bit, I2C compatable A/D converter and a 16 bit, I2C compatable A/D converter for about US$1. That would get your resolution down to 0.8 - 0.05C. Idon't know why you'd want more resolution than that. Errors in otherpart of the circuit would be much higher than that.

Given that I know how to make I2C comms work and next to nothing about op amps, that's the direction I'm likely to take.

Headaches

Sunday, 29th June 2008 by Forrest Higgs

Electronics and firmware development proves to be much easier than software...

I've pretty much got the EEPROM print data buffering scheme working andcan see my clear to how I need to get the extruder controller board anddesign to do what I want. The next obvious task was to get someproper printing data so that I could test out the whole PC/Tommelisedatalink.

My first impulse was to dump my VB.NET Slice and Dice software infavour of Enrique Perez' Skeinforge script that he'd written for Art ofIllusion. I hadn't been paying attention recently, however. It turnsout (note from Nophead) that the beanshell Java scripts aren't exactlywinners in the computational efficiency race. What that meant is thatit could take AoI executing the Skeinforge script sometimes over a dayto get you a set of G-Code print instructions.

Enrique rewrote the script in Python and it apparently executes in amagnitude less time now. It is no longer, however, directly compatablewith AoI and several steps are required to take an AoI STL file intothe the Python version of Skeinforge.

I was very disappointed with all this. Once I started really readinginto Enrique's progress with Skeinforge, I discovered that while it isexcellent coding it is still a work in progress. Because of that I've decided to give Skeinforge a miss for now and stick with Slice and Dice.

I've done this for a couple of reasons. First, I know the code fairlywell and can get into it and make fixes without the overhead of gettingfully up to speed with Pyton. Second, I'm reluctant to make part ofTommelise 2.0's operating package dependent on yet another open sourcelanguage (Python). I accepted Beanshells Java scripts because Art ofIllusion is written in Java. Throwing Python into the mix just makesthe whole thing a bit too complicated. When Skeinforge stabilises Imay well port the logic into Visual Basic.NET and replace Slice andDice with it.

Finally, Nophead was able to isolate a problem with the Boolean intersection and union module in Art of Illusion that I'd long suspected but was unable to isolate. I'd had so much trouble with wonky code in Slice and Dice in the past that I'd gone to considerable trouble to make it able to graphically play the slices so that I could check them for problems before committing to a print. That pretty much decided me on sticking with Slice and Dice for now.

Now I have to get reacquainted with Slice and Dice's many peculiar little ways again.

Nophead's recommended extrusion temperatures

Sunday, 29th June 2008 by Forrest Higgs

Excerpt from

http://builders.reprap.org/2008/06/making-extruder-extrude-followup.html

Only this weekend did I find a " Guide for Newbies" that specifies some temperatures that the wise and skilled nophead has been using:

HDPE.raft_temp = 200
HDPE.first_layer_temp = 240
HDPE.layer_temp = 220
PCL.raft_temp = 0 // no raft
PCL.first_layer_temp = 130
PCL.layer_temp = 120
ABS.raft_temp = 200
ABS.first_layer_temp = 215
ABS.layer_temp = 230
PLA.raft_temp = 0
PLA.first_layer_temp = 180
PLA.layer_temp = 160

Alas, when I cranked the temperature up to 200C (I'm using HDPE), theextruder extruded just like it was supposed to. Piece of cake. :)

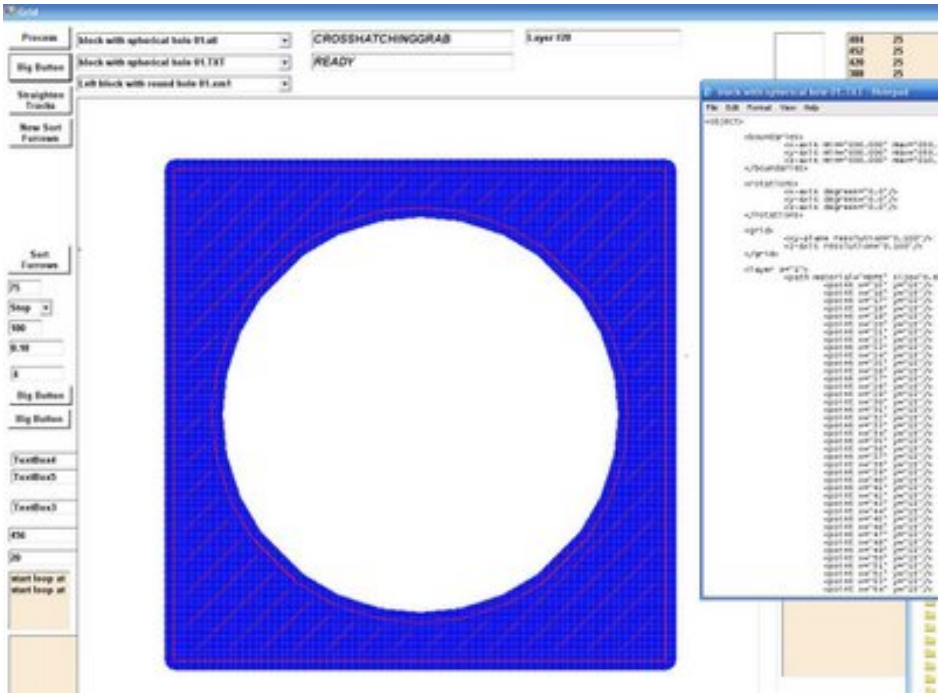
Dusting off Slice and Dice

Monday, 30th June 2008 by Forrest Higgs

Slice and Dice is resuscitated for use with Tommelise 2.0...

Ihadn't worked with the Slice and Dice code that I developed originallyfor Tommelise 1.0 for nearly a year. Just before I set all that aside,I did a massive bit of reworking of the code. As a result of that, thelatest versions of Slice and Dice were not what I remembered havingworked with and took had a bit of a learning curve to them.

Most of the getting reacquainted had to do with the resolution of theanalysis grid. Once I adjusted that for Tommelise 2.0 things began tosort themselves out fairly well. This morning I created a simple testobject consisting of a 50x50x10 mm block from and subtracted a 40 mmcentered sphere from it. You can see the top layer of the graphic ofthe print instructions here.



On the right of the graphic you can see the XML rendering of the printinstructions. Interestingly, I spent a lot of time collecting lines sothat I could create GCode-like line descriptions. With the buffereddata format that I am using all I had to do was take a step back fromthat final description and I pretty much had what I needed withoutfurther transformations.

A bit more about Zach's opto-endstop

Wednesday, 23rd July 2008 by Forrest Higgs

A small sharing of experience in incorporating Zach's ZD1901 opto-endstop into a controller board...

Hooking up Zach's opto-endstop was atrivial exercise. You provide ground, +5 v and get back a 5 v signal.To receive the signal from the opto-endstop, I set PortB.5 to an inputby setting TrisB.5 =1. I then rewrote the test program for the y-axisto reset the start point for the axis when it got a good signal fromthe opto-endstop.

It didn't work the first time out forreasons which caused me to include this short note. I tend to dotemporary lashups with sticky tape and the like to test things. Thethinking is that once I have everything running I can design propermounts and such using Tommelise 2.0 and print them out, replacing allof the bits of wood and tape. This opto-endstop mounting was nodifferent.

I had, of course, to use something tobreak the signal in the H21LOI. For that I cut a strip out of a pieceof red construction paper, reasoning that it was thick enough andopaque enough to do the job. I was wrong. It wasn't.

From there I selected a much thicker piece of white paper of heavy poster quality with a heavy clay fill. No joy.

When I took a piece of carton(cardboard) off of the back of a shrink package of 16 pin sockets fromRadio Shack and dipped that into the gap of the H21LOI I got a goodsignal back. I then cut a long strip of similar carton off of the sideof a package of hack saw blades and taped it onto the y-axis. I alsotaped that opto-endstop to the baseboard of Tommelise 2.0 as you willsee.

Resetting the y-axis with theH21LOI opto-endstop from Forrest Higgs on Vimeo.

That done, the test routine performedperfectly. The moral of this story is never underestimate thepenetrating power of the IR beam in a ZD1901. It may not look likemuch, but it is nonetheless powerful.

One other thing gave me pause which Iwill mention for completeness. Ordinarily, my controller boardsinclude a L9805CV voltage regulator that converts 12 v from my ATXpower supply to 5 v for the logic circuits on the controller board. Idon't use a heat sink on the regulator simply because the load that thelogic chips put on it is not substantial.

When I put new circuitry on a board,however, I tap it briefly at frequent intervals to check to see if

itis heating up, that being a sure sign that I have a short somewhere onthe board. This evening I noticed that it was warmer than usual. Without the opto-endstop the regulator was relatively cool to thetouch. With it it was warm. Not hot, but warm. I checked my wiringadditions and there were no shorts. I also tapped all of the chipschecking to see if any of them had gone bad with similar negativeresults.

Being obsessive, I checked theamperage going into the board via the 12 v input while the y-axis plusopto-endstop was in operation and got 190 milliamps. I also checkedthe 5 v supply going to the opto-endstop board and got 20 milliamps.

Finally, I used my IR thermometer tomeasure the temperature of the regulator and got 36 Celsius. Roomtemperature was 21 C. Apparently the drain from the endstop board iswarming the regulator up a bit, but not alarmingly.

In future, though, when designingcontroller boards I will place the L7805CV somewhere else on the boardwhere I can slap a heat sink on it easily. A heat sink will let it happily handle loads up to 1 amp.

Tommelise 2.0 x-axis operational

Thursday, 24th July 2008 by Forrest Higgs

Aused Haydon 26000 series linear stepper has been installed on thex-axis of Tommelise 2.0 and connected to the controller board andtested...

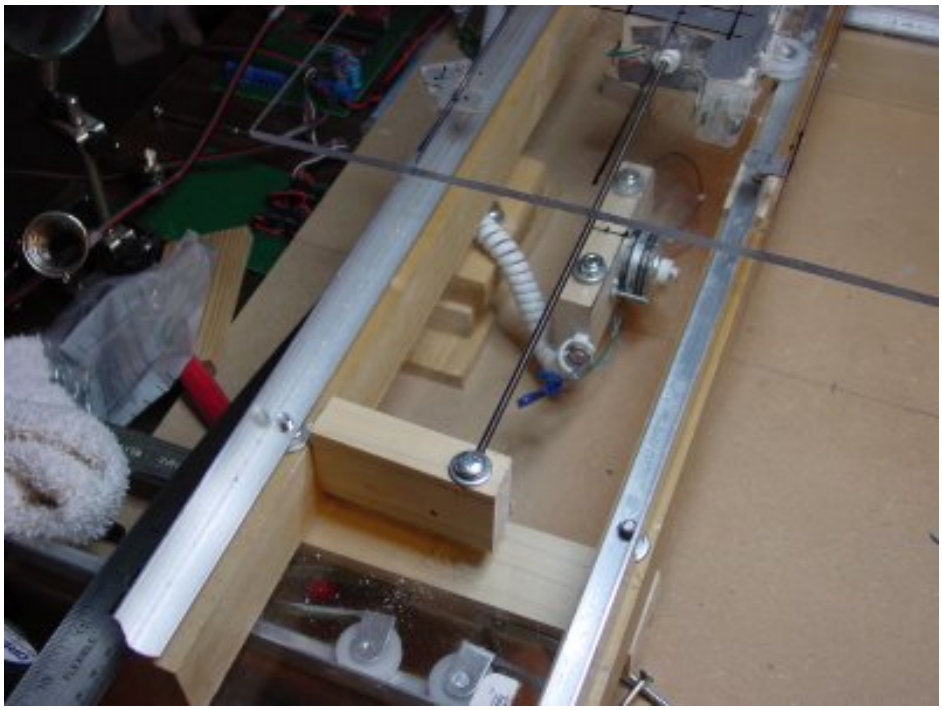
This is just ashort note that I've connected up the x-axis on Tommelise 2.0 with oneof my many used Haydon linear steppers and run it in with a testprogram from the controller board. The 26000's that I bought have alead screw with precisely half the pitch (0.05 mm/step) that the 36000has. Because of that I have to run it at twice the step rate of the36000 to achieve the same transition rate as the y-axis with its muchlarger 36000. Happily, the 26000 has proved to be well up to thatchallenge.

The 26000 makes a hum, not a loud one but it isthere. I could comfortably sleep with this thing running in my bedroom.It is about as loud as an old-fashioned electric clock that uses a walloutlet for power. All the same it is still much quieter than the fan onmy power supply. I'm going to have to do something about that powersupply, I suspect.

Final assembly of the Tommelise 2.0 x-axis from Forrest Higgs on Vimeo.

I was able to pile 4 lbs of basmati rice onto the x-axis while itwas operating. Six pounds caused an occasional step skipping at thespeed that I am pushing the 26000 at.

Securing the lead screw for the x-axis was trivial. I simplyscrewed a scrap of poplar wood into the side of the guide rail box forthe axis, grooved the top of the scrap and secured the lead screw witha screw tightened washer as you can see here.



I've run in the new assembly for several hours without incident. Iwill integrate an opto-endstop

with the axis tomorrow, time and day jobworkload permitting.

Using (or not) telephone coil wire

Thursday, 24th July 2008 by Forrest Higgs

Saveyourself some grief. Don't try to do connections with movingelectronics on your positioning system with coiled telephoneconnectors...

I have a very strong recommendationto make to anyone tempted to use coiled telephone connector wire tomake connections to moving parts on your positioning system.

Go lie down till the feeling goes away.

I managed to make it work to connectthe axis-mounted linear stepper motor for the y-axis to the base of mypositioning system on Tommelise 2.0, yesterday. It took me about halfan hour to make eight soldered connections successfully.

This morning I was trying for arepeat performance with my x-axis. Four more connections. Afterthirty minutes of unsuccessfully trying to solder four connections Igave up in disgust.

Here is the problem. The conductorsin those coiled wires are braided out of copper filament that is lessthan hair thin (My hair, anyway. I actually compared the two under amagnifying glass). The whole braid is a #26 or thinner. The coilingis done by a tough plastic coating over this braided conductor and thewhole shebang looks like it is then coated with something like a PVCouter layer.

The PVC is easy to strip. Thatcoiled inner plastic layer, however, is a bastard to strip off thebraided conductor without either breaking it entirely or damaging it. When you've managed to get it stripped properly you find yourself thenhaving to solder it to an inevitably much thicker wire, either braidedor solid. It's like trying to glue fuzz to a wall. The coiled plasticcoating of the wire has to be held against the larger wire with eitherelectrical tape (NOT) or an alligator clip (this barely works). Onceyou manage to successfully solder the connection you still worry aboutwhether you are going to accidentally stress these soldered joints andbreak the connections.

I've replaced the coiled telephoneconnector for the x-axis with a bundle of braided bell wire. It tookme about 5-10 minutes to make the bundle and solder the connections. It doesn't look so elegant, but it works and it didn't make me crazierthan I already am. :-)

Tommelise 2.0 x and y axes operating in concert

Thursday, 24th July 2008 by Forrest Higgs

Both the x and y axis of Tommelise 2.0 are not running in concert with a stand alone firmware test program...

I finished the wiring for the x-axisand reconnected the y-axis. After getting a few things backwards (Ireally must mark left and right on the pieces), the whole xy printingtable began doing transits as you can see here.

X and Y axes running in concert from Forrest Higgs on Vimeo.

I have had the x-axis drivercircuitry wired for months but hadn't tested it due to the long delayin getting the USB and EEPROM firmware working. This morning I justslapped a driver chip (SN754410) in the socket and took a chance thatit would run correctly. It didn't. In fact, it didn't run at all.

I'd just picked up a driver chip offof my worktable and slapped it in the socket. A vague memory came backto me from months ago that just before I'd started working seriously onthe USB firmware and wiring I'd accidentally ruined such a chip. Iswapped the one in the y-axis socket out and the x-axis startedperforming immediately.

After tossing the bad driver chip inthe rubbish bin (finally), I pulled a new one out of inventory andinstalled it. At that point both axes started running.

All that remains to do is to mount the opto-endstops properly (both are working).

At that point I'm facing big time firmware writing to trigger stepper pulses via interrupts.

Off to the races

Friday, 25th July 2008 by Forrest Higgs

It appears that the the faster you run the Tommelise 2.0 xy table the quieter it is...

While assembling the xy positioningtable I've been running it at 12.5 mm/sec. This morning, I decided tosee how fast I could make it go with a practical load. That hum thatthe 26000 was making bothered me, so I slowed the positioning tabledown and it got louder. Oddly, the noise level dropped as I speed the25000 up.

The highest rate that I could run thecomplete table without losing steps with no load turned out to be 22.5mm/sec. The 26000 is as quiet as the 36000 at that speed. Interestingly, the noise level for the 36000 seems to stay about thesame whatever speed I run it at.

With a load of 2 lbs I was able toreliably run the xy positioning table at 20 mm/sec. It startedskipping at 4 lbs. It might have been able to handle a load between 2lbs and 4 lbs but I only had 2 lb bags of basmati rice so I wasn't ableto explore that possibility.

What that means is that for adiagonal infil, which is most of a print job, I can print at (20mm/sec) (2^0.5) or roughly 28 mm/sec.

The performance curves for the Haydonlinear steppers suggest that I could get a lot more thrust out of themfor a particular stepping rate with a better driver circuit. Thatsuggests that I could run them a lot faster. Frankly though, I'm farmore interested in just getting reliable, high definition prints atthis point than matching the print rates of commercial printers.

Mind, I was prepared to be very happywith 6-7 mm/sec transition rates, given that Tommelise 1.0 with it'slittle Solarbotics gearmotors was managing 3.6 mm/sec with fair skiesand a stiff wind behind it. It looks like I am going to be doing a lot better than that.

Stepping with interrupts

Sunday, 27th July 2008 by Forrest Higgs

The rough test modelfor running the xy positioning table is being evolved into firmwarewhich will ultimately run Tommelise 2.0...

Heretofore, the test program thatruns the x and y axis stepper motors and senses when boundaries areexceeded consisted of a simple endless loop. Timing for steps wasachieved by using WaitMs commands. While that is handly, it is also avery limited approach. For print mode I want to generate interruptswith the Pic 18F4550's internal clock which trigger the stepper steps.

I had code of that sort left overfrom previous projects, so I adapted it to this purpose and had thewhole thing running in interrupt mode by noon. Here you can see the xand y axes operating in interrupt mode.

X and Y axes running on interrupt from Forrest Higgs on Vimeo.

The Haydon 26000 linear stepper motorwhich runs the x-axis (the top one) is a used one which was originallywound for 7 v. The little green circuit board with the big white powerresistors lying on the near end of the x-axis assembly compensate forthe fact that the 26000 is being fed 12 v. power instead of 7 v.

You can also see the microcontroller board and the EEPROM board lying on the floor of the y-axis (the bottom one) to your left.

Here you can see the limits detection system for the y-axis in pretty much it's finished form.

Testing the limits for the y-axis from Forrest Higgs on Vimeo.

I've pulled the limits in to 6 inches(155 mm) so that I won't have to do the final adjustments that I willneed to do to get the full 11 inches (280 mm) of work space out of theaxis.

These videos give you an idea of thesimplicity of the Tommelise 2.0 concept. Once I decided to

go withlinear stepper motors the non-printable "vitamins", to use Dr. AdrianBowyer's happy term, consist largely of the electronics and the linearstepper motors and their lead screws plus a few bits that are needed tomake the extruder.

The rest of the bootstrap system canbe made from pretty much whatever comes to hand. I've chosenchipboard, a piece of acrylic plastic and milled poplar plus a few bitsof aluminum extrusion and fasteners largely because they can be formedinto what I need with a minimum of effort and relatively few handtools. While I am quite proud of the clean feel of the xy positioningtable, it could have been as easily built of quite differentmaterials. Aside from the linear stepper motors no part of what yousee was mandatory. While I used aluminum "L" extrusions, you couldhave as easily used, say, steel rods. The acrylic was used because Iliked to the look that it gave, not because it was critical to thesuccess of the design. Ditto, the brass shim. It looks great, but anytruly opaque material, such as stiff plastic or cardboard, which I usedearlier, would have done as well.

That is the strength of the Tommeliseapproach to positioning. It uses what, for now, is a relativelyexpensive stepper and lead screw arrangement. There is, however,nothing intrinsically expensive about them. Similar linear stepperscan be had out of Chinese factories for less than $5/stepper.

First steps towards printing a stepper motor

Sunday, 27th July 2008 by Forrest Higgs

Printingstepping motors, or parts of them at least, may be the next step toreducing the "vitamins" required to grow self-replicating 3Dprinters...

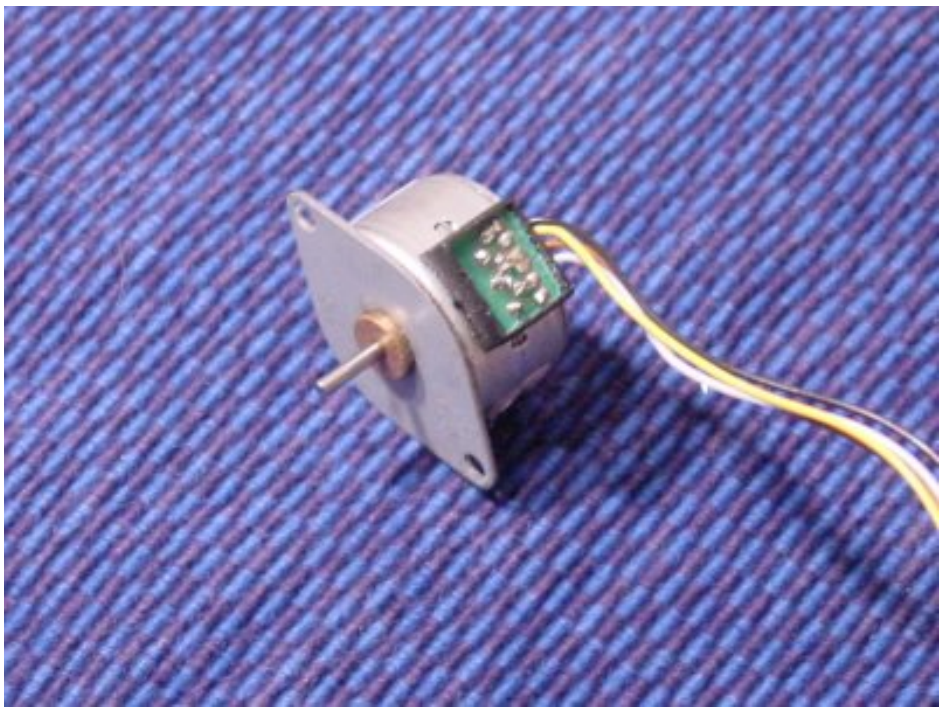
Right now, Darwin's self-replicableparts inventory consists mostly of fasteners and mountings plus partsof the extruder. There has been talk about printing circuit boards andreducing the amount of steel in the system by printing such things astie rods.

Tommelise 2.0 comes at positioning via lead screws andlinear steppers instead of Darwin's belts and NEMA 23 steppers. Exceptfor screws and other simple fasteners there is very little additionalsteel in the structure. Most of Tommelise is made of plastic andparticle board. While this could be replaced with equivalent structuresmade of ABS, that would do little towards reducing the overall cost ofthe system.

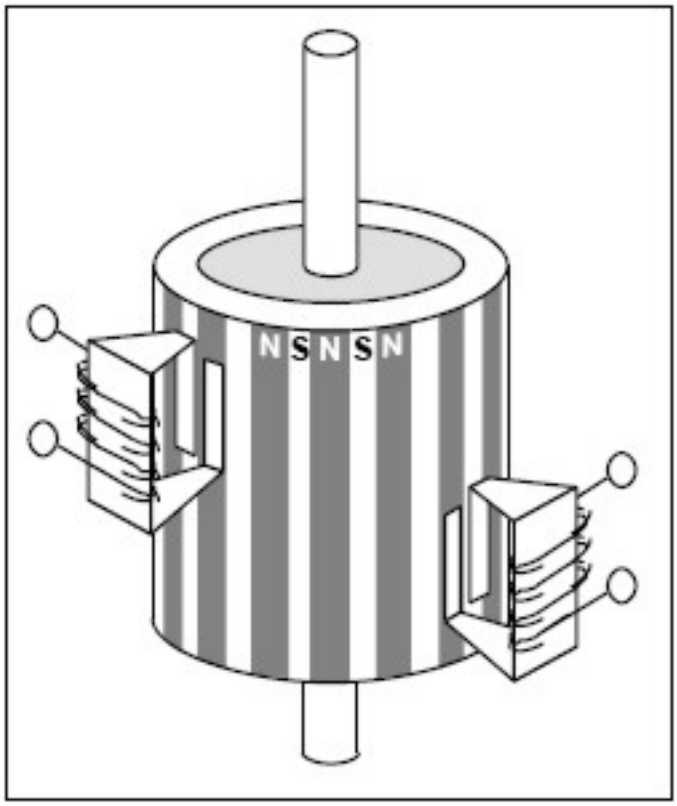
Until a cheap Chinese linear stepper supplier isidentified, there are several candidates, mind, the most expensiveparts of Tommelise are the linear stepper motors and their lead screws.That led me to wonder how much drama would be entailed in printing mostof the parts for a linear stepper motor.

Dissecting a tin can stepper motor

Mostlinear stepper motors are what are known as "tin can" or permanentmagnet steppers. Tin can steppers have a very distinctive appearance asyou can see with this photo.



This schematic of a bipolar tin can stepper gives you a very rough idea of what goes on inside of one.

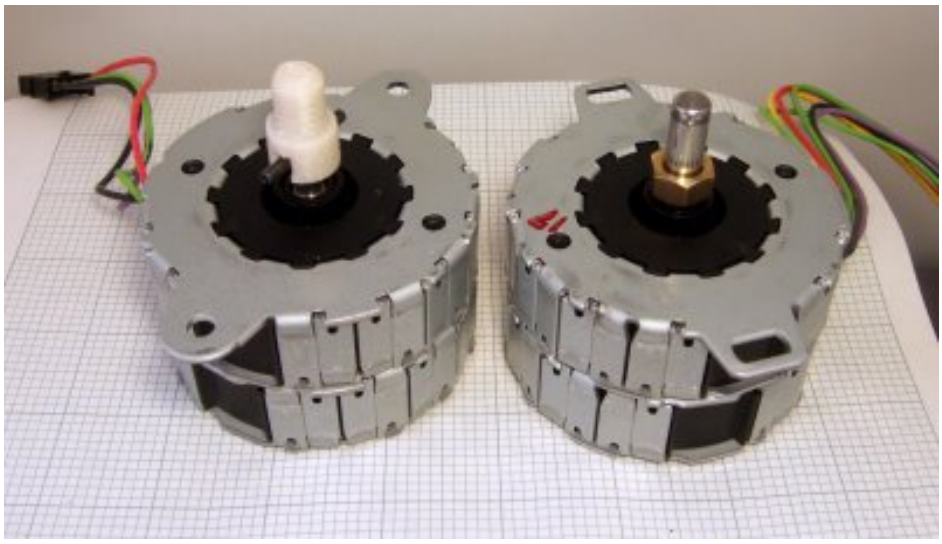


Youcan clearly see the drive shaft and the electromagnets for the twocoils. The nature of the permanent magnet, however, is not as obviousas it could be.
Some months ago, I purchased several tin cansteppers of the sort shown in the first pic in this article. In workingwith one of them, a flaw in the drive shaft caused most of the shaftwhich would have connected to the application to crack and break offnear the tin can, making the stepper motor effectively useless. To geta better idea of what goes on inside I decided to take that one apart.Basically, what I did was to saw one end of the motor off.
Oncethat was done the drive shaft and permanent magnet were visible. Itconsisted of a steel drive shaft surrounded by what looked like analuminum spindle which had a thick coating of what looked like thickvinyl plastic. While the assembly was magnetized, it wasn't clear fromwhat part of the assembly the magnetism was coming from.
I decided then to remove the outer coating to see if the spindle was indeed aluminum.

Hereyou can see pieces of the plastic sheath surrounding the drive shaftand spindle. The spindle was non-magnetic and very likely aluminumwhile the steel drive shaft was not magnetized at all. The plasticsheath contained what looked like a metallic grit which was the sourceof the magnetism. Interestingly, the magnetism of this sheath was notparticularly strong.
Looking at it, though, solved the mysteryof the odd magnet shown in the schematic. The coating was a binderwhich contained magnetizable material. If you think magnetic recordingtape, this was effectively the same thing, the difference being thatinstead of a long tape you had a short loop of magnetic material,instead.
The tin can proved just as interesting.

Itcontained two narrow, large diameter spools containing very fine magnetwire. You can see part of the coil on the lower left of the tin canthat was exposed when I sawed the can open in the

photo.
What isfascinating, however, at the six little steel spikes on the end piece.If you look in the can you see that each spool has one of these piecesof steel on each end. On one spool the six points are offset so thatthere are twelve spikes on the interior of the spool, six coming fromthe north end of the electromagnet and six coming from the south end.The two spools' spikes are again offset with respect to each other sothat you have 24 distinct north or south stripes down the interior ofthe stepper motor. That agrees, broadly, with the schematic.
Theway it is set up also explains to me, at least, why half-stepping oreven full stepping was not much of a success with this kind of stepper.Only wave stepping seemed to work reliably with it.
Tin cansteppers tend to have relatively large steps, viz, 7.5 and 15 degrees.The one I took apart had a step size of 360 degrees/24 north or southstrips internally = 15 degrees. While the ones I've worked with arequite small, tin can steppers can be made quite powerful as this pic,lifted from Nophead's (Chris Palmer) blog demonstrates.



You can see with these big Johnson-SAIA tin can steppers that thenorth/south iron strips are on both the inside and outside of themotor. There are 24 strips on each winding for a total of 48 with thetop and bottom windings offset by 7.5 degrees, giving this stepper a7.5 degree step. Chris used four of these steppers to drive the z-axison the Darwin he is building instead of one NEMA 23 stepper driving abelt to power all four vertical lead screws.
Thoughts on printing a tin can stepper
Looking back on what I learned, it would appear that printing most of atin can stepper is not an impossible undertaking by any means. The twowindings are basically plastic spools. Those should be a trivialprinting job.
In my case, with a linear stepper I use a threaded nut (thrust collar)instead of a drive shaft. It should be relatively easy to print aspindle that slips over that. Mind, I need to make it up as ananti-backlash thrust collar, which means that I actually need two nutskept in compression or tension by a spring to cancel backlash effects. That's a detail of no particular importance to this discussion. Plastic anti-backlash nuts are very common in the industry.
I researched the magnetic media question and discovered that from it'svery beginning at the dawn of the 20th century in Germany it has beenferric oxide (Fe2O3), thus the red colour of magnetic recording tape. The nature of the grit that I found in the plastic sheath in thedissected tin can motor leads me to wonder whether the Chinese havebeen using simple iron filings. Given the low

level of magnetism Iencountered I doubt that they were using rare earth materials.
In either case, both ferric oxide and fine iron filings can be had in in the US in one lb lots for about $6/lb.
It should be no big deal to make such a coating out of the magnetic media mixed with an epoxy like our old standby JB-Weld.
Magnet wire rated up to 400 degrees Fahrenheit of the proper gaugescosts about $24 in one lb lots and $11/lb in 10 lb lots in the US.
Obviously, you will need something like brass bushings and thrustcollars for drive shaft steppers.
There is obviously some sort ofarrangement for counteracting thrust in the linear steppers. I expectthat I will have to dissect one of my used Haydon 26000 steppers to seehow that's done. In that I own twenty of them now, that will be no bigloss.
From looking at the 26000 it appears from the outside that most of that is done with plastic.
Proper lead screws and thrust nuts can be had from any number ofAmerican suppliers.
Alternatively, electrical discharge machining(EDM) techniques could be use to make such lead screws on a small scale.

A design charette for a printed linear stepper motor, part 1

Tuesday, 29th July 2008 by Forrest Higgs

In which your narrator tools up to design a printable linear stepper motor, however clumsily...

The discussion following the blog entry "First steps towards printing a stepper motor" was very useful in focussing my thoughts about what was needed for putting together a printable linear stepper motor.

I had been concentrating on the notion of somehow duplicating the magnetic coating on the rotor that I had found in the Jameco 25BY2414-R tin can stepper motor that I dissected. Nophead suggested that...

Ratherthan have multiple magnets in the rotor, make it with interleaved polepieces, like the stator, with a single cylindrical magnet inside them.I.e. it magnetises alternate poles north south just like the coils do.

While I had thought of this approach,I had been discouraged by the relatively high prices of the cylindricalmagnets that I had seen on offer on the web. I had seen a fewcylindrical magnets with a center hole at ridiculous prices. As well,none of the holes were big enough to slip a thrust collar for a leadscrew of any useful diameter into. I had tried drilling a rare earthmagnet with discouraging results.

Greenarrow pointed me at a German vendor of magnetsat very reasonable prices. While snooping around in that website I ranacross a selection of magnets with holes which could at least makeregular tin can stepper motors with a few having holes approaching thesize that I needed. More importantly, the German vendor gave me theproper phrase for searching for such magnets, viz, ring magnets.

Once I had a proper name for what Iwas looking for I quickly found several American suppliers for bothinexpensive magnets and, much more importantly, ring magnets with largediameter holes. K&J Magnetics had pretty much what I was looking for at a price that I could live with.

Two of those could make for a very nice little linear stepper.

At that point I wanted to run out andbuy a bunch of magnet wire and start making a pair of windings forthese magnets. Fortunately, it was late at night by that time and Idecided instead to see what I could find out about designingelectromagnets on the web instead. While there was a lot onelectromagnet design to be found there was one particularly useful site put together by Rick Hoadley.Rick is nuts about magnets and just the sort of mentor I needed. Hissite is well worth going over in detail. His page of practicalexperiments demonstrating Lentz's Law is especially fascinating.

More to the point, however, Rick has a section on designing air core electromagnets which is very useful indeed. He even put together an Excel page that does air core electromagnet design.

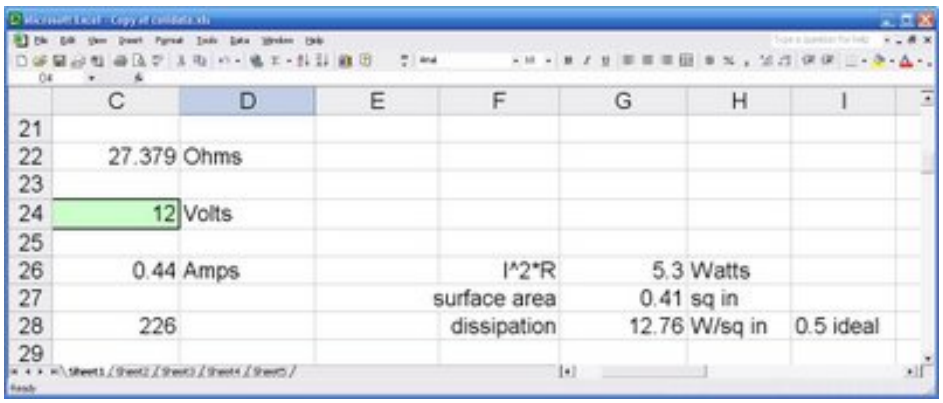
One never knows just how reliablesuch design tools are, so I decided to see if I could back into thedesign parameters for the winding of the little Jameco 25BY2414-R tin can stepper motorthat I'd taken apart. The coils appeared to have been dipped into somesort of plastic which was allowed to partially harden and then equippedwith the stators and slipped into the tin can. I scraped off theplastic exposing the wire coil and took their dimensions.

The tin can on the Jameco stepper wasone inch in diameter and one half inch thick. The core radius of thewinding was a quarter of an inch and the outer radius was roughly 0.41inches. The outer layer of plastic and the steel of the tin can broughtthat outer radius up to 0.5 inches. From the catalog listing and mymeasurements of another identical Jameco stepper, the resistance ofeach winding was 27 ohms. Looking at the catalog page for the 25by2414-R, I could see that it drew 440 milliamps. The stepper was rated at 12 volts.

Just for fun I plugged these numbersinto Rick's Excel sheet and then started plugging in wire gauges to seeif I could get the listed resistance and amperage.

	A	B	C	D	E	F	G	H	I
1	**Electromagnet Design**								
2		ALL DIMENSIONS ARE IN INCHES							
3	**Coil Data**				for max gauss				
4									
5	core radius	r1	0.25	inches	0.25				
6	core diameter	B	0.5		0.5				
7	outer radius	r2	0.41		0.75	r2=3*r1			
8	coil diameter	D	0.82		1.5				
9	coil thickness	N	0.16		0.5				
10	core length	L	0.16		1	L=4*r1			
11									
12	**Wire Data**	**awg**	**dia**	**nom dia**	**turns/in**	**turns/sq in**	**Ohms/1000'**	**Current**	**Fusing**
13	wire size, enamel	35	0.00562	0.00620	141.90	20150.0	329.00	0.03	4.28
14									
15	number of turns/layer	t	22.7		25.8	max			
16	number of layers	y	22.7		25.8	max			
17									
18	total number of turns	T	515.8	turns	666.0	max			
19									
20	total length	x	998.6	inches	83.2	feet	0.01	lb	
21									
22	total resistance	R	27.379	Ohms					
23									
24	applied Vdc	V	12	Volts					
25									
26	rated current	I	0.44	Amps		I^2*R	5.3	Watts	
27						surface area	0.41	sq in	
28	amp*turns		226			dissipation	12.76	W/sq in	0.5 ideal
29									
30	Gauss at center	G1	168	Gauss	(center of air core)				
31									
32		G2	154	Gauss	(center of edge of air core)				
33									
34		G3	123	Gauss	(inches from RH edge of air core)				
35		a	0.08	inches from RH edge					
36									
37	Inductance with air core	L	3.5	mH			Coil		
38									

Sure enough, with a American WireGauge (AWG) of 35 the numbers fell right into place. I had spentseveral hours toying with Rick's Excel page trying to design coils forthe ring magnet before I tried to reverse engineer the Jameco windings.One of the things that bothered me about Rick's calculations was thathe numbers that I kept getting for heat dissipation were considerablyover Rick's 0.5 watts/square inch "ideal". Backing into a real steppermotor would, I felt, give me a feeling for just how realistic that"ideal" really was. Here you can see what sort of heat was actuallygetting generated in that winding and what Rick considered "ideal".



I had noticed when working with thisstepper that it heated up quite impressively if you simply left thewinding energised while it was not actually moving the rotor. This hasbeen a common observation by Reprappers with a variety of steppermotors.

Going back to the dissection of thebroken Jameco stepper, the rotor with the magnetic coating was on theorder of 0.4 inches while the coated coils were running right at 0.5inch, which left an air gap of something like 0.05 inches (1.27 mm)between the rotor and the coils.

The radius of the K&J ring magnetis running 0.375 inches. If we put Nop's mirror image of the statoronto the ring magnet and wrap a stator around the windings we probablyought to give about a 0.2 inches between the magnet and the coil.

Here is a first cut at a winding for the proposed stepper.

	A	B	C	D	E	F	G	H	I	J
1	**Electromagnet Design**									
2		ALL DIMENSIONS ARE IN INCHES								
3	**Coil Data**				for max gauss					
4										
5	core radius	r1	0.6	inches	0.6					
6	core diameter	B	1.2		1.2					
7	outer radius	r2	1.25		1.8	r2=3*r1				
8	coil diameter	D	2.5		3.6					
9	coil thickness	N	0.65		1.2					
10	core length	L	0.355		2.4	L=4*r1				
11										
12	**Wire Data**	**awg**	**dia**	**nom dia**	**turns/in**	**turns/sq in**	**Ohms/1000'**	**Current**	**Fusing**	**lb/1000'**
13	wire size, enamel	26	0.01594	0.01700	52.35	2741.0	40.81	0.26	20.50	0.76
14										
15	number of turns/layer	t	18.6		20.9	max				
16	number of layers	y	34.0		38.2	max				
17										
18	total number of turns	T	632.5	turns	798.4	max				
19										
20	total length	x	3529.4	inches	294.1	feet	0.22	lb		
21										
22	total resistance	R	12.903	Ohms						
23										
24	applied Vdc	V	11	Volts						
25										
26	rated current	I	0.92	Amps		I^2*R	10.1	Watts		
27						surface area	2.79	sq in		
28	amp*turns		580			dissipation	3.62	W/sq in	0.5 ideal	
29										
30	Gauss at center	G1	158	Gauss	(center of air core)					
31										
32		G2	149	Gauss	(center of edge of air core)					
33										
34		G3	126	Gauss	(inches from RH edge of air core)					
35		at	0.1775	inches from RH edge						
36										
37	Inductance with air core	L	12.6	mH			Coil			

I allowed for a voltage drop of about1 volt across the dual half-H SN754410 driver chip and tried to doseveral things. First off, I needed to keep the required amperagebelow the 1.1 amp max that the SN754410 has as its upper limit. Myfirst cut design came in at 0.92 amps.

As well, I tried to keep the amountof magnet wire that I used so that I could get the full use of a one lbspool. In this case, I used about 300 feet of #26 AWG magnet wire,which is a shade less than a quarter of a spool. That means that Icould do two motors with one spool. I also tried to keep the Gaussrating at about the same as I encountered on the Jameco tin canstepper. Jameco's stepper was getting 168 Gauss while my first cutdesign is getting 158.

I'm going to let this coil designrest for a few days and then do it over taking into account anycomments I hopefully get from those out there who have had muchexperience than I have.

The biggest departure from the Jamecotin can stepper that this design represents that I can see at presentis that the rotor is going to be a have a much higher inertia than theJameco stepper does. I've got roughly 0.8 oz (~23 grams) of magnetalone and it is in quite a substantial diameter. It's not going to beas nimble as the Jameco stepper as best as I can see.

All the same, it's possibly good enough for a first cut at designing a printable stepper.

A design charette for a printed linear stepper motor, part 2

Tuesday, 29th July 2008 by Forrest Higgs

In which the narrator possibly figures out a few things... or not...

Early this morning before work Ispent a little more time with the coil design Excel sheet. Afterfiddling with the parameters for a while I discovered that the smallerthe inner radius of an air core electromagnet the more intense thefield becomes, all other things being equal. That led me to noticethat the major difference between the Jameco tin can stepper and whatI am contemplating building was radius of that air core. Morespecifically, the rotor inertia was the big difference between thetwo. I did a rough calculation of what the rotor inertia would beusing two of those ring magnets and came up with somewhere between40-50 gm-cm^2.

I also decided that I should figureout what the parameters were on the Haydon linear steppers rather thanthe Jameco, since I was ultimately planning to make a linear stepper. Of particular importance was the Gauss level that was happening in theHaydon steppers.

I repeated the Jameco exercise on both the Haydon 26000 series stepper and on the 46000 series stepper. I chose the 46000 because with a rotor inertia of 25 gm-cm^2 it is isas close to what I am planning as I can find.

There was no problem backing into thewire gauge and Gauss levels. Both the 26000 and 46000 linear stepperswindings were generating about 115-120 Gauss.

I then designed coils for that ratingand a much higher level at 190 Gauss, just in case. I then ordered alb spool each of #26 and #28 AWG, the two wire gauges indicated in mymodels. The Haydon 46000 was apparently using #30 wire, but thedifferent geometry of what I was designing seemed to argue for aheavier gauge.

From the looks of things the stepperI am designing will pull about 0.9 amps per winding and use about 9-10watts at 11 volts. That level is safely within the level of what mySN754410 can deliver. It looks as if will be twice as powerful as the46000. I hope that I can get enough speed and agility out of it, butif not, then I can do a redesign using one of those ring magnetsinstead of two.

I ordered 10 of the ring magnets. BTW, I found magnet wire at a very cheap rate(~$12/lb), about halfprice from other vendors from a company in New Jersey.

Now, back to other ongoing projects till the supplies get here in about a week.

An affordable alternative to JB-Weld and BBQ paint?

Thursday, 31st July 2008 by Forrest Higgs

Cheaper alternatives to Cerastil for securing nichrome heater wire to extruder barrels may be in the offing...

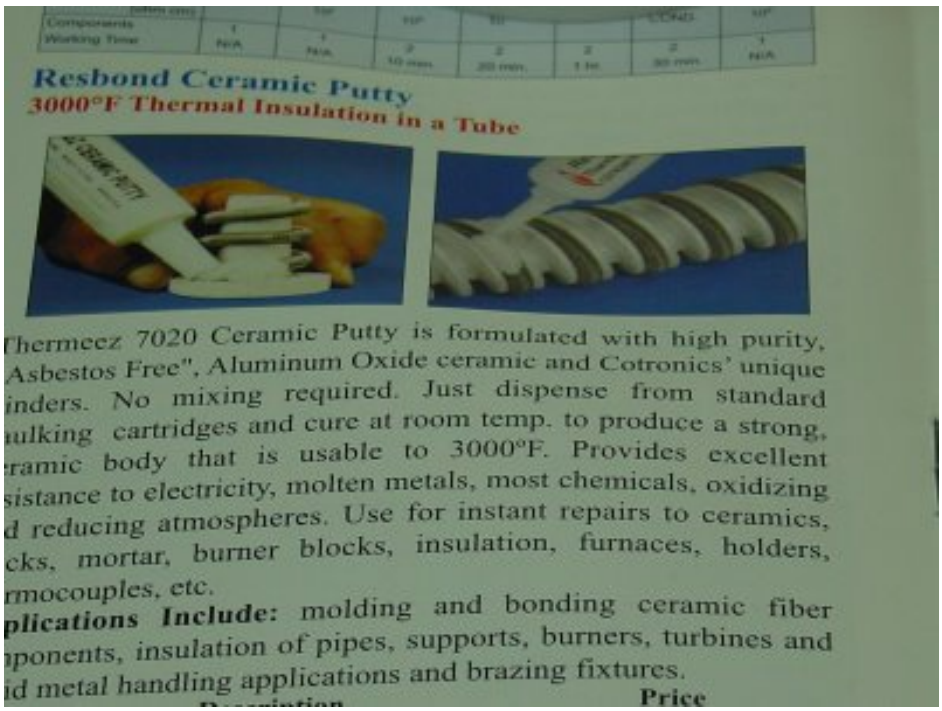
Nearly two years ago when Adrian invented the Mk II extruder, the basicenabling technology for Reprap and pretty much any other low cost 3Dprinter project you'd care to name, he dealt with the problem ofattaching the nichrome heater wire to the extruder barrel by wrappingthe wire in PTFE tape, a very unsatisfactory solution. I foundinsulated nichrome wire which helped matters a bit but that was neverquite a satisfactory solution and before too long we were experimentingwith things like BBQ paint, which I still use, and JB-Weld, a hightemperature epoxy which is in wide use in the Reprap community. Theproblem is that BBQ paint has to be refurbished about every hundredhours and JB-Weld simply perishes in a similar time period.

Recently, Chris Palmer (Nophead) discovered Cerastil,a German high temperature ceramic cement, which gives wonderful resultsbut costs about $200/kg. The problem with Cerastil is that the minimumsize is a kilogram and you only need a few grammes to attach nichromeheater wire to your extruder barrel.A few weeks ago I went hunting for a cheaperalternative to Cerastil. I ran across an American firm, Cotronics,which offered what may be a close match for Cerastil for about$50-60/kg. I took delivery on some this afternoon.

I will be trying to use Resbond 989FS to build up one of my thin-walled extruder tubes within a week or two.

Resbond 989FSseems to be slightly superior to Cerastil H-115if the spec sheets are to be believed and much, much cheaper. On thenegative side it comes premixed and the bucket is listed as having ashelf life of 6 months at room temperature. I immediately put it in aziplock bag and tucked it into the fridge to see if I could do betterthan that.While cheaper, Resbond still comes in what amountsto 1 kg lots, which is enough to finish hundreds of extruder barrels.Cotronics was kind enough to send across their catalog, however, andpaging through it I discovered that they

had another product, Thermeez7020, which isn't quite as strong, but handles heat equally well and,more importantly comes in both standard caulking tubes for about $18and in 4 oz tubes for $26.50/(package of 3). That last looks to me likeit may be about what an individual Reprapper might want to be using.



Resbond Ceramic Putty

3000°F Thermal Insulation in a Tube

hermeez 7020 Ceramic Putty is formulated with high purity, Asbestos Free", Aluminum Oxide ceramic and Cotronics' unique inders. No mixing required. Just dispense from standard ulking cartridges and cure at room temp. to produce a strong, ramic body that is usable to 3000°F. Provides excellent sistance to electricity, molten metals, most chemicals, oxidizing d reducing atmospheres. Use for instant repairs to ceramics, cks, mortar, burner blocks, insulation, furnaces, holders, rmocouples, etc.

plications Include: molding and bonding ceramic fiber nponents, insulation of pipes, supports, burners, turbines and d metal handling applications and brazing fixtures.

Price

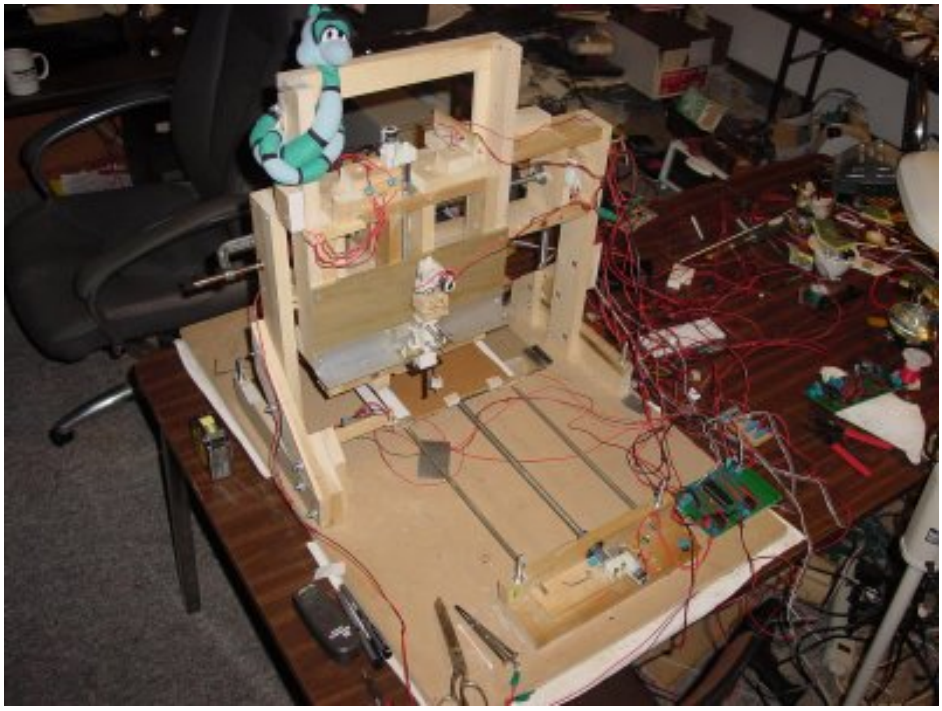
Check out the pic. Those are, unless I am mistaken, heavy dutynichrome heating coils that they are applying that Thermeez to. Thatspeaks well for it's applicability for our purposes.
Now. Who's going to buy and try this stuff? Me again? :-)

Z-axis positioning table takes shape

Thursday, 14th August 2008 by Forrest Higgs

The narrator takes off a few hours and knocks a z-axis positioning table together for Tommelise 2.0...

Tommelise had an effective, but far too big z-axis (vertical).



I had originally designed it to be able to handle two extruders. The slides were all wood construction.

For Tommelise 2.0 I wanted the z-axispositioning table to be much more compact. Obviously, all woodconstruction wasn't going to get that for me, so I originally acquireda u-profile aluminum extrusion that fitted 3/4 inch wood snugly. When Iwent to actually build the z-axis, however, I discovered that theu-profile by itself really wasn't adequate to do the job.

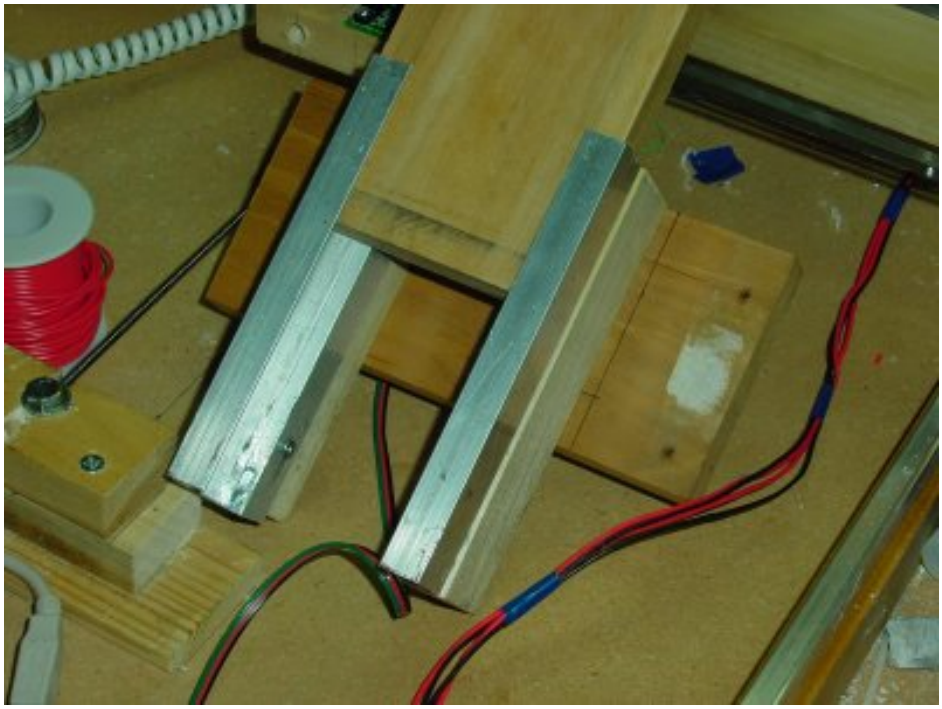
The alternative was to bolt somethingonto the u-profile. Bolting, however, would have to be flush in orderto avoid adding extra friction to the wooden z-axis positioning table.It finally struck me that I might be able to braise an L-profile ontothe u-profile and achieve what I needed. The problem with that approachwas that I didn't seem to be able to braise aluminum extrusions.

I posted a query in the Reprap forumsand soon learned that the very smooth extruded sides of the profile hadto be roughed up before they would bond with molten braising aluminum.Once I understood that (thanks guys!) I used a rotary rasp in my Dremeltool to rough up the surfaces that I intended to braise. I was able todo a fair job with fillet braising and had the profile I needed.

With that obstacle surmounted, it was a straightforward matter to create my more compact z-axis.

The z-axis positioning table is madefrom a 12 x 3 1/2 x 3/4 inch piece of milled poplar. Once I've finishedsanding down the surfaces that contact the aluminum slides, I'llinstall the Haydon 26000 linear stepper and lead screw. This assemblywill be secured onto a simple gantry above the xy positioning tablewith a pair of bolts and wingnuts. The idea is to have a completez-axis plus

extruder for different kinds of extruders using differentkinds of plastic filament. Since I own some 20-odd used Hayson 26000series linear steppers this is no big deal for me.



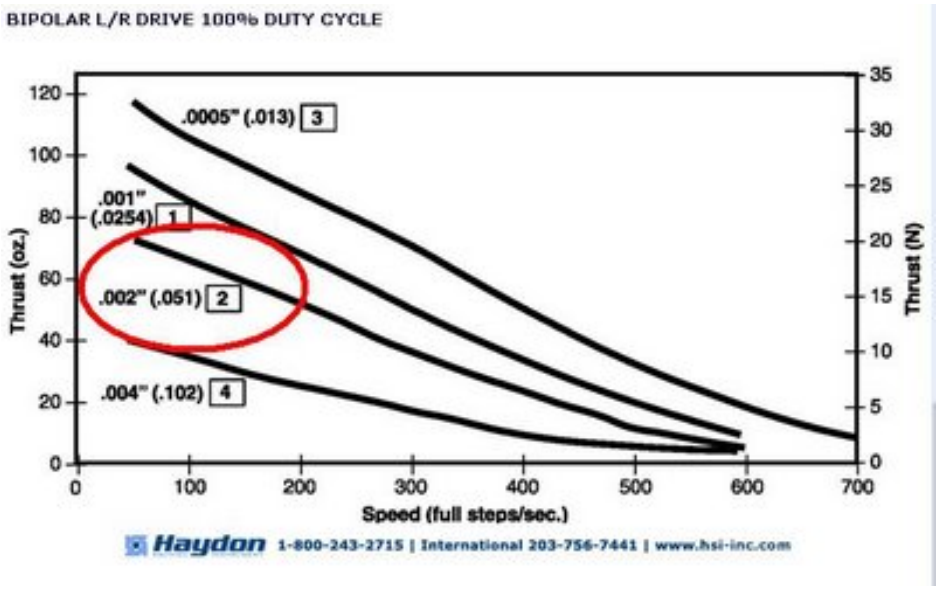
I'm hoping that will let me swap extruders much more easily than I was able to with Tommelise 1.0.

Selecting a linear actuator for the T2 z-axis

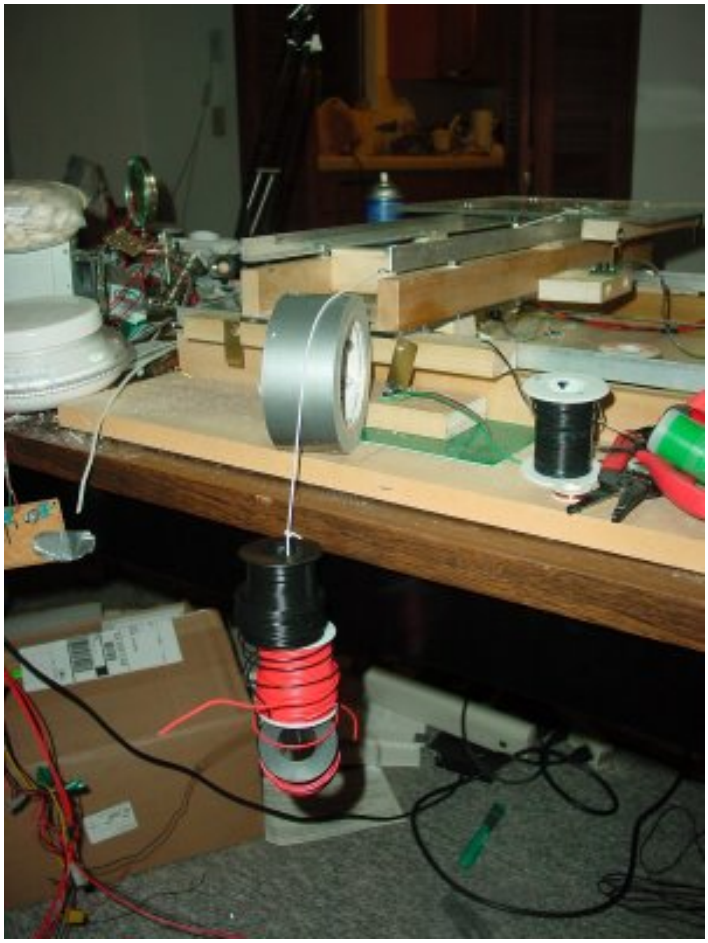
Friday, 15th August 2008 by Forrest Higgs

In which the narrator measures the lifting power of the Haydon 26000 linear stepper...
WhenI designed Tommelise 2.0, a major consideration driving my thinking wasto make the torque requirements as small as possible. My main reasonfor doing this was to keep the energy demand by the printer to aminimum. One of Dr. Bowyer's original dicta for Reprap was that theprinter should be able to be run off of a few 12v automobile batteries.I've kept that as *Ratio decidendi*in my subsequent designs. As an example, the ability to switch off thePC and let the printer run autonomously for long periods of time is adirect result of the 12v battery proviso.

When I finished the z-axis yesterday the friction component ofthe thrust requirement was about 24 oz. Given that the z-axis table towhich the extruder would be fastened weighed another pound and theextruder, perhaps, another 8 oz., we are looking at a thrustrequirement in lifting the z-axis ensemble of components of over 2.5lbs.



Nowin theory the 26000 ought to be able to handle that kind of load withease. In fact, however, when I have measured the thrust that thislinear stepper could deliver I consistently got about 24 oz instead ofthe 60 oz indicated on the performance. This evening I informallymeasured the practical thrust that the 26000 just to be certain Ihadn't missed anything.



I hadn't.
I got to wondering again about the discrepancy between the performancechart and what I measured. Heretofore, I'd been putting it down to thefact that I use a relatively primitive drive

circuit employing anSN754410 quadruple half-H driver chip. Somehow, this evening thatassumption just didn't feel right, however. For some reason thespecification of the drive employed to make the chart at Haydon caughtmy attention this evening in a way it had not before. It specified aBipolar L/R Drive.
Chasing that up on Google left me with this very illuminating passage.

L/Rdrives employ two series power resistors to decrease the L/R-timeconstants of the windings; for example, a 45Ω power resistor in serieswith each of two 15Ω winding resistances divdes the L/R-time constantsby four and allows the rated supply voltage to be increased by a factorof four. Both the crispness of the response and the high speed torqueare increased. While the rotor holds position or moves at low steprates, the series power resistors protect the motor by holding thewinding currents to the rated limit. Since both the L/R-time constantsof the phase windings and the rated supply voltage were increased by afactor of four, the example drive would commonly be referred to as anL/4R drive.

Haydon has buried a huge fudge inthe presentation of the thrust capabilities of their linear steppersfor novices like myself in the off-hand mention of their having used anL/R drive. It is interesting that they do this in that I'd hadto do something similar, viz, attach power resistors in series with thewindings because the windings were rated for 7v while I was supplying12v.
In any case, this evening I took a few minutes off and with carefulsanding and smoothing of the aluminum slider housings I reduced that 24oz of friction losses down to about 4 oz without losing the tightnessof the fit of the z-axis positioning table. Even with that reductionI'm still about half a pound too heavy for the 26000. Looking aroundmy worktables I found some nylon pulley wheels that I'd used on thelong-disassembled Godzilla printer.
Those I intend to use in a counterweight scheme that will effectivelyneutralise the weight of the z-axis table and the extruder leaving the26000 to cope with a mere 4 oz of friction. That, it should be able tohandle without a hiccough.

Testing the Tommelise 2.0 z-axis

Saturday, 16th August 2008 by Forrest Higgs

Haydon 36000 and 26000 steppers are tested in situ on the new z-axis...

Tommelise 2.0 z-axis stepper test from Forrest Higgs on Vimeo.

Ihad the notion that I ought to be able to get around having to rig acounterweight system to take the load off of the z-axis linear stepper.I measured the thrust available on the y-axis Haydon 36000 anddiscovered that it would handle about 2.5 lbs, which was just aboutwhat I needed. I didn't want to commit my other new Haydon 36000, so Itried to use my old used, unipolar 36000 that I'd originally boughtwhen I started getting interested in linear steppers, instead. Ithought, perhaps, that the 36000 would give me a better safety marginthan the 26000.
It worked just fine in the z-axis, but didn't give me any more thrust than the smaller 26000 which I tested next.

Tommelise 2.0 z-axis Haydon 26000 stepper test from Forrest Higgs on Vimeo.

Oh well, it was worth a try.
I'moff to my hardware store to get some extra bits so that I can rig thegantry for the z-axis now that it's built. After that I've got to getthe limits switch working on it.

Gantry finished

Sunday, 17th August 2008 by Forrest Higgs

The gantry supporting the z-axis on Tommelise 2.0 is finished...

I basically bolted two 4" x 24 " x3/4" pieces of milled poplar onto the edges of the table and thenbolted the gantry onto that. Now that it's done it looks simple,however, figuring out where everything belonged and getting thingslined up properly took me most of yesterday evening and this morning.

There was a second agenda in thedesign of Tommelise 2.0. The notion was to make the whole thing fitinto the 62" (W x L x H) limitations for regular luggage on airlines.If you look at the gantry you will notice that both it and the z-axisare mounted with wingnuts. I have the dimensions right with this. As T2stands now its W x L x H equals 60" with the gantry demounted andstacked inside. Put a lock-down top on it and you have 61".

It wasn't the intention to gete afull-blown checked luggage version of T2 working this round, but ratherto see if the pieces could be made to work in that mode. They do. In afully operational travel

printer I'd have enclosed the ends with 4"poplar and shortened the y-axis rails by 1 1/2" to accomodate that. I'dhave also recessed the lock nuts and rounded the corners and put padson them. The x, y and z axes would be locked down inside and paddedwith foam. The electronics and power supply would go in my otherchecked bag.

The most fragile parts of T2 are thelead screws. Those need to be removed from the axes and secured in agrooved block so that they don't get bent.

Since I started working on T2 I'vejust about decided that it would be much better to create atravel-sized printer that met the 45" limit for carry-on so that itwouldn't have to face baggage handlers. While this smaller one wouldn'tbe practical for serious printing, it would be an excellent workingprop for lectures.

One thing that I added while I wasbuilding the gantry was a set of removable feet for T2 that can beadjusted to level the system.

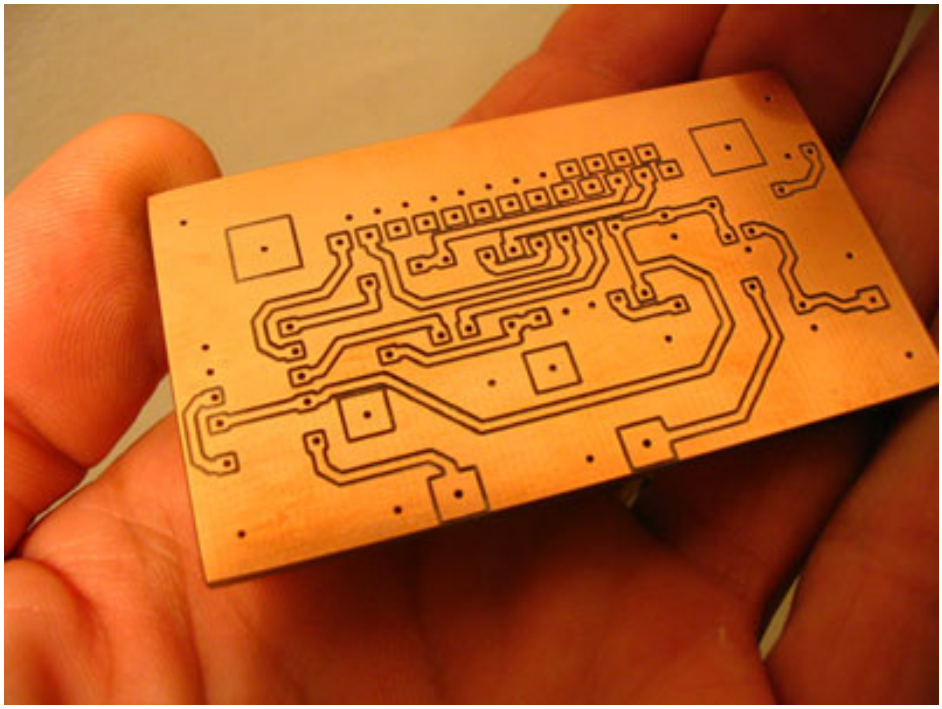
I don't know why I hadn't thought of those before.

There are a number of odds-and-endstasks to do on T2, like installing the opto-endstop for the z-axis andsuch not. The remaining big task, however, is to cobble together a newextruder. For now, I will probably just use the old 18F4610 board fromTommelise 1.0 to do the job. It has everything I need already on boardand the only thing I have to do would be to establish an I2C link fromthe 18F4450 controller board to the 18F4610 board. That shouldn't beany big deal.

Aside from printing stuff, one thing I will be trying very soon is to mill my own PCB's. I own a Dremel MultiPro 395.

Dremel's flexible drive shaft willlet me mount the 395 on the gantry and let it drive the tool head onthe z-axis work table without having the entire load thereon.

I'm looking forward to moving my nasty looking stripboard prototypes to something more like this.



Although Nophead probably veryrightly says that it is not possible, I also want to take a crack

atmilling a set of extruder parts out of HDPE with this sort of rig.

Milling plastic with a Reprap machine?

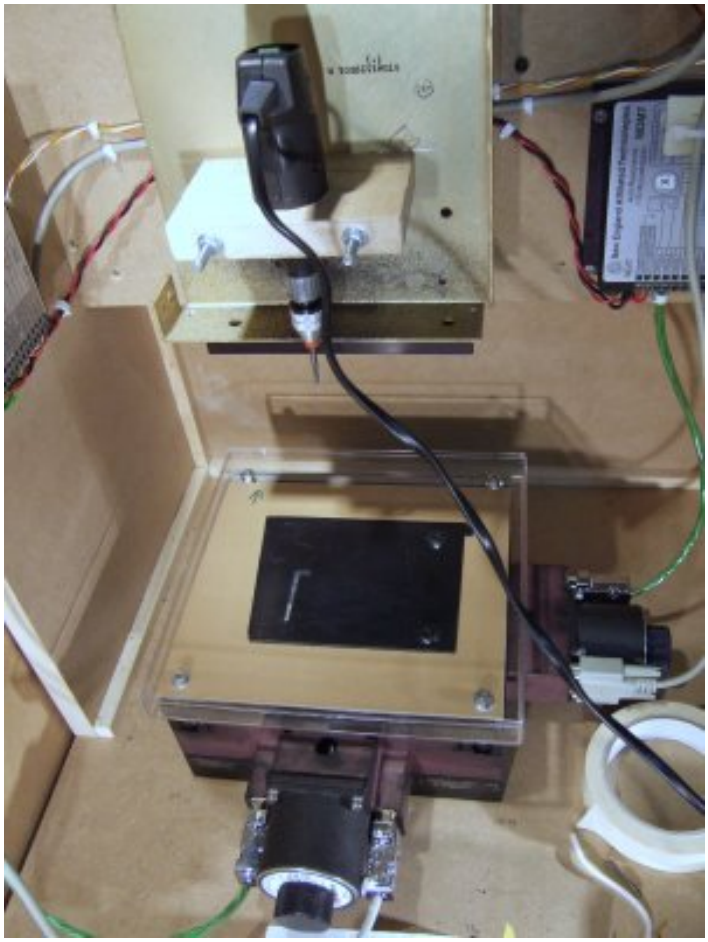
Monday, 18th August 2008 by Forrest Higgs

It might not be impractical after all...

Back in April and May of last year, Nopheadtook a look at milling plastic on his repstrap machine. Unlike most ofus, Nop simply bought a proper xy platform and went from there. Thatdecision, combined with his very thorough and thoughtful way ofapproaching problems has made him the treasure of a resource to theReprap project that he has become in the time since.

Parts being almost impossible to comeby in those days, Nop resolved to mill his own extruder. I wasn'tfollowing his blog regularly at that time and missed much of what hewent through. At first he bought a spiral saw which could rev up to 30K rpm. For some reason, though, it proved inadequate, so he acquired a minidrill that could hit 20K rpm from his aunt. He used this one thereafter.

That decision made, he proceeded to try to mill some styrene...



... with indifferent results. He noted at the time that...

Imight be wrong but I think the snatching is due to the set up not beingrigid enough. I had already identified that as a weak point and the newmotor mount was aimed at improving it. The first drill was supported atboth ends but this one is too short so it needs a much stiffer mount,which is what I was trying to make! Perhaps the end mill bit is notsuitable for styrene, or perhaps styrene is not very machinable, orperhaps the RPM is too high, or too low.

...and decided.

Onething I can do to improve stiffness is to replace the 2mm aluminiumplate with the 6mm slab I already bought for the job. I was puttingthat off until I got the new mount so as not to have to drill two setsof holes in it.

And subsequently managed quite nicelywith the string of results we've all benefited from today. It also leftus, however, with the impression that has since turned into a receivedwisdom that it isn't really possible to do milling work on things likeplastic with a light framed Reprap machine.

Changing firmware compilers

Friday, 22nd August 2008 by Forrest Higgs

The shift to a more powerful firmware compiler for Tommelise 2.0 comes a bit sooner than the narrator expected...

I've happily used the Oshonsoft BASIC IDEfor Pic microcontrollers for just over two years now. Just over a yearago, though, young Vladimir Soso, who runs the Oshonsoft companysingle-handedly had a massive setback with his health. In present-daySerbia, that is no laughing matter. Since then he's been trying to getout of debt and get well. That's not easy for a small businessman evenhere in the States, much less Serbia.

As a result of that Vladimir hasn'tbeen much help with improvements or support for his Oshonsoft compilersand IDE's. Support has never really been an issue with his apps,because they are rock solid and if you've used them for a little whileyou rarely have a support question that the Yahoo forum dedicated to his compilers can't answer in very short order.

I'm feeling really bad this morning. This morning at about 01:00 I realised that I had spent four man-daysover the past two weeks trying to coax Oshonsoft Basic into letting meuse I2C in slave mode and was no nearer to a piece of working code thanI was when I began.

There's something about 01:00 in themorning with a hot cup of tea that gives me a different perspective onthings than I have at other times. What I realised was that if I hadspent those four man-days doing billable consulting work, I could haveeasily bought the most expensive Pic programming environment on themarket and had a lot of money left over.

Before Vladimir got ill, thecapabilities of his compilers easily outpaced my needs. Since then,however, I've slowly outgrown their capabilities. My little epiphanythis morning has been a long time coming, but I've finally been forcedto conclude that for me I've outgrown Vlad's product. I've beenspending a lot more time trying to get his compiler to do what I needit to than I really have any business doing.

This is not to put his compilersdown. They are rock solid and have the shortest learning curve in theindustry. I recommend them to just about anybody as an inexpensive,painless way to get into writing firmware. It took me two years oftrying to do some pretty incredible things and succeeding with Vlad'scompilers before the new challenges I am trying to confront began tooverwhelm their capabilities.

Anyhow, at about 02:00 I finallypurchased a copy of the Mikroelectronika Basic compiler forevaluation. It doesn't appear to be as intuitive a developmentenvironment as Vlad's Oshonsoft. That said, if it's library functionsare any indication it is a lot more powerful. I need that power now.

Changing firmware compilers (part 2)

Sunday, 24th August 2008 by Forrest Higgs

Wherein your narrator decides, perhaps foolishly, to rewrite Tommelise 2.0's firmware in Mikroelektronika Basic...

This morning I decided to take acrack at getting some distance down Mikroelektronika Basic's learningcurve. It seemed to me that a good first step would be to see if Icould establish USB comms between the extruder controller board and myPC. Mercifully, Mikroelektronika included an example program that woulddo just that for the 18F4550 microcontroller that I use in bothcontroller boards in Tommelise 2.0.

I was able to load the example, asimply echo routine that bounced back at the PC anything that it sentto the microcontroller. Mikroelektronika even included a compiledversion of that one in hex code. Unfortunately, it was written for aMikroelektronika development board that for some reason used a 8 MHzclock crystal. Actually, as long as you have some sort of clock crystalfor the 18F4550 the frequency isn't too important. The chip has aclever set of postscaler values that you can set in the configurationregisters that allow you to run the chip at the highest possible speed,viz, 48 MHz, regardless of kind of crystal that you use.

What that meant, though, is that Ihad to make alterations to the default configuration settings for theprogram. Mikroelektronica's help for you in that regard was a coylittle...

- Be VERY careful about the configuration flags for the 18F4550 - there's so much place for mistake!

Which shows a serious Serbian talentfor understatement that I hadn't suspected heretofore. If I hadn't hadthe configuration settings from my Oshonsoft code that Vlad hadoriginally set, I'd STILL have been trying to get their code to work.

After a half-dozen tries at changingthe configuration settings I finally got them all right at the sametime and my PC gave its little weird doorbell acknowledgement that ithad spotted my board when I plugged in the USB cord. I then used Jan Axelson's (author of USB Complete) generic USB code for the PC-side of the USB communications. You can get this code at...
http://www.lvr.com/hidpage.htm#MyExampleCode
While I like to use Visual Basic...
http://www.lvr.com/files/GenericHid_vb.zip
She has generic PC-side comms code in Visual C# and half a dozen other languages.

Jan's USB comms app found the boardimmediately and was able to echo one byte transmissions very handily.Another forty-five minutes of poking around in the descriptor file ofthe sample program and I discovered how to reset the firmware sidebuffer specifications to handle multibyte sequential transmissions.That enabled me to duplicate the behaviour of Vlad's Oshonsoft USBmodule which transmits 8 byte packets of data at a time.

The lack of an ability to specify thesize of the read/write buffer for USB comms was what made my

coding ofT2's USB routines to program the EEPROM print buffer such a trial. Ihad to shunt print data from the PC in 7 byte packets and load the 128byte read/write buffer in the EEPROM chips before I could burn the dataonto the EEPROMs. While I eventually got the code bullet proof, it isstill messy and inelegant, to say the least.

With the new Mikroelektronika Basiccompiler I can simply specify the USB read/write buffers to be the samesize (128 bytes) as the EEPROM read/write buffers. That would simplifythose routines tremendously.

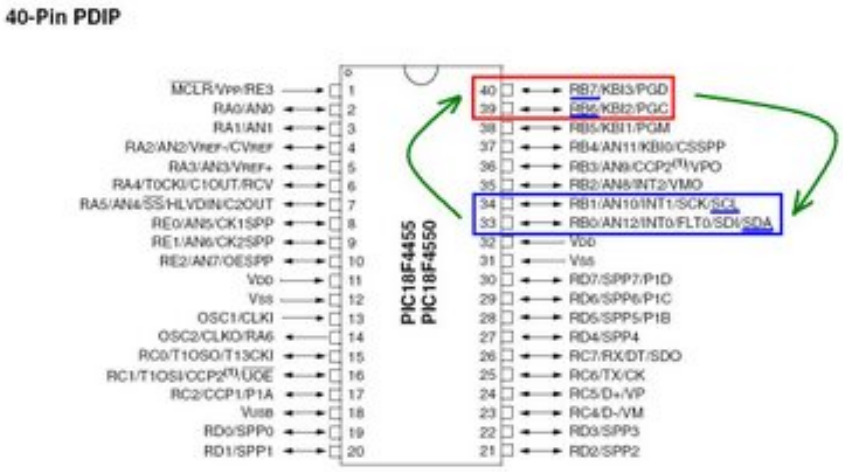
That done, I took a hard look atMikroelektronika's I2C facilities. They have both hardware andsoftware (bit banging) I2C routines. Implimenting the I2C connectionbetween the 18F4550 and the EEPROM bank in Oshonsoft Basic had been soeasy when I did it that I hadn't put much thought into the nature ofI2C comms. I hadn't been aware of the distinction between hardware andsoftware approaches in comms except for RS232 (serial comms). In thatVlad's Basic let you put the pins you assigned to the SDA and SDLfunctions wherever you wanted, it appeared that he was taking thesoftware route. That was fine with me in that it worked fast and gaveme considerable flexibility in pin selection.

The hardware/software distinction inMikroelektronika's I2C library, however, got me to wondering aboutexactly what Vlad was actually doing. I pulled out the assembleroutput from my firmware and had a look at what sorts of things werehappening at the assembler level with Vlad's I2C code. From the looksof it he is using the registers for I2C comms as you would in ahardware app, but reassigning the SDA and SDL pins as you would with asoftware app. I'm not quite sure what that is all about. It works,though.

While at first I thought I'd just usethe software I2C routines to mimic what was happening in OshonsoftBasic, I soon ran across a caveat in the language manual that put thatidea to rest...

Note: This library implements time-based activities, so interrupts need to be disabled when using Soft I2C

So much for that idea. My firstfeeling was that it was going to take some serious rewiring ofTommelise's positioning robot controller board to get I2C workingproperly with Mikroelektronika Basic. When I actually sat down anddrew it out, however, I discovered that I needed to only swap two pairsof pins connections.



When I thought back on what I'd triedto do with the master/slave I2C comms connection it occurred to me thatI'd been trying to use interrupts on the extruder controller board whenI was trying to establish master/slave I2C comms between the positionerand extruder controller boards. That may well have been why I wasunsuccessful.

In the morning, now that I have USBworking, I am going to swap those two pairs of wires on the positionerrobot controller board and have a go at establishing I2C comms with theEEPROM buffer with Mikroelektronika Basic. If that goes well, I'mgoing to see if I can port that slave I2C code that is written inMikroelektronika C to Mikroelektronika Basic and get the two boards totalk to each other. Should that work I will be porting the rest of thepositioner robot code over to Mikroelektronika Basic and be effectivelyat the end of using Oshonsoft Basic on this project.

Limits to growth

Monday, 25th August 2008 by Forrest Higgs

Inwhich the narrator discovers the dangers of trying to do too manythings at once with 8-bit Pic microcontrollers...

In the forums recently there has been a lot of loose talk about makingour Reprap machines more and more independent of PC's. I've certainlybeen in there pushing the idea and have gone some distance towardsreducing the dependence of Tommelise 2.0 via a large print buffer madeof EEPROMs which enables it to print for many hours withoutintervention from a controlling PC. There was also talk recently about how nice it would be to usesomething like the much roomier SD card storage units instead ofEEPROMS that I use on Tommelise as well. This weekend I found myselfshifting over from the Oshonsoft to the Mikroelektronika Basiccompiler. The Mikroelektronika Basic compiler provides support for SDcards. While I was at Circuit City this afternoon it struck me that itwould be fun to go ahead and cobble together an SD slot on my prototypeboard and see, thereby, what the problems would be if and when I decideto shift away from EEPROMs.

I went ahead and bought a 1 GByte SD card and confirmed that my XPS 720H20 PC supports this kind of card without my having to buy a USBconnected SD card reader. This evening I decided to do some duediligence before I ordered an SD connector breakout card to do theprototyping with. I'm rather glad that I did.

Here is what I discovered. I'm going to start from the microcontrollerthat I am using with Tommelise 2.0, the Pic 18F4550. Right now I amusing a hardware I2C link between the microcontroller and the EEPROMprint buffer. Because I2C is a bus-oriented comms standard I can handany number of devices off of the bus as long as they have differentidentifier codes. That lets me seriously talk about having themicrocontroller that drives Tommelise's positioning robot talking toboth the EEPROM print buffer and another 18F4550 microcontroller boardwhich I am using to control the extruder using the same I2C bus. Thatis very handy.

While I could replace the EEPROM buffer with the SD card, in theory atleast, I still need that I2C link to the other 18F4550 that controlsthe extruder. That's where the trouble starts. The SD card is supportedby the Serial Peripheral Interface (SPI) while the EEPROMs aresupported by the I2C bus. The problem with that is that for the 18Fline of Pic microcontrollers both of those are created with the samemodule. You can have one or the other but not both at the same time.

In theory, I could simply shift over completely to SPI when I installthe SD card interface and drop the EEPROM bank. The problem with thatis that while on paper SPI can handle multiple slaves, letting me talkto the extruder bord as well as the SD card, it isn't obvious how youcan make that happen with the specs of the 18F family of chips.

Now Microchip makes 16 bit Pics with separate SPI and I2C modules. None of them, however, come in a 40 PDIP configuration. That makes thecost of developing a new board skyrocket. Mind, there are a number of ways for achieving Pic-to-Pic comms rangingfrom serial to one-wire connections. From personal experience, it doesnot appear to be possible to do USB and serial connections at the sametime. Virtually all of the bit-banging methods get into trouble if youare running an interrupt scheme. I am and I can't really run Tommelisewithout interrupts.

The point is that I am going to give the SD card idea a rest for awhile and do all the other things that I CAN do with Tommelise. :-D

Fixing a software problem with hardware

Thursday, 28th August 2008 by Forrest Higgs

In which your narrator solves the problem of making a microcontrolleroperate in I2C Slave mode via a hardware rather than a software fix...

Over my increasingly long andmisspent life I've found, more often than not, that the solution toproblems are much more often found not in creativity and genius as muchas figuring out how to frame the question of the problem correctly. This was one of those situations.

Here is how things developed. Iwanted to dedicate one native-USB 18F4550 microcontroller to runningTommelise 2.0's cartesian positioning robot and another to running herextruder. I won't repeat the whys and wherefores of why I wanted to dothings this way, their having been discussed at length previously. Thetwo controllers obviously needed to be able to talk to each other.

My first idea was to exploit the UARTmodule that the USB comms between the cartesian positioning robot andthe PC had made superfluous. For some reason the 18F4550 didn't wantto run both USB and UART comms at the same time. I can't say that Iwas much surprised at that after looking at the chip's schematic.

My next idea was to use any of theseveral bit banging comms protocols like software serial comms and thelike. It turned out that those didn't like running in an interruptdriven environment. That didn't surprise me much, either.

I'd had a lot of success with the I2Ccomms format in hooking up the EEPROM chips to the cartesianpositioning robot to its microcontroller and thought that that might bea good protocol to use in making the two 18F4550's exchangeinformation. While the chip data sheet made this look like a distinctoption, none of the compiler suppliers had actually made it possible tomake a Pic operate in I2C Slave mode in their I2C library functions. My old, reliable Oshonsoft Basic compiler certainly didn't.

I eventually found some good-lookingcode that supposedly did the job in the MikroElektronika C compiler andresolved to transfer that over to MikroElektronika Basic. On thestrength of that I bought their Basic compiler. The porting was easy,but it turned out that MikroElektronika was having problems with theirUSB library functions in any both the most trivial code examples. Intheory their new 7.0.0.3 beta release fixes their USB problems. At thepresent time I've installed the new beta release and am awaiting a newactivation key from Belgrade.

Annoyed that it has taken me nearlytwo weeks to establish rather simple comms between two Pic chips, I satdown and thought about the whole I2C phenomena. Turns out that Philipsin the Netherlands invented the I2C standard. Years ago I'd done somework with what was then known as ***N.V. Philips Gloeilampenfabrieken*** and as a result had, perhaps, some small insignt into Philips' corporate thought process.

Once I thought about I2C inconjunction with Philips it seemed impossible to me that they wouldhave evolved a standard and then not designed a chip to helpnon-Philips circuit designers in

using the standard in the productionof new I2C capable chips operating in Slave Mode. If such a chipexisted, its existence would probably explain why none of the compilervendors thought it necessary to implement a software Slave Mode intheir I2C libraries.

Framing the question in that manner to Google quickly found the chip for me.

http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf

Both Philips and Texas Instrumentsmake the chip at a cost of US$1.58 and US$1.75 in one-off quantities,respectively. I put in an order to Mouser for ten.

Problem solved, with hardware instead of software.

The only problem, if it can be called that, with the chip is that it requires that a full 8 bit port onthe slaved microcontroller be committed. With my extruder board, whichwill be operating in slave mode, that isn't much of a problem.

Tooling up to mill plastic with Tommelise

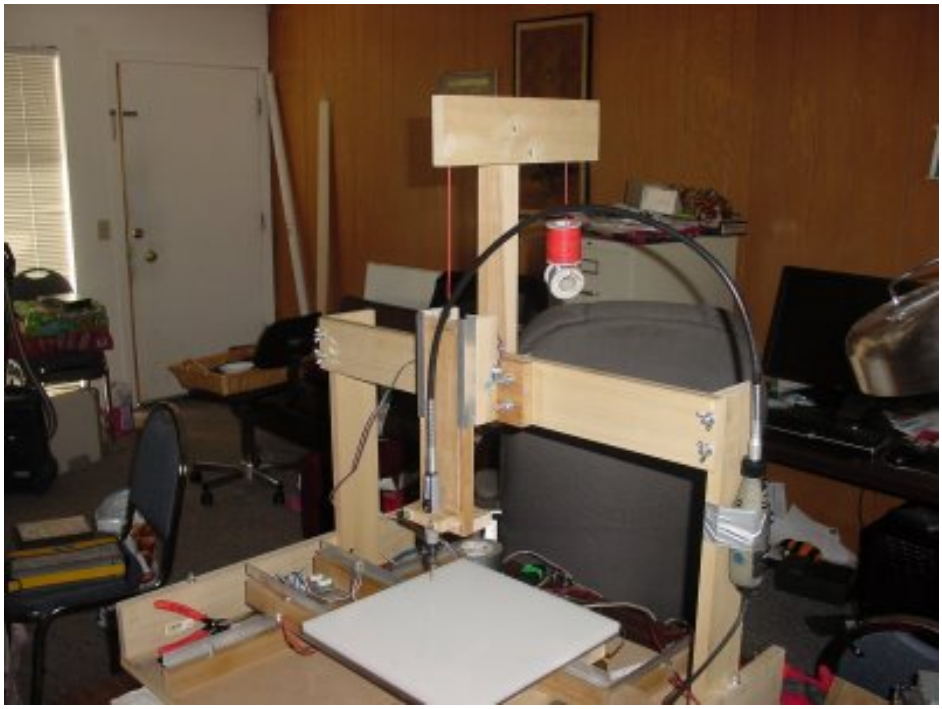
Friday, 29th August 2008 by Forrest Higgs

The narrator puts the finishing touches on Tommelise 2.0 to do plastics milling...

I'm really excited to see if I can mill parts requiring high strength levels out of solid stock.

I just got in two square feet ofhalf-inch HDPE (left side of the picture) and polypropylene (right sideof picture). Sheet stock costs me $2.80 and $2.10/lb for HDPE andpolypropylene, respectively. Compared to what filament costs that's avery attractive price.

This morning I rigged Tommelise with the Dremel tool plus flexible drive cable with the notion of milling both PCBs and important parts of Reprap machines like extruder parts sets out of solid stock.



The counterweight system for thez-axis positioning table reduces the load on the linear stepper drivingthat axis from over 2 lbs to the friction load of 4-8 oz.

The small tool head at the end of theDremel flexible drive cable weighs a mere 4 oz extra, which will becompensated with counterweights. I currently have a 1/8-inch cutterhead mounted.

I will be buying smaller diametercutter heads capable of milling printed circuit boards after I get someexperience milling plastic.

For PCB's buying two-sided blankPCB's costs the same $0.12/square inch that the American Reprapfoundation quotes. The difference is that I will be able to prototypeboards and correct errors without having to come up with the front-endcosts that professional PCB firms demand for one-offs. I'm hoping thatthat will accerate and democratise the development of controlelectronics, too.

I think it is way past time that wedemocratised the whole parts supply situation. This might just be theway forward in that regard.

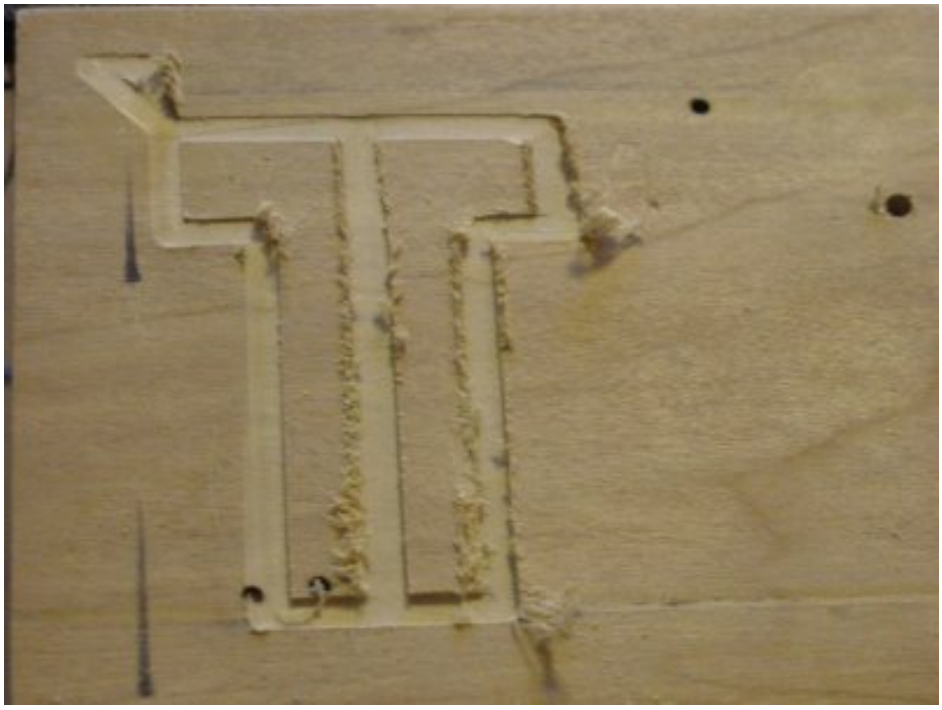
Milling Stuff

Monday, 1st September 2008 by Forrest Higgs

In which Tommelise 2.0 goes operational in a mode not anticipated by the narrator...

Since the slave I2C chips haven'tarrived, I decided to see if I could get Tommelise 2.0 runing as a 2.5DCNC milling machine since in that mode an extruder controller boardisn't required. I began by milling blanks of poplar.

In this mode, I was able to debug the T2 positioning robot without using my 1/2-inch plastic sheet...



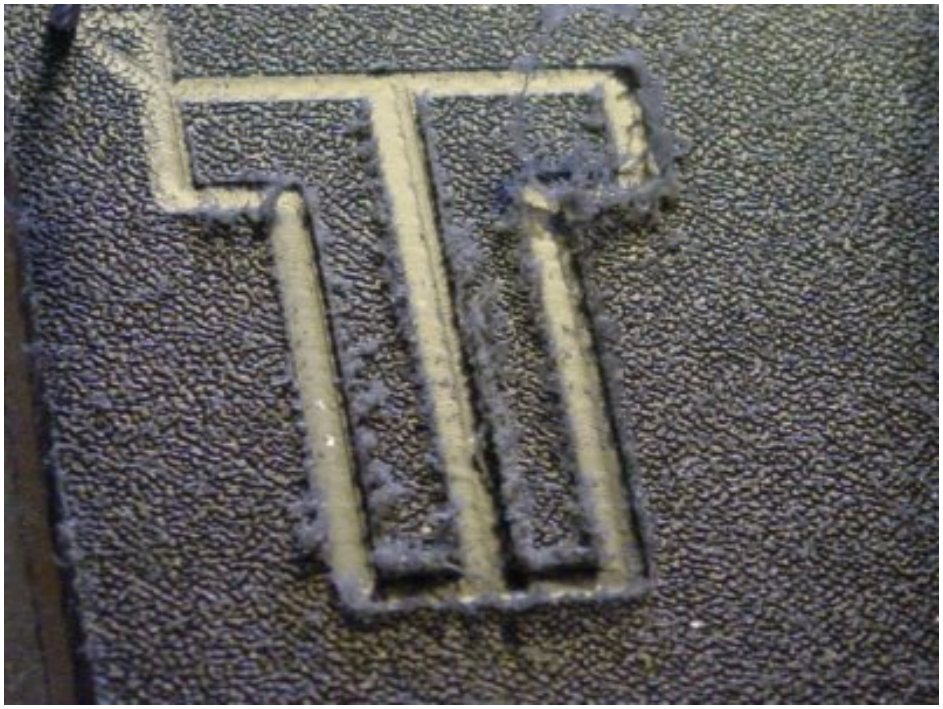
I was able to run the positioning robot at 12.6 mm/sec with 0.05-0.1 mm milling depths.

Once of the things I discovered wasthat I hadn't locked down the y-axis opto-endstop. The vibrationsgenerated by the Dremel that I'd duct taped to T2's gantry caused theloose opto-endstop to move a bit which meant that the y reset pointmoved with time. Another was that the friction fit that I'd made forthe flexible Dremel toolhead wasn't stiff enough.

After securing the endstop and thetoolhead and tightening the z-axis motion in the xz plane. I began touse the scrap of black HDPE cutting board that I had left over fromearlier work. I used this

piece to determine the operating parametersfor this plastic.

After a while I was able to settle on a toolhead speed of 12.6 mm/sec and a 0.25 mm/sec.



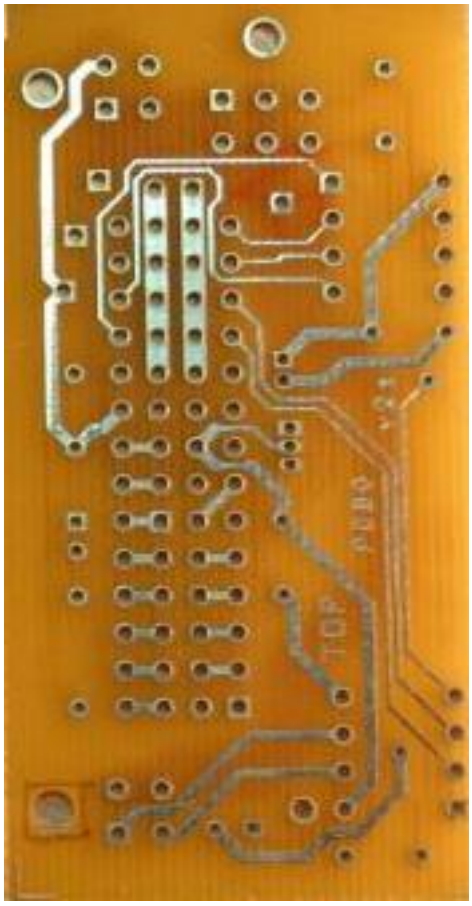
The milling was quite clean.

That setting seemed to run fine. Just to be sure I pushed the cut depth down to 7 millimeters.

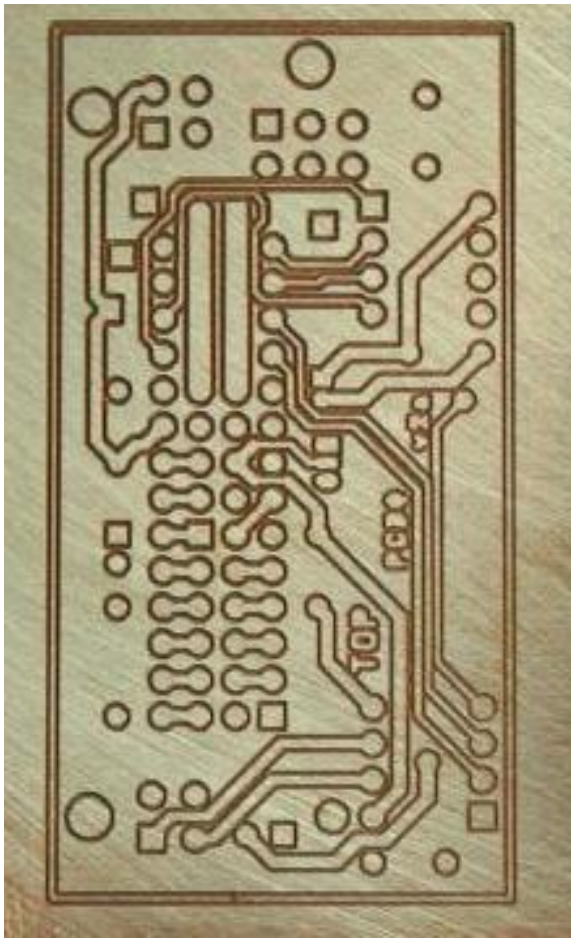
As you can see milling HDPE generatesa wealth of swarf. For the past few days I've been researching theopensource PCB design program KiCAD. KiCAD uses Gerber formatted filesfor generating toolhead paths for etching PCB boards rather than GCode.I read up on the Gerber format and discovered that it was trivial toconvert it into XML instructions usable by T2.

While I was looking all that up I discovered two interesting people at MIT, Marsette A. Vona and Daniela Rus. Vona and Rus came to my attention when they were using Gerber files to generate a particularly interesting kind of milled PCB.

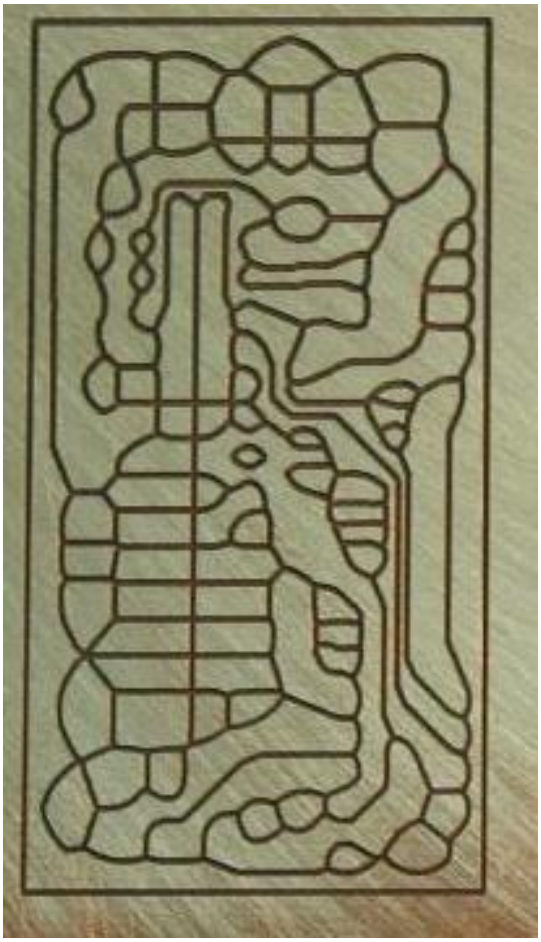
A traditional photochemical processed PCB looks like this.



An equivalent milled PCB looks like this...



Vona and Rus applied the notion of Voronoi regions to generate a much easier-to-mill equivalent PCB.



Oddly enough I used Voronoi regionsin applications to designing architectural spaces years ago. Sadly,Vona and Rus generated a 27 page Java algorithm to do the conversioncomplete with bugs. The grid approach that I use with Slice and Dicehave 95% of the routines that I need to convert KiCAD Gerber and drillfiles into Voronoi region PCBs. I'll blog more about this as I get into the software development.

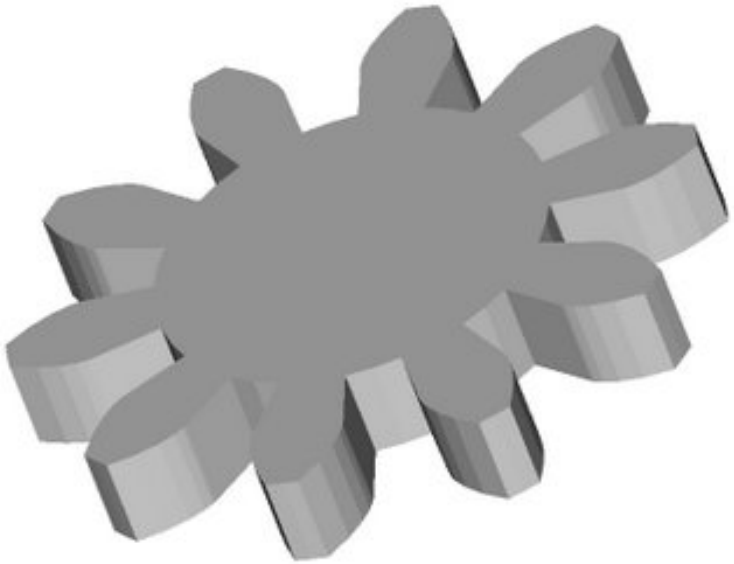
Shifting gears

Sunday, 14th September 2008 by Forrest Higgs

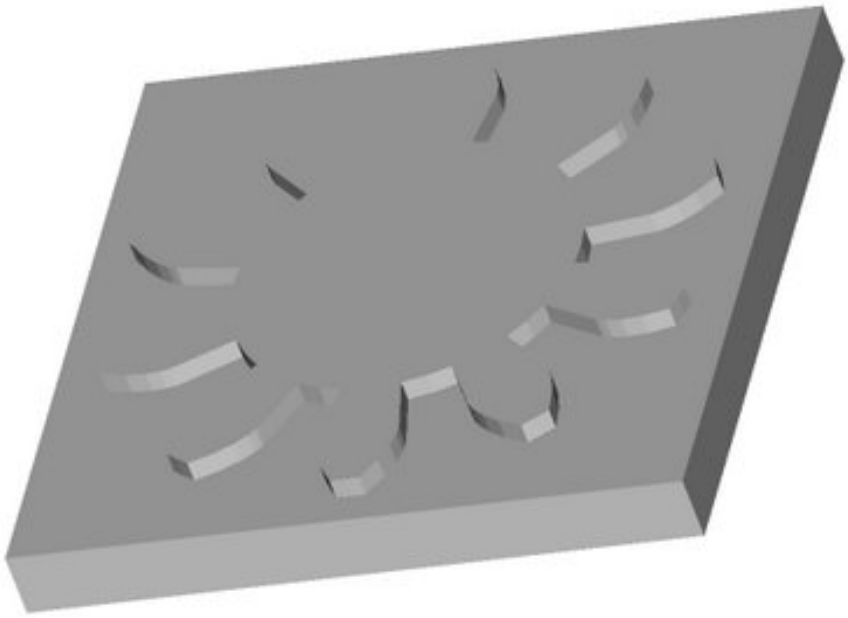
Wherein the narrator's long-held dream of building cheap gear trains becomes reality...

Once I had gained confidence that I could accurately mill engineering plastics with Tommelise 2.0, I had to do a considerable amount of work on both the Slice and Dice software that I'd written for the PC-side of Tommelise and the firmware side as well. I've been doing that work in my spare time during the past two weeks which is why you haven't seen me posting during that time.
The big problem I faced was the fact that when I wrote the slice and dice routine for Tommelise 1.0 over a year ago, I was in a hurry. It worked fairly well, but was too fragile to be easily shifted over to handling milling, which requires that you have to remove material from a block of plastic from which you are trying to make something rather than building it up by adding more plastic to it.

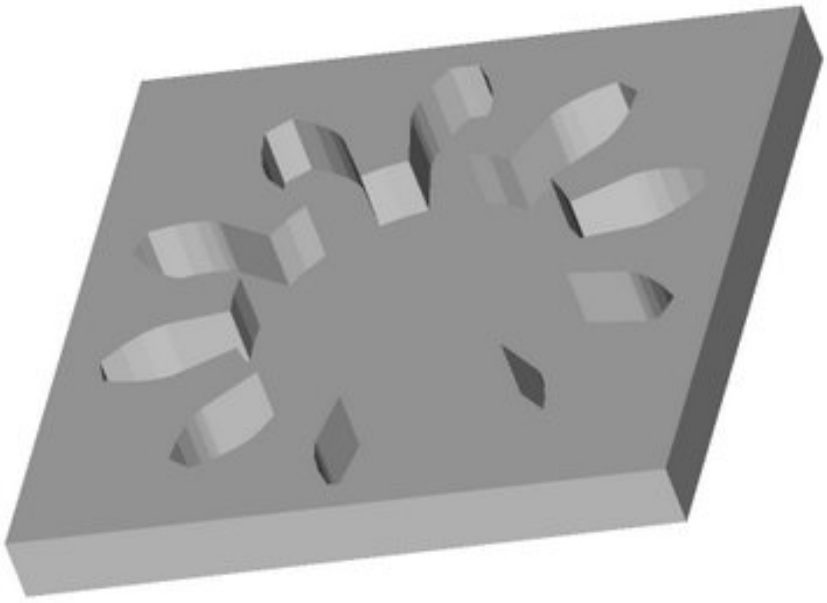
Skeinforge's creator, Enrique, pointed the way for me some time ago, viz,
Also in theory for simple shapes, you could subtract the desired shape from a surrounding box, then use slice and dice software to make the infill gcode pattern for the subtracted shape, and drive the milling head with that gcode.
As I discovered, you were neither restricted to "simple" shapes nor was Art of Illusion's Boolean operations mode not up to the job, as Nophead has asserted from time to time. I first created a gear...



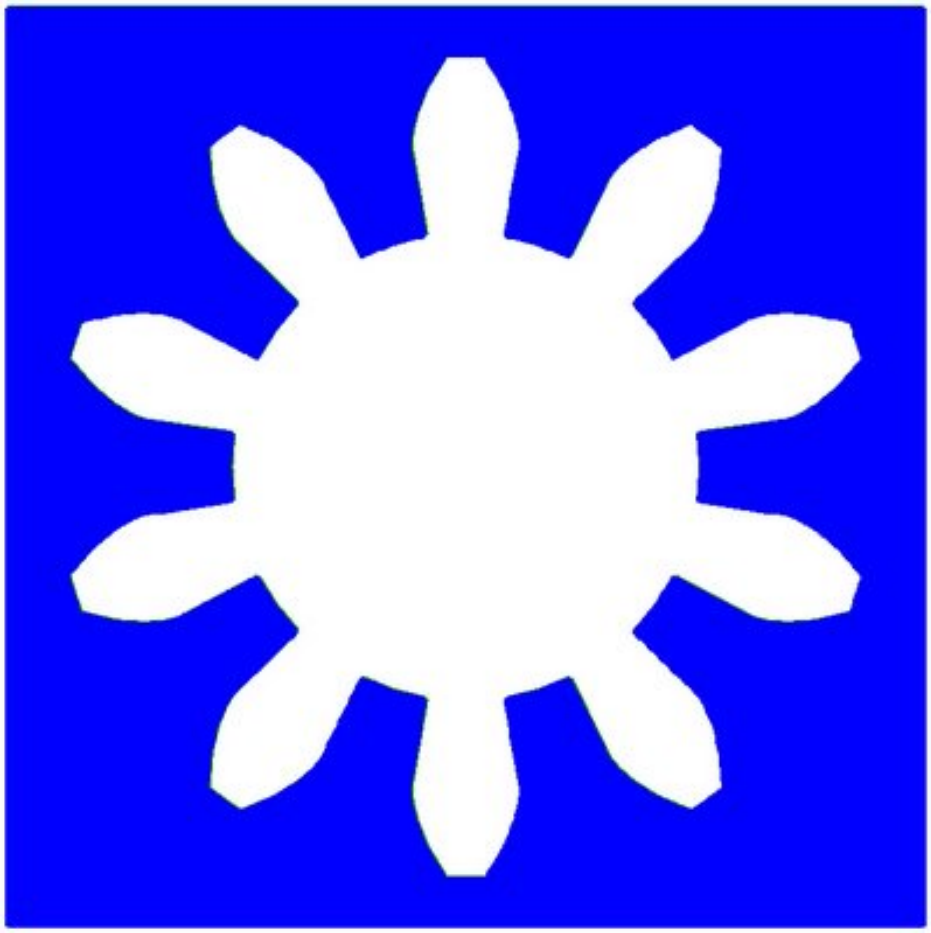
...then settled it into a cube...



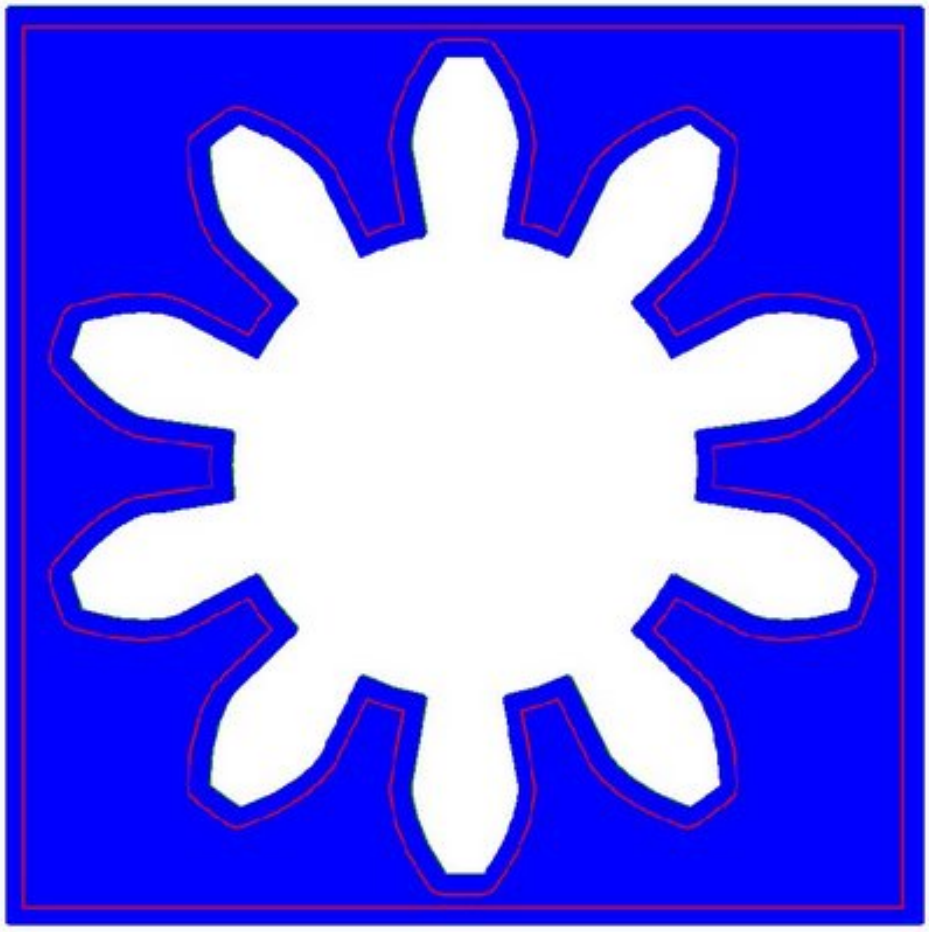
...and subtracted the gear form from the cube.



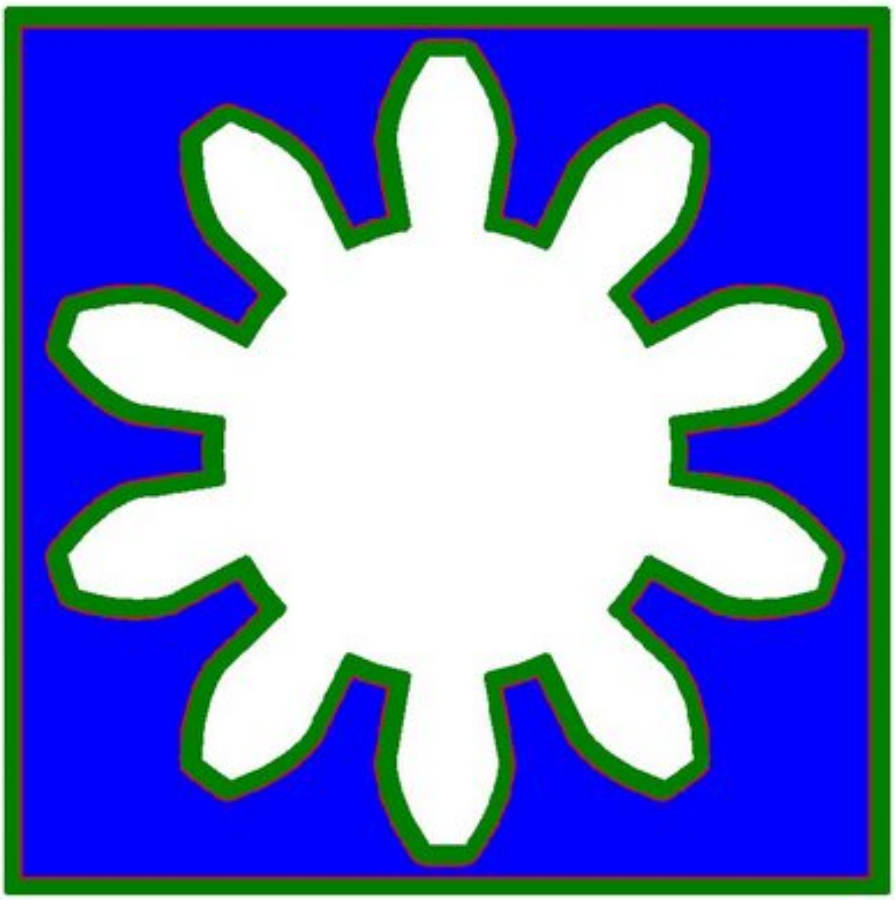
After an enormous amount of work on my Slice and Dice routine, I was able to calculate toolpaths for milling. First, I took a slice of the solid that represented the material I had to remove from my sheet of plastic...



...then I plotted a cutter path around the inside and outside perimeters of the material to be removed. Since my cutter was 1/8th inch in diameter, I set the center of the cutter path 1/16th inch from the perimeter.



Once that was done, I was able to identify the perimeter side of the cutter roads in green.



The remaining blue zone was where I need to remove material with a variation on the infil routine that used to fill the interiors of the extruded prints that I made with Tommelise 1.0. I haven't converted this piece of code yet. One nice feature of milling, as opposed to printing an object, is that you are not facing the issues of getting too much or too little plastic extruded over a complex road trajectory. With milling, it is no big deal if your cutter head goes over the same piece of plastic that you need removed twice or more times as long as you keep your cutter from intruding on the object you want remaining after you are finished milling.
Once I'd got to this point, I decided that it would be good to see if I could mill a gear by its perimeter as I had when I milled the polymer pump of the old Mk II extruder week before last. It was good that I did that, because the shape intensive nature of the involute profile gear that I was trying to mill revealed several software problems on both the Slice and Dice and the firmware side of the Tommelise 2.0 control system that hadn't revealed themselves with the Mk II exercise.

Finding and fixing these problems required that I spend a great deal of time upgrading the GUI of the Slice and Dice routine so that there was a one to one correspondence between what I represented graphically on my GUI and what I could click on with my mouse cursor. Once I had that capability, chasing down the code peversities was relatively straightforward.

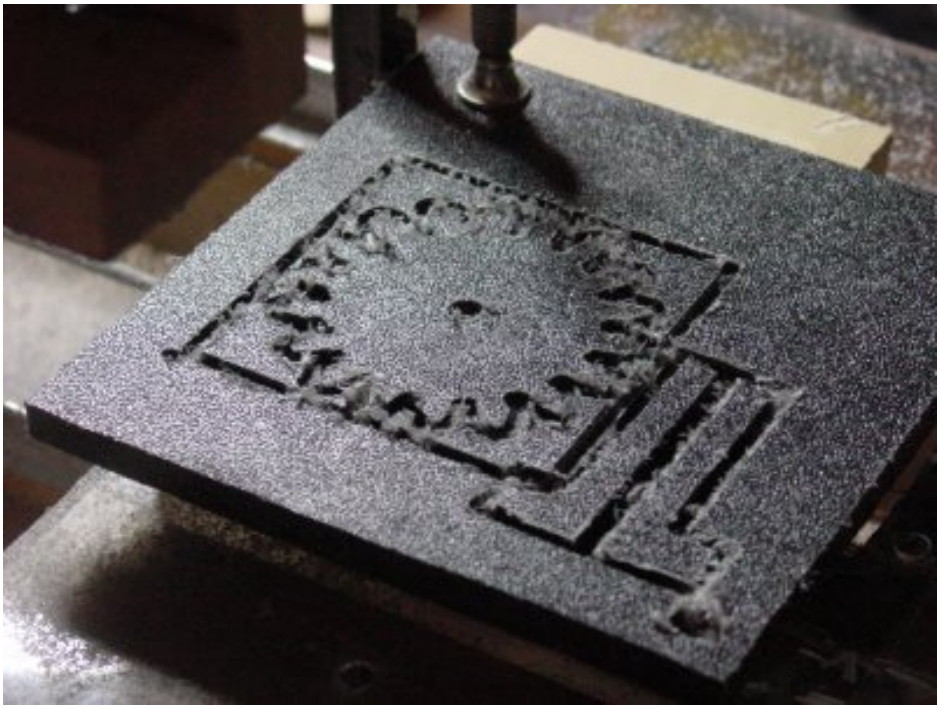
Because it takes me a week to get plastic sheet delivered as milling stock and 15 minutes to get milled poplar board, I did the testing of the milling exercise with the gear with poplar rather than plastic.

If I ever get bored working in plastic, I know that I will be able to make quite respectable all-wood cuckoo clocks.
:-D

18 teeth - living and learning

Sunday, 14th September 2008 by Forrest Higgs

In which the narrator discovers that milling has only a passing resemblance to additive printing...



I cooked up an 18-toothed involute profile gear in Art of Illusion this morning, processed it in slice and dice and decided this time to try cutting it in a scrap of black HDPE. The cutting went quite slowly, using about 8 minutes per layer.

This is not a function of Tommelise being slow but rather a question of the slice and dice routine not linking up the segments of the gear profile very efficiently. Most of the time in milling a layer is spent lifting the milling cutter at the end of one string of line segments and moving it across the gear then lowering it to start another string of line segments. I expect that a fully linked set of line segments would reduce the profile cut to about 30-40 seconds/layer with a 0.2 mm cut depth.

I had cut down to about 1/8th inch into the 3/8ths inch thick scrap and was beginning to prepare supper for my visiting son and daughter when I heard the cutter begin to really labour.

What happened was that the hours of vibration had loosened the chuck holding the cutter and allowed it to fall into the cut road with lateral but no vertical support. In a very few inches it had cut all the way through the 3/8ths inch scrap and into the poplar below it.
In doing that it marred the profiles of two of the teeth thus ruining the milling job.

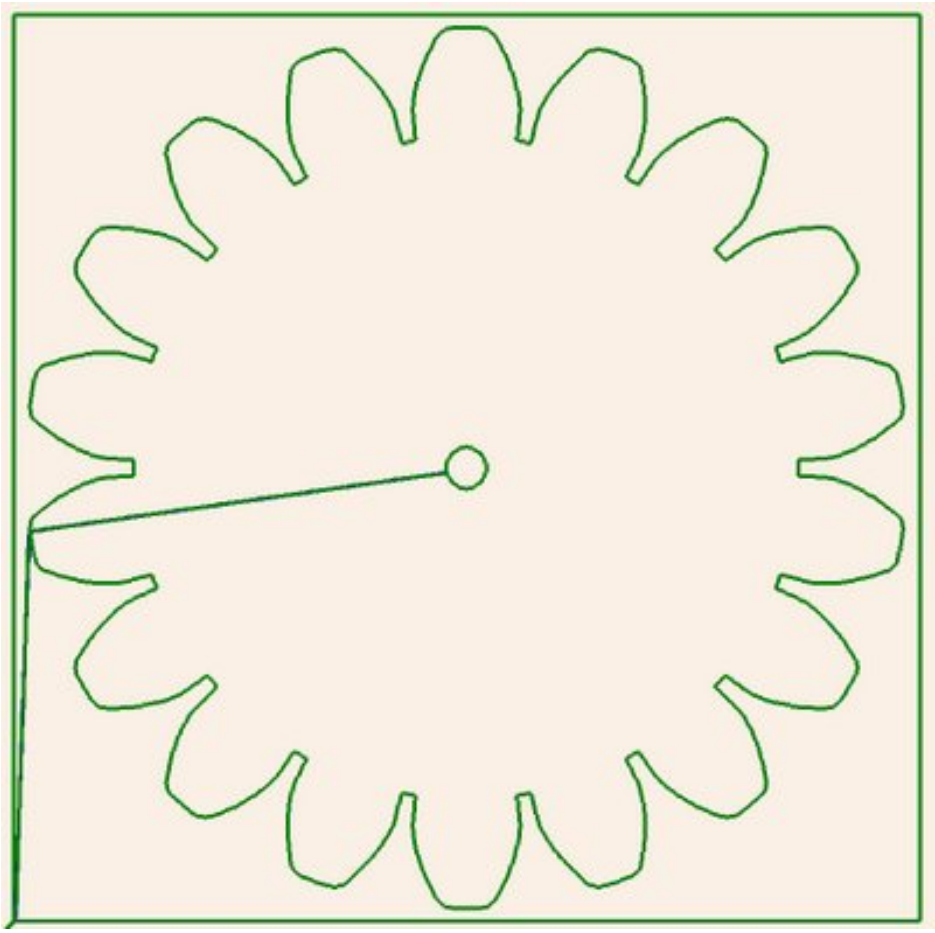
While I was unable to complete the gear, I did learn a few good lessons. I need to stop the milling job and check the seating and tightness of the chuck from time to time. That is pretty obvious.

What I also discovered was that I can mill beautiful holes with Tommelise. When I developed the STL file for this gear I put a hole in the middle for the drive shaft and dimensioned it to 1/4 inch. Making small, tight holes, as long as they are larger in diameter than the milling cutter is no problem whatsover.

Milling around objects

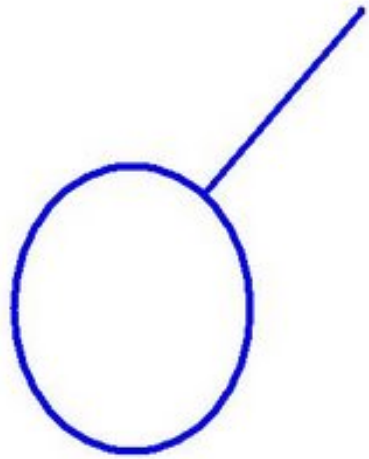
Sunday, 21st September 2008 by Forrest Higgs

While I have been able to mill gears with large tooth counts with milling data from Slice and Dice the number of times that Tommelise has to raise the head, move to a new part of the milling job lower it and start cutting plastic again has, not to put too fine a point on it, been excessive. A simple 18-tooth gear profile cutting job that ought to have taken 30-40 seconds was taking 8 minutes. My efforts for the past few days, as a result, have been aimed at reducing that.
Actually, it didn't take too long to get Slice and Dice to minimize the number of milling segments drastically.

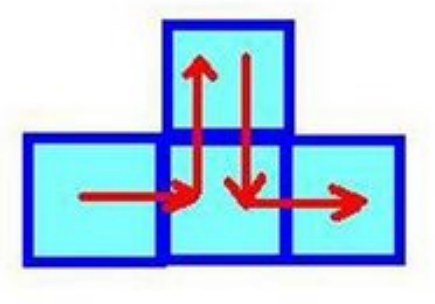


I was able to reduce the 18-toothed gear job down to three loops; the outer square perimeter, the gear perimeter and the drive shaft hole. ***read more***
As it happened the software problem I'd been having getting complete loops had a topological origin rather than a simple coding or logic flaw. As you know, I use a grid rather than a vector method for developing toolhead paths. While the grid method tends to make for simpler code, it is also more CPU intensive. What I hadn't realised is although it is much simpler than vector code it does have a few problems specific to the method itself.
The one that was causing me trouble had to do with this sort of thing. If you describe a vector loop that looks like this.

If your grid is sufficiently coarse, it will interpret this vector loop into something like this instead.



Looking at this at high resolution in the grid you see this sort of thing going on.



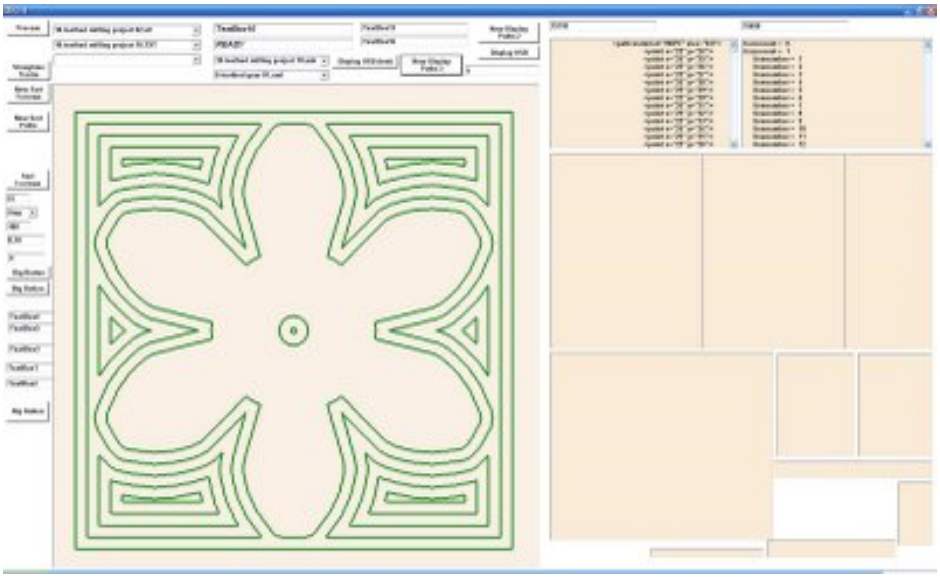
This makes for some very complicated coding, indeed, if you are not careful. While I rarely had a narrow tool cutting road collapsing into a single path going in, dead ending and coming out by the same route, I did quite regularly have situations of a long line more or less straight path having a little hiccough like the one shown above. I solved the problem by the simple expedient of just dropping those, since they were only 0.1 mm hiccoughs in any case.

At that point I had to tackle a much more daunting problem of how to generate efficient tool paths that would enable Tommelise to clear an area in plastic around something, like a gear, that it was milling out of sheet stock.

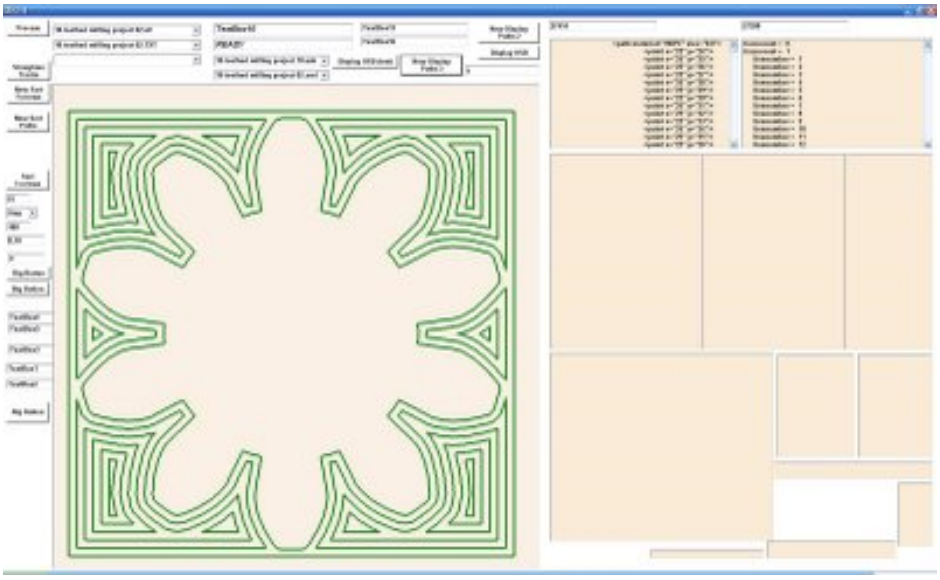
As a practical matter, I wanted to do was to make two fused gears with different tooth counts which rotated on the same shaft. This sort of object is very commonly used in gear trains where one wishes to gear up or gear down the rotational speed of a DC motor. While I had been cutting single layer gears with gay abandon making such a two gear assembly in one piece meant that I had to cut the smaller gear first and then excavate and clear plastic down to the level of the second, larger gear. That meant that not only did I have to be able to create the proper toolpath to cut out a gear, but that I also had to be able to efficiently clear large amounts of material that I didn’t need to get to the level where I could cut the second gear.

To do that I had to change the Slice and Dice software so that Tommelise became a true 2.5D milling machine rather than just the equivalent of a 2D laser cutter. After a few days trying to Google prior art I gave up and "invented", not necessarily originally, my own. What I did was to simply leverage the code I'd used to develop the perimeter tool paths to keep cutting away the remaining grid till nothing was left.

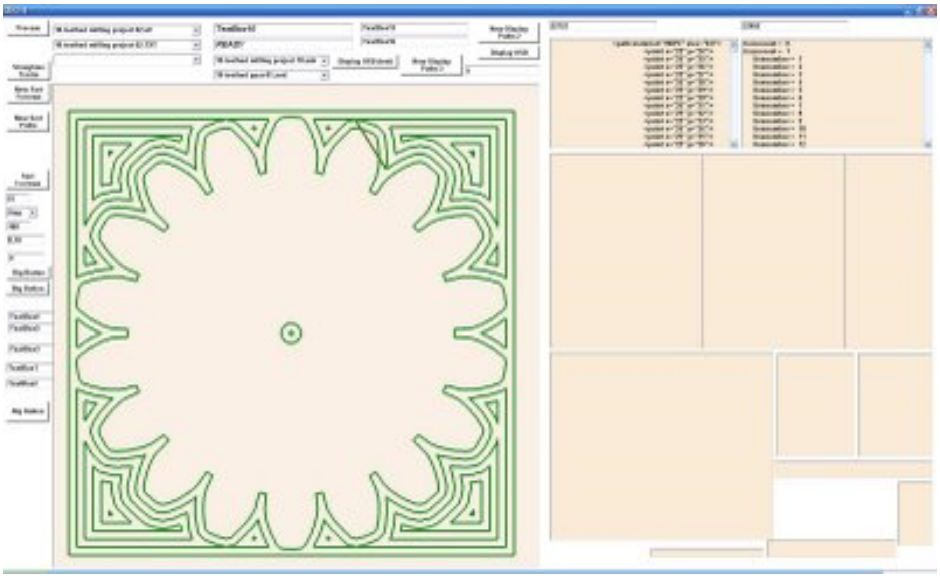
Here is what you get with a 6-toothed gear.



I still have a bit of work to do to organise the loops that I've generated in the most efficient order to minimize the time that the cutter is not actually cutting plastic, but the method is essentially robust. Having had success with the 6-toothed gear, I started hammering the code a bit to see if I could find other flaws in the code. The 10-toothed example worked perfectly.



I finally found a flaw with the 18-toothed example which you can see at the upper-right-centre part of the toolpath display.



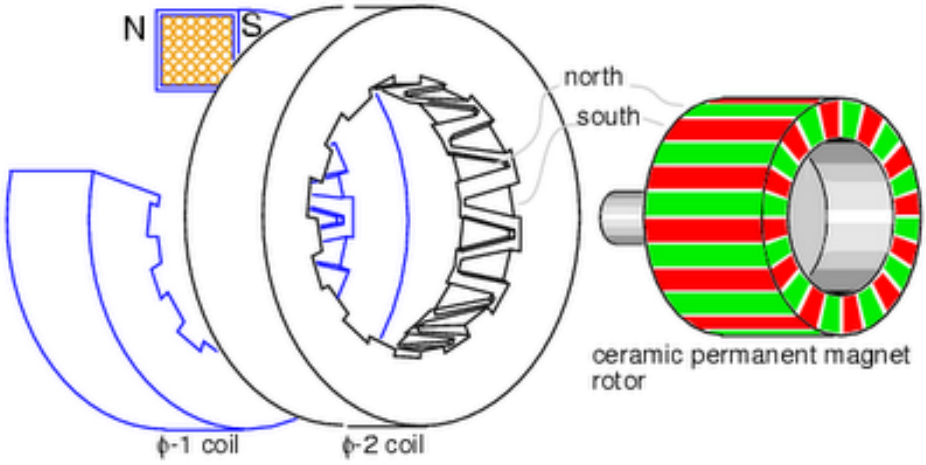
Last week, I made a special order of thinner HDPE sheet (1/4 and 38ths inch thick) suitable for cutting gears. UPS is set to deliver it tomorrow, so I should be able to test the USB print code generated very quickly.

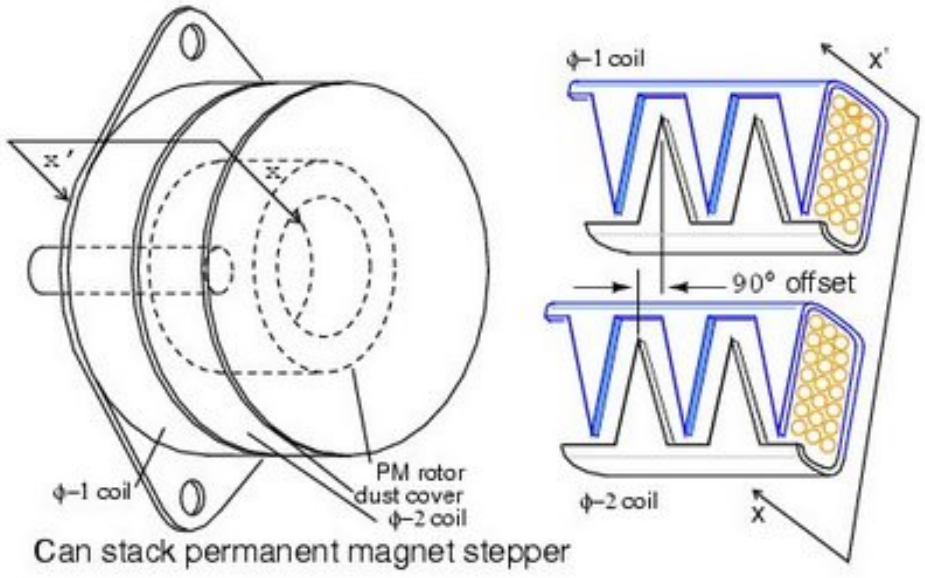
First try at cutting steel

Wednesday, 24th September 2008 by Forrest Higgs

Some time ago I resolved that I would take a crack at making Tommelise cut the thin sheet steel to do the field offsets for the linear tin can stepper that I've designed. I bought a little diamond cutting head for about $9 that works with my Dremel.

Field offsets are how you turn an ordinary air core electromagnet into a multi pole magnet suitable for making what is known as a "tin can" stepper motor. You can see here what I am talking about.



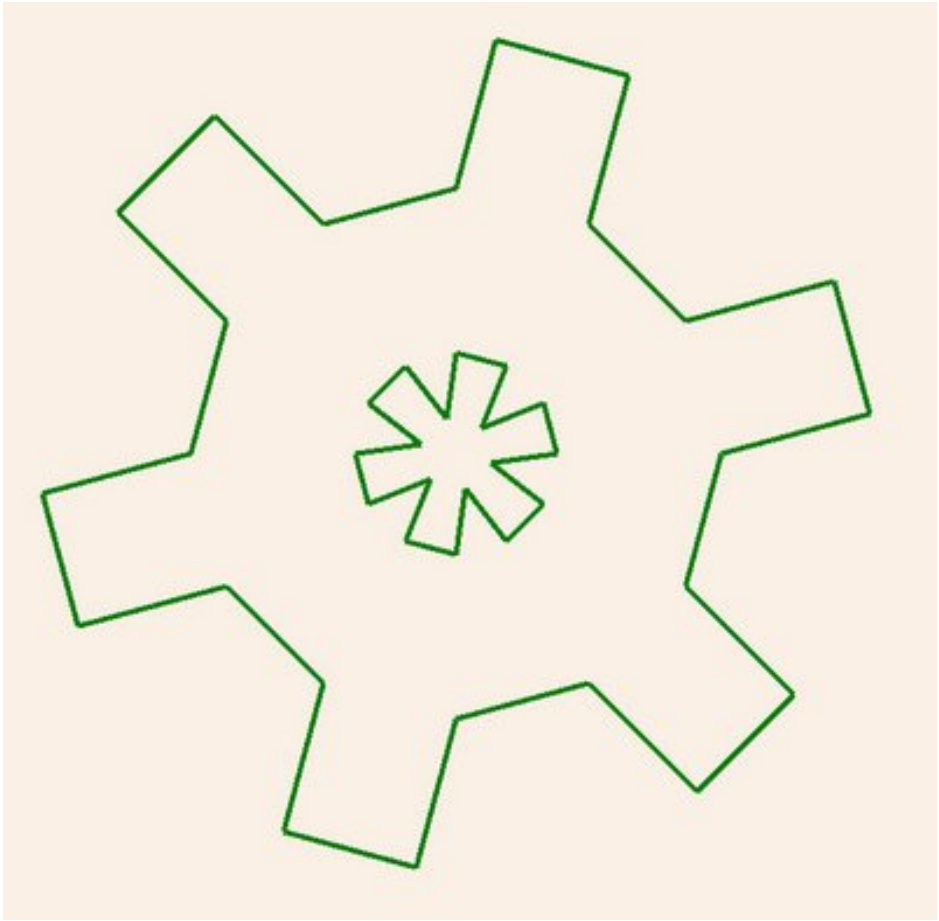


Can stack permanent magnet stepper

I plan on using a 3/8-24 inch threaded rod for a lead screw for now till I figure out how to cut my own lead screws. You can buy that kind of threaded rod in most good hardware stores. One of the gripes that I have with the Haydon linear stepper motors is that the 3 mm lead screw is far to flimsy for the lengths I'm using. Making my own linear steppers lets me fix that problem as well as controlling the costs.

First, I had to generate a flat steel expression of what the field offsets would look like (Shades of

my misspent years in engineering school discriptive geometry studio).



I then purchased a 1/16th inch round diamond cutter for the Dremel tool that I've installed on Tommelise 2.0 and went to work.

The first problem that I encountered was the flimsiness of the thin gauge steel needed for the field

offsets. It tended to bow and this caused problems for the cutter. Duct tape finally secured it sufficiently for an attempt at cutting to take place.

While the diamond cutter was making progress it was far too slow for the 0.05 mm vertical resolution of Tommelise 2.0. When I advanced the cutting depth I tended to get chattering and juttering as the cutter grabbed the edges of the cut in places.

I finally called it a night and attempted to cut a sample of the steel by hand with the diamond cutter. In fact, it was possible and went relatively quickly. The problem was, however, that the diamond cutter is most efficient when held at a 45 degree angle with respect to the material being cut. That makes all kinds of sense for a person using a Dremel tool by hand, but is not much help for me.

I'll be shopping around for a different cutter geometry to see if I can ameliorate this problem.

Meshlab

Monday, 29th September 2008 by Forrest Higgs

In which your narrator, unable to find the "simplify mesh" command in a newly downloaded copy of Art of Illusion rashly asks the rest of the core Reprap team where it is to be found these days...

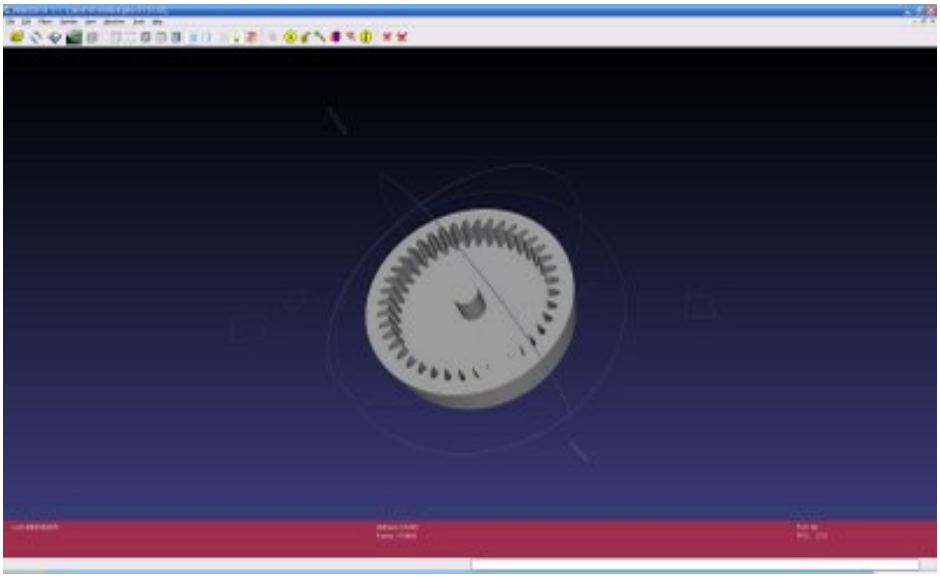
Since getting the positioning robot of Tommelise 2 running properly some time ago, I've spent most of my time milling plastic and steel. Doing that quickly revealed the lamentable state of my PC-side Slice and Dice software which means that for every hour I've spent milling this or that I've spent 4-5 hours sorting out problems in Slice and Dice.

The slicing module started out being reasonably stable, if a bit slow. Most of the development action has been happening in the dicing routines. Efficient milling of objects requires that Slice and Dice do things that aren't necessary for additive extrusion like FFF.

Last night, I pretty much got everything running using my usual software development paradigm, viz, writing a horrifyingly complex piece of code (not necessarily the same thing as a BIG piece of code) and spending weeks getting it running properly only to realise that there is a much simpler way of doing the job. The code in question was one which collected and linked together minute line segments generated by my grid approach to dicing. About dark on Sunday, I realized that a simple agent routine could probably do the dicing job that I'd been struggling to make work. Three hours later, the agent was written, tested and commissioned in Slice and Dice.

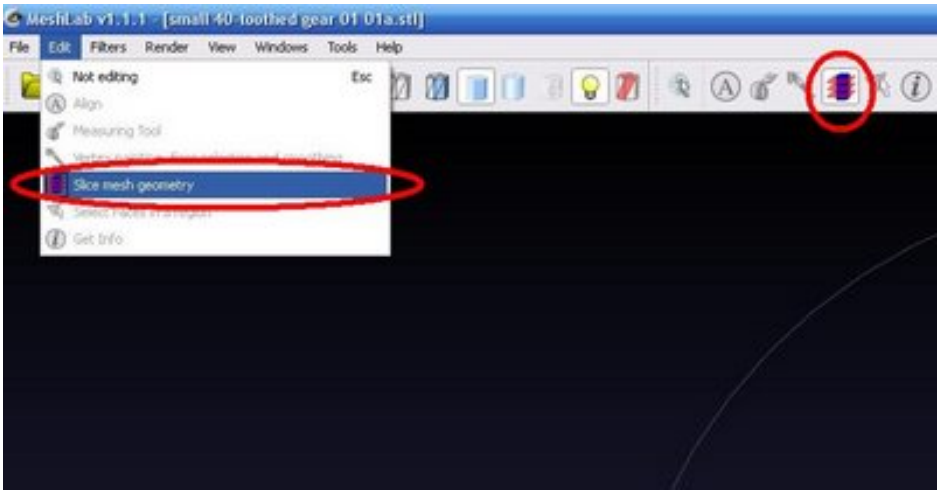
In testing the agent module, I began to run up against an Art of Illusion-related problem. While AoI does a fair job of converting solids descriptions to STLs it does tend to rather proliferate surface meshes. A small, 40-toothed involute profile gear that I'd designed required about 175,000 mesh triangles to describe it. Much of this complexity comes as a result of doing the #D boolean operations needed to convert a gear profile into a solid object description.
There used to be and may still be, for all I know, a command in AoI called "simplify mesh" which could greatly reduce the complexity of such AoI surface meshes. Unfortunately, I've been working on the positioning robot for Tommelise 2.0 for a long, long time and had bought a new PC during that time. As a result, I'd not run AoI and had got quite rusty with its use. When I downloaded and installed the latest version. That's when I got rash and put out a RFI to the core Reprap team asking for assistance in finding the "simplify mesh" command in the newest AoI release.

Vik, ever helpful, responded within a few hours that I ought to take a look at a new open-source package that he'd discovered the previous week called Meshlab. I quickly downloaded and installed Meshlab and had a go with it. I'd hoped for a simple routine that simply ate big STL files and spit out smaller, more efficient ones. When I imported my 40-toothed gear milling project into Meshlab, I realised that I'd got considerably more than that.

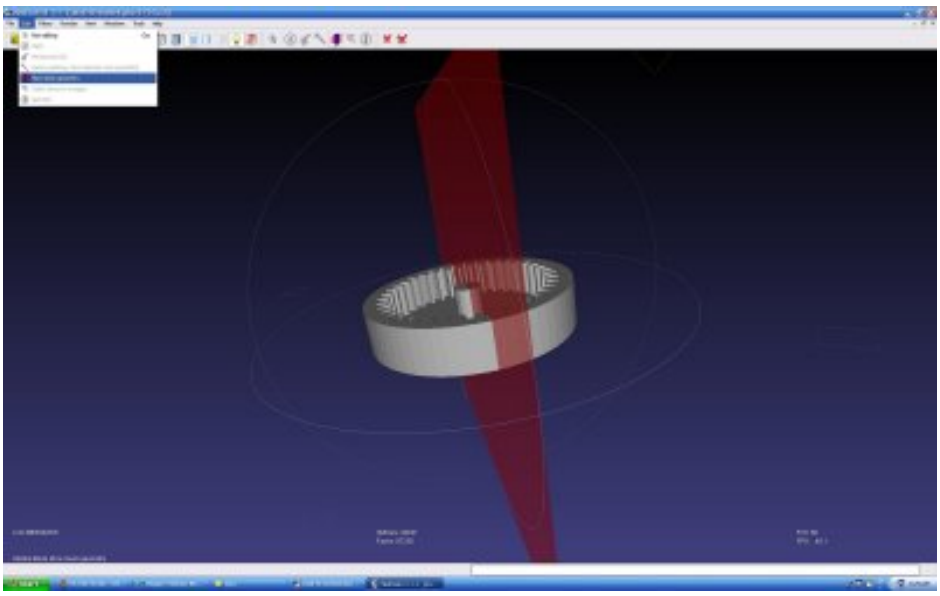


Following Vik's instructions I was able to reduce the complexity of the mesh described in my STL file by an apparent 75%, judging from the file size. When I tried to open this simplified STL file in Slice and Dice, the input routine crashed. I quickly discovered that whereas my Slice and Dice routine required that STL files be expressed in ASCII format, Meshlab spit them out in binary format. Finding no way to change settings in Meshlab to export ASCII format STL files, which is not the same thing as saying there are none, I solved the problem by re-importing the simplified STL file into AoI and then saving it again in ASCII format.

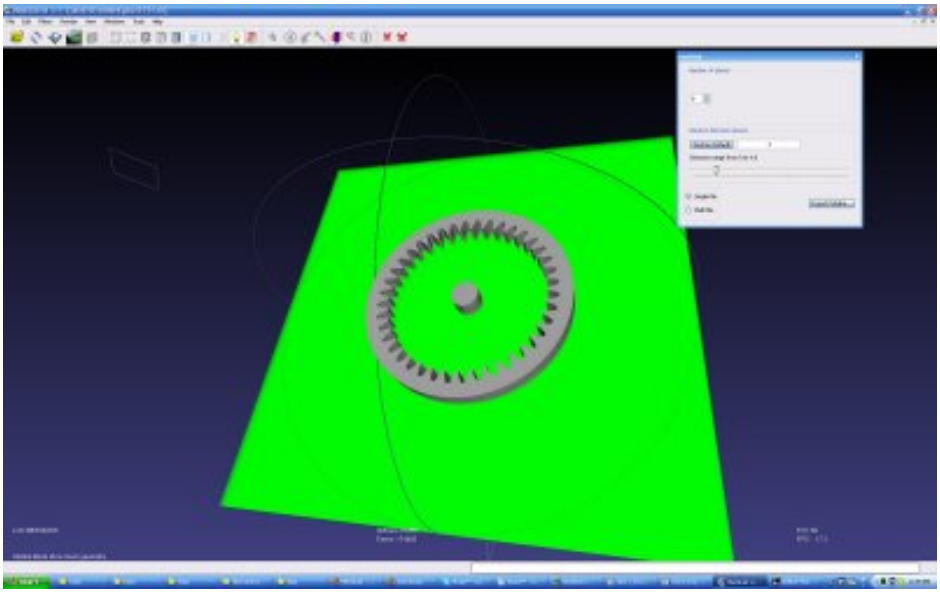
That done, I discovered that Meshlab had reduced the complexity of my original mesh by 50% instead of the 75% that I'd originally thought it had.
While poking around looking for an option in Meshlab to save STL files in ASCII format I noticed a little button consisting of a cylinder with a number of planes cutting through it.



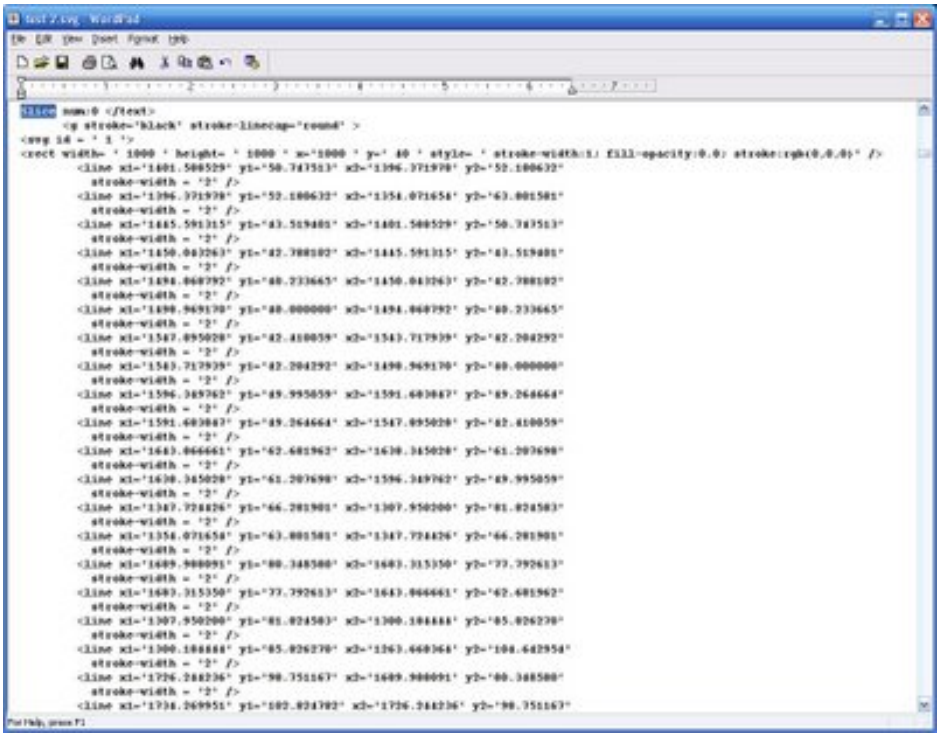
This command had the intriguing name of "slice mesh geometry". Clicking on it just for fun cut a plane through my gear project.



I quickly discovered that this cutting plane could be oriented and specified in some extremely interesting ways.



A small window popped up that allowed me not only to cut my object with a plane, but instead let me slice it with a set of parallel planes set at standard distances from one another which could be easily set. Not only that, it allowed me to save the resulting information in SVG (Scalable Vector Graphics) format, a variety of XML.



The biggest shock was the speed with which it could do this slicing. I'd been rather proud that my Slice routine could cut a slice something of the complexity of my gear project in a matter of a couple of minutes. Imagine my mortification when Meshlab did a dozen slices of the same project in the blink of an eye. Reading a bit more about the Meshlab project, they have some 32 programmers working on the code. From the looks of things they're also quite good programmers who know their geometry.

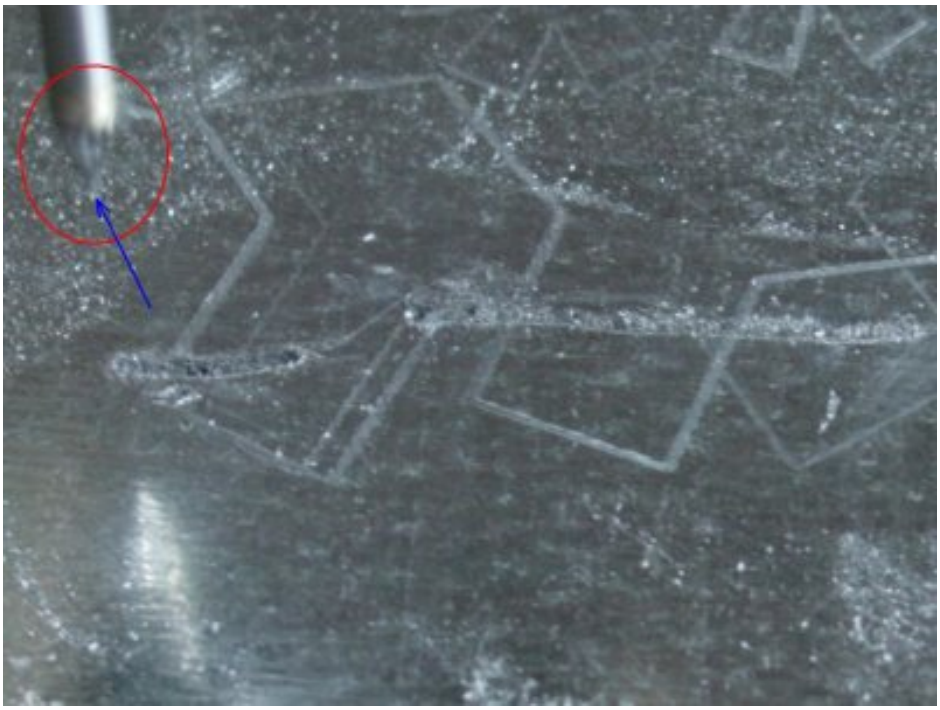
While I tend to trust my own coding, largely because if I need it to do something special I can get in and alter it to do that something special with relative ease, I know when I'm beaten. If the slicing routine in Meshlab is doing what it appears to be doing, I'm retiring by slice module and using Meshlab instead.

Second try at cutting steel

Monday, 29th September 2008 by Forrest Higgs

In which your narrator actually manages to cut the thin steel plate...

The round diamond engraving cutter that I used before was clearly not useful for cutting steel, only engraving on it. I expect that I could have actually cut it if I'd had many, many hours and a lot of patience. I went back and found a pointed tungsten carbide cutter head suitable, they said, for cutting things like stainless steel and granite.

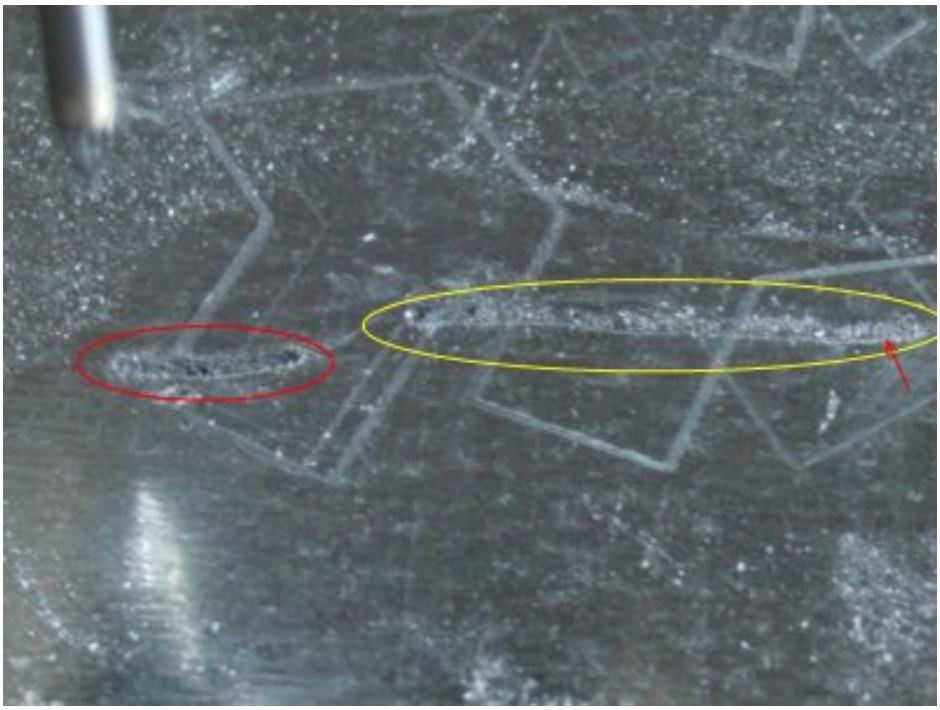


As I went through the many flavours of cutters on offer, I kept seeing the caveat, "do not use this cutter as a drill, cut from the side of the cutter." I decided to reduce the number of variables that I had to deal with so I glued a strip of the 0.01 inch steel (0.254 mm) that I was dealing with onto a piece of milled, 3/4 inch poplar with white Elmer's glue. White Elmers glue is not recommended for joining metal and wood. That's why I used it. I wanted to be able to remove the steel from the wood backing after I was finished cutting. In that regard this gambit worked very well. I was able to cut the steel without worring about the steel warping or resisting the cutter laterally with stored energy from bending. I also didn't have to worry about pieces of cut steel abruptly flying off when the cuts were finished.

In spite of that I found that the pointed cutter had problems, different ones mind, but just as serious as the diamond engraving cutter had. Instead of the steel sheet warping and storing energy in deflection, the torque induced by the high speed cutter tended, when the cutter grabbed at the edge of a cut at irregular intervals, to move the z-axis housing and the xy cutter plane fractionally. This energy would store until a grab of the cutter occurred. Then the cutter tended to cut off at an angle and drill all the way through the steel abruptly.

It was clear that quick, shallow cuts at high speeds, the method that worked so well with HDPE, weren't going to work with steel. At that point I abandoned trying to cut out the templates for

effecting the field offsets and decided to do a set of controlled linear cuts, not unlike what I did by hand initially with HDPE.
I quickly confirmed that quick shallow cuts were not going to work. At that point, having noted that the cutter would drill through the thin steel, I decided to see if I could drill through the steel and then cut laterally at a slow rate through the whole of the steel sheet.
That worked.



I made an initial cut (red ellipse) at several steps (0.1 mm) per second. The lateral force from the cutter head caused the z-axis mounting to flex slightly along the x-axis, a known weakness in the present x-z positioning table. When I dropped the cutting speed to about 0.1 mm/sec (yellow ellipse), I was able to cut through the steel successfully. The deflection at the right end of the 0.1 mm/sec occurred when I doubled the cutting speed.
At 6 mm/sec and given the noise levels, I don't think that I'm going to be cutting a lot of steel sheet without an acoustical hood over Tommelise or at least acoustically isolating the Dremel tool from the x-z gantry frame. All the same, I should be able to cut my field offset plates for my home-built linear stepper motor design.

Tommelise 2.0 goes into production

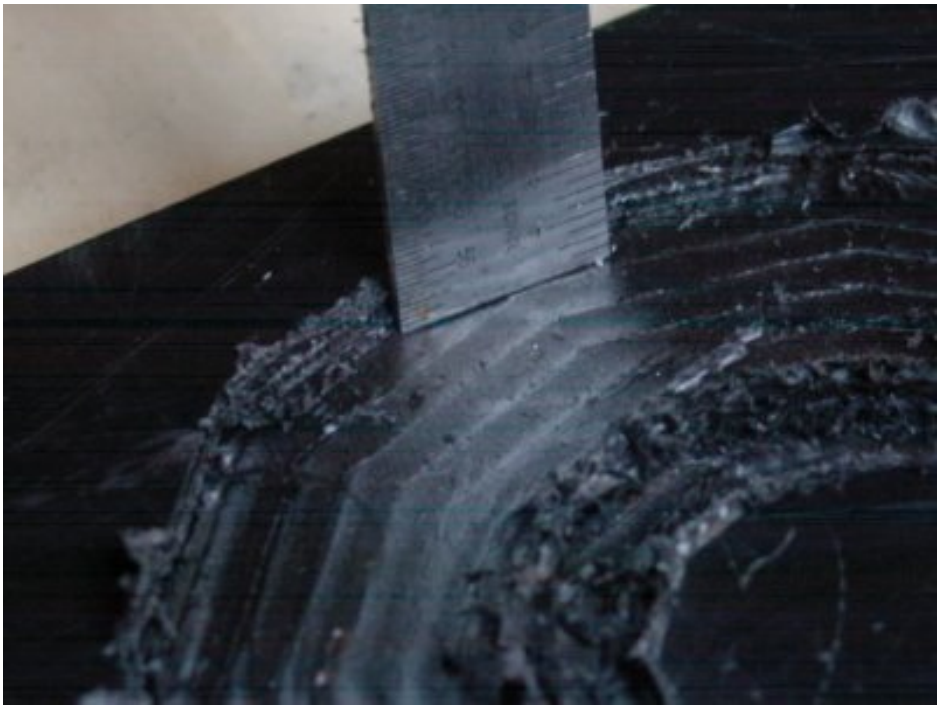
Thursday, 2nd October 2008 by Forrest Higgs

You won't be hearing too much about Tommelise 2.0 after this. It went into regular production last night making parts for other projects. There are a few things that want sorting out about it, viz,

- a less that rock solid USB link (not that that matters a whole lot given the EEPROM print buffer built into the system
- a z-axis that could be a bit more stiff and a z-axis stepper that could use a bit more torque (for milling work)
- a bit of workon the PC software side

All that said T2 is doing very good work right now. Last night I started proving the preliminary design for the electromagnet spools for the linear stepper motors that I hope to use in Tommelise 3.0. Here you can see one of the two halves for a spool being milled out of 3/8ths inch HDPE.

Here you can see the cut 5 mm into the 8 mm internal width of the spool.



The inner spool wall will be made of two 1.5 mm walls, one fitting into the other for a total thickness of 3 mm. The wide lip of the spool will be 1.5 mm thick. The air-core electromagnet it will hold 62 meters (220 grams) of #28 copper magnet wire. The electromagnet will draw 0.84 amps at 11 volts and produce a magnetic field of 128 gauss at the center of the air-core.

As it is presently designed the linear stepper for Tommelise 3.0 will use a common 3/8-16 threaded rod as a lead screw. In combination with the 15 degree step of the stepper motor itself the motor will advance a linear distance of 0.053 mm/step. Drawing, as it does, 9.2 watts, the

stepper will be pulling 240% of the electricity that the largest stepper on Tommelise 2.0, a Haydon 36000 series.

I've deliberately overdesigned it hoping that I can get something that will work acceptable. I can easily size the design down, should the design specifications I've made are even in the same ballpark with the stepper's actual performance.
.

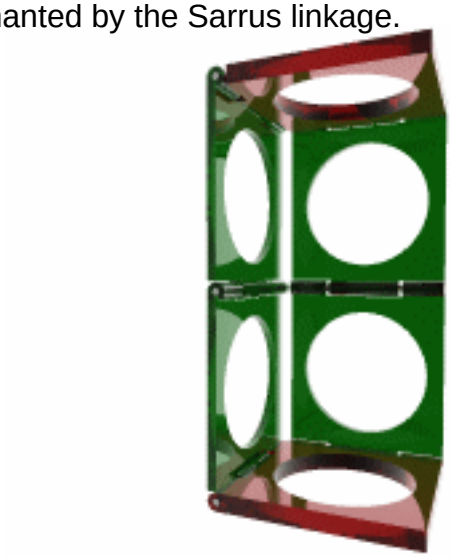
Notes on the Sarrus linkage

Monday, 6th October 2008 by Forrest Higgs

During the 18th and much of the 19th centuries accurate guide rails for machinery were simply not available. For that reason, the machines of the First Industrial Revolution tended to rely on linkages rather than guide rails. James Watt of the Watt steam engine fame, designed one of the first practical linkages for providing straight line guidance for his engines. Watt's linkage was followed by a succession of other, different approaches. In my opinion, the Peaucellier linkage is easily the most elegant.

It has seemed to me that a self-replicating 3D printer might well utilise linkages, which are easier to construct, rather than accurate guide rails

The problem with most of the linkages is, however, that they are 2D. By the time you either reinforce them to provide guidance in three dimensions they become either complex or bulky. For that reason, I've long been enchanted by the Sarrus linkage.



I've found watching it move is positively hypnotic. Since commissioning Tommelise 2.0 and carrying it over to production work I've had some time to consider what sorts of concepts belong in a third generation Tommelise. Of course, the Sarrus linkage came up almost immediately. Rather than invest a lot of emotional energy in designing a Sarrus linkage reprap machine, I decided that wisdom lay in knocking together a mockup Sarrus linkage that I could hold in my hands and see, if I could, what it was like. I began to have misgivings when I arrived at the hardware store and discovered that the hinges on offer tended to have a considerable amount of play. They also weren't particularly cheap for what you got. A Sarrus linkage uses six hinges and you are paying a minimum of $3.00 for a pair of poorly made ones.

This evening, while watching a made-for-TV docudrama on John Adams, I knocked together a Sarrus linkage. At first blush, it appears that the kinematic ideal and the reality match rather closely.

When, however, you begin to check the linkage for stability you discover that the play in the six hinges is additive and you are left with a tremendously unstable platform.

This was a disappointing denouement to what looked like a very promising line of development.

Notes on the operation of T2

Thursday, 9th October 2008 by Forrest Higgs

Now that I've been running Tommelise 2.0 for a while, I've begun to see a few pressing matters that want fixing...

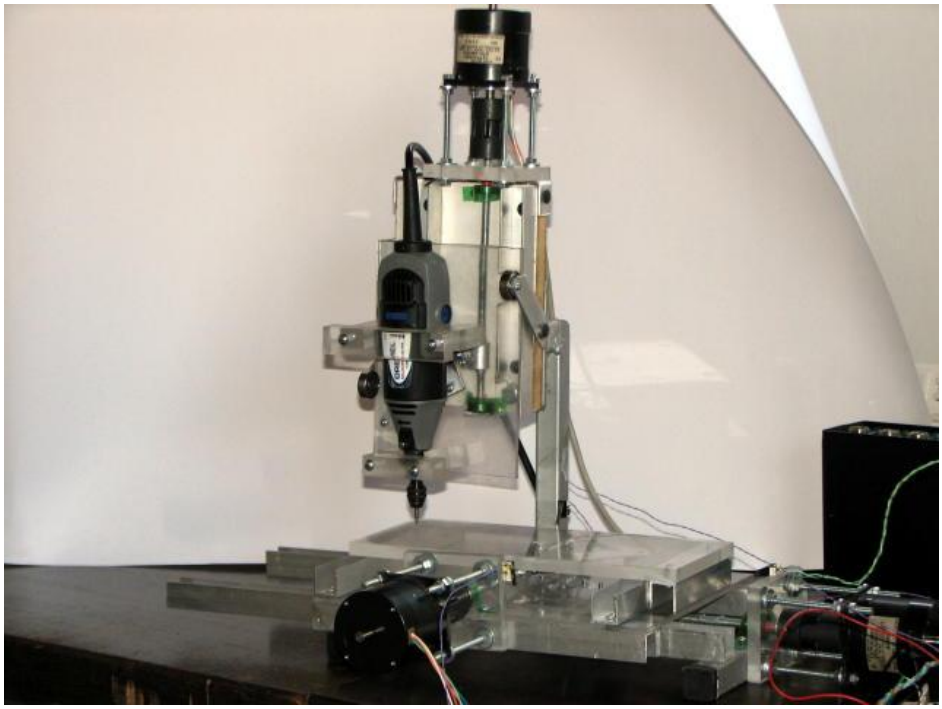
The most serious has to do with the Hayson 26000 stepper that runs the z-axis which positions the xz positioning table that seats either the extruder or the business end of the Dremel milling head. The short description is that it hasn't got enough torque. It tends to miss steps for a few minutes after I've started it up. This problem goes away after the z-axis slideways have warmed up a bit along with the multifilament wire that supports the counterweights for the z-axis.

I considered for a while simply replacing it with the z-axis off of T1 which is run by a Solarbotics GM2 and monitored by an Austria Microsystems rotary encoder chip. Doing that entails a bit of a rewrite of the firmware which I am not anxious to undertake just now. Instead, I am going to upgrade the 26000 linear stepper with a more powerful 36000 that I have in hand that isn't being used for anything else. That will require only that a single line of firmware code be changed.

It appeared that I was having some EMI/RFI (electromagnetic/radio frequency interference) caused by the Dremel hand tool that I am using for milling. This was expressing itself via the USB connection being dropped with rather depressing frequency. After some discussions with the core Reprap team and a great deal of Googling, I discovered that the USB cable that I was using was a HID, low speed cable which isn't shielded. I bought a shielded USB 2.0 capable cable as a replacement and the frequent USB connection drops have stopped. Those shielded USB 2.0 cables are fiercely expensive. I paid $27 for a 15 foot cable. That's as much as one of Darwin's stepper motors costs by way of comparison.

I also was having a situation where at very infrequent intervals the xy positioning table would simply go skating off merrily cutting through what I was trying to make. I initially thought that that was also a problem with EMI/RFI. In fact, it was caused by a dry joint in my controller board that caused the y-axis stepper to occasionally cut out after several hours when the board warmed up. I was able to determine this when, after the y-axis stepper had gone dead, it magically restarted when I measured the voltage across two of the poles of one of the screw terminals serving the stepper. The slight downward pressure from the voltage probes caused the dry joint to close.

In fact, although the Dremel hand tool has no EMI/RFI protection built in, it is apparently a remarkably quiet device electromagnetically. I ran it at 35,000 rpm under load right next to my radio on both AM and FM bands and heard no static whatsoever. In a way, this made sense in that I'd seen Dremel's locked into CNC routers which milled printed circuit boards. If it were tremendously noise electromagnetically, I doubt it would have been used for that.



After getting the z-axis fixed my next priority is to start milling printed circuit boards. I've had a few requests for T2 controller boards and don't want to put such people through the suffering of having to build up strip boards.

A new use for J-B Weld...

Sunday, 12th October 2008 by Forrest Higgs

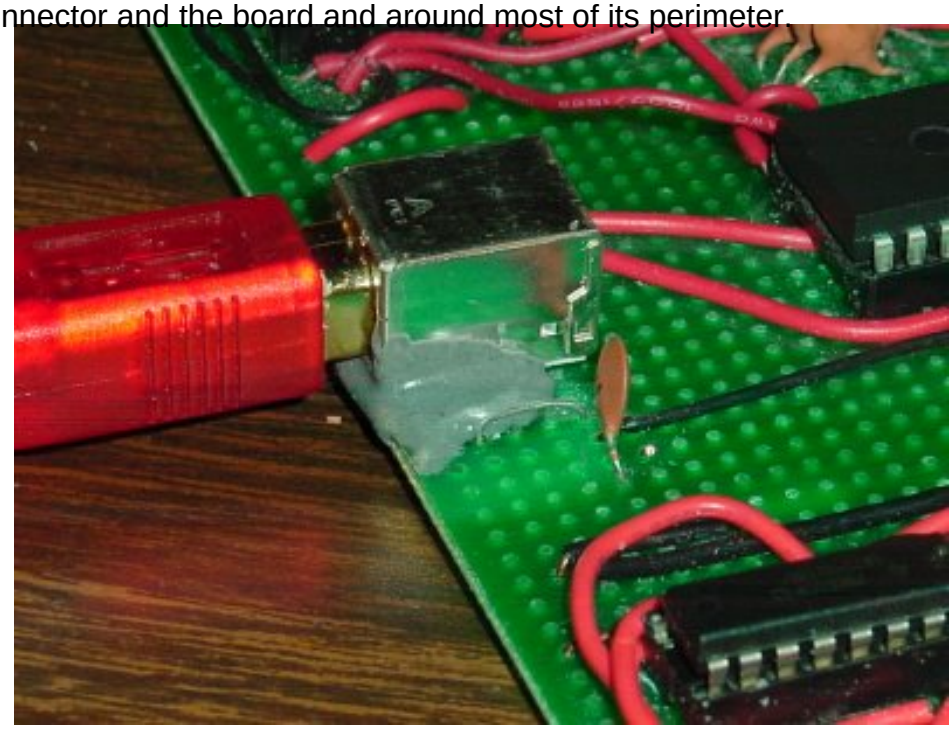
...and other adventures on the way to debugging T2's controller board...

As I mentioned in my last posting I'd identified a few bugs in the operation of Tommelise 2.0. While the shielded USB 2.0 cable cured a lot of the problems, it by no means fixed everything. Because of that I resolved to carefully chase down and fix all of the bugs on my punch list.

One very mundane problem I went after immediately. USB Type B connectors, the tall squarish ones, don't really have much of a clear way to be attached to the stripboards that I use to prototype circuits with. Because of that, the whole mechanical connection between the Type B connector and the board resides with the four hair-thin pins that connect D+, D-, +5v and GND to the rest of my circuitry. At the suggestion of several of the Reprap core team, I'd disconnected the +5v between my PC and T2's controller board some time ago. I'd left GND connected, also at their suggestion.

Getting back to the Type B connector, however, after connecting and disconnecting the USB port literally hundreds of times the connector seemed to be getting floppy, though it still seemed to work. I figured that it was only a matter of time before one or another of those four pins broke, however, so I decided to see if I couldn't fix that. I first removed the old connector and got out a new one. Putting that one on and making an effort to shove a pair fo the little mounting pins into some of the holes on the board didn't seem to help matters mechanically.

It then occurred to me that this might be a job for JB-Weld. I mixed up a tiny bit and put it both between the connector and the board and around most of its perimeter.



I had to wait most of the day for it to harden, but it REALLY fixed the problem of the floppy USB connector.

That fixed, I decided to document every connection on my controller board so that when I began to try to mill controller boards I'd have a definitive description of what was connected to what. I approached this task with considerable trepidation as I'd done this sort of exercise before with

other projects and found it to be a humbling experience at the best of times.
Once again, I stood amazed that my controller board worked as well as it did with so many errors in the wiring. I had chip grounds that weren't grounded, comms lines that were cross-wired and any number of other sins caused mostly by my dyslexia and occasionally by simple forgetfulness or pure incompetence.
That took most of Saturday. With all that sorted out I discovered that given the multiplexing of pins on the 40 pin 18F4550 depending on all of the Port A bits to be available for input or output, multiplexed with odds and ends like A/D circuits and the occasional clock, was a rather vain hope. Some of the problems with my y-axis stepper controller stemmed from that. I moved the two offending pins A.3 and A.4 down to D.0 and D.1, respectively, and sorted that out.
Preparing to do an undate on the firmware to take the new pin assignments into account, I electrocuted myself.
My flat was built in the early 1960's and most of the wiring dates from that era. Residential wiring back then was predicated on the charming notion that appliances could be run assuming A/C neutral and ground were, more or less, the same thing. There are a few properly grounded service outlets in my flat, mostly in the kitchen and the bathroom. While I have T2 connected via a long, heavy extension court to a grounded outlet in the kitchen, my PC was connected to an ungrounded wall outlet in my little office in my dining room.
I should have known better. I'd run into the problem a couple of times in South Africa under identical circumstances though with considerably more voltage involved. For you newbies without a lot of electrical experience, this is important, so pay attention.
The electrical power lines in your neighborhood, which are underground if you are nearly anywhere except the USA, provide what is called 3 phase power. When the electrical utility company hooks up your house, they usually only take power from a single phase of the line. For apartment blocks, however, they will usually bring in two or all three phases.
Here is the first thing that happens. When the utility people hook up a new building they have a nasty tendency to open up the transformer box and hook your service to the easiest phase bussbar that they can reach. That makes sense, but what happens is that you'll have a neighborhood transformer with, perhaps, three-quarters of its connections hooked to one phase. What that does is distort the distance between the neutral line and ground. At my home in South Africa this had got so bad that there was 180v difference between neutral and ground. I accidentally grounded my PC there and fried the whole thing back in the early 1990's. ESCOM, the state power company, brought out a team and completely rebalanced the transformer connections. They didn't pay for my toasted PC, however.
What happened today was similar but a bit different and something that I'd only had happen in labs before. Some nitwit electrical engineer way back when decided to save some wire and put some of my flats circuits on one phase and others on a second phase. Because of the aforementioned out-of-balance tendency at the transformer, there was a 63v difference between the two phases in my flat.
A voltage of 63v is enough to tickle but not injure as long as you are not severely grounded (like you're standing in water, for instance). I was chasing circuit voltages in T2's and had one hand on the board's GND and accidently touched the ground attached to the USB connector shielding which was detached from my controller board but connected to the my neutral grounded PC. I thought I was mistaken the first couple of times I felt the shock but quickly figured out what was happening.
Now I have both my PC and T2 connected to the same properly grounded circuit in the kitchen and the problem has disappeared.

I'm going to have a few quiet words with the fine people at Pacific Gas and Electric about their unbalanced 3-phase power during business hours tomorrow.

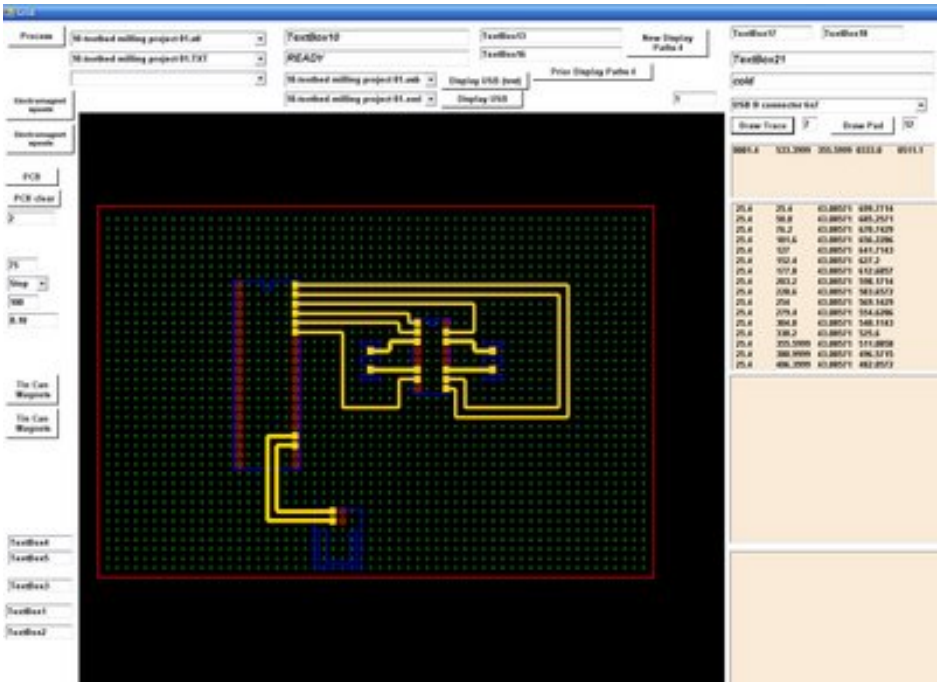
Just as an aside, I've returned T2 to full operational status this Sunday evening. Everything is clicking along beautifully. The bugs seem to have been resolved, for now at least.

J'en ai marre!

Tuesday, 14th October 2008 by Forrest Higgs

In which your narrator shares one of the better known sentiments of the French chanteuse, Alizée.

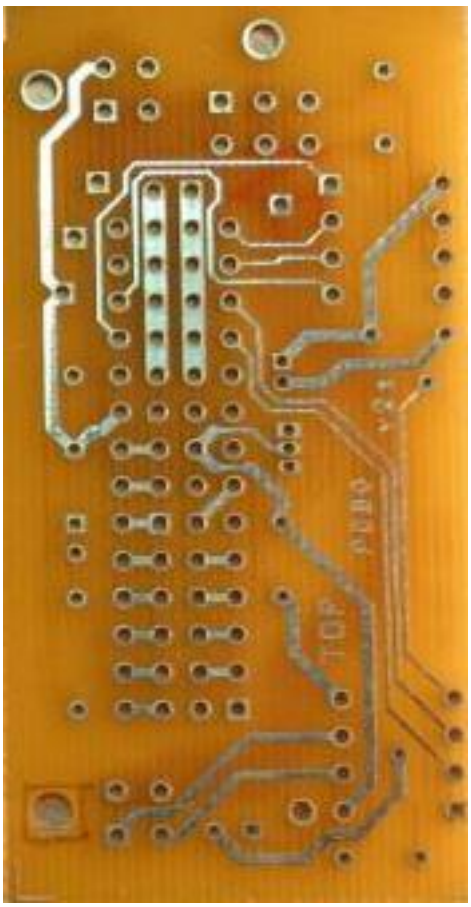
I had what was either a late night yesterdayor an early morning today, depending on how you want to look at it. By about 1830 yesterday, I was, like Alizée, ... *fed up ... pissed off ... tired.* For the past several weeks I have been diligently working my way through the tutorials for Eagle, KiCad and, finally, FreePCB. Voicing a few of the more pungent Afrikaans explicatives under my breath, I shut down FreePCB and opened Visual Studio in Visual Basic .NET mode. By 0130 the next morning, there was this...



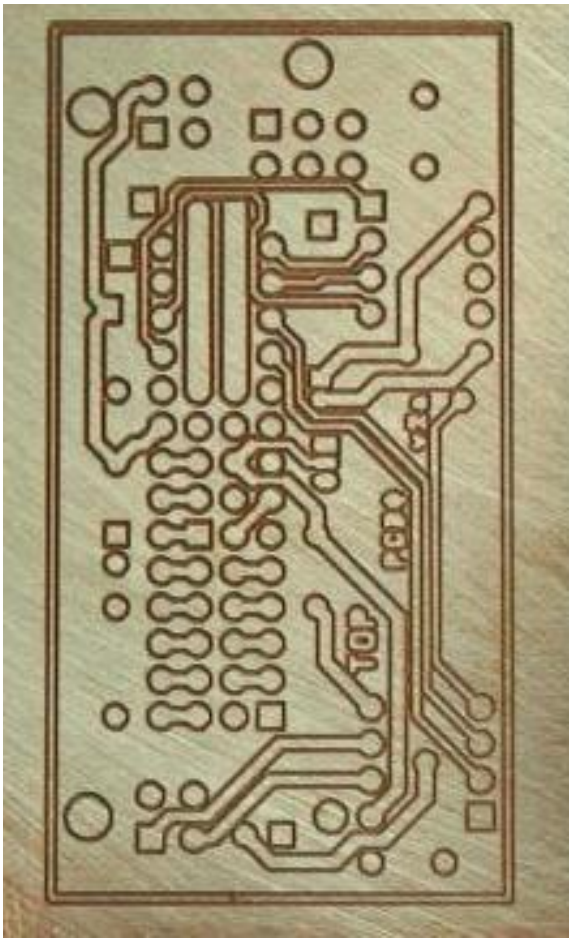
I'd decided some time ago that Tommelise 2.0 was, as part of it's ability to print parts, going to be able to mill the printed circuit boards (PCBs) for Tommelise 3.0. One of my long-standing goals for the whole Tommelise Reprap project has been that a reasonably bright twelve-year-old should be able to build one from printed parts received from another Tommelise owner.

In order to make that happen I needed to be able to convert the output from a PCB design app to something Tommelise could understand. On the face of it this was no big deal. Eagle can spit out G-Code via a plug-in while KiCad and FreePCB can handle Gerber format plus a few others. As I've discovered, however, appearances can be deceiving.

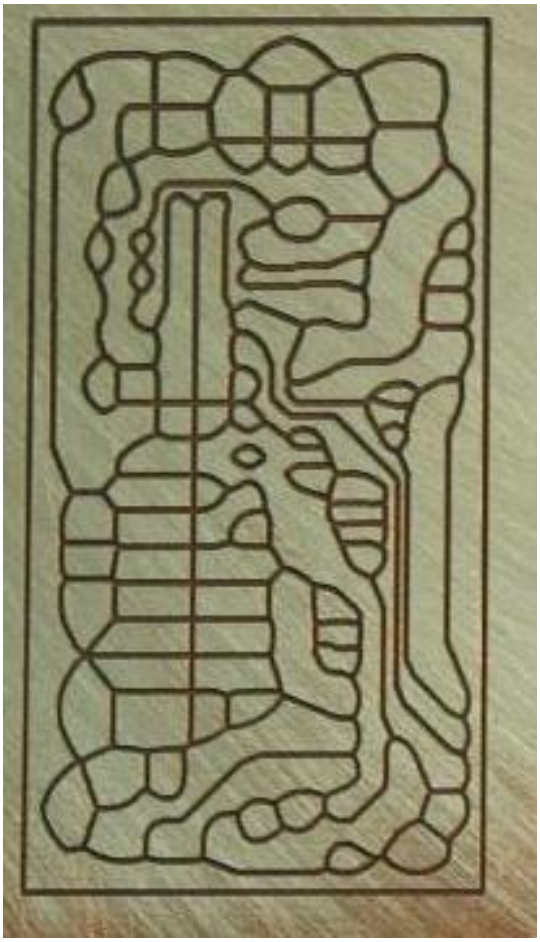
A few months ago, I ran across a lovely paper on applying the Voronoi Isolate notion to the calculation of toolpaths for milling PCBs. Recapping briefly, most PCBs are made by one of several chemical etching processes. The end result tends to look something like this.



When you convert the G-Code or Gerber files over to toolpaths for a milling machine you get an end result that looks like this.



If you look closely several helpful pieces of text that were included on the chemically etched PCB have been translated, more or less, whole cloth into milled paths which, incidently, have nothing to do with connecting one pin or component to another electrically. Traditions which developed over the years in chemically etched PCB design encourage this sort of thing. Eagle, KiCad and FreePCB all do this sort of thing. While that's helpful, when Vona and Rus at MIT popped this Gerber file into their Voronoi Isolate app, this resulted.



If you look on the board in the position where printed text appears on the etched and conventionally etched PCBs, you see that the text has been transmogrified into something looking like vaguely like an insect larva. While that was somewhat distracting, I began to see real problems in their coding when I stumbled across this little bon mot in their report.

As it relies on the above-stated assumption about the Gerber input, Algorithm 1 does not work in all cases. Some parts of some traces may not have segment endpoints and flash center points which precisely match-up, even when most parts do. An example is shown in Figure 3 (left), in which Algorithm 1 has already run and assigned primitives to traces as best as it could. Such cases can only be detected by searching for geometric overlaps.

My original plan was to use the MIT Voronoi app along with KiCad to generate nice toolpaths for Tommelise. When I read the paper closely, however, I discovered that the authors had apparently implimented Algorithm 1, which had the sorts of all-to-common problems caused by less than perfect Gerber formatted input files. As well, they'd used a rather complex geometric approach to creating Voronoi Isolates and implemented it in, the good Lord save me, Java. ~~***While they proposed another, presumably more bullet-proof, approach and even sketched out how it ought to work in some detail, there was no evidence that they had ever coded their Algorithm 2, much less tested it.***~~

~~***I've played sleight-of-hand in enough research reports over the years to know that if they'd***~~

~~***actually written and tested Algorithm 2, the authors would have been bragging about it and have written subsequent papers about it. As it is, nothing is out there. That means that it either never got written or the authors are keeping it under wraps in hopes of commercialising it, quod erat demonstrandum (QED).***~~

{Update, please note: Marty Vona has corrected my misapprehension about the state of Algorithm 2. Indeed, Algorithm 2 was coded and is available at the link...

http://www.mit.edu/~vona/Visolate/visolate/processor/TopologyProcessor.java

correction and update made on 17/11/08}

I wrote some code that did Voronoi Isolates years ago using a much simpler grid approach and had most of the routines in Slice and Dice to do the job again.

That left me with learning how to design PCBs with one of the available, preferably open source, apps. As I spent more time with the apps, I tossed Eagle in that although it was very good, it was a commercial app. While they let you use a severely cut-down version of Eagle for free, if you needed more than that, you were looking at paying $750 for just their standard version. While I've never shunned software just because it's commercial, I've got a severe allergy to paying more than a very few hundred dollars for one app.

It was a matter of a few moments to locate a torrent for a recent cracked version of Eagle on Pirate Bay which would have got me around this obstacle. Thinking about it, though, it became clear that it was just bad business to predicate the success of an open source project on a commercial app that users would be tempted to pirate. Thus, Eagle became history.

The more I worked with learning KiCad and FreePCB the more I kept thinking about that hypothetical bright 12 year old that was my professed target audience. If he or she had the slightest touch of attention deficit disorder, the learning curve on either of the PCB apps was well over a magnitude too long. It was like learning Algebra in Middle School, it takes far too many hours "learning" things that turn your mind to slag before you get anything back to convince you that it's worthwhile going on.

The authors of both packages began to write their apps years ago with the notion of creating a simple app that would draw in a lot of new users. Over the years, however, both apps became more and more like commercial bloatware, and their original simplicity deteriorated with each upgrade even while their power and flexibility increased dramatically.

As my spare time, hour by hour and day by day, evaporated, I began to realise that I was caught in the old, old situation which is most efficiently described as...

...when you are up to your ass in alligators, it is difficult to remember that your initial objective was to drain the swamp.

Briefly, I felt like I was trying to drive a lawnmower from the cockpit of the Space Shuttle. I'd only wanted to mill single, or at most double layer PCB's. I'd gone to a LOT of trouble to keep my controller designs simple and using as few parts as possible. By 1830 yesterday evening it had become crystal clear that the situation had spun completely out of control.

With considerable trepidation, I began to write my own app.

I got acquainted with KiCad, a French app, some years ago and had got quite used to and liked Eeschema, the schematic module. It was quite useful for documenting my controller designs. As well, I'd generated my own component library to get me somewhere between schematics and physical board layouts. Ordinarily, chip pins in the Eeschema libraries are grouped, more or less, by function rather than actual physical position on the chip. I'd rewritten those chip descriptors to reflect the real pin positions on the perimeter of the chips that I was using. I found being able to look at my board and look at my schematic and see, more or less, the same thing was very handy for quality assurance.

Having come on the Reprap project quite early on, I'd got, with considerable pain, into using stripboards, specifically Euroboard stripboards, as a development tool. Stripboards are rather handy for blocking out new circuits and are more permanent than a powered breadboard. Recently, however, my use of hex inverter chips to reduce the number of microcontroller pins needed to run T2's stepper motors have rather overwhelmed the ability of the Euroboard stripboard to cope and the poor things have been covered with a mass of jumper wires.

When I thought about it, I wanted to mill PCBs not much more complicated than stripboards before the mare's nest of jumper wires began to collect on them. I want there to be a one-to-one match between my schematics and my boards whenever possible. What I coded last night pretty much does that.

I can...
- place footprints of the components I need on my board
- put solder pads under the pins
- connect the pins with traces
- generate an XML formatted equivalent of a G-Code file that my milling module in Slice and Dice can make sense of
- geneate an XML formatted drill file that tells T2 where to drill the holes for the component pins

I've got maybe one or two more man-days to add in a few editing and file save/refresh capabilities and then one or two more to interface the file formats with the milling modules. I should be able to cut some test PCB designs that actually mean something next weekend.

Sort of like designing with 2D strip board

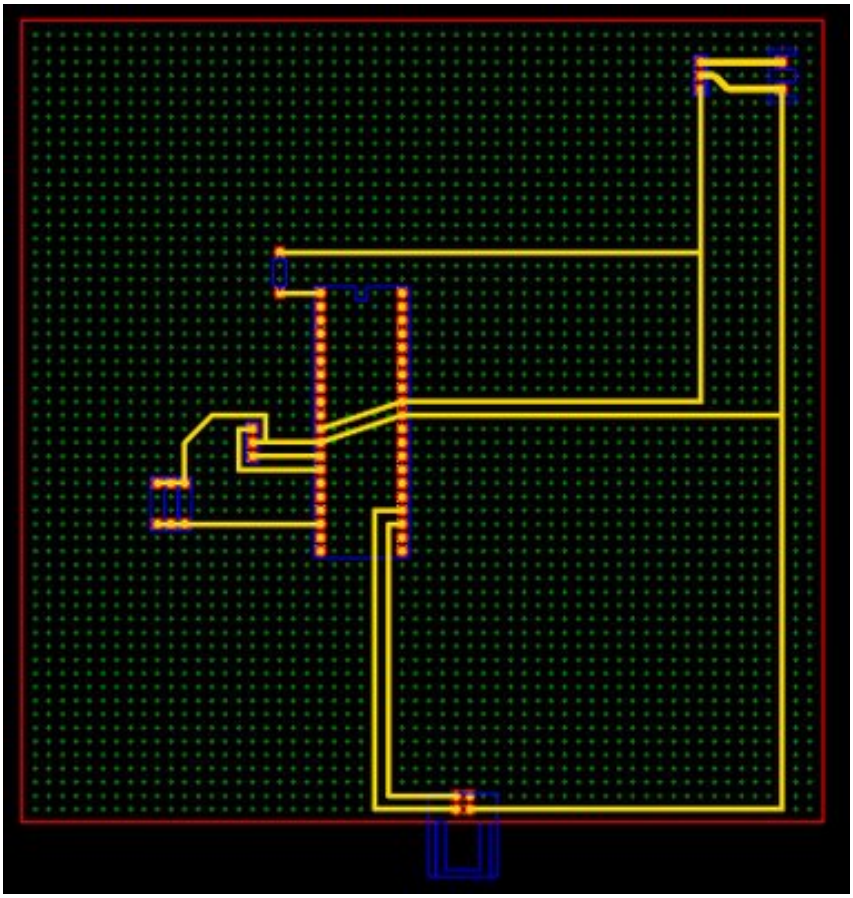
Friday, 17th October 2008 by Forrest Higgs

In which your narrator overestimates the difficulty in knocking his PCB layout app into shape...

I was very pleasantly surprised with the ease with which my PCB layout app came together. I visualised getting it working just well enough to generate some milling files for T2 this weekend. I got a LOT further than that this evening.

The biggest job I has was to sort out the graphics scaling equations that I was using. I ALWAYS have trouble with that. Of course, I got something that looked like it was right that turned out, once I started trying to test its envelope, to be be profoundly wrong. The difference this time, however, was that I was able to sort out the code snarls in just an hour or so this evening. Heretofore, I often spend days on this sort of task because of my dsylexia.

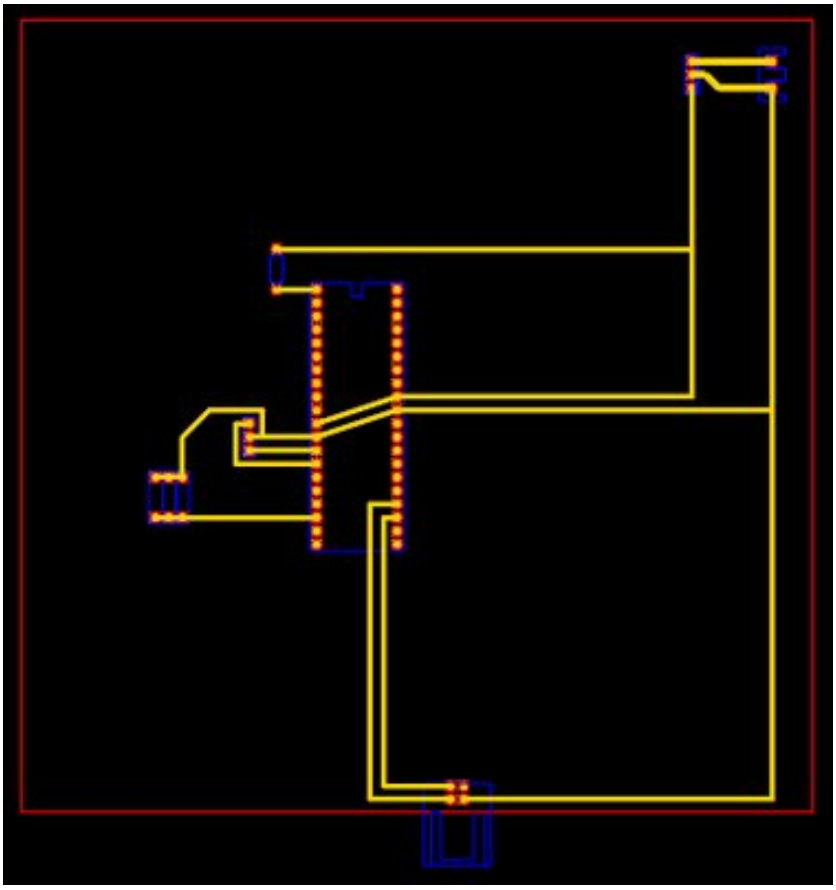
Once the scaling equations were right the rest of the work, things like being able to delete traces and components, storing and reloading layouts and the like was all done in about two hours. Just to put the app through it's paces, I did a basic layout of the 18F4550 plus its USB connection on a 150x150 mm board. I've left off a couple of capacitors that go with the voltage regulator in the upper right-hand corner of the board. It wouldn't have been a big deal to put them in the footprints file. I just haven't got around to it yet.



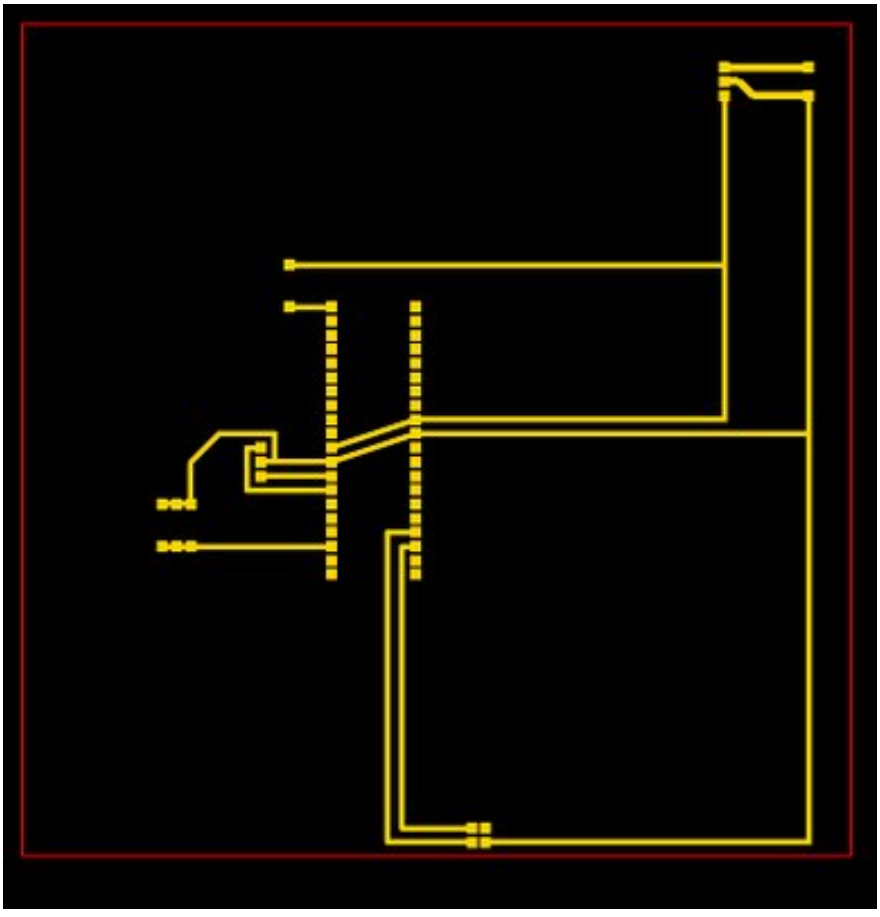
Here you see the edit and build window. You choose the footprints of the basic components from a pull-down menu and click the pin 1 of each onto your grid. The grid resolution is standard 0.1 inch (2.54 mm). It works sort of like stripboard, but in two dimensions instead of one.

You can see from the layout that the 18F4550 is an extremely easy chip to set up communications for.

Once you have your components laid out and connected up to your satisfaction you can just look at the layout sans the grid.



Then you can click to just look at the copper.



I will scan that window directly in VB.NET to generate the array that the milling routes app needs. I suppose I could also simply put it out as an image so that anybody who wanted to do chemical etching could do so. It would be easy enough, but it's not something I am personally interested in. Once you're milling your own PCBs, the economics change rather dramatically. Aside from not having to worry about front end costs and placing a substantial order to spread my overheads, I can get single or double sided board material for about $0.6/square inch. A 150x150 mm board costs a bit over $2. Because of that, the incentive to make the boards as tight as possible simply isn't there.

Some good advice about milling PCBs from Fred

Friday, 17th October 2008 by Forrest Higgs

Fred, at North Bay Technical, give some good advice and demystifies the problem of wear on milling bits...

I've got T2 set up for milling PCB's and got the board design app pretty much done last night.This morning I ordered my milling bits and a half dozen each of 6x6 inch single sided and double sided blank boards to practice on. I had a nice chat with Fred at North Bay Technical about what I should be buying and he was a font of information.

First off, according to Fred, Adrian's report about having burned up his cutter doing one board is no fluke, but is avoidable. His diagnosis that the cutter was ruined by the fibreglass was spot on according to Fred. Reading through a variety of websites dealing with milling PCBs they all stress that you've got to have the boards VERY flat on your xy plane. One site had plans for a little vacuum table while another guy had a flat steel plate he was using to tack his boards onto using a drop of superglue in the center of the board. He used acetone, which migrated under the board by capillary action, to release his board afterwards.

Fred, however, had a much neater trick. He also designs and sells milling machines. The way he maintains cutter depth is to put his boards on a flexible surface and then has the cutter head equipped with a foot very much like what you see on a sewing machine which contacts the surface of the PCB while you are milling and presses the board slightly down into the flexible surface while maintaining the cutter depth.

Doing it that way gets you something like 2,000-2,500 inches of milling per milling bit.

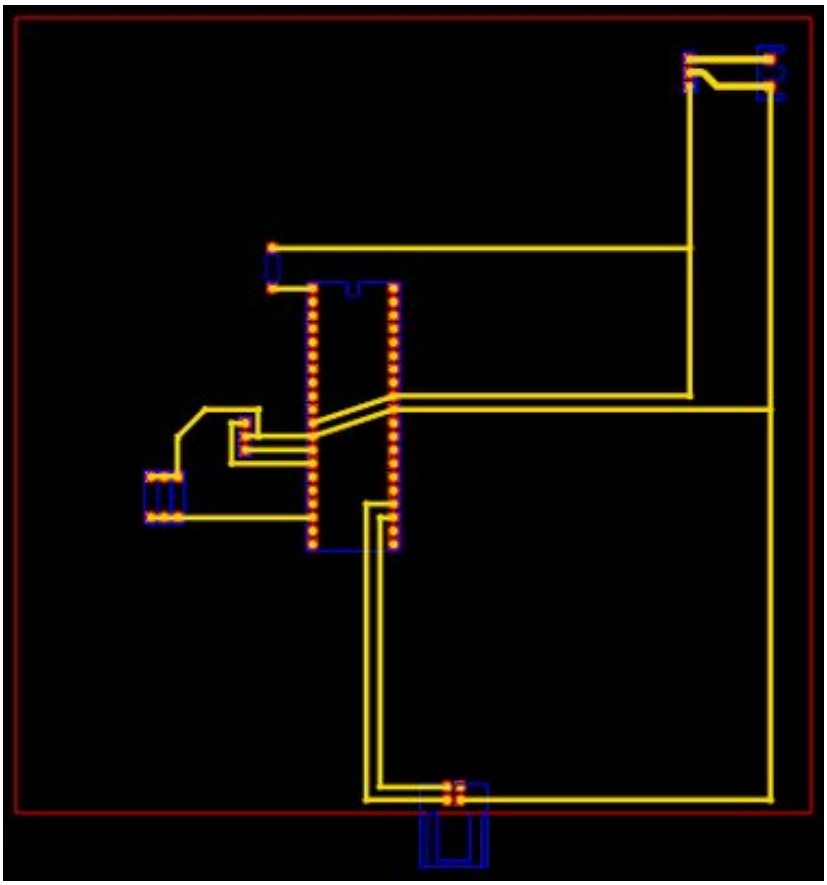
Anyhow, I'll be milling a foot for T2 out of HDPE this weekend.

Life imitates art far more than art imitates Life - Oscar Wilde

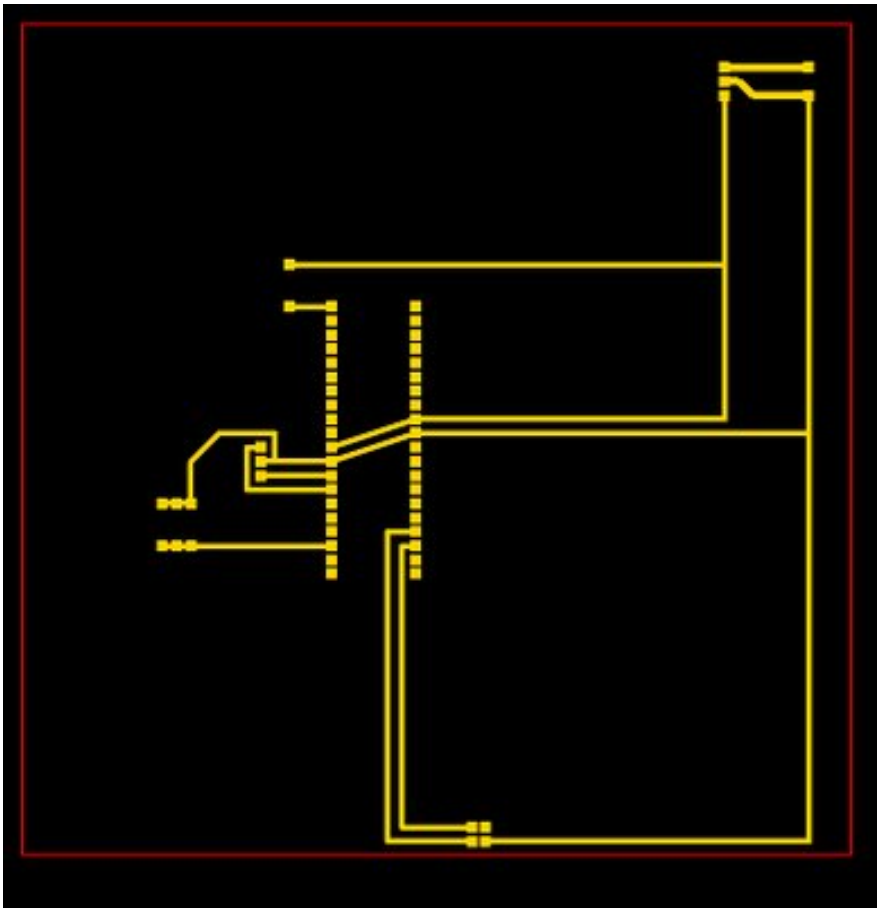
Tuesday, 21st October 2008 by Forrest Higgs

In which the narrator tries to solve a practical problem and finds the solution to be esthetically pleasing as well...

I had a few hours this morning while I was waiting on some information before I could do the next task in my day jog, so I used the time seeing if I could carry my PCB layout app through to doing Voronoi Isolate layouts. I had got as far as doing the layout of a typical USB/18F4550 circuit board.

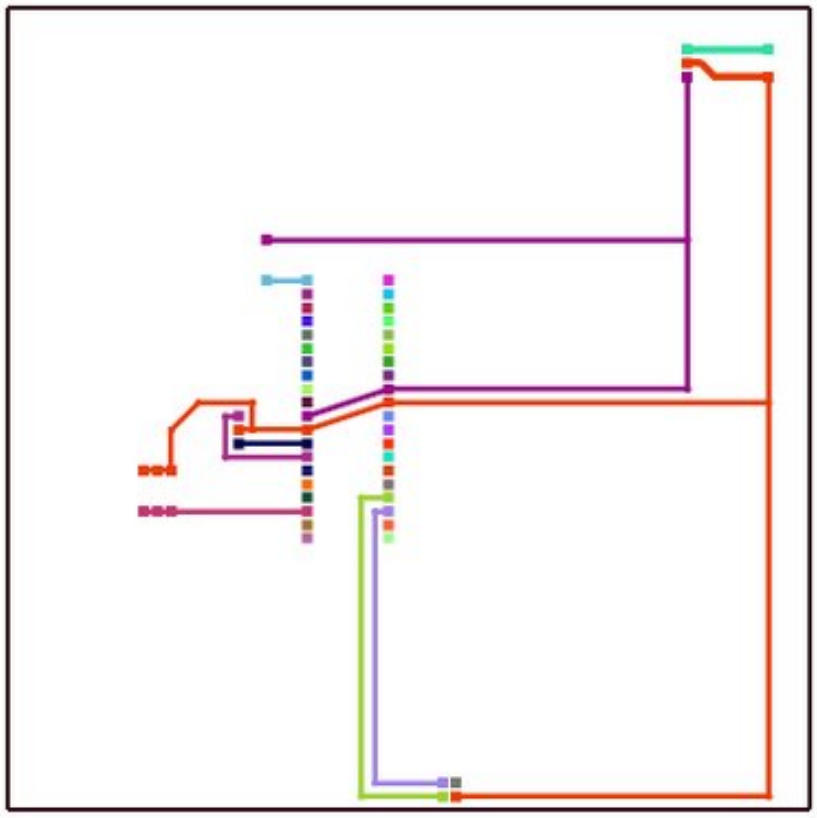


and was able to get that down to the copper traces which would be suitable for developing conventional milling toolpaths, thusly.



All that I reported some days ago.

Having got to that point the next thing I had to do was identify the individual traces so that I could develop the toolpaths around them. Experience with ordinary milling toolpaths made that a simple enough exercise.



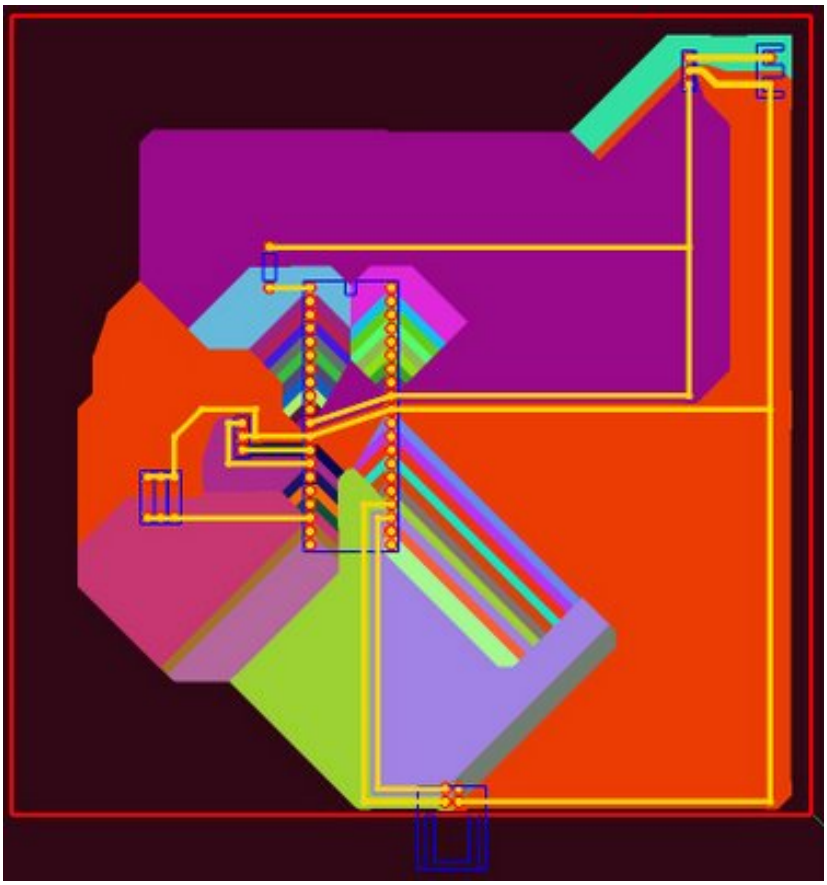
As you can see, I've colour coded the individual traces. I used a random number generator to assign the RGB values for each trace.

It should be mentioned that early on in his career, our own Dr. Adrian Bowyer developed an algorithm for doing Voronoi, also called Dirichlet, tesselations. I'm not sure what the algorithm he developed looked like in that it was published in a journal in 1980 and I wasn't able to get at it on the web. I was going to ask Adrian for a copy of the paper, but hadn't got around to it. This morning, not having the paper in hand, I decided to have a go at writing my own and seeing how it came out.

I use a finite element rather than a geometric approach in all of the Slice and Dice apps that I've written. The PCB app I've written follows the same prejudice. Coming from that prejudice, it seemed to me that Voronoi tesselations could be generated rather simply from a bitmap of the traces, suitably identified by trace, by simply wrapping each trace in successive layers of pixels until the accreting traces ran up against one or more other traces. At that point growth along those boundaries would stop and growth would continue by accretion until the PCB was filled.

Here is what the USB/18F4550 circuit board looked like when you applied that rule.

It may not be completely kosher, but worked fairly well. As abstract art, it wasn't too bad, either.



Here, I've overlaid the original components and traces on top of the Voronoi expansions of the traces so that you can see how things fit on the board.

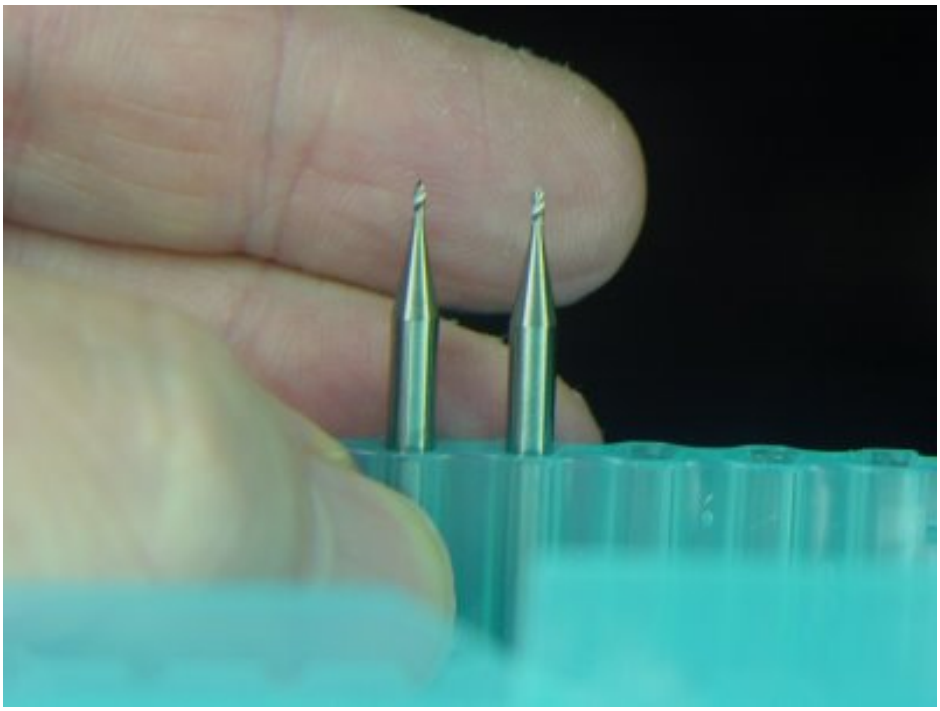
My next task is to generate a drill file consisting of the centres of the little red circles on the layout. Those are where the chip pins fit. That should be trivial.

PCB supplies arrive

Wednesday, 22nd October 2008 by Forrest Higgs

The PCB supplies from North Bay Technical arrived this afternoon...

As usual it is one thing to see the specification for a cutter on a website and quite another to be holding it in your hand.



Or, in my case, looking at it through a rather immense, illuminated magnifying glass so that I can see the cutting surfaces on these things. I took the photo through the magnifying glass or you wouldn't be able to see them, either.

I bought six, single-sided 6x6 inch boards and six, double-sided boards. The quality looks to be very high and the price, very reasonable.

Generating PCB milling toolpaths for Tommelise 2.0

Tuesday, 28th October 2008 by Forrest Higgs

In which your narrator moves from text and pictures to a more or less podcast approach to demonstrating the performance of T2's toolpath generation system for conventional PCB traces ...

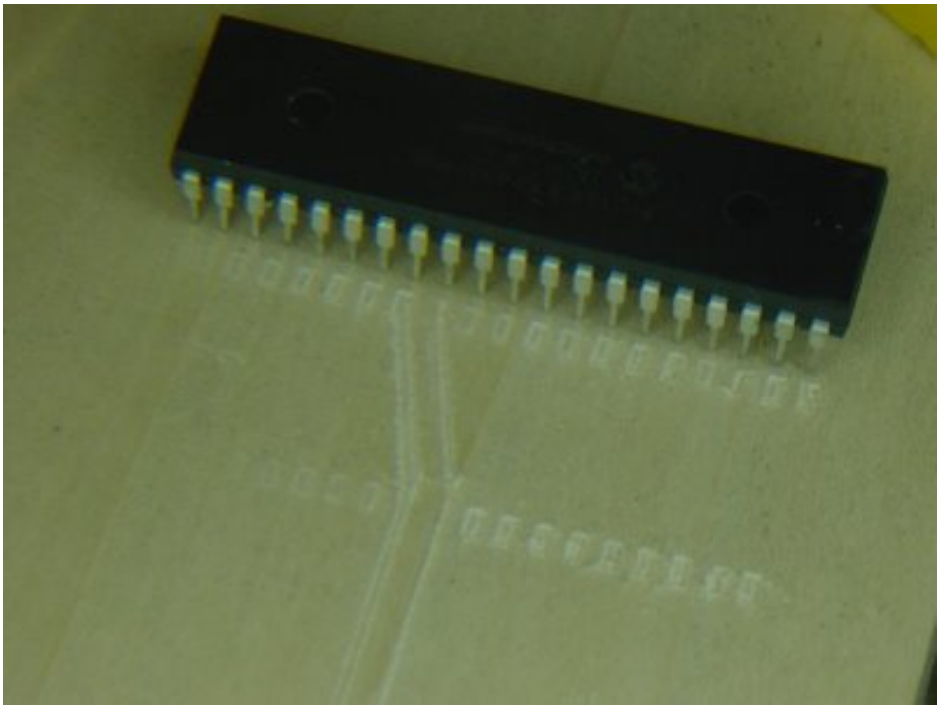
This is a 7-8 minute presentation and will stream about 128 MBytes to your PC. If you do not have broadband, you should not attempt to view this podcast. I am also going to suggest that you make sure that the HD option is "on" and expand the podcast to full screen to achieve the best understanding of what is happening.

Toolpath generation for PCB milling on Tommelise 2.0 from Forrest Higgs on Vimeo

A tale of two rulers

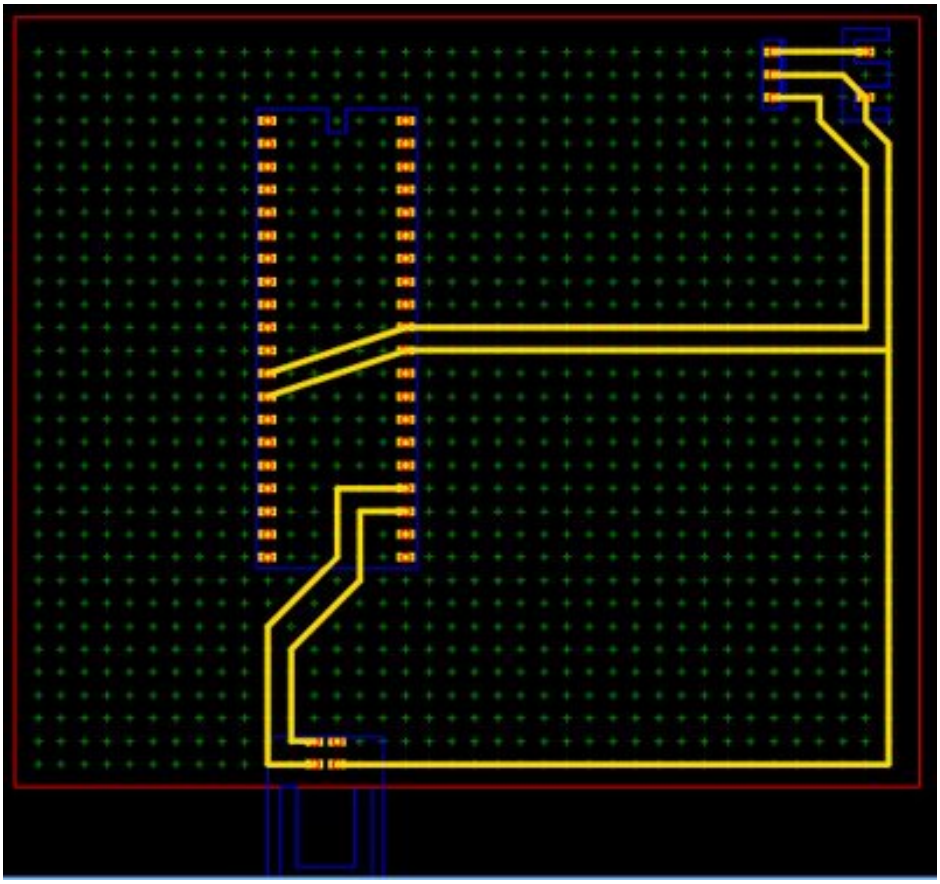
Sunday, 2nd November 2008 by Forrest Higgs

In which your narrator gets mired in the no-man's-land between SI and U.S. customary units...

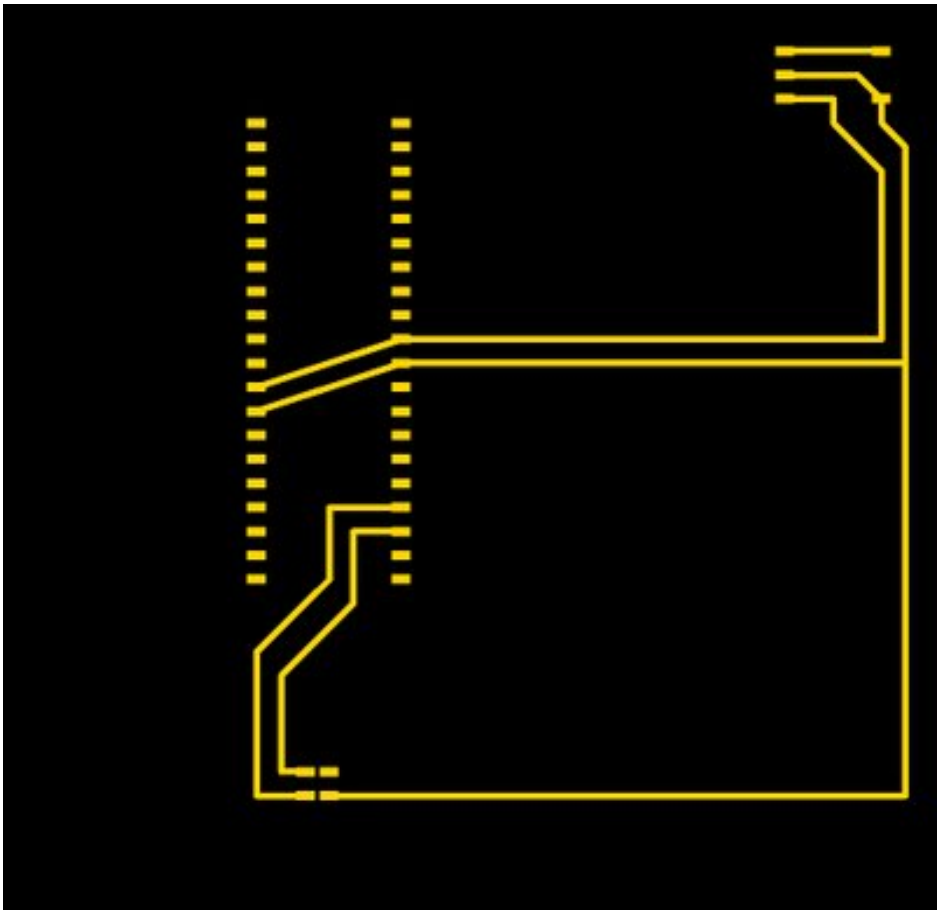


My spare time in the past week has been devoted to sorting out the mismatches between the various bits of PC-side software that I use to run Tommelise 2.0. Today, finally, I felt like I was ready to actually try to cut a PCB design, in a piece of milled poplar, mind, I'm not made of money.

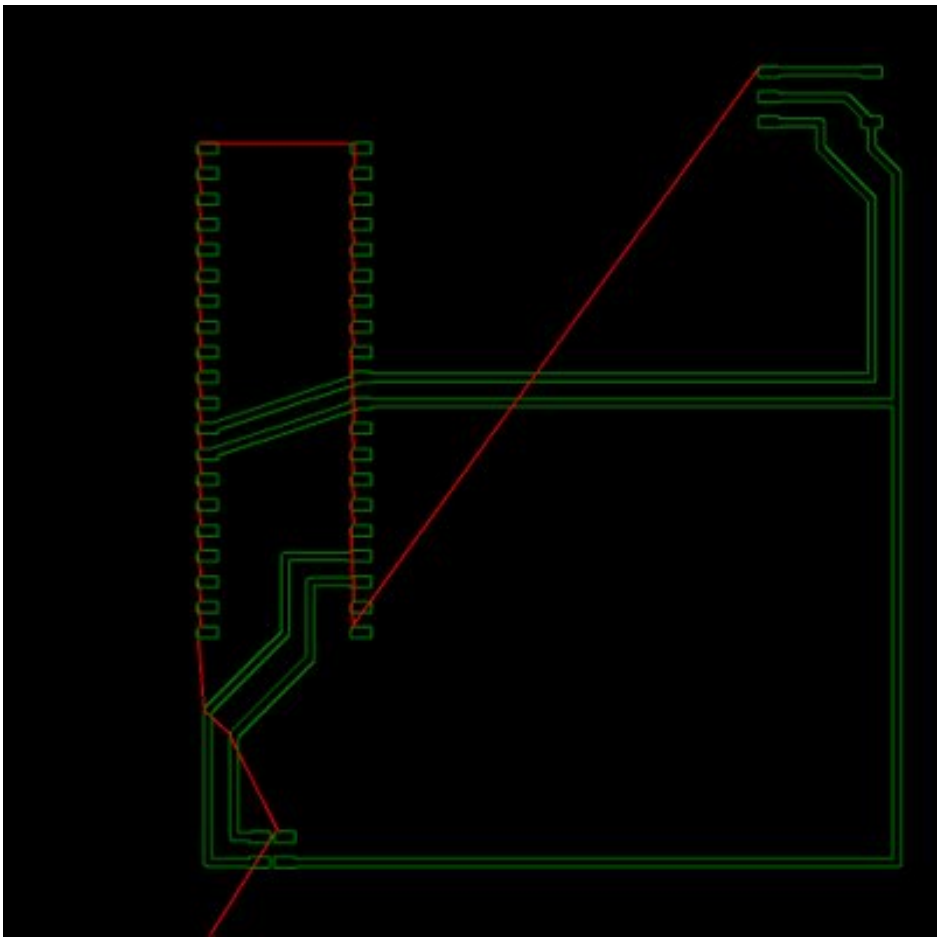
The first problem I faced was designing something that would fit on a piece of poplar. I settled on a 100x85 mm design with just a few connections made. Here is the layout.



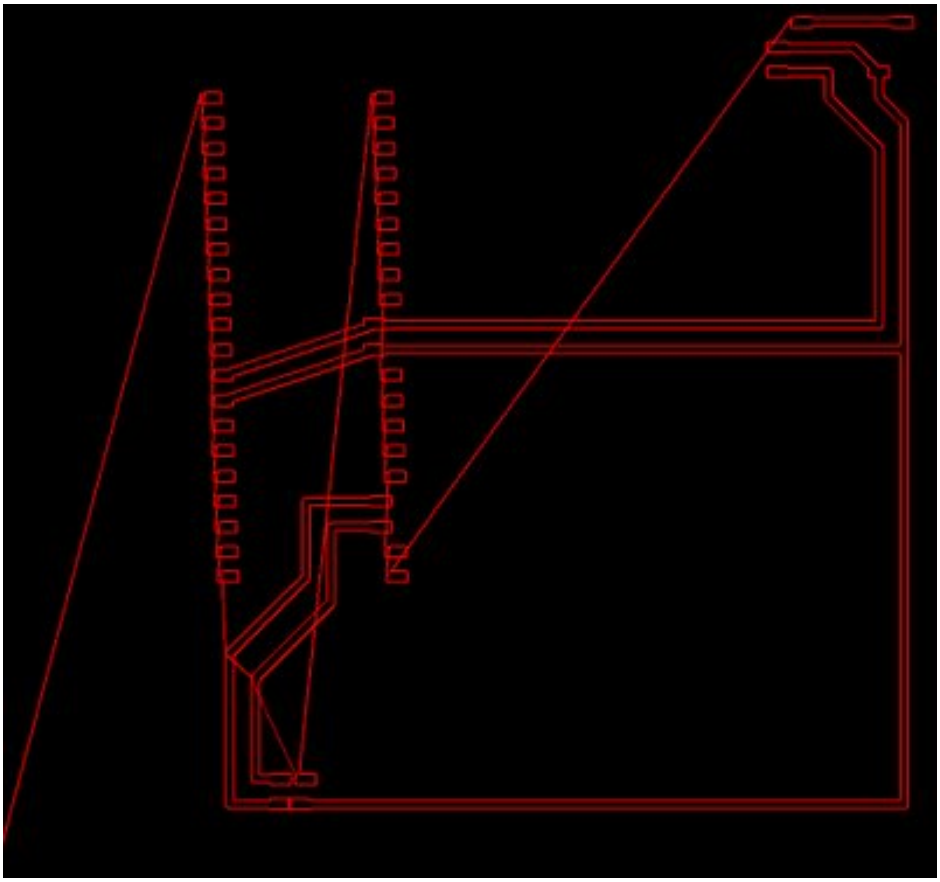
Once I had that, I processed the copper traces.



At that point I did the identification of the individual traces and then the loop tracks around each track. That left me with an XML description of the traces on the board. The module that converts the XML to USB print instructions also redraws the traces as a quality assurance check.

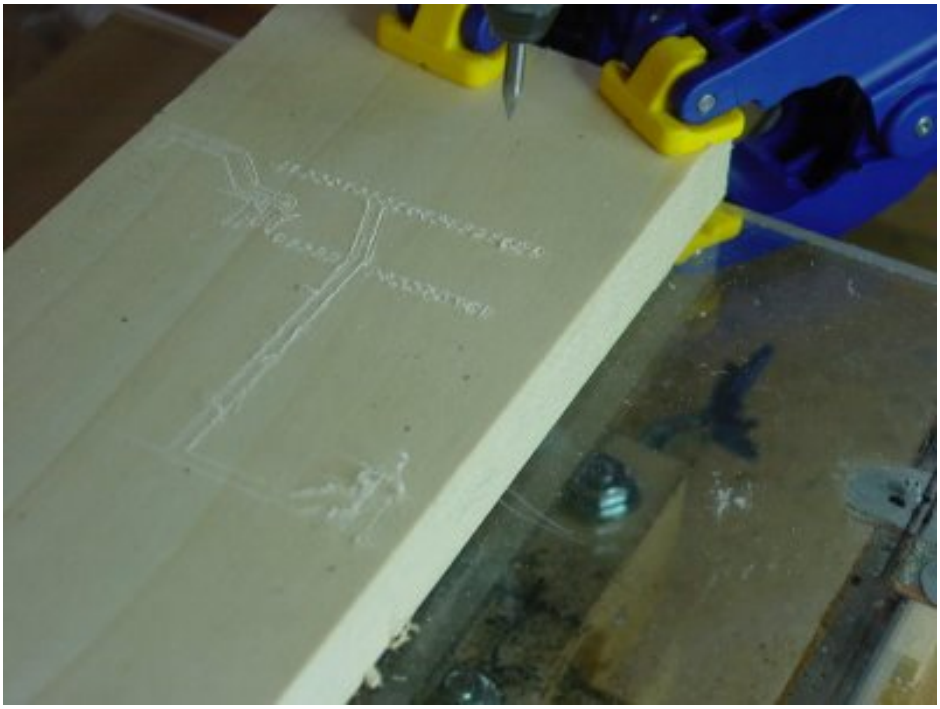


There is a final check before I send the USB format file over to Tommelise 2.0's print buffer. I hadn't used this moduel in a long time and was surprised to see that the USB graphic readout looked creepy.



I've done a lot of code revisions in recent days, so I decided to just go ahead and try to cut the USB file to see if the problem was in the USB formatting or the USB graphics readout code.

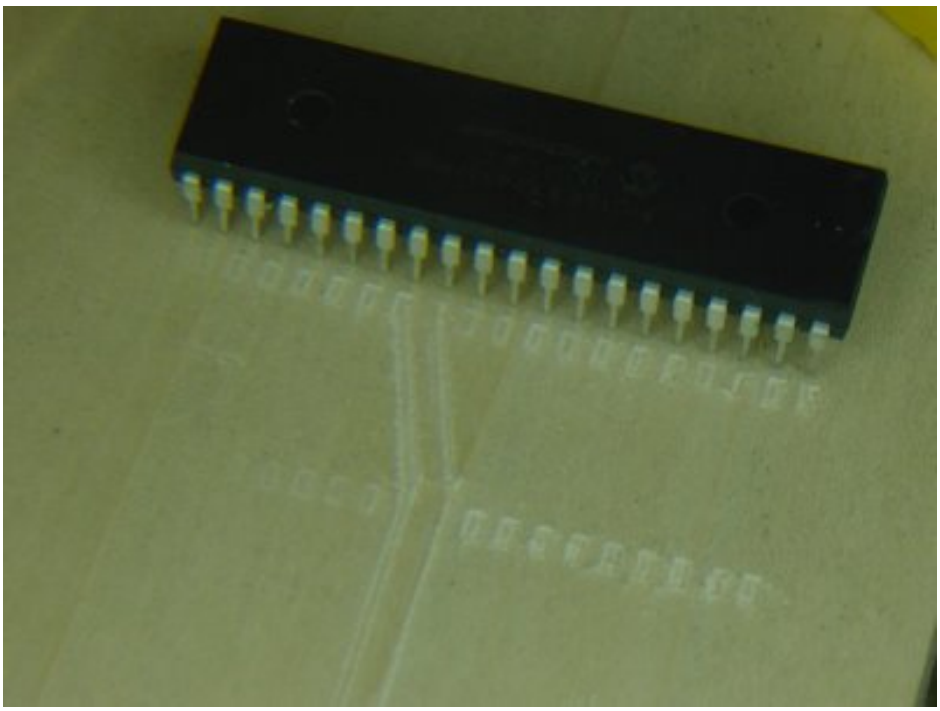
It turned out that the problem is in the graphics readout, because the actual milling of the traces onto poplar looked pretty good. I discovered that the speed setting that I was using for the first test was far too high.



Reducing the milling rate to 25% of what I use for cutting HDPE gave me a much better looking set of pads for the 18F4550.

Here you can see the ground input trace from the two pole screw terminal and the 5 volt output from the voltage regulator.

Once I had the traces milled I laid a 40 pin 18F4550 chip alongside the pads.



The fit was pretty good, except that it didn't fit perfectly.

I'd known that the Haydon linear stepper motors that I was using weren't stepping precisely 0.1 mm/step, but a bit over that. In fact, the lead screw was cut in U.S. customary units rather than SI and was achieving 0.004 inches/step, which works out to 0.1016 mm/step rather than 0.1 mm/step. While the difference doesn't look big, it is significant. The cumulative error after cutting 20 pads amounted to a bit over 0.8 mm. If you look at the picture again, you will notice that the pins are lined up on the right side and out of alignment just a touch under 1 mm at the other.

What that means is that I am going to have to set my grid to 0.004 inches instead of 0.1 mm. That means that it will take 25 steps between one pin centre and the next. Ah the joys of trying to cope with competing systems of measurements. :-D

Chemically grinding ABS scrap

Monday, 24th November 2008 by Forrest Higgs

It appears that acetone can be used to reduce ABS scrap to a fine powder for reprocessing into new filament...

I did a little exploratory work into grinding HDPE scrap with rotary files and rasps some months ago. While I was able to demonstrate that it was indeed feasible to grind HDPE down into a fluffy powder, it involved a considerable expenditure in electricity and created quite a lot of noise.
This last weekend I recalled a conversation that went on at the very beginning of the Reprap project back in July 0f 2005, well before we actually had anything that worked. As a timeline, Adrian created the Mk II extruder, the first practical one, in November of that year. In any case, the conversation got into the notion of dissolving plastics with solvents and the potential for their use in the printing of plastics. Steve DeGroof was participating and in the time that followed I'd managed to attribute the solvents idea to him. A note to Steve this morning set me straight on that matter and enabled me to recover the blog posting that had the conversation.

Most of the solvents notions were put forth by a guy named Mike, who I've not heard from since. Somewhere along the way he'd got the idea that you could dissolve HDPE in acetone. I have an affinity for printing with HDPE, probably a vain hope, but one I nurture all the same. This morning I had a few moments so I purchased a pint of acetone and made an attempt to dissolve some HDPE scrap.

Sadly, after being submerged for three hours in pure acetone, the scrap came out in pretty much the same shape it went in.

I almost wrote off the experiment as a total failure, but decided to give ABS a try as well. Six inches of 3 mm ABS works out to about 1 cm^3. I clipped the filament into 1 cm lengths and put them in a quarter cup of pure acetone.

To my delight, the acetone turned milky with dissolved ABS within a few seconds. After stirring and leaving the mix for about half an hour I saw that about half of the ABS filament had dissolved. Another half of an hour found the mix in about the same state, so I presumed that some sort of equilibrium between solid ABS and ABS in solution had been reached.
Mike had suggested that...

Another way would be to add an excess of acetone, make a slurry and then drip the slurry into water to form the powder. the finer the drops the finer the powder.

While he was dead wrong about acetone dissolving HDPE the method seemed to me worth trying with the dissolved ABS. I poured the dissolved ABS/acetone mix into a pyrex tray containing about four cups of water.

A froth of finely divided ABS formed on top of the water. I was a bit nonplused about this since ABS is supposed to be more dense than water. I got a bit of the undissolved ABS out of the measuring cup and tossed it in to the water bath and it floated, too. Mind, not all of the powder floated. Some of it appeared to be neutrally bouyant. Most, however, collected at the top of the water bath. This did not appear to be a surface tension issue.
Just to be certain, I cut a small piece of ABS off of my roll and tossed it into the water bath. After poking it with an old spoon to break the surface tension, it promptly sank. I'm not sure what to

make of all of that.

The undisolved residue of ABS quickly dried into a few lumps and a thin, transparent film.

I carefully collected the floating ABS in the waterbath with a ceramic spoon and ran it through a coffee filter. We appeared to have caked, powdered ABS.

A few moments with a hot air paint stripper on its lowest setting yielded an ABS powder with the consistency of fine cake flour.

It appears that dissolving scrap ABS in acetone and then separating it with water yields a very high quality ABS powder that could easily be converted back into filament or, for the more optimistic, used directly for printing.

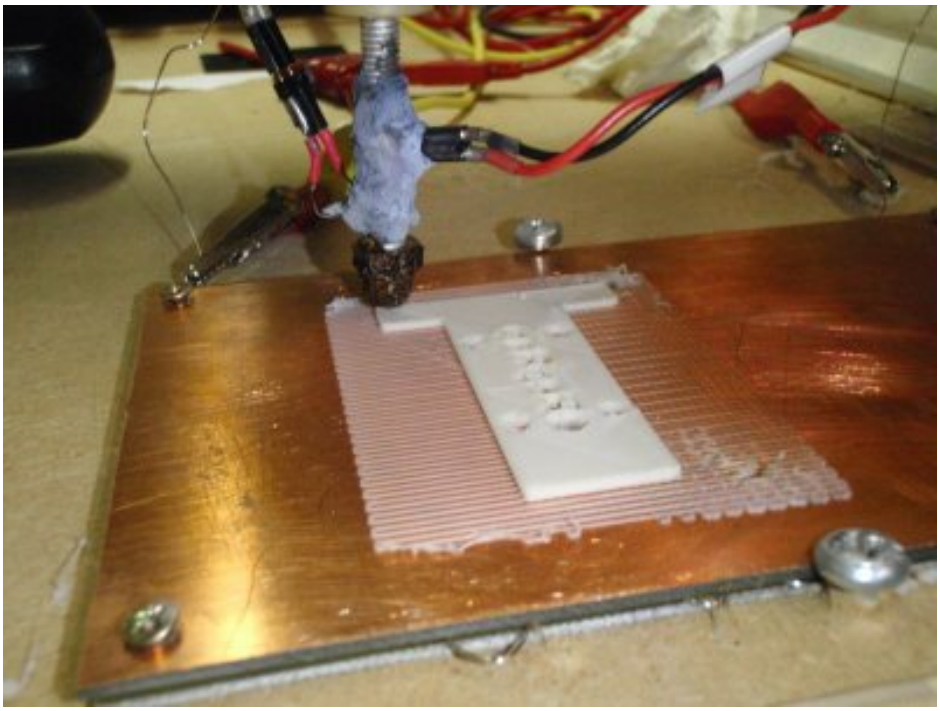
I did, however, a few back-of-the-envelope calculations. It appears that I could dissolve something like 25-40 cm^3 of ABS scrap in a gallon of acetone. Given that I can buy a gallon of acetone at a retail price of about $15.00, it seems obvious to me that I must have some method of recovering the acetone in order for the method to be economic. Fortunately, acetone boils at 52 C, so simply boiling it out of the water bath and collecting it in a condenser should be no problem. It's latent heat of vapourization is about a quarter of water's so the energy cost of recovery should not be excessive.

Heating the print surface: doing the numbers

Thursday, 27th November 2008 by Forrest Higgs

Scaling up Metalab's heated printing surface...

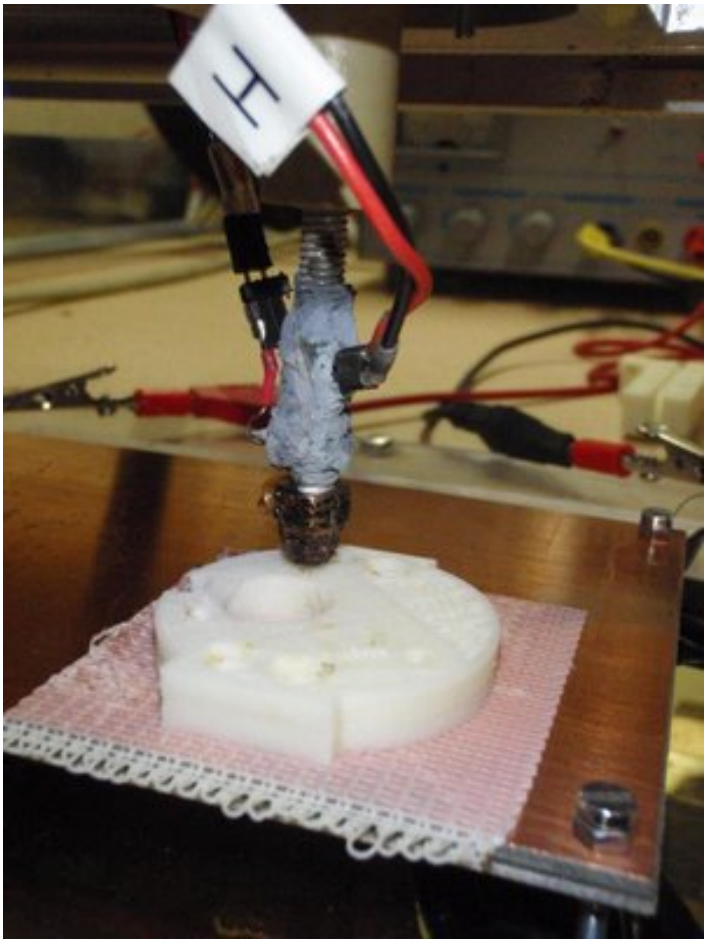
Recently, Marius, at Metalab in Austria, reported that printing onto a heated surface solved the problem of getting plastic to stick AND greatly reduced warping of the printed object. As you might imagine this report drew a considerable amount of interest, since warping has been one of the most serious technical challenges faced by the Reprap project.



Metalab simply placed a layer of what appears to be mineral fiber cloth on top of their particle board working surface, placed what appears to be heavy gage nichrome wire on top of that and then placed a single sided raw printed circuit board, copper side up, on top of that. From the placement of the power clips it appears that they attempted to heat the whole surface.

My first problem was to determine the area that they were trying to heat. The configuration looked familiar, and in this picture from Marius' blog he was attempting to print a standard old-style Mk II filament guide. I prototype my controller boards, as a rule on Euroboard scaled stripboards, those being easy to acquire where I live. I pulled a spare Euroboard and a filament guide from my stocks and quickly determined that Metalab was also using a Euroboard scaled printed circuit board.

What that means is that Metalab was heating a 10x16 cm surface. At the end of Marius' blog entry he demonstrates a second pass at the design problem.



Here you can see that they've simply sandwiched the nichrome between the fiberglass sides of

two Euroboards. It's an elegant solution at that scale if not particularly energy efficient or scalable. What is important, however, was that for this design they gave parameters for the nichrome heating element that they used successfully.
The base plate consists of two standard epoxy/copper PCBs with a ca. 9 Ohm nichrome wire sandwiched between them and fed with ca. 15 volts. This resulted in a base plate temperature of around 120 degrees celsius.
Applying Ohm's Law one quickly finds that this board draws 1.67 amps at 15 volts or roughly 25 Watts of electricity to heat it.

Doing a quick scaling of that load to Tommelise 2.0's working surface, 12x12 inches or ~930 cm^2, the scaling factor works out to...
930/(10x16) = 5.8

That means that to heat Tommelise's full print surface I am looking at...

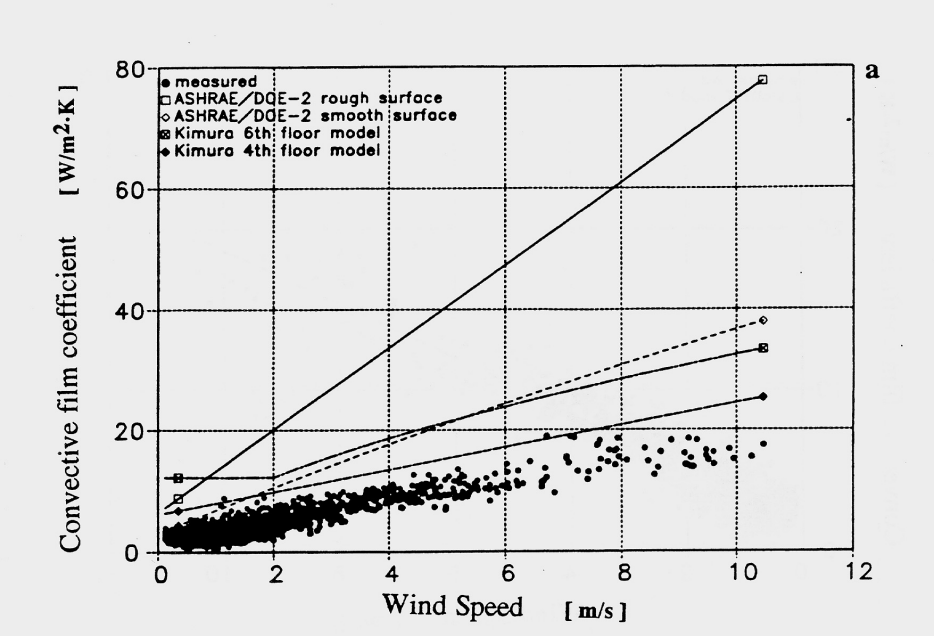
(25 Watts)(5.8) = 145 Watts.

Given that Tommelise 2.0 only draws about 20 Watts max while operating, that is a rather huge additional amount of energy to use.

Because of that, I decided to do some quick calculations to see if I could see a way clear to get a better handle on this problem. To begin with, Metalab is radiating off of a clean copper surface. That gives you an emmisivity of about 0.1. Assuming an ambient temperature of about 20 C, a heated surface of 120 C and applying that information into an expression of a formula for radiant exchange which assumes an ambient emissivity of 1.0, you plug numbers into an expression like this...

$$\dot{q}_{\text{net 1 to 2}} = \frac{\sigma(T_1^4 - T_2^4)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}.$$

You get an energy requirement due to outgoing radiation of about 4.4 Watts. Please note that this is for one radiating surface, not two as Metalab has., so the outgoing radiation from a Metalab configuration would be about 9-10 watts. This is over a magnitude less than what Metalab reports. At a similar temperature differential one would expect natural convection losses in a still air environment



M. Yazdanian and J. H. Klems, Measurement of the Exterior Convective Film Coefficient for Windows in Low-Rise Buildings, ASHRAE transactions, v100, Pt. 1, pp 1087-1096, 1994.
to be running something on the order of 2-4 Watts K/m^2. For our work surface this would be on the order of 10-20 Watts/side. That would mean that we would be looking at a total energy demand of around 30-50 Watts for a Metalab configuration in a still air situation. Kick the convection level up to, say 4 meters/second as we do when we direct an air stream at the top of the printed surface and your convection contribution would increase by a factor of five on the top side of the heated surface. With such a configuration you'd be looking at...

9 Watts (radiation) + 10-20 watts (bottom side convection) + 50-100 Watts (top side convection) = ~70-130 Watts

Metalab chose copper as their print surface. Providentially copper has a very low emissivity (~0.1). If they had used anodized aluminum instead, they would have been looking at a radiation fraction for a single side of around 70 Watts. Similarly, if they had printed a raft across the whole top side they could expect to be an effective emissivity of the top of the raft of 0.95 which would have raised the radiation contribution of that side to just over 80 Watts.

Obviously, we can greatly improve on the performance of the Metalab design, which was purely exploratory in nature. Insulating the bottom half of the printing surface would effectively reduce the energy demand by a bit less than half.

With respect to materials, while the printed circuit boards have the rigidity to handle surface deflection on a Euroboard scale, I doubt that they can be reasonably expected to do so at wider spans given the depth of the boards. Cross-bracing under the printed surface assembly should, however, put that right. Another question remains as to how the fiberglass printed circuit boards would handle 100 degree heating and cooling cycles over a long period.

The one thing to remember is that while the heated printing surface offers us a technical fix to the warping problem that is unlike anything developed heretofore, it has substantial energy requirements. Taking such an option would pretty much put paid to the notion of Reprap machines operating off batteries in a third-world environment.

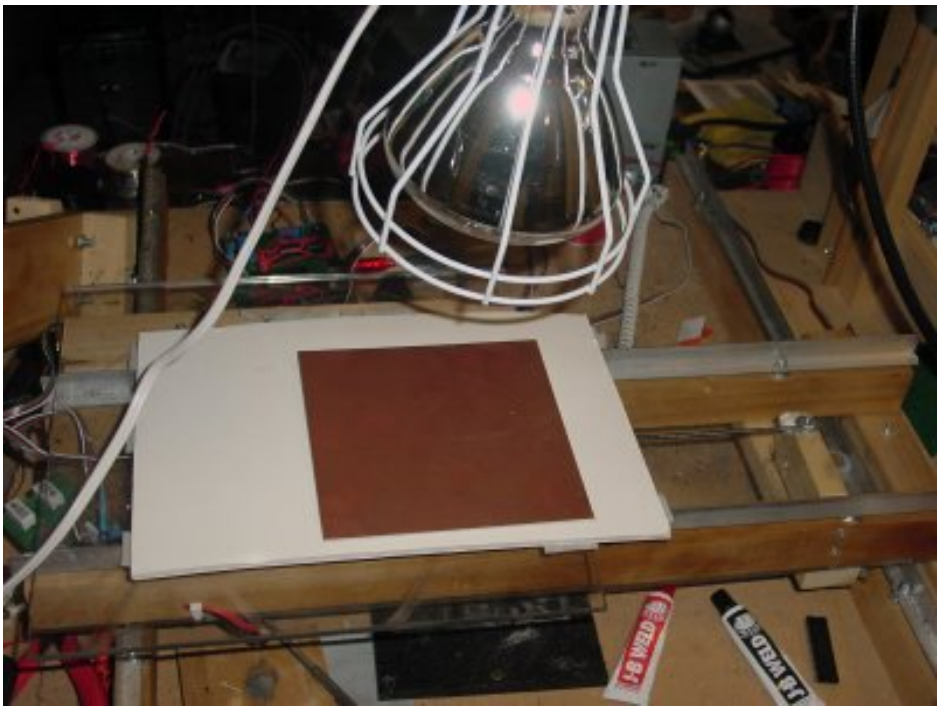
Another approach to heating the print surface

Sunday, 30th November 2008 by Forrest Higgs

In which the adventures of your narrator in his quest to create a heated print surface are related...

The more I thought about the way that Metalab was heating their printing surface, the more problems I saw with the approach. The most important one, in my opinion, was the amperage requirement that heating a 12x12 inch working surface with 12v electricity would entail. Roughly, we are looking at 150 Watts to heat a 12x12 inch surface. That works out to about 12-13 amps of 12v power.

Yesterday, it occurred to me that one might well achieve more or less the same effect by using a conventional IR heating bulb instead of a nichrome heater between two heat resistant layers with a copper printing surface. This morning I acquired a 150 Watt IR bulb from my hardware stockist and decided to see what was implied by using this approach.



I bought a standard cage holder for the rather fragile IR bulb and used the clamp to attach it to Tommelise's gantry with no problem at all.
My first experiment was to simply turn it on and focus it on a 6x6 inch single-sided printed surface board that I had in stock. I put down a piece of foamboard to keep the xy printing table from heating up and then laid the printed circuit board copper side up on top of it.

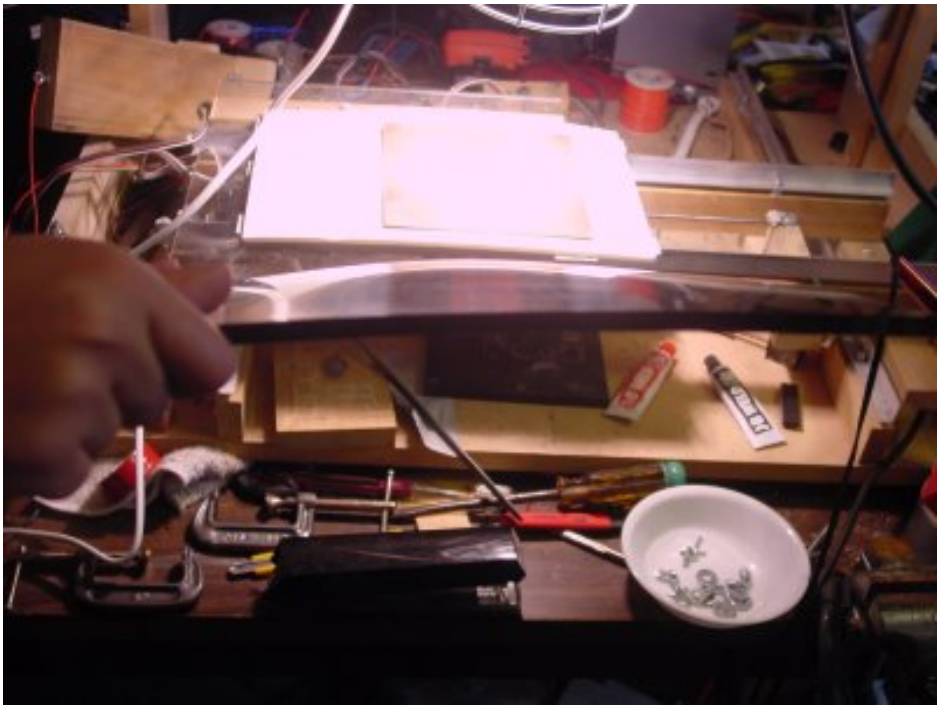
As I expected, the emissivity of the copper was very low. While the air temperature in my flat was running 20-21 C I was unable to get the copper to reach much more than 27 C measured from the top. . The IR radiation was not only reflecting off of the copper, but it was doing so in a highly

specular manner as well. If I aligned my IR thermometer so that it saw the reflection of the bulb itself, I saw temperatures of 100-110 C. If I changed the alignment a few degrees, the temperature dropped down to 27 C again.

An interesting aspect which, in retrospect was obvious, was that what the copper side of the printed circuit board was seeing, vis a vis radiation, was the surface of the IR bulb, not the filament or anything like that. Thus, the surface of the bulb pretty much stabilized at about 110-120 and the center of the printed circuit board stabilized at about 100-105. It is worth noting that when I rapidly flipped the board and measured the temperature of the fiberglass-expoxy backside the temperature at no time exceeded 32 C.
It seems that Metalab was able to achieve their temperatures pretty much as a result of the low emissivity of their copper-sided pcb's.

At that point, I recalled that several developers had talked about printing onto HDPE sheets. I decided to try that in that I had a lot of quarter inch 12x12 inch sheets of black HDPE. That exercise gave me a result that made sense as well, in retrospect at least.

Focussing the IR lamp on the HDPE sheet resulted in a slow warming of the center of the sheet. After about 20 minutes the center of the sheet was measuring 105 C while the edges were running about 40 C. The unexpected result was that the differential heating created a substantially raised dome of HDPE in the middle of the sheet. This photo was taken a few minutes after the sheet was removed from the xy print table.



The degree of warping is quite clear in the photo. While the temperature on the top side ran 105 C

the temperature of the bottom side read 80 C. Cooling off the sheet with tap water returned the sheet to its original state in a matter of moment.

Several things can be taken from the exercises.

- the surface on which we print needs to be insulated underneath
- the surface needs to have very good conductivity in order for the heat to spread laterally
- a high emissivity would be nice as well as a diffuse, rather than specular, response to directed IR radiation.

Another thing that might be useful is to not use IR bulbs but rather use a radiant coil, so that a much higher radiant temperature could be achieved. The glass face of the IR bulbs kept the temperature from getting much above 110 C for the wattage used.

Building a printable stepper for a next generation Tommelise

Friday, 5th December 2008 by Forrest Higgs

In which your narrator gets down to cases in the quest to design and build a more or less printable stepper motor...

Back in July I started a project to build a tin can linear stepper motor for my next generation Reprap machine, Tommelise 3.0. Before I went on to other projects I established that it ought to be technically feasable. I had two motives for wanting to do this

- to increase the percentage of a Reprap machine that it could build
- to get around the pig-headedness of the Chinese engineers who were refusing to sell me lead screws for their very inexpensive (~$2.50) linear stepper motors in the lengths I needed

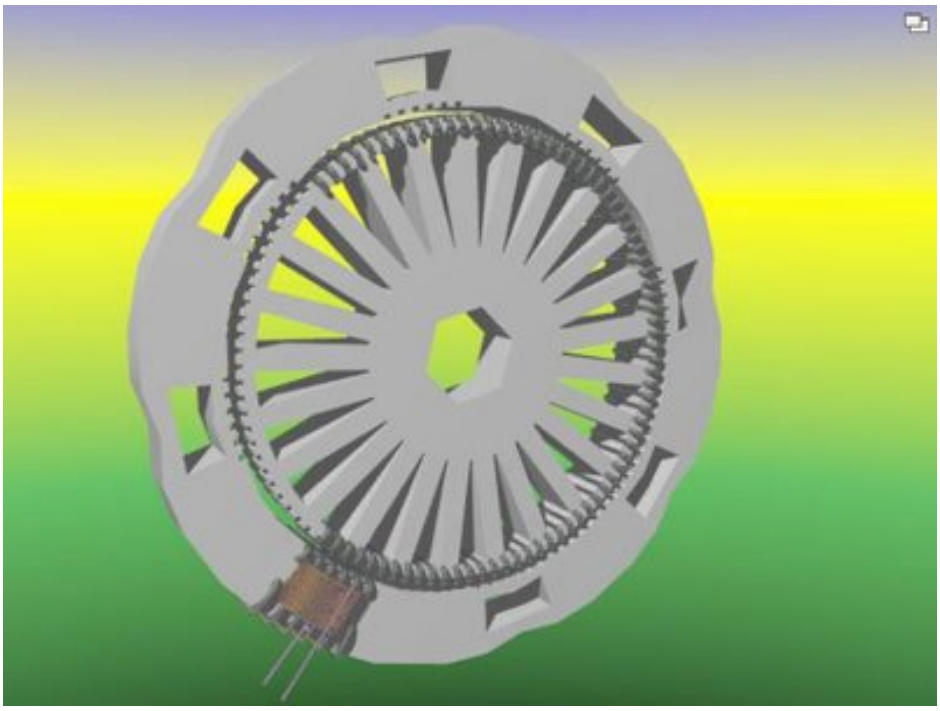
I bought the original linear stepper motors for Tommelise 2.0 from Haydon. While their product massively simplifies the design and construction of accurate positioning robots, their pricing structure obviously factors that ease in to the point where a Haydon linear stepper and lead screw costs about $80/axis. The Chinese were much more attractive, pricewise. Unfortunately, they had had some customer complaints about the tendency of their small diameter lead screws to bend under load. While I tried to explain that I wasn't using their linear steppers to anything like their design loads, they still flatly refused to sell me the lead screws in the 12-18 inch lengths that I needed.

Having lived in China and liking the Chinese people immensely, I was altogether shocked and finally angered at this situation. In the months since July this annoyance began to inform my efforts at building a printible tin can linear stepper. The big problem with making a tin can linear stepper lies in the rotor. If you go for a one for one match you have to have an aluminum spindle and coat it with either a polymer or epoxy layer with an admixture of magnetic media. While this isn't technically a really difficult problem, you then have to create a special gaussing device to create the alternating magnetic stripes of opposite polarity on the rotor.

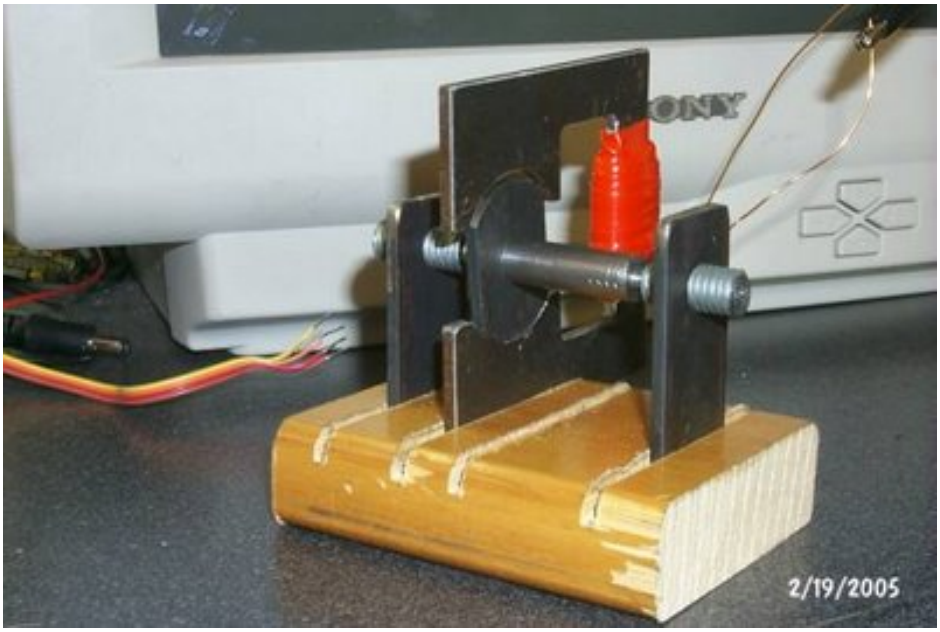
I came up with an alternative design which used thin steel strips and a ring magnet to simplify this process. There were two problems with my solution. First, the retail price of the ring magnets was on the order of several dollars. Second, and more importantly in terms of my annoyance with Chinese suppliers, was the fact that the Chinese these days pretty much make ALL of the rare earth magnets in the world. This situation has got to the point that the GPS guided bombs that the US uses to pound terrorists use Chinese magnets. Finally, I discovered that the nickle plating that the Chinese put on their rare earth magnets perishes after a few years leaving you with a really nasty mess where your magnet used to be.

All this kept me from continuing the effort I'd started to build a linear tin can stepper. A few days ago, however, I ran across what began to look like a much more suitable alternative. While, I'd

read about variable reluctance steppers in a variety of stepper references, I hadn't really run across one in practice. Recently, however, I was nosing around in the Reprap forums "Things to Print" section, a place that I rarely visit. Back in May, Tim Atwood came up with a printable variable reluctance stepper, although he doesn't quite call it that.

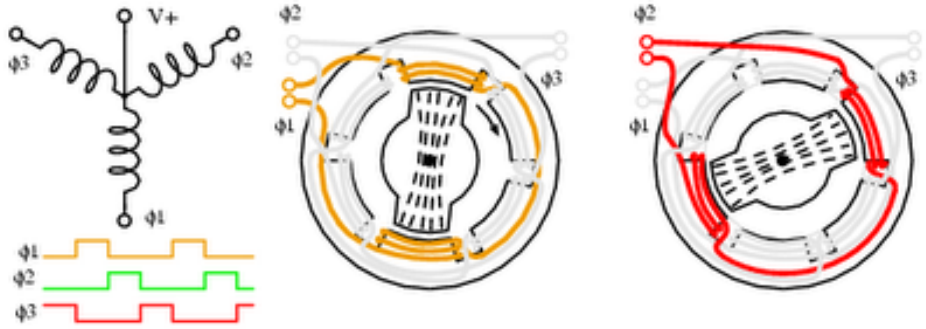


While Tim was aiming at producing a stepper with a very small step angle, the Tommelise approach needed nothing this complicated. What really got me moving, however, what this little project.

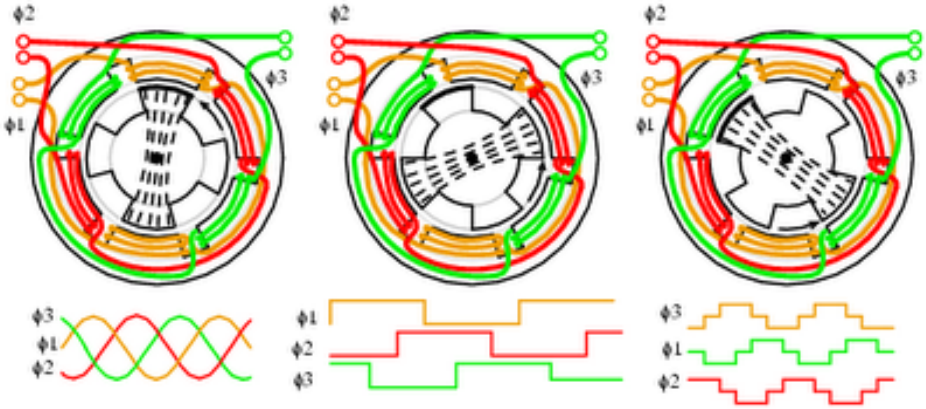


The motor you see is a one phase variable reluctance motor. As you can see in the video it is a bit of a pig to get started and keep going. Watching its inventor work with it, however, I was

reminded of an illustration I'd seen some time ago.



While this design had a 60 degree angle step, I could get it to give me 30 degrees by either half-stepping it or using a cruciform configuration rotor rather than a simple bar rotor.



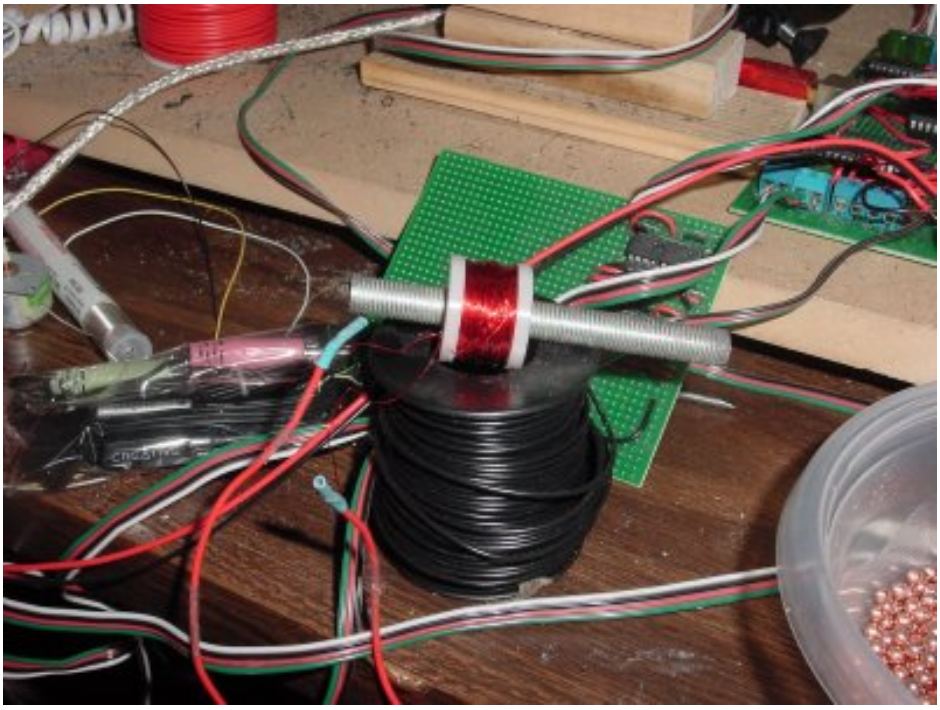
Further, if I used the general design of the single phase variable reluctance project that twigged my imagination, I could reduce the complexity of the fabrication dramatically.

With all that in mind, I dug out Rick Hoadley's Excel app for designing electromagnetic coils and had a go at designing a coil for a first cut at such a stepper out of stuff that I had at hand.

Electromagnet Design										
	ALL DIMENSIONS ARE IN INCHES									
Coil Data				for max gauss						
core radius	r1	0.25	inches	0.25						
core diameter	B	0.5		0.5						
outer radius	r2	0.5		0.75	r2=3*r1					
coil diameter	D	1		1.5						
coil thickness	N	0.25		0.5						
core length	L	0.5		1	L=4*r1					
Wire Data	**awg**	**dia**	**nom dia**	**turns/in**	**turns/sq in**	**Ohms/1000'**	**Current**	**Fusing**	**lb/1000'**
wire size, enamel	28	0.01264	0.01360	65.44	4282.0	64.89	0.16	14.50	0.48
number of turns/layer	t	32.7		36.8	max				
number of layers	y	16.4		18.4	max				
total number of turns	T	535.3	turns	675.8	max				
total length	x	1180.2	inches	98.4	feet	0.05	lb		
total resistance	R	6.382	Ohms						
applied Vdc	V	5.4	Volts						
rated current	I	0.85	Amps		I^2*R	4.6	Watts		
					surface area	1.57	sq in		
amp*turns		453			dissipation	2.91	W/sq in	0.5 ideal	
Gauss at center	G1	252	Gauss	(center of air core)					
	G2	179	Gauss	(center of edge of air core)					
	G3	74	Gauss	(inches from RH edge of air core)					
	at	0.25	inches from RH edge						
Inductance with air core	L	3.7	mH			Coil			

This was a nice little coil. I got considerably better gauss ratings than my linear steppers gave me and kept the amperage down to something managable. The only worry I had was that I was generating 2.91 watts/square inch of heat to dissipate while Rick suggested that 0.5 was ideal. I had no idea what "ideal" meant when Rick used the term.

So... I built the coil. The spool for it was made from two nylon sleves epoxied together.

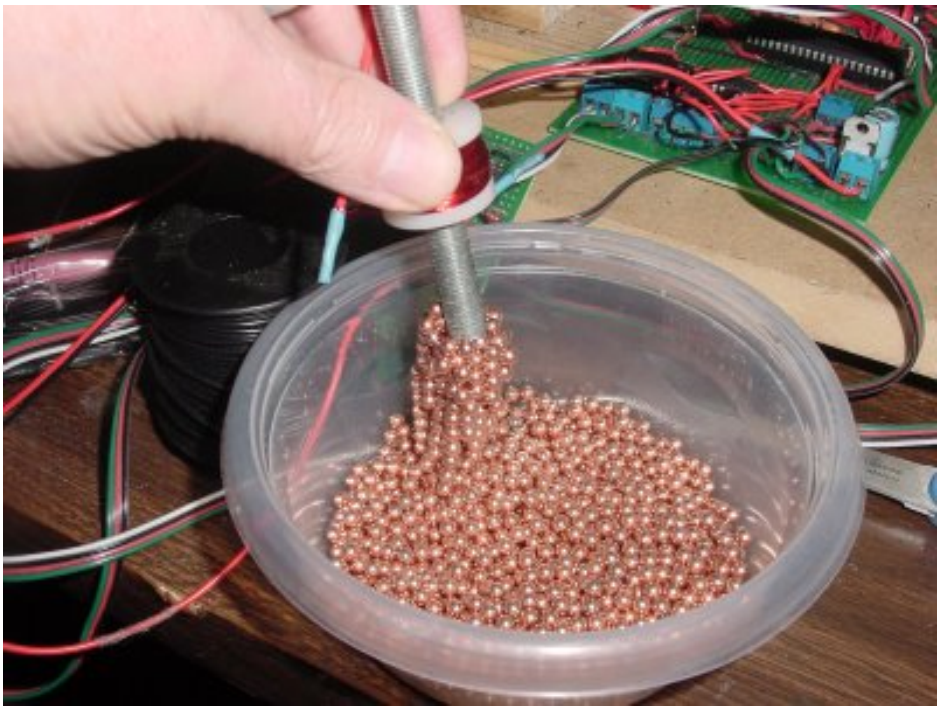


To give you an idea of scale, that's a piece of 3/8ths inch threaded rod running through it. I pulled 5.4 and 12.2 volt power off of my ATX power supply. When I ran it at 12.2 v, it got quite hot within

a few moments. I monitored the coil's surface temperature and shut it down when it nosed over 100 C.

I then applied 5.4 volt power. At that level the coil stabilised at about 71-72 C, hot, but not outrageously so. Apparently Rick's "ideal" dissipation rate heats the coil to only a few degrees above ambient.

Rick also mentioned another little game you could play with your electromagnet. The game was to see how many steel BBs (4.5 mm copper washed steel pellets used in American air guns) your electromagnet could pick up. Although Rick didn't mention it, it struck me that the number of BBs a magnet could pick up may be directly related to it's gauss rating. I had a go with this.



The only trick to this experiment is that to lift the maximum number of BBs you have to be very careful. Also, if you lose power before you get your BBs from the dish to another container things can get a little messy. In fact, though, I discovered that the pickup power of the electromagnet related to the gauss level predicted by Rick's Excel sheet such that...

1 gauss = 1.8 BBs

One end of my electromagnet would pick up 42 grams of BBs at 5.4 v.
I'm quite excited about going forward with my variable reluctance stepper motor project. Freescale described this class of motor in these terms.

The reluctance motor, commonly referred to as either a variable or switch reluctance (VR or SR), offers the simplest, lowest cost motor available to date. This motor consists of a shaped, highly permeable material for the rotor and two- or more-phase windings in the stator. The shape of the

rotor concentrates magnetic flux lines into "poles" on the rotor which then interact with the rotating field being developed in the stator windings. The VR and SR motors must be electronically controlled to start turning and to produce the torque needed to continue rotation. Torque pulsations generated by this design tend to produce undesirable audible noise, leaving much research still to be performed. The potential for its low-cost application into many areas, however, makes it an extremely desirable motor to use in a broad range of products.

This looks promising.

Chemically grinding ABS scrap: part 2

Thursday, 11th December 2008 by Forrest Higgs

In which your narrator continues his efforts to get something for Adrian to put in his granule extruder...

This time I prepared 5 cm^3 of ABS from 3 mm filament by cutting it into 4-5 mm bits

To this I added 20 cm^3 of pure acetone at room temperature (~18-20 degrees C).

After about an hour with periodic stirring to keep the softening bits from sticking to the bottom of the container about 20-30% of the ABS was dissolved.

At that point I put the jar in a water bath 43 degrees C. (~110 F). Raising the solvent temperature allowed the rest of the ABS to dissolve within another 20 minutes. Thus, at this point, I know that I can dissolve ABS in pure acetone in volumetric proportions of 1:4. I expect that it would be possible to improve on this. The limiting factor seems to be the need to wet the pieces of ABS with acetone more than anything.

When I decanted the ABS dissolved in acetone in the previous experiments I skimmed the floating precepitate off the water and was able to filter it quite easily with an ordinary coffee filter. Unfortunately, I soon discovered that much and possibly most of the precipitate was heavier than water and settled, over a period of days to the bottom of the water.

As a result of that, this time I simply filtered the whole water/acetone mix through the coffee filter so as to get all of the ABS precipitate instead of just the lighter fraction. While the light fraction of ABS did not jam the coffee filter, apparently the heavier fraction was of a fine enough grain size that it did and I had extreme difficulty using the filter to separate water and ABS precipitate.

Cleaning the filter several times and changing it once did allow me to recover all of the precipitate eventually. I know now, however, that coffee filters aren't going to be a practical way of recovering the ABS. I'm beginning to think that some minaturised variation on a spray drier is going to be needed in which we can skip the need to precipitate the ABS in water.

Upgrading T2's z-axis

Saturday, 13th December 2008 by Forrest Higgs

Tommelise 2.0 gains a more rigid z-axis...

New Tommelise 2.0 z-axis uploaded from Forrest Higgs on Vimeo.

While the poplar and aluminum extrusion z-axis that I originally built for T2 was adequate for the sorts of lateral loads generated by ordinary extrusion printing and milling of such materials as HDPE, it proved too floppy for the more exacting requirements of milling thin steel sheet and printed circuit boards.

The original z-axis used a poplar z-axis work table with aluminum U-profile extrusions braised onto L-profile extrusions for guides. The aluminim braising proved both difficult to do and fragile, as well. Further, the friction load between the aluminum guides and the poplar z-axis work table proved to be excessive. Finally, the 12" poplar work table itself proved to weigh over a pound.

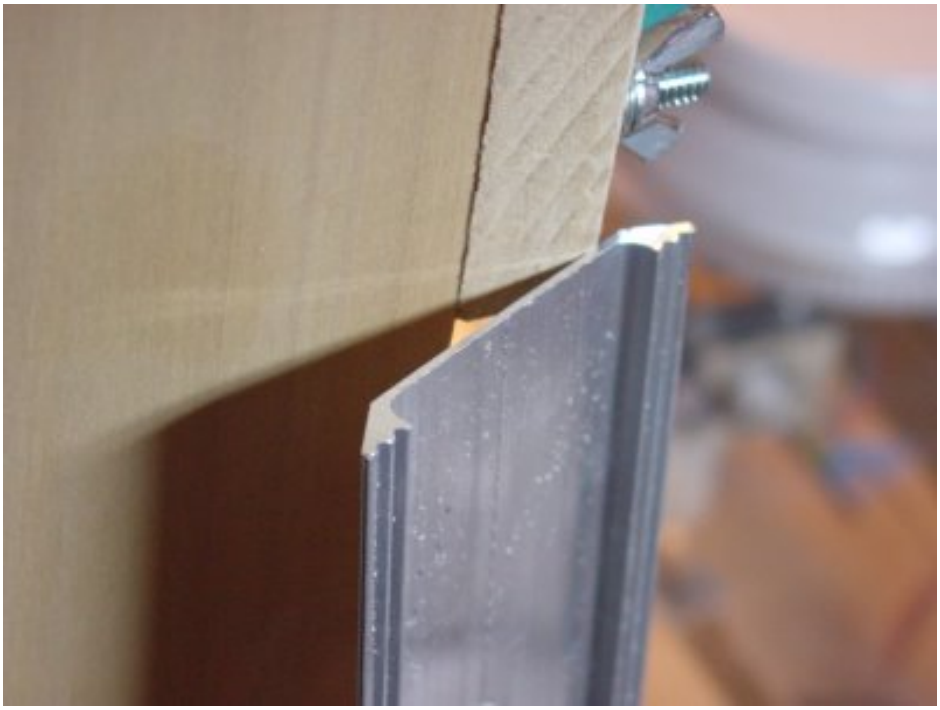
Fully extended only about 6 inches of the work table remained withing the guides. While increasing the tension on the guides kept the table stable, the consequent increase on friction load tended to overwhelm the z-axis linear stepper.

My first design decision was that the overall length of the z-axis work table needed to go to 18 inches so that fully extended there was still a foot of the table controlled by the guides.

I used JB-Weld epoxy rather than screws to make rigid joins.

While the longer z-axis work table was non-negotiable, the 50% longer work table was also going to weigh about 50% more. In that the existing z-axis required a counterweight, it was obvious that something was going to have to be done about lightening the worktable. 3/8ths inch HDPE was about as thin as HDPE could be and still be rigid enough. Sadly, although HDPE is half the depth of the poplar, it is nearly twice as dense and weighed approximately the same.

There the situation remained for some weeks until I happened across a 2.5 inch aluminum door threshold at my local hardware store.



The width was right and with a length of 36 inches, half of it was just what I needed for my z-axis table. The biggest plus was that the whole thing weighed four ounces where HDPE or poplar would have weighed 24.

Retaining rails were made with extruded aluminum L-profiles; one with the L-profile up to give lateral and normal guidance and the other with the L-profile down to provide normal guidance only.

Standard M8 (skateboard) bearings were used to provide alignment compression. On the side opposite the lateral guide the bearings were positioned at 45 degrees from normal to provide both vertical and lateral compression. Bearings on the other side provided normal compression only.

Here you see the front side of the z-axis in situ.

And here from the back side where you can see the tensioning spring for the bearings providing the normal compression.

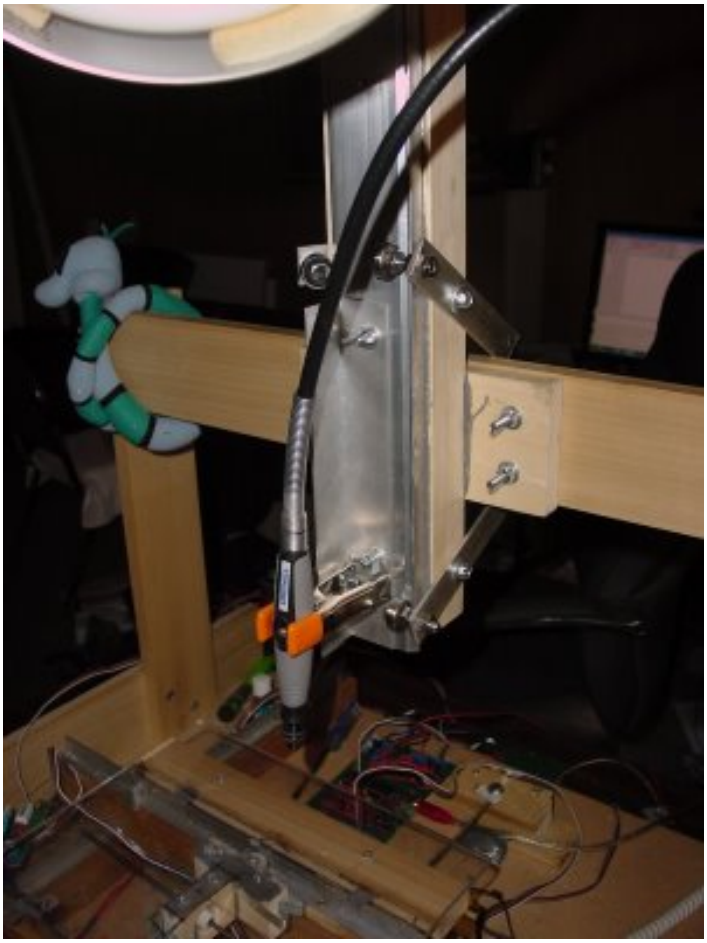
Tommelise 2.0 operational again

Sunday, 14th December 2008 by Forrest Higgs

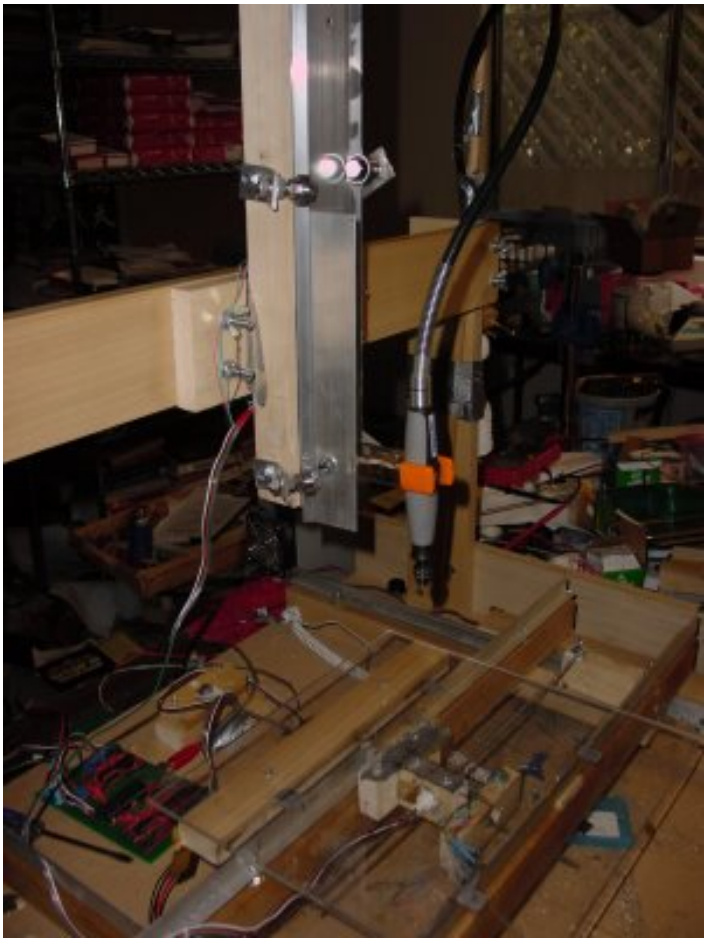
In which your narrator finds what may be a very clever way of attaching toolheads to the z-axis worktable...

My first impulse was to mount a piece of poplar onto the z-axis worktable and secure toolheads onto that. I'd already designed the z-axis worktable so that it could easily be demounted from the z-axis base. It seemed reasonable that toolheads should be similarly easy to mount and demount.

Recently, I'd bought some spring clamps and my son, this morning, was noting how much force it took to force the jaws apart. As he was working with one of the larger plastic ones I had, it occurred to me that it could clamp the Dremel toolhead as well easily as it could clamp two pieces of poplar together while I drilled a common hole through them.



A quick trip to the hardware store gave me a spring clamp that I was able to mount on a 1.5x1.5 inch L-profile aluminum extrusion. That I attached to the z-axis with bolts and wingnuts.



It seems to work pretty well and I don't need anything resembling a counterweight system.

Developing HDPE milling know-how on Tommelise 2.0

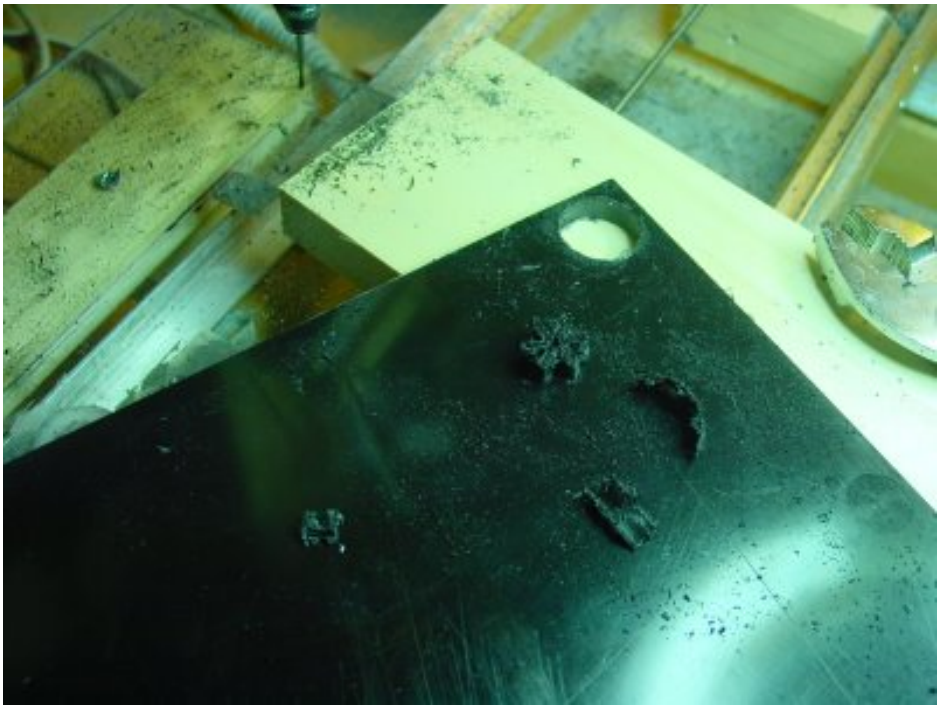
Wednesday, 17th December 2008 by Forrest Higgs

There is nothing like actually milling things to learn how to mill things...

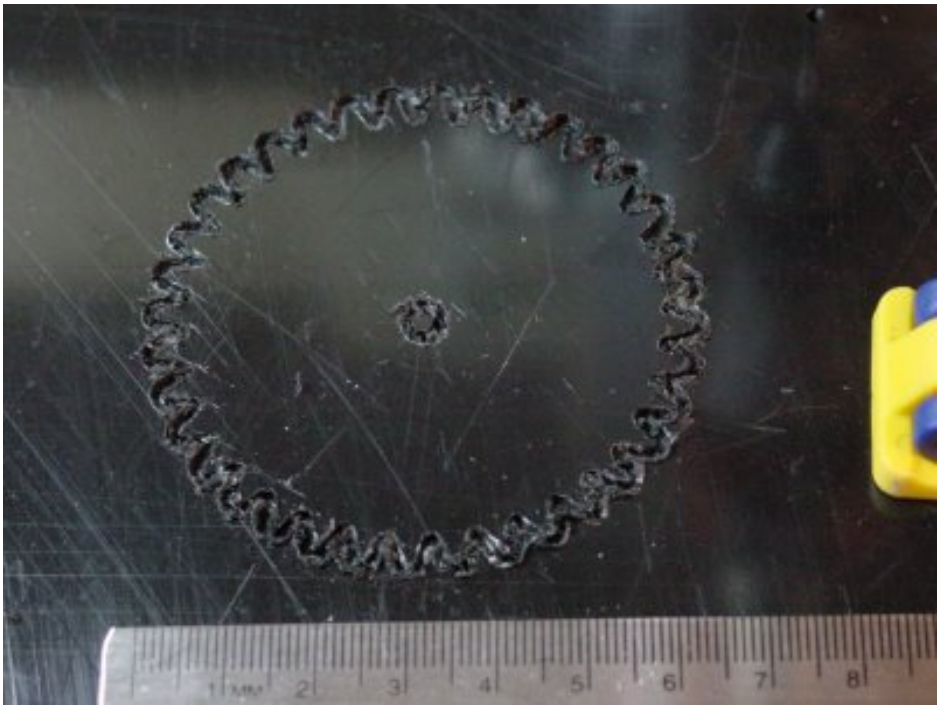
Prior to rebuilding T2's z-axis last week I'd done a few bits of milling, enough, I thought, to show me the inherent problems with milling on a light frame system. I should have known better.

I began milling experiments using a 1/8" cutter head. The swarf tended to form long ribbons and remain attached to the HDPE block at the cutter level. I would tear this stuff of periodically to keep the milling area clear.

I'd done very little work with the 1/16" cutter head when I stopped milling to rebuild the z-axis. I demonstrated that it was possible to mill quite small sprocket gears, however.



I nursed this particular gear cutting through and didn't think too hard about what I'd done. When I decided to cut a companion gear to this one, a 30-toothed monster that did a 5:1 reduction. After sorting out a few bugs in the pathfinding routine that generated cutter paths I began to cut such a gear in earnest.

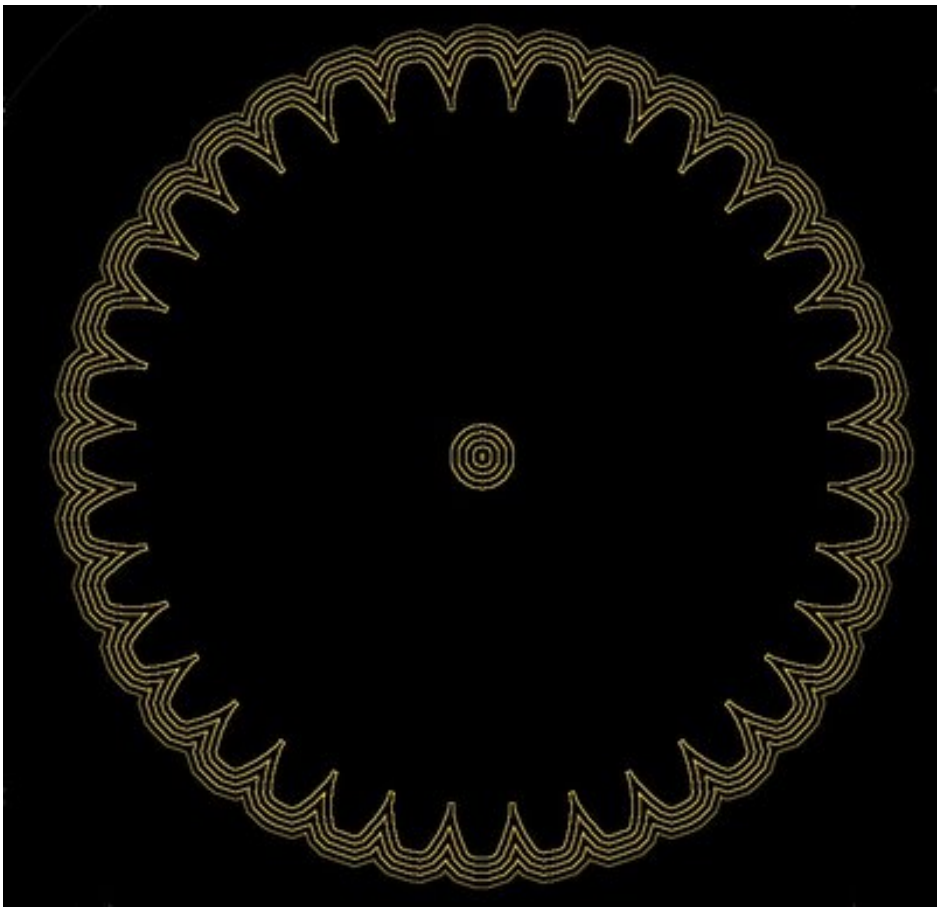


I quickly found that I was unconsciously attempting to use T2's milling capability like a laser cutter. The 1/16" cutter head created ribbon swarf just like the 1/8" cutter head. Although, I'd noticed that I had to clear the milled groove regularly when I was making the 6 toothed gear, I hadn't worked out the full implications of that for bigger milling jobs. As a result, I sort of neglected doing that.

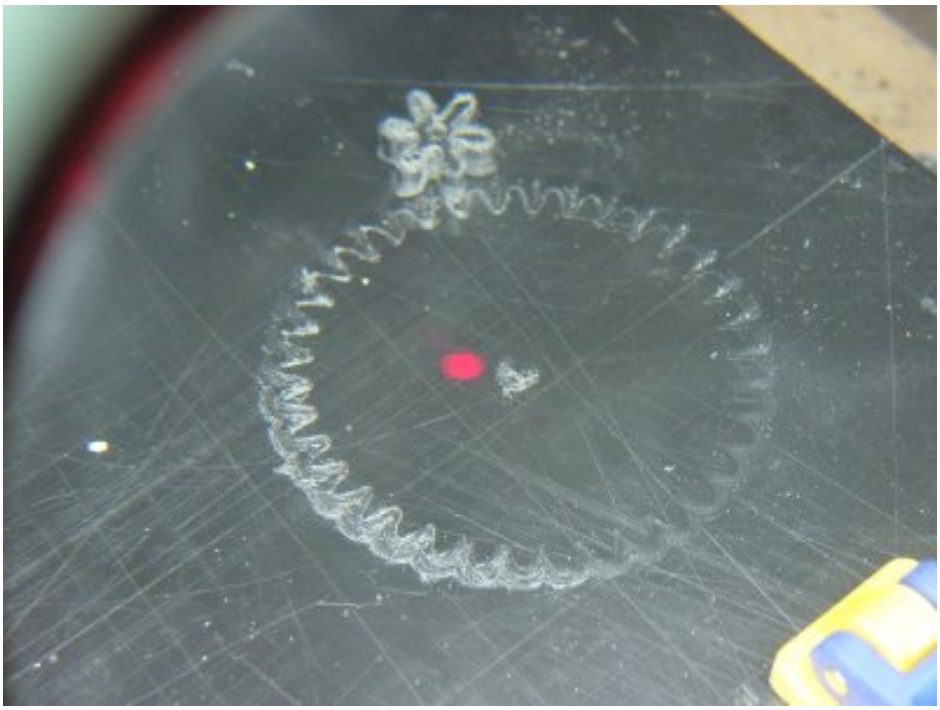
It wasn't long before I noticed that the cutter head was deflecting and damaging the emerging gear in spite of the fact that the new z-axis was much more rigid than the one it replaced. What was causing the deflection was rather simple. Bits of ribbon swarf in the narrow cut groove was getting wrapped around the cutter head. This expanded its diameter slightly which meant that the swarf was getting rubbed against both sides of the at high speeds. This melted it onto the cutter head, filling the cutter grooves. This would keep happening until friction from the melted swarf slowed down the cutter and caused it to grab against the sides of the cut. That caused a deflection and damaged the gear.

If I kept cleaning the cut the problem didn't get out of hand. Doing that was a pain in the neck, however, and not really practical as a regular procedure.

Fortunately, I had expanded the milling capability of the Slice and Dice routine to do excavate as well as cut. I decided that if I did several cuts and made the cut wider, I could keep the fine resolution allowed me by the 1/16" cutter while giving the ribbon swarf less possibility of fouling the cutting job. It would also be a LOT easier to get the ribbon swarf out of a wider cut.



Widening the cut to 5/16" gives me room to get into the cut with needle-nosed pliers to clean it out, if necessary. With a 1/16" cut I couldn't even see if the cut was clogged, much less clean it effectively.



Now it is just a matter of putting in the hours required to cut this beast.

Making a thrust collar in Art of Illusion

Thursday, 8th January 2009 by Forrest Higgs

In which your narrator walks you through the differences between preparing a design for printing as opposed to preparing one for milling in Art of Illusion...

Non nobis Domine

Thursday, 15th January 2009 by Forrest Higgs

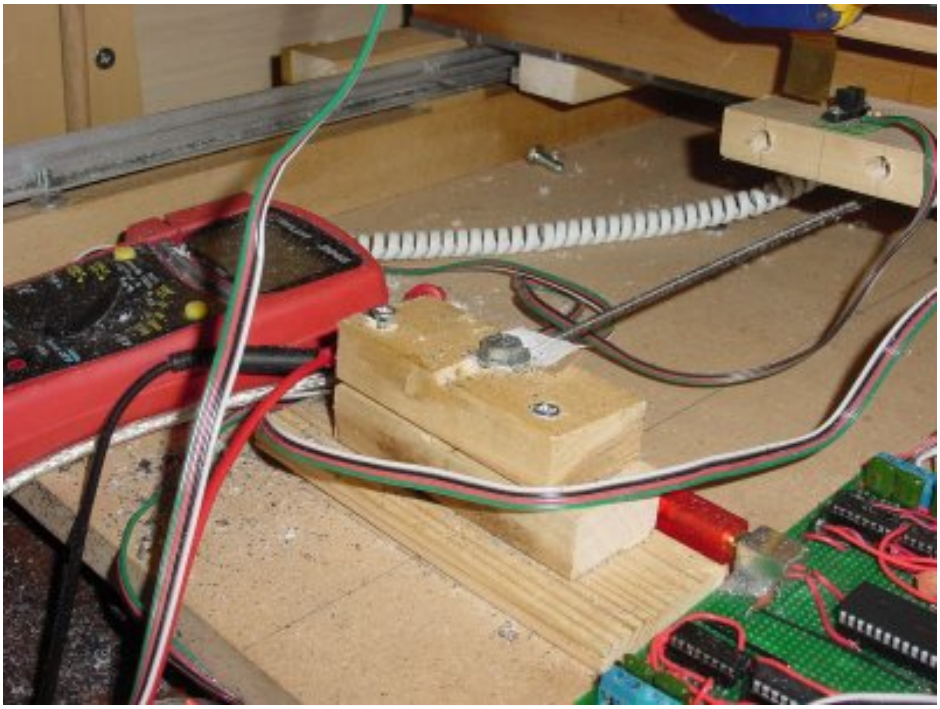
In which Tommelise 2.0 successfully mills it's second gear...

I am so tired. Little did I know when I coaxed my first involute profile gear out of Tommelise at the end of September last year how long it was going to be before I could do good milling as a matter of course. This evening I achieved that.
Rather than talk about all of the trevails I went through getting here I will only tell you about the last two. After having done a massive rewrite of my Slice and Dice software, I was still having T2 go crazy after many hours of milling, ruining the product. A few days ago I got 96% of the way through milling a thrust collar gear for a project only to have T2 go mad and cut the gear to pieces as you can see here.

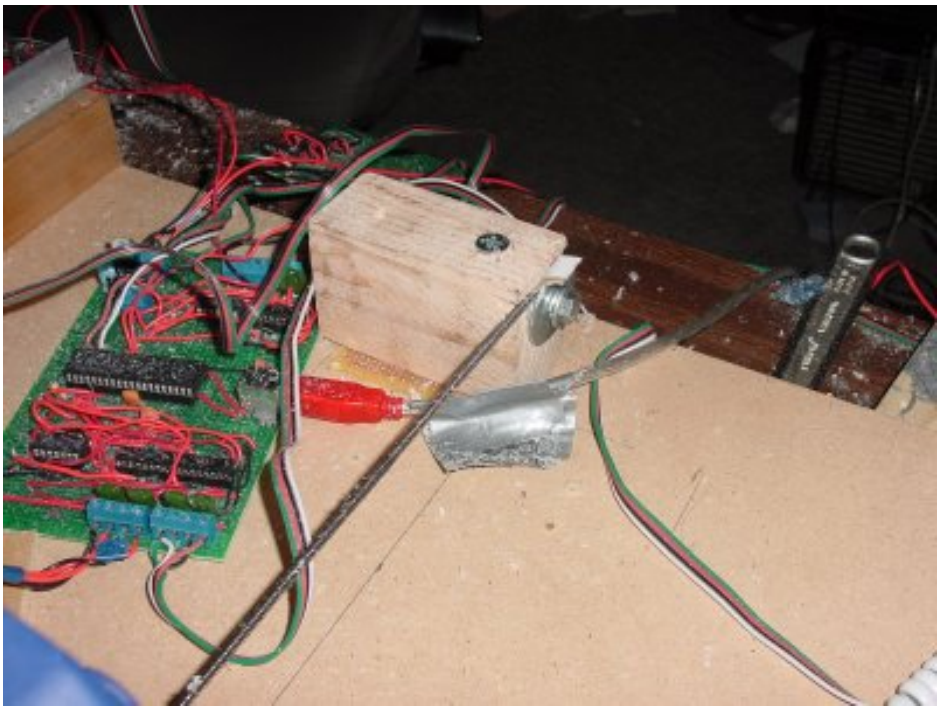
I seriously thought about just buying a Ponoko extruder and dropping the whole milling idea. After two days of depression, however, I went back at it again.

My first break happened when, while cutting another gear, I observed T2 suddenly drop the end mill over a millimeter into the HDPE and start grinding away. That told me that I had a problem in the interrupt routine and more specifically, I had a problem with the z-axis code in the interrupt routine. I rewrote that routine and seemed to have got rid of that problem.

My second and final break was my serendipitous discovery of just why it was that periodically the end mill would just grab at a milled wall and then start digging through it. When I knocked T2 together originally I used a few scraps of lumber to make a mounting block for the y-axis lead screw.



While the block was firm enough when I built it many hours of operation had loosened the screws in it to the point that when the y-axis was making rapid direction change the block was deflecting rapidly over a millimeter at times. That was more than enough to account for the grab and grind problem. I replaced that scrap block with something more substantial and bolted it to the particle board base of T2.



That sorted out the deflection problem.
I then began to cut a 10-toothed gear for my new tin can linear stepper ensemble. The cut went both smoothly and relatively quickly.

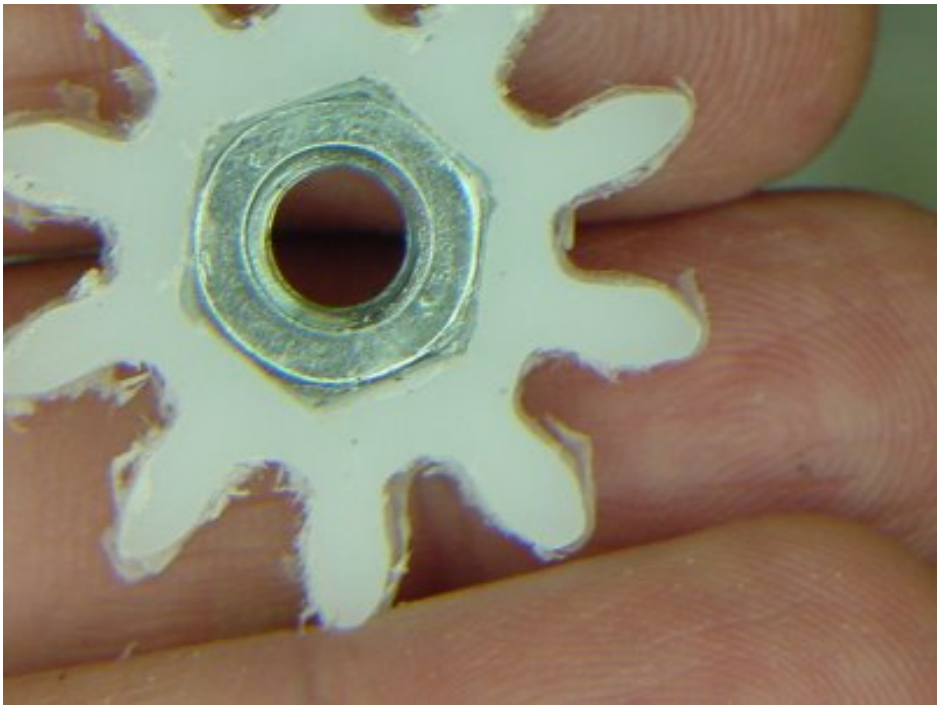
I cut the gear in 3/16" (4.76 mm) HDPE sheet. When I reached 4 mm I began to see the darker wood under the translucent HDPE.

The detail of the wood became clearer and clearer as I milled within a few tenths of a millimeter of the bottom side of the HDPE sheet.

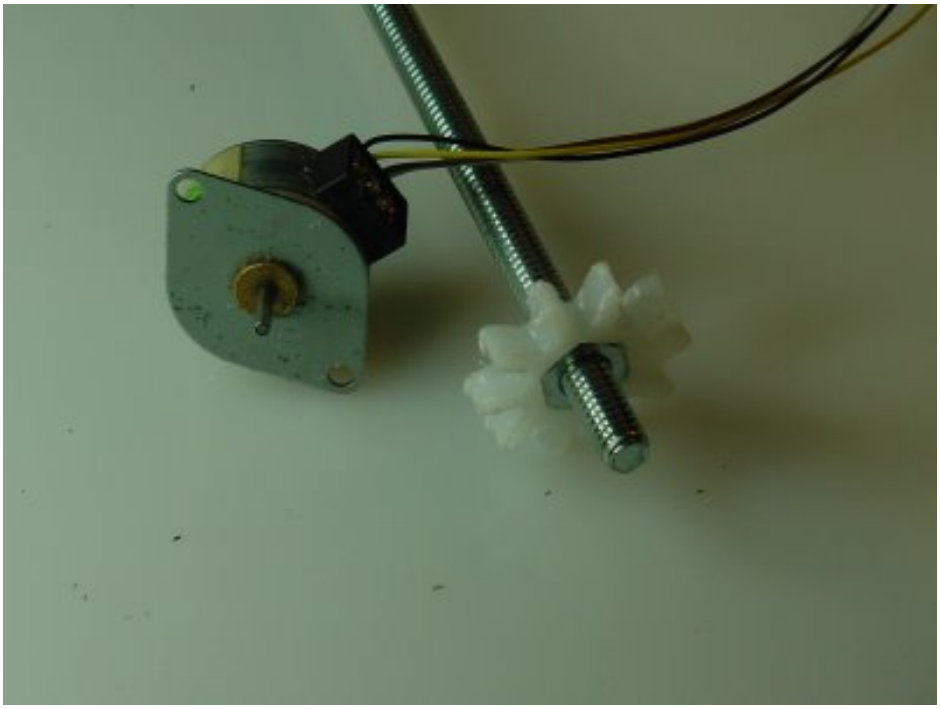
Finally, I could see the actual grain of the wood through what was rapidly becoming totally transparent HDPE. Basically, my gear was sitting on a tough film of HDPE. The core of the lead

nut cut popped out first. I had noticed in September that with a light cutting the end mill tends to simply throw the product out of the cavity without damaging it. I spent considerable time rewriting Slice and Dice code to put tabs on products that would keep the end mill from damaging the product when it broke loose. The film effect turned out to be far mor effective.

A few moments later the gear popped out, too, with just a few fragments of film on it as you can see here.



I was able to pressure fit the 1/4-20 lead nut in the gear with a vise.



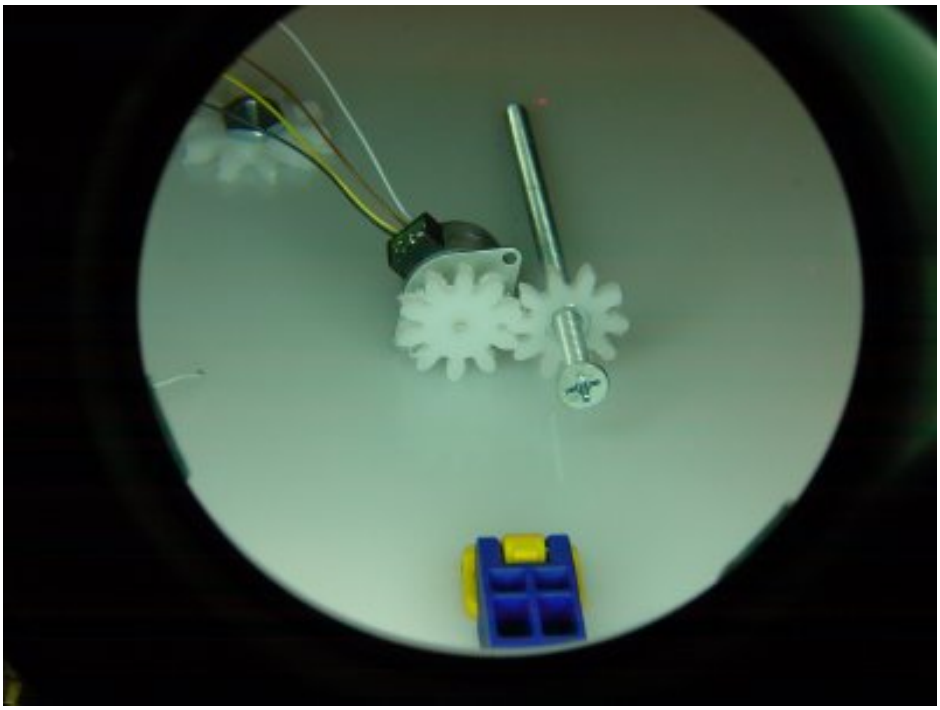
Tomorrow, I will cut the drive gear for the tin can stepper and, hopefully the housing for everything. I realise that I must do a second design iteration to deal with backlash. At this point, however, I just want to see if everything fits together as I imagine.

So far, so good.

And then there were two

Thursday, 15th January 2009 by Forrest Higgs

Cutting the stepper gear for the linear stepper ensemble was routine, almost boring...



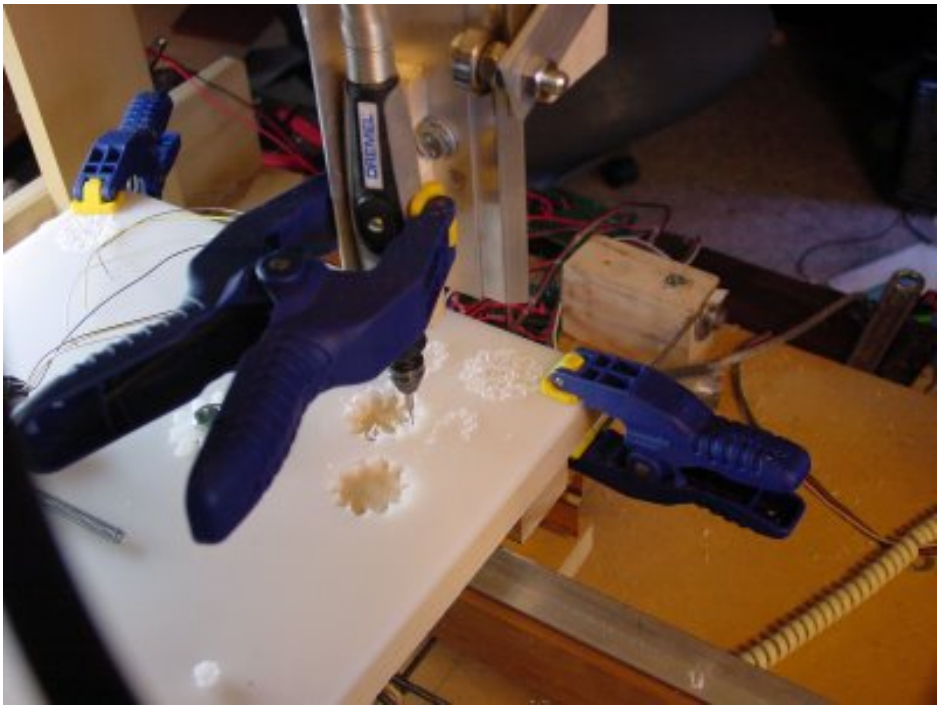
This morning I designed a companion gear for the one that fits over the thrust collar nut that fits onto the little Jameco tin can stepper. One thing that milling does that printing isn't so wonderful at is doing holes. The milled products are isentropic with respect to material qualities, so there is no chance that a mounting hole is going to be a weak point on the product unless you design it that way.

I was able to stack the data for cutting the gear so that it did five cuts (.102x2 x5 = ~1 mm) before having to be restarted. When I had cut 4 mm into the HDPE I reverted to single cuts as I approached the poplar underlying the HDPE sheet. I used a measured probe to confirm how deep I had cut at any given time. It was easy enough to see the poplar grain through the rapidly thinning HDPE. When the end mill finally broke through the last of the HDPE I simply put a finger on the gear until it had finished the cut and then turned the Dremel off. T2 creates very little lateral thrust while cutting so this simple expedient was quite adequate for keeping the end mill from possibly marring the gear.

I designed the drive shaft hole for the gear a touch smaller than the shaft and then press fit it onto the drive shaft with my vise as I did with the thrust collar nut.

One other little milestone for me was that this was the first time that I had two mated gears to check to see if my involute profile gear script worked properly. It does. The two gears meshed quite smoothly.

Arvin Evans posted a comment to my last blog entry asking about the clamp mount for the Dremel tool on the x-z working surface. When I upgraded T2 I changed how that was done to reduce the floppiness and to make it easier to get the Dremel and other tool heads on and off the "L" profile on the x-z working surface.



I demounted the old clamp that stuck out and used a plastic Arvin (interesting coincidence, actually) clamp to bind the Dremel directly to the "L" profile. I also carved a little wooden seat that keeps the vertical orientation of the Dremel vertical. You can just see it behind the Dremel in the photograph.

I can highly recommend those Arvin clamps. They are feather light and very powerful. I use them to clamp both the toolhead and the sheet HDPE. They work beautifully in both roles.

After Darwin: Should Mendel be a specific 3D printer or a technology toolbox?

Friday, 16th January 2009 by Forrest Higgs

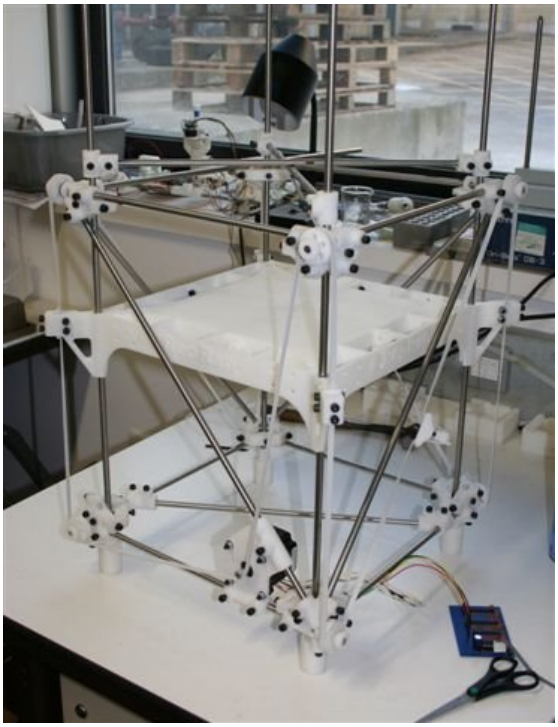
In which your narrator suspects that we are using the wrong model of evolution...

Darwin has been a roaring success by virtually any measure you'd care to name. In April of 2008 we had exactly five reprap and repstrap printers more or less operating. Last month, Dr. Bowyer estimated that several thousand were either operating or under construction. That's not exponential growth. It's hyperbolic. I, personally, have no doubt that in 5-10 years Dr. Bowyer is going to find himself on the Queen's annual Honor's List. Mind, in my opinion he richly deserves it. That said, I wanted to share a few ponderings that I've been nursing on the nature of change in the reprap movement for the past few months.

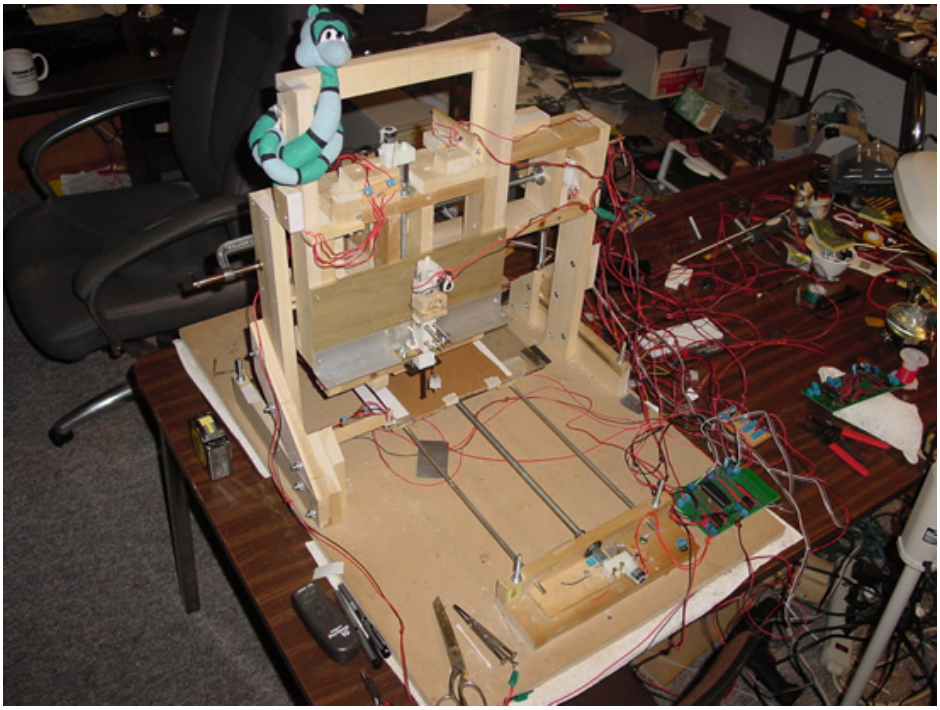
Basically, Reprap has two models of evolution going on in parallel. In the core team, we see what I will call a punctuated animal model of evolution while the larger reprap community is a very vibrant bacterial model.

Back at the end of 2006 and the beginning of 2007, the Reprap Core Team had several prototypes for a first release Reprap machine; Da Witch in New Zealand,

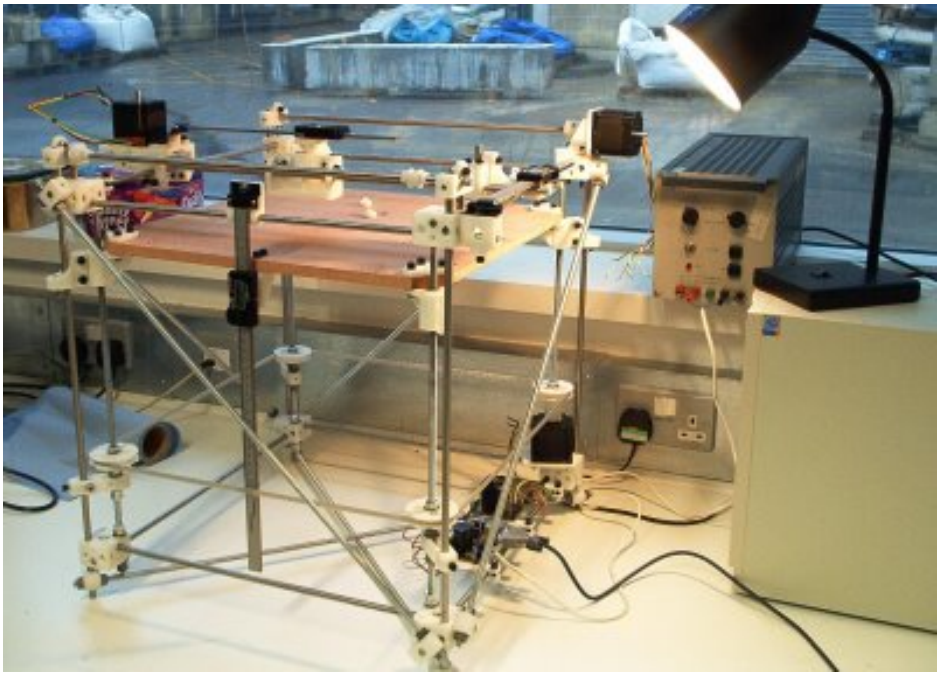
A.R.N.I.E. at Bath



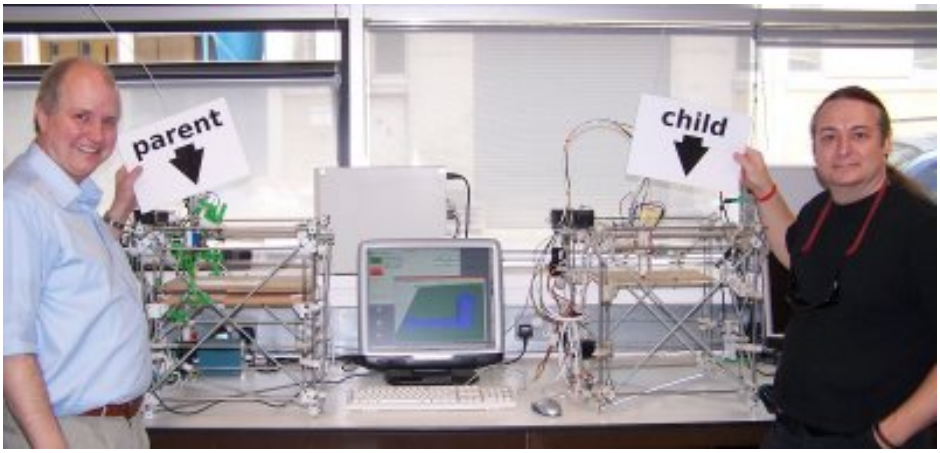
and Tommelise 1.0 in the US.



By March of 2007, Darwin, which was to become the first official release reprap machine had grown out of the A.R.N.I.E. prototype...

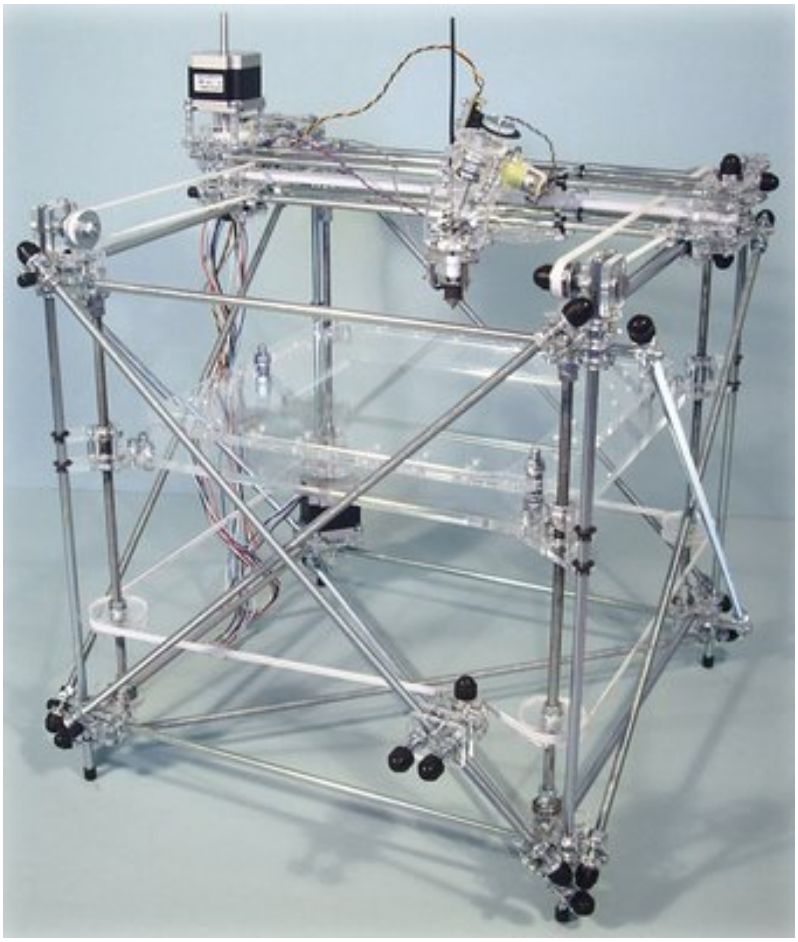


...and by the second quarter of 2008 Vik Olliver had managed to print a full parts set for a Darwin with his Darwin machine.



Then a very curious and totally unexpected thing happened. Fully 6 months went by before a Darwin replicated again, this time in Canada.

By that time, however, by Dr. Bowyer's estimate the population of Darwins were in the low thousands. What had happened?

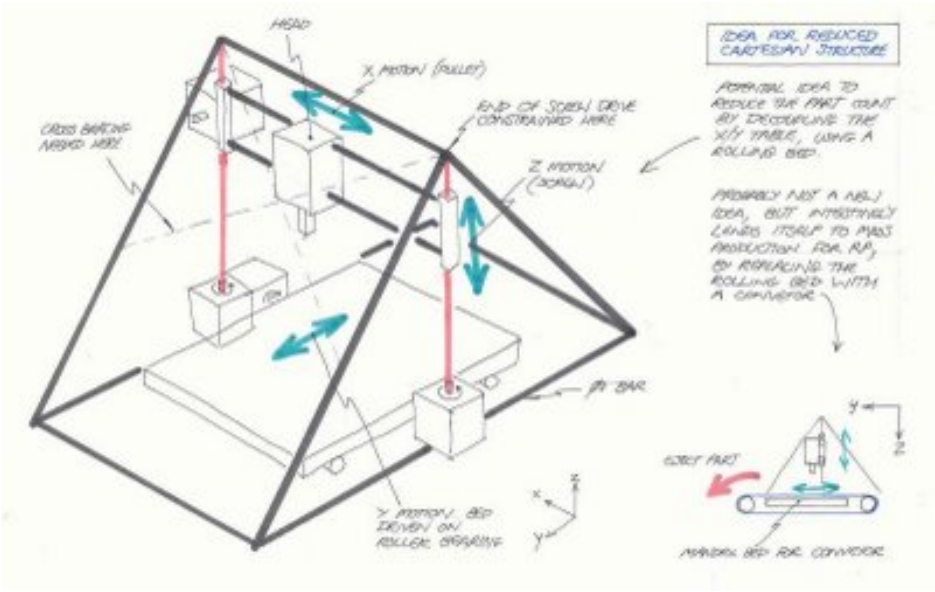


Basically, Darwin morphed into a fully industrial product. It began with the controller boards being outsourced for production by the Reprap foundation and has culminated with a shippable kit purchasable for US$1,100 {shipping not included} requiring little more than the sort of assembly you'd be expected to apply to something bought from Ikea. What is getting built out there in its

thousands, to use Dr. Bowyer's metaphor, is 100% vitamins - 0% replicated parts.

These aren't really Darwins, they're repstrap machines. You could use one to make a Darwin, but they aren't. Further, because they are repstraps, the machine you print with one isn't even the machine you're printing. The parts in it are laser cut, not printed. Now I suppose you could, if you wanted, print the laser cut parts. I haven't seen anybody attempt this, though.

A few tentative attempts have been made towards coming up with a second generation mainstream reprap machine. The most notable effort in this regard has been undertaken by eD at Bath university.



eD's new design promises to be considerably simpler to build and less materials-intensive than our current Darwin design.

We've found ourselves increasingly sliding towards designing with industrialisation in mind from the onset. Instead of designing simpler electronics that would be easy to cobble together, we have been designing boards with parts like surface-mount chips friendly to outsourcing and short production runs. Design, for absolutely the best of reasons, has been getting more and more centralised and unified.

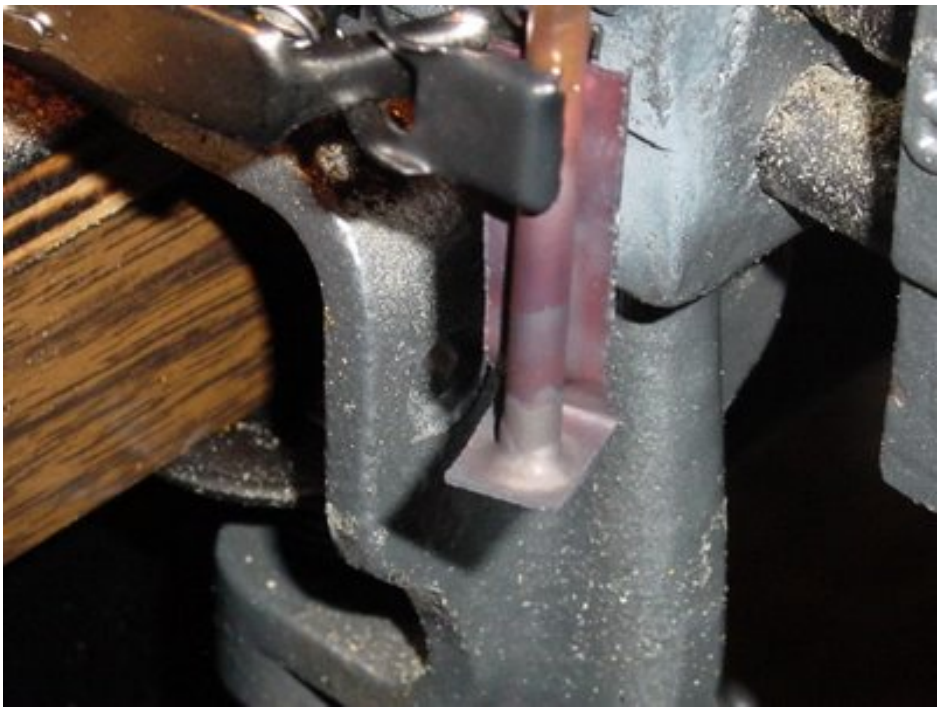
I think that this has been happening in large measure because of the model of evolutionary development that the core Reprap team have been using. While we have been giving lip service to Darwin's original theory of natural selection, in practice we've been using Richard Goldschmidt's Saltation theory in which very large changes happen from one generation to the next.

I think, and this is my personal opinion, that we really haven't thought about the implications of what we are doing when we come up with a single specification for a particular generation of Reprap machines. Single specifications presume a particular environment of parts availability. As practice has shown us there ain't no such thing as a general availability of a particular set of parts

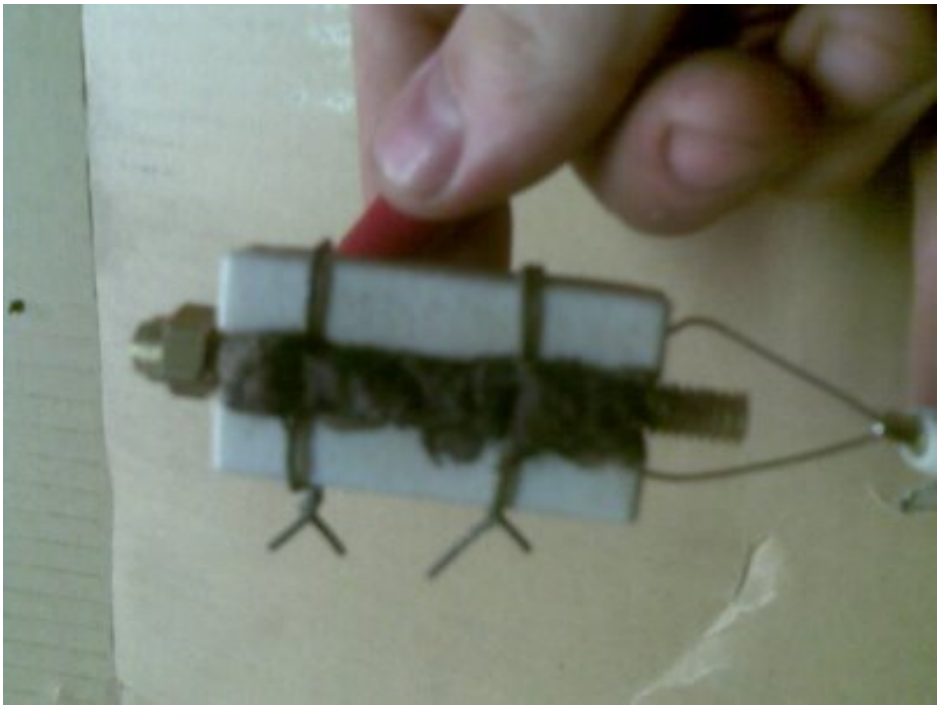
for the Earth as a whole or even large parts of Earth.

The only place that sort of thing can economically happen is on a factory production line variously defined. That's why there are factory production lines and that's why it makes sense to make something at one point on the planet and then ship it everywhere.

If we look, however, at the larger Reprap community we see a very interesting thing happening. For example, lathes are relatively expensive and have a learning curve associated with them so in one place we see things like this



In another place nichrome wire and ceramic potting material is difficult to obtain so magical solutions like this occur...



While the Reprap Core Team is practicing Saltation, the larger Reprap community is practicing Horizontal Gene Transfer, a method of exchanging recipes that enhance adaptability and communal robustness that bacterial populations employ to manage extremely rapid evolutionary rates and exploit new environments.

If you look at Darwin, for example, you can subdivide it into a set of different parts or systems, depending on how you look at it. A quick partial tree structure of Reprap Darwin looks a bit like this...

- 3D modeling software
- Slice and Dice software
- PC-side control software
  - communications
- Positioning system
  - mechanics
  - firmware
- Toolheads

I think, and this is again my own, very personal opinion, is that the core team needs not so much to dictate the technologies used as manage and facilitate the interstices/interfaces between the different parts and systems while letting the community at large provide the microenvironment-specific solutions to problems.

There are, for example, at least half a dozen solutions in the community to controlling a Reprap machine. Some depend on open source compilers, some don't. Similarly, there are a variety of kinematic solutions to the question of positioning the toolhead.

As a practical matter this means that the Core Team needs to climb down from more than a few articles of faith that it holds about what belongs in a Reprap machine and concentrate more on the

much more difficult task of managing and interfacing the rich diversity of parts and pieces that have been developed by specific community members trying to solve specific problems in particular environments.

If we do that, as I see it, we have a chance of creating a prototyping/production capability that truly can diffuse into the furthest reaches of the planet. As it stands now, we're developing technology that is easy to make in a factory, put in a box and ship worldwide. In my opinion, what we have slipped into doing now is the antithesis of what we set out to do so many months ago.

Anti-backlash nut

Saturday, 17th January 2009 by Forrest Higgs

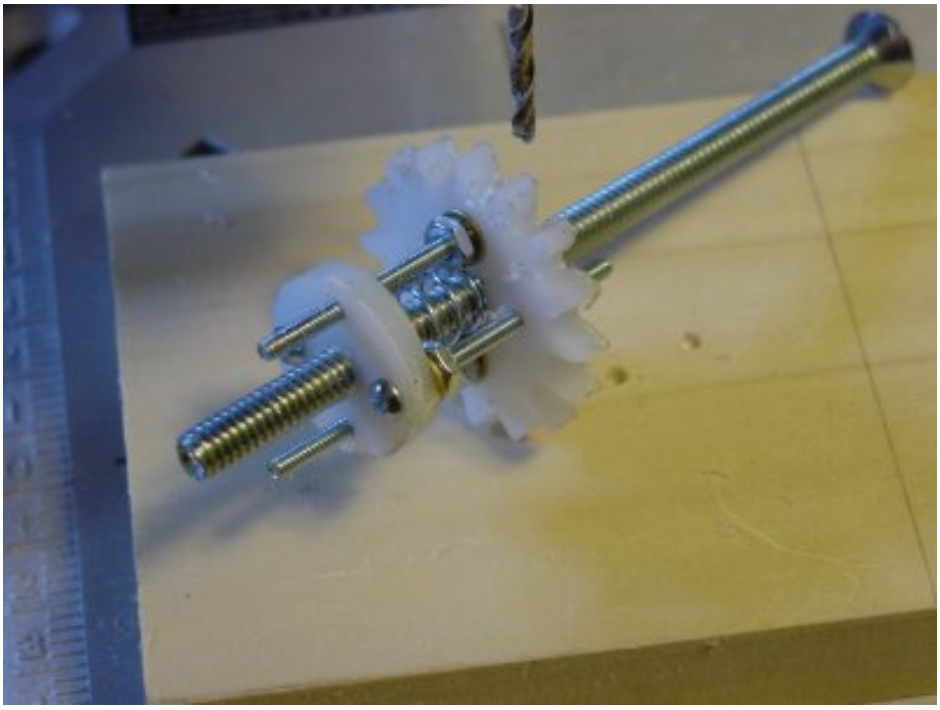
In which your narrator mills an equivalent to an expensive antibacklash nut on Tommelise 2.0...

I started out cutting a 1/4 inch nut shape in my thrust gear to seat the drive gear on the 1/4-20 threaded rod that I am using for a lead screw. The steel nuts had about half a millimeter of play in them so I dug around for a simple anti-backlash system.

Noll, Inc., just down the road from me in San Luis Obispo has a nice, simple design...

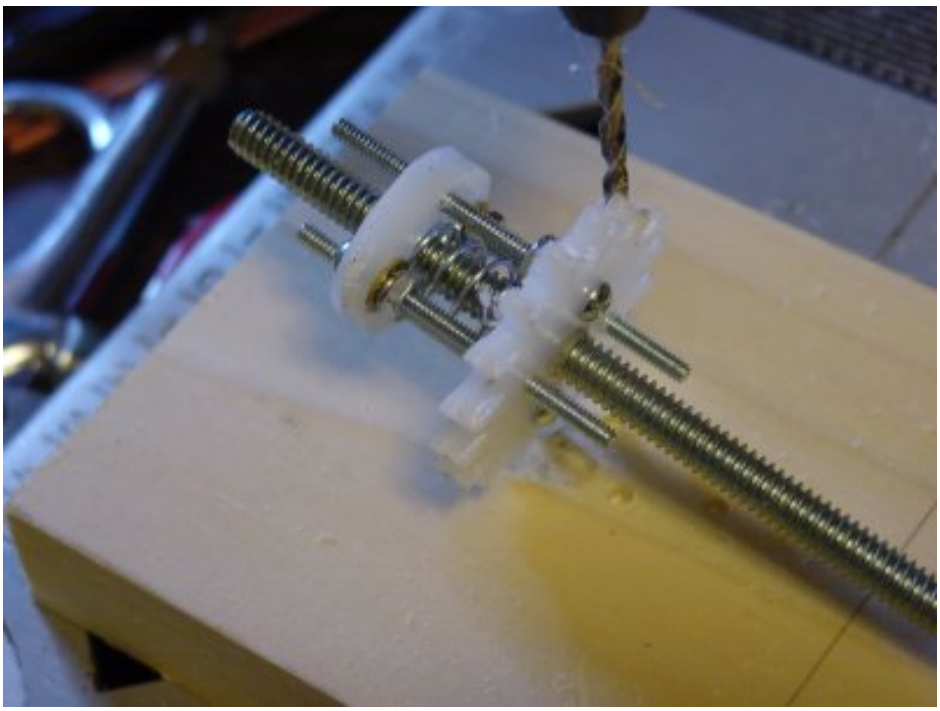
Image not found.

which I redesigned for my purposes.
A day later and I'm done.



The irony is that while I was at the hardware stockists I got to thinking about the fact that my drive

gear was HDPE and that it wouldn't be rocket science to just tap the gear for 1/4-20 threads. I bought a proper tap and designed in a #7 hole into the gear to receive the tap. When I'd cut the threads in the gear I discovered that an HDPE gear has no play, so there is little point in having an anti-backlash assembly associated with it.



The 1/4-20 threaded rod has a metric pitch of 1.27 while the Kysan tin can stepper had a 15 degree step length. My drive gear has 16 teeth and the stepper gear, tentatively, has 12. That gives me a linear resolution of a touch over 0.0375 mm/step. If that isn't suitable it is a simple matter to change the tooth count for the stepper gear.

Going high risk steampunk

Tuesday, 20th January 2009 by Forrest Higgs

In which your narrator finally gets stuck into exploring other prime movers for future reprap machines...

Some time ago, Tigertaler, one of our more creative Reprap fanatics, did a series of design charrettes for printable stepper motors which he published on Youtube. His designs {example shown above} looked at using solenoids as a motive force for cleverly designed kinematics to achieve stepping motion in a highly heterodox manner.

His logic is impeccable. It is a lot easier to wind a solenoid coil than it is to do the windings and rotor construction for virtually any kind of electrically powered motor you'd care to name. I'd mooted this notion around the Reprap Core Group before. Nophead noted that the frequencies that such a solenoid might have to reach to provide a useful combination of step size and rotational speed could well be beyond what could be achieved. Given that Nophead has likely forgotten more than I'll ever know about electronics, this tended to dampen my enthusiasm at the time.

Last weekend, however, a concatenation of events caused me to revisit Tigertaler's work. The previous week I'd very successfully milled quite a handsome anti-backlash assembly for a tin-can stepper powered linear drive. I built the assembly around a 1/4-20 threaded rod with the intention of driving it with a Kysan 25BY2414 rated for 12v and 0.2 amps/winding. When I got the assembly together it became obvious that the Kysan was not going to have adequate torque to propel the thrust collar at meaningful speeds.

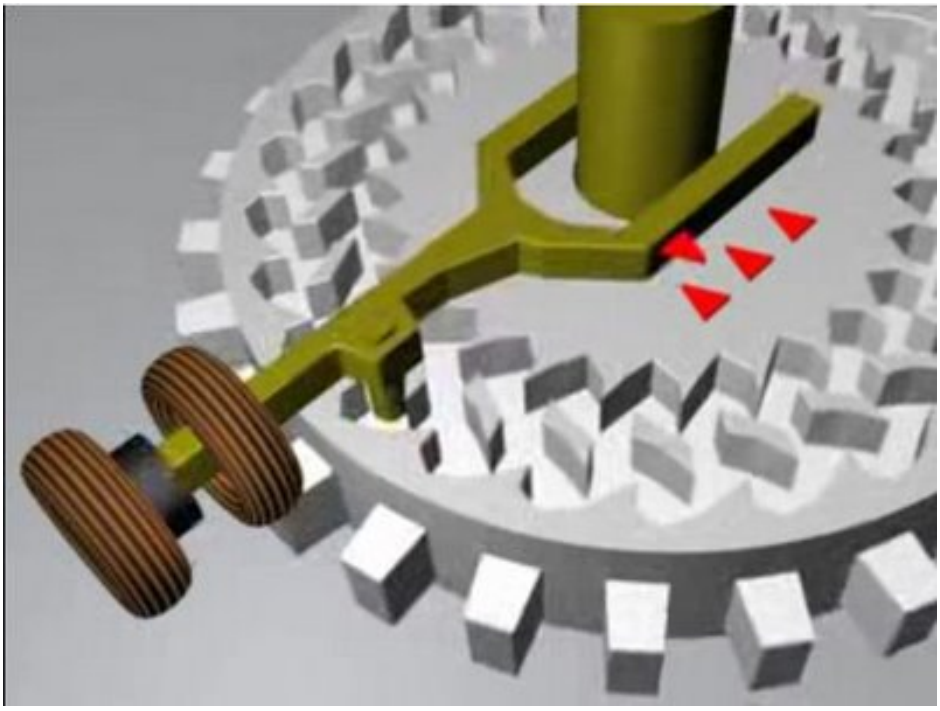
I had two choices at that point. I could redesign the assembly for #6-32 threaded rod or I could buy a beefier tin can stepper. Anaheim had beefier tin can steppers at beefier prices for one-off. Both options were going to take more time while going for a beefier stepper was going to increase the price of the assembly. None of that pleased me to any great extend so I make a third choice and decided not to decide for a few days.

The whole experience had brought one lesson home to me. When designing something that takes power it is not always that obvious, to me anyway, just how much power it will take to make it go around. For testing purposes it's always nice to have a lot more torque than you need just to test concepts.

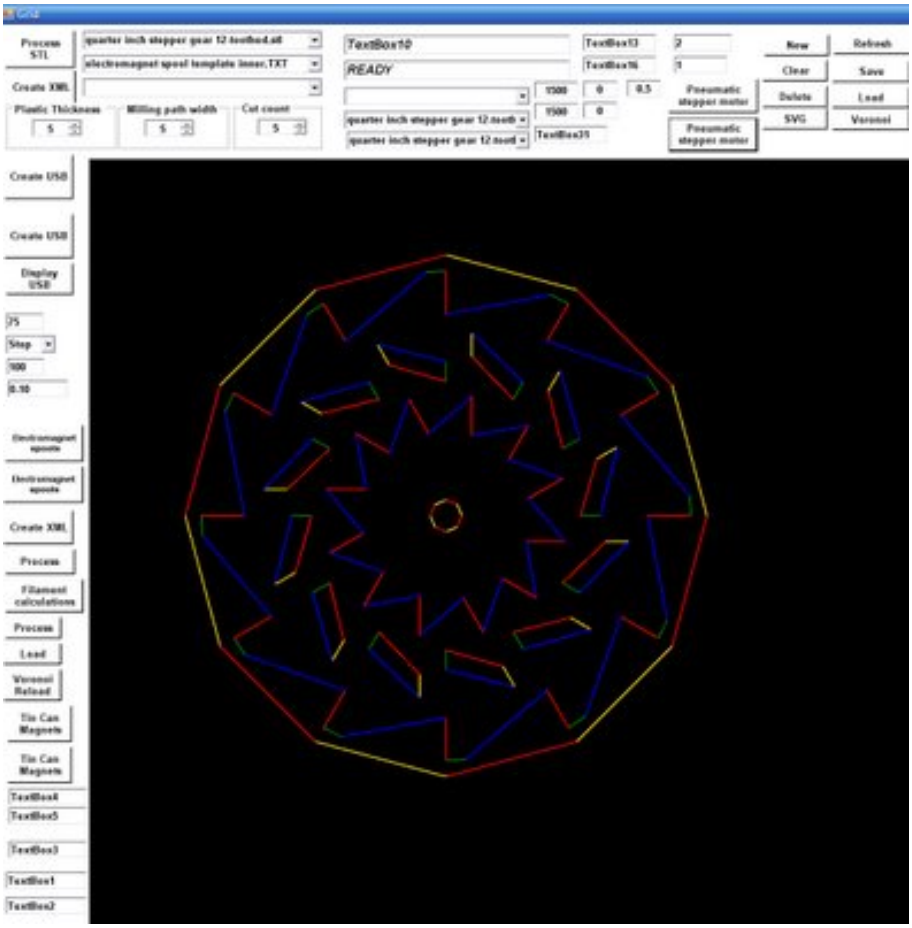
That was when the notion of using pneumatics came back to me. Pneumatic actuators power output depends on the pressure of the air supply. I'd bought myself a compressor for Christmas with something like that in mind. To date it has been doing yeomanry duty keeping swarf out of my

milling jobs. It is, however, capable of providing up to 150 psi (~10 atm) of air pressure and can easily drive nail guns and pneumatic wrenches. Pneumatic actuators are very small for their power output. As well, I should be able to mill them out of HDPE without a great deal of drama.

At that point I began to give Tigertaler's most ambitious printable stepper a very close look.



While it looked quite complex at first glance, in fact it was merely a repetitive presentation of a relatively simple mechanism. At that point I set about to develop a script that could do a slice description for milling.



Now it remains to do a few modifications to my slice and dice app to accept a slice directly instead of insisting on generating it out of an STL file. I hope to be milling my first Tigertalar Steam Pumk stepper motor in the next day or so.

Conserving MCU pins via the I2C bus

Tuesday, 27th January 2009 by Forrest Higgs

In which your narrator finally gets around to demonstrating something he's been talking about for the better part of a year...

About a year ago I found myself beginning to work with the I2C bus when I decided that I needed a decent print buffer on the Tommelise 2.0 reprap machine. I ran across a 512Kbit EEPROM chip at quite reasonable prices that connected to the I2C bus rather than requiring a committed port on my 18F4550 MCU. For me that was wonderful in that I was committed to a single board controller solution rather than the multiple board scheme that the mainstream Darwin used.

I was astonished at how easy it was to integrate I2C EEPROMS with my existing board. They've worked brilliantly as a print buffer ever since then.

At that point I began to look around at other I2C enabled chips and discovered a wide variety of things that I could do with my MCU via the I2C bus. There were some things, however, that were simply not available in I2C format. After much grumbling, I finally had a "DUH!" moment. It seemed reasonable that Philips Electronics N.V, who had developed the I2C standard would have HAD to have created a generic I2C slave chip that could be interfaced to ordinary port-enabled IC's. If they hadn't, the I2C bus would have never got off the ground.

Sure enough, a short catalog search revealed that Philips had indeed created several such chips and that several other suppliers had developed their own I2C slave chips as well. I then ordered a dozen of their excellent PCF8574I2C slave chips, manufactured by Texas Instruments for a few dollars apiece. The PCF8574 gives you access to a single 8-bit port through the I2C bus while its bigger brother, the PCF8575, gives you access to two, 8-bit ports. You can hang eight of either chip type off of a single I2C bus. Mind, most firmware IDE's offer you software emulation of I2C comms which will let you assign more than one I2C bus.

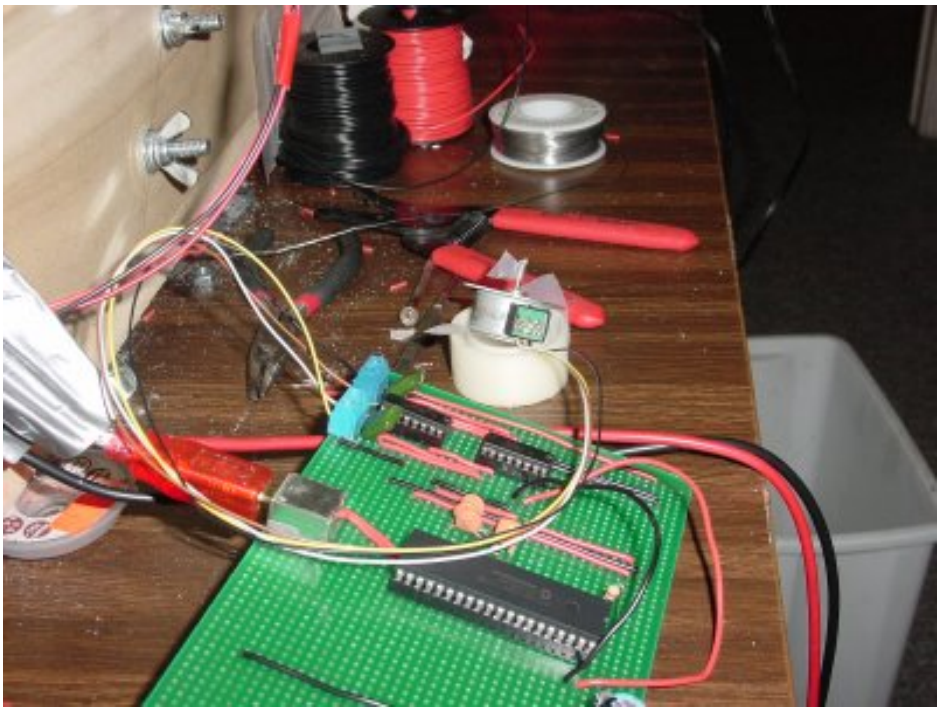
The demands of my day job and other, more pressing issues with Tommelise 2.0, like making it mill plastics and printed circuit boards kept me from actually using the generic I2C slave chips on anything for many months. Last week, however, I got Tommelise 2.0 milling things properly and was able to take a deep breath and look at my "to do" list.

One big item on my "to do" list was to build an infrared ranging system, very short range radar if you will, for a telepresence robot that I eventually want to get running. Since the ranging system is a spot detector, I have to have an alti-azimuth positioning system which, of course, requires running a pair of stepper motors. I had been putting off designing new boards until I'd got the

printed circuit board milling system working. Last weekend, however, I saw the pile of Euro board format stip boards and decided that I might as well use one of them and see, as well, if I could prove out the notion that the PCF8574 I2C slave chip could be used to run the 754410 high-current half-H driver that I would be needing to drive the steppers.

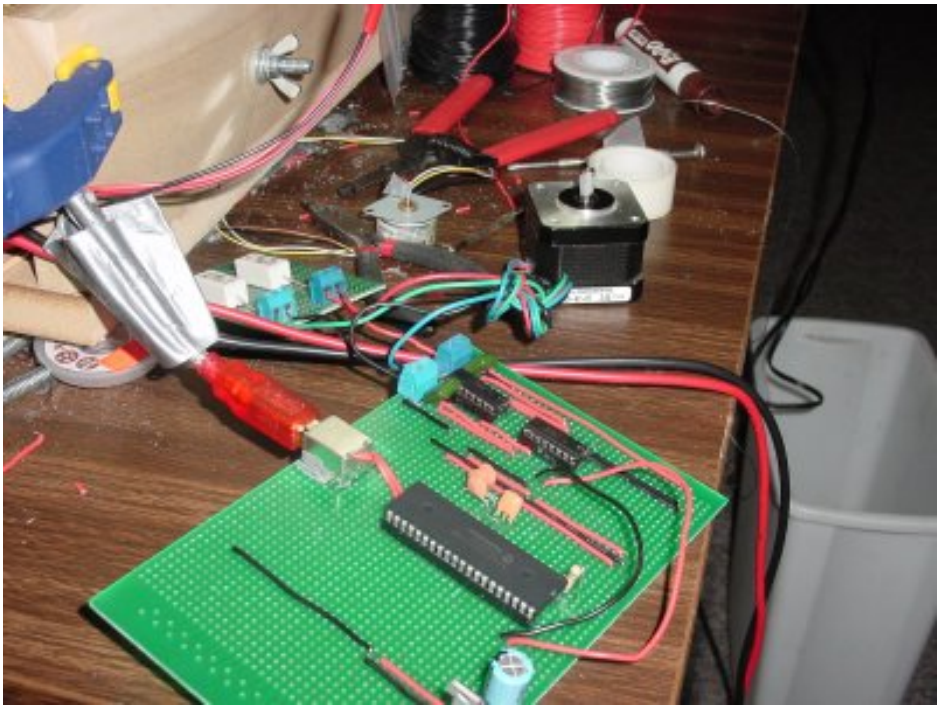
I'm a very conservative board designer. I used the 18F4550 MCU that I have loads of hands-on experience and firmware from the Tommelise 2.0 project. I also have PC-side USB support software for that chip as well. It took me a couple of hours to design and wire up the board and about the same amount of time to chase down and fix the wiring errors that I made on it.

This morning, I took a few hours off and copy/pasted up the firmware to drive one of my little Kysan 25BY2414tin can steppers that I keep for testing. Oddly, I was not able to get it running with wave (single phase) stepping. That is puzzling in that wave stepping tends to work on anything. When I switched over to full (two phase) stepping, however, it ran without so much as a qualm.



What was even more interesting was that I was able to push the Kysan up to 2,000 pps, six times faster than I had heretofore been able to achieve with wave stepping.

At that point, I decided to see if I could use one of my Lin Engineering 1.8 degree step Nema 17's that I'd bought in bulk when I first started working on the Reprap project. I wouldn't be using this stepper ultimately on my IR radar system because it was two heavy and drew far too much power (2.5 amps/phase) for what I needed. To choke it down to something my 754410 chips could handle (less than 1.1 amp/phase) I put 15 ohm power resistors in series with each phase.



I was able to push the Nema 17 up to 2,000 pps as well before it started stuttering.

There had been some talk that I2C and steppers wouldn't be happy together. That appears not to be the case. The data sheet on the PCF8574 says that a read/write takes 4 microseconds. That's not long enough to matter even if we're running three or four steppers in concert.

What using 12C slave chips like I have lets you do is avoid having to go over to surface mount chips which let you have many more pins on your MCU which are directly accessible. Through the hole DIP chips are usually limited to a maximum of 40 pins with the rare 48 pin offering making an appearance from time to time. What happens with DIP chips is that the chipmaker makes one pin do duty for sometimes as many as six of the pins that would have been available on a surface mounted package for the same chip.

I've tried to use some of these stacked up function assignments on 40 pin chips. It seems at times that you have to do a dance around a fire on a full moon and sacrifice a goat to get at some of the functionality of stacked DIP MCUs at times. What I suspect is happening is that IDE vendors get tangled in their own compiler coding trying to make one set of code fit dozens of slightly different models in the same MCU family. It's certainly a pain for an unwary board designer.

In any case, hanging output and not-interrupt input functions that would ordinarily occupy IO ports on the I2C bus leaves you with very clean board landscape around your MCU. That's a big plus for me in that I'd like to keep my controller board design to a milled single-sided or simple double-sided board and avoid having to send my designs out to a board etching firm and pay set-up and minimum run charges.

It's been pointed out on any number of occasions that learning how to surface mount chips is no big deal. I have no doubt that that is true. What I do know is this. When you insist that audience know how to use Linux or at least how to do Linux-like builds on a Windows machine in order to get a mainstream Reprap machine running at all and when you insist that in addition to that that they learn C, not exactly the most user-friendly compiler out there to make changes to your firmware, adding to that learning how to do surface mount soldering seems to me to unnecessarily create a positively Himalayan set of barriers to people building and modifying mainstream machines.

That is not at all to say that there are not a lot of people out there who either already possess either some or all of those skills who are also interested in 3D prototyping. There palpably are or Reprap Darwin wouldn't be the amazing success story that it is.

I just keep coming back to my archtypical bright twelve-year-old. While there are a lot of them out there, the number that have knowledgeable father figures who can help them over the bumpy bits on the path to success is limited and getting smaller by the year as our society ... um ... "progresses". That's why I stick to Windows, which the twelve-year-old probably already has on his or her PC, Basic, which is trivial to learn for all its limitations and DIP chips.

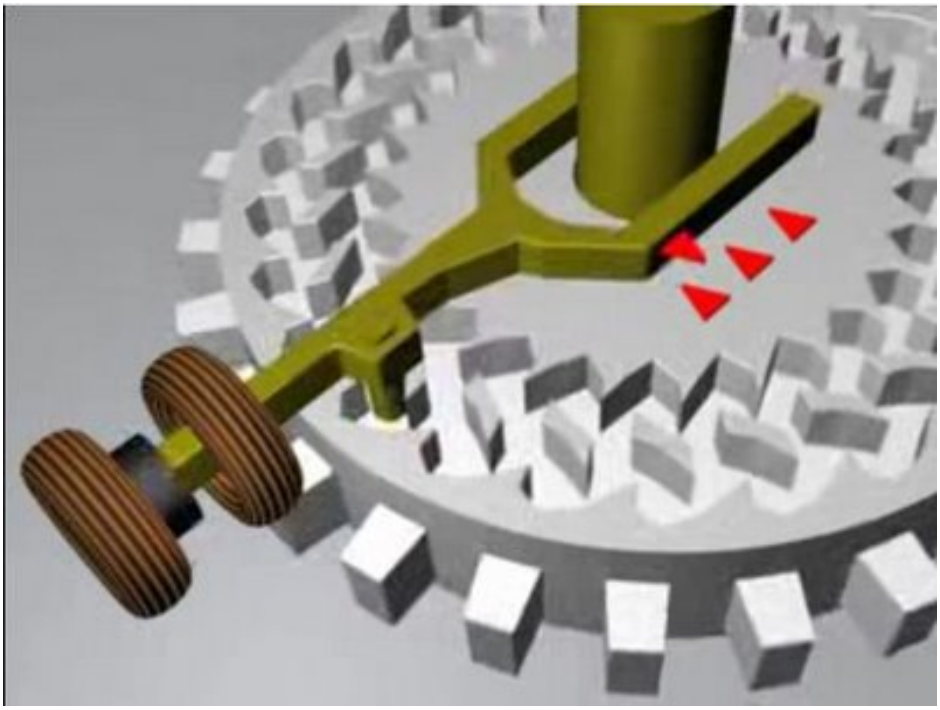
Going for surface mount chips, for all their advantages and for all that the chip industry sees them as the future, for me is just adding another layer of difficulty to an enterprise which is already more than difficult enough.

Pushing the limits

Wednesday, 4th February 2009 by Forrest Higgs

In which your narrator pushes tests the envelope of both Art of Illusion and himself in pursuit of a printable stepper motor...

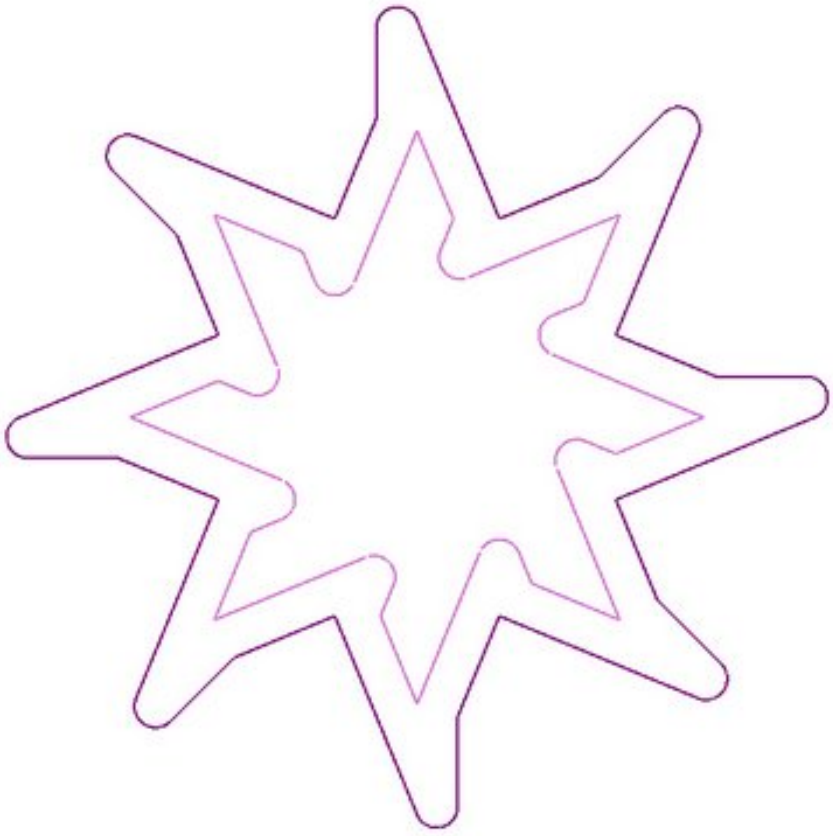
I spent a week's worth of spare time translating Tigertaler's vision of a printable stepper motor's thrust plate into a VB app with the intention of translating that into a Java toolscript for Art of Illusion. By the time I got finished I understood his design well enough to be dissatisfied with it.



Tigertaler's design is brilliant for a fine resolution stepper motor. I found, though, that when I translated his tiered thrust path design, which had the lower tier for one direction and higher tier for the other, to the much lower resolutions that I needed that the difference in the force needed to move the stepper in one direction was very different than that required for the other. As well, assuming a constant velocity for the solenoid, the rate of movement in one direction would be very different than the rate of movement in the other.

As Tigertaler designed his, the differences would not be terribly significant. Anyhow, having used Tigertaler's design for a learning exercise, I sat about to try to design something more suited to what I want to do. In doing so, I developed a very healthy respect for what Tigertaler is capable of. That's HARD work, for me anyway.

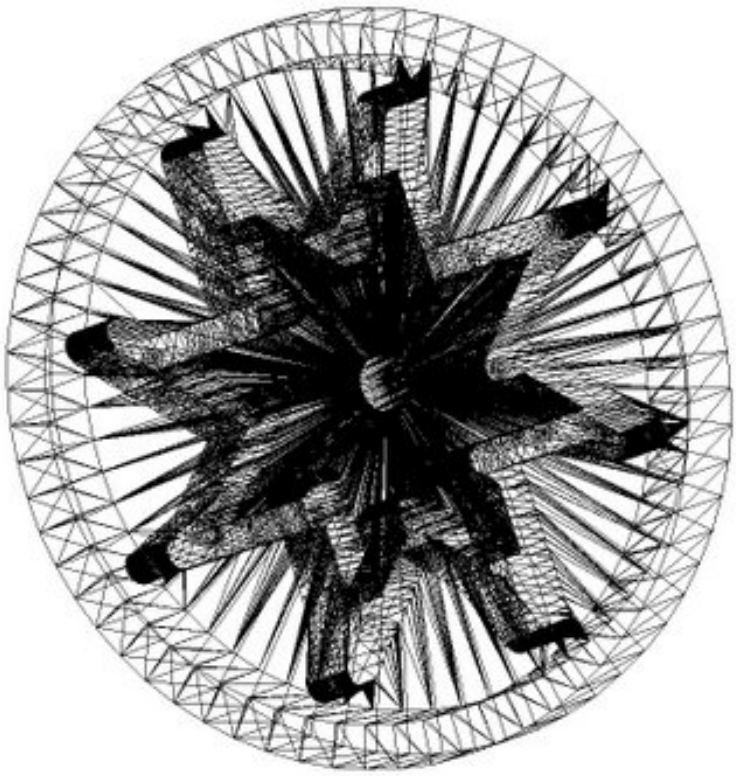
For starters, I designed a 45 degree step angle thrust path. Instead of a two-tiered thrust path pattern, I went for a design that had a mirror image of the path for one direction on the other side of the thrust plate. My design selects direction by setting down the solenoid thrust nub into the thrust path of the appropriate side.



The thrust path looks deceptively simple. It isn't. It took me over a week's worth of evenings to generate the code to properly describe it. As well, it took moving Heaven and Earth to get Art of Illusion's boolean option to work with it. I've come to the conclusion that I am going to have to grit my teeth and admit that the critics of AoI like Nophead and many others have got the right of it. It just isn't powerful enough to do what I want to do.

In any case, I finally got an assemblage made in AoI that I can use with the milling options in my Slice and Dice app. Here you can see a shaded expression of the milling assemblage.

To give you an idea of just how nasty this thing is, I'm including a wire frame view of the same assemblage.



I've tried running it through Slice and Dice and I'm going to have to a little more cleaning of that code to get a good set of milling roads, too.

Milling the starfish

Friday, 6th February 2009 by Forrest Higgs

In which your narrator chooses between getting on with designing the millable stepper motor or learning another 3D CAD system...

In our last episode, I'd finally got AoI to generate the STL files that I thought I needed to get the 45 degree step size stepper motor milled. In fact, when I pulled the STLs that you saw then into my Slice and Dice app, more often than not Slice and Dice simply wouldn't generate clean end mill roads. Although the STL files visually looked pretty good, they weren't clean enough for Slice and Dice.

Several well-wishers suggested different 3D CAD apps and I reviewed several of them last night. I can see that I might need to switch apps, but decided that I'd much rather get on with the millable stepper motor design project first. First, I went in and did a bit of housekeeping on the transfer file of loop coordinates from the VB app and then I increased the step size to 72 degrees from 45 in the hopes that the simpler boundry profiles for the thrust plate would be within AoI's present capbilities.

Indeed, that was the case. I don't really think that this five step/360 degrees stepper will be useful as such, but it will allow me to explore the possibilities of this design approach and perhaps even result in a working stepper.

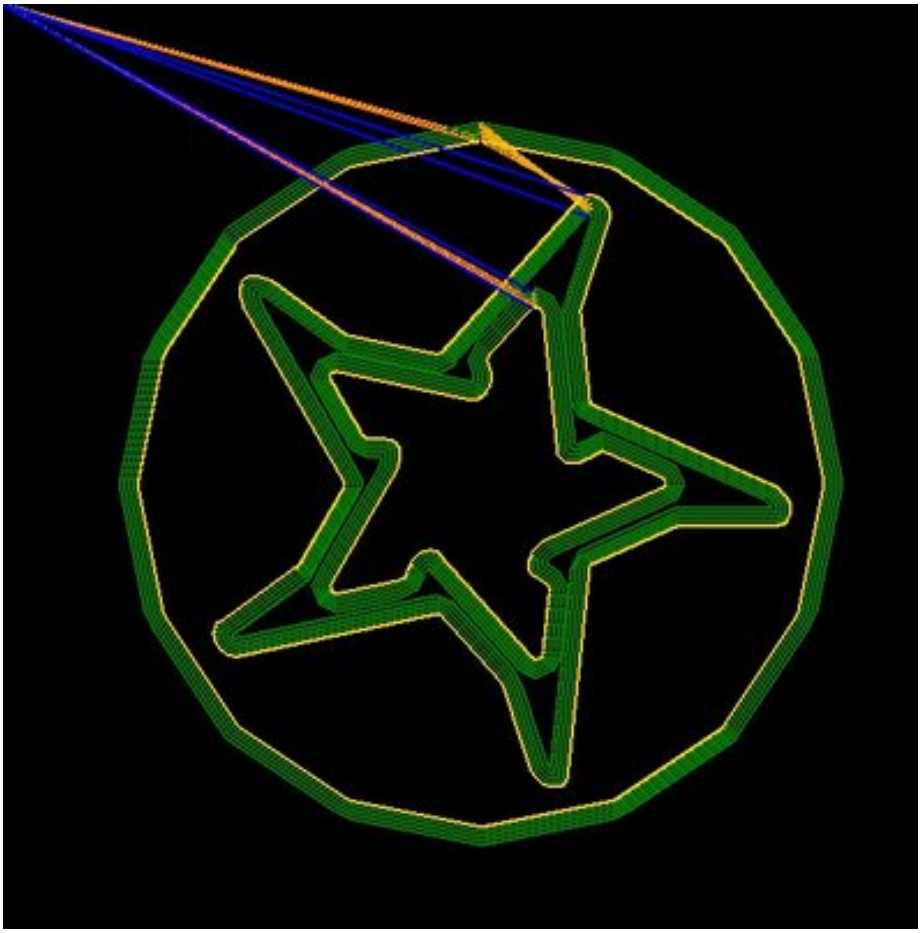
Once I brought this starfish shaped thrust plate into Slice and Dice, I discovered that the code that I'd written to develop milling roads, while adequate for cutting involute profile gears, was really not up to scratch for milling this thrust plate. Whereas with a gear I could cut away from the gear profile and then do a polishing cut after resetting, with the thrust plate I had a groove which required an accurate surface on both boundary walls instead of one.

To complicate matters, the groove was of varying widths. This tended to create interference between milling roads being developed from either side of the groove and also to leave blind artifacts which had to be treated as separate milling exercises. Needless to say, I had some work to do on the milling code.

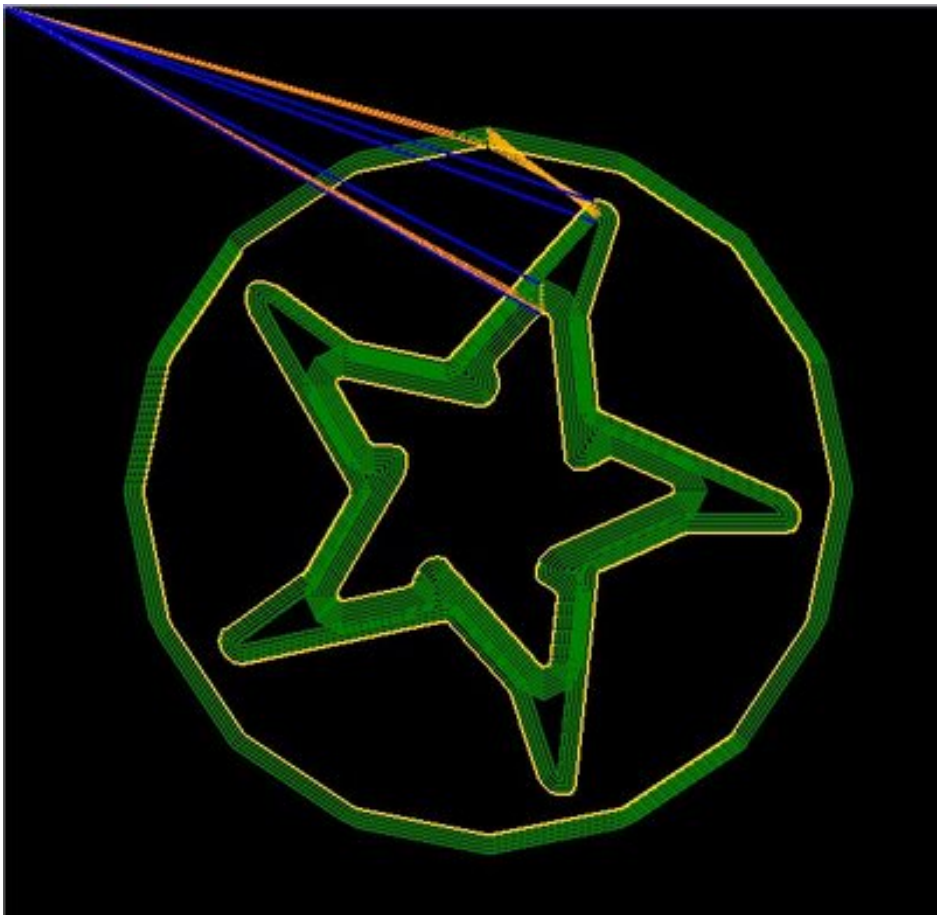
My first feeling was that the code was going to need a major rewrite. A moment's reflection led me to realise that this was the same sort of situation that I'd earlier dodged with the inadequacies of AoI. I didn't want to do a massive rewrite of the milling app. I wanted to get the bloody stepper going.

Heretofore, the milling app had spend quite a bit of time moving from one set of milling paths to another. Indeed, it spent about as much time doing that as it did actually milling. It struck me that if I treated each perimeter in the thrust plate as a separate milling exercise I might afterwards be able to overlay each separate operation and get a proper milling job done.

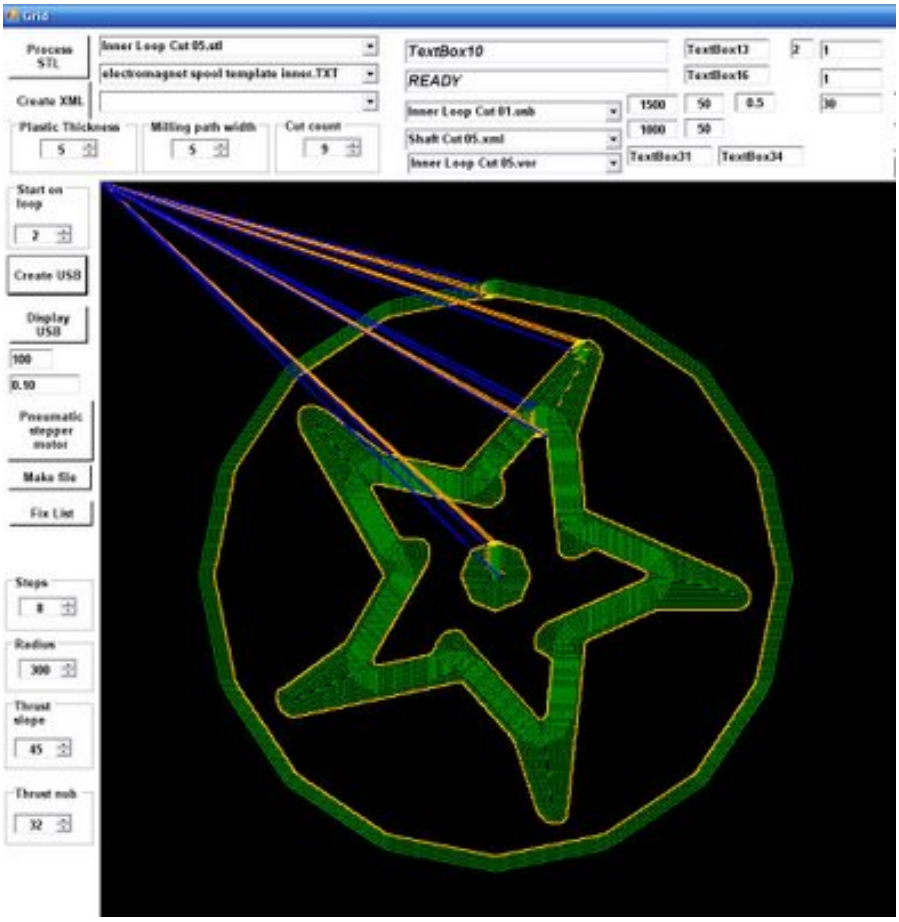
Here you can see my first attempt at overlaying milling exercises.



I didn't specify enough milling roads in this attempt. It does serve, however, to show how the area that needed milling out tends to eventually generate a variety of blind guts as the two sets of milling paths meet. I upped the path count for the inner and outer loop of the thrust path and had another go.



At this point I discovered that the outer perimeter of the thrust path degenerated into pointy paths after several cuts. These sharp changes in direction challenged the cluster analysis routine that I used to find and determine a single direction for a milling path. Once I loosened up a parameter in that routine, things went swimmingly.



For this particular milling exercise, I now have a good set of milling instructions. I've also contacted Peter Eastman, the head guru of the Art of Illusion project, about the problem with AoI's boolean routine. I posted a description of the problem on AoI's help forum and he began a dialog with me with me in about 15 minutes. One very big thing in AoI's favour is the committment of the developers to the problems of the user community. With a little luck, I will be able to look forward to a more powerful boolean op function in AoI.

AoI gets extra innings

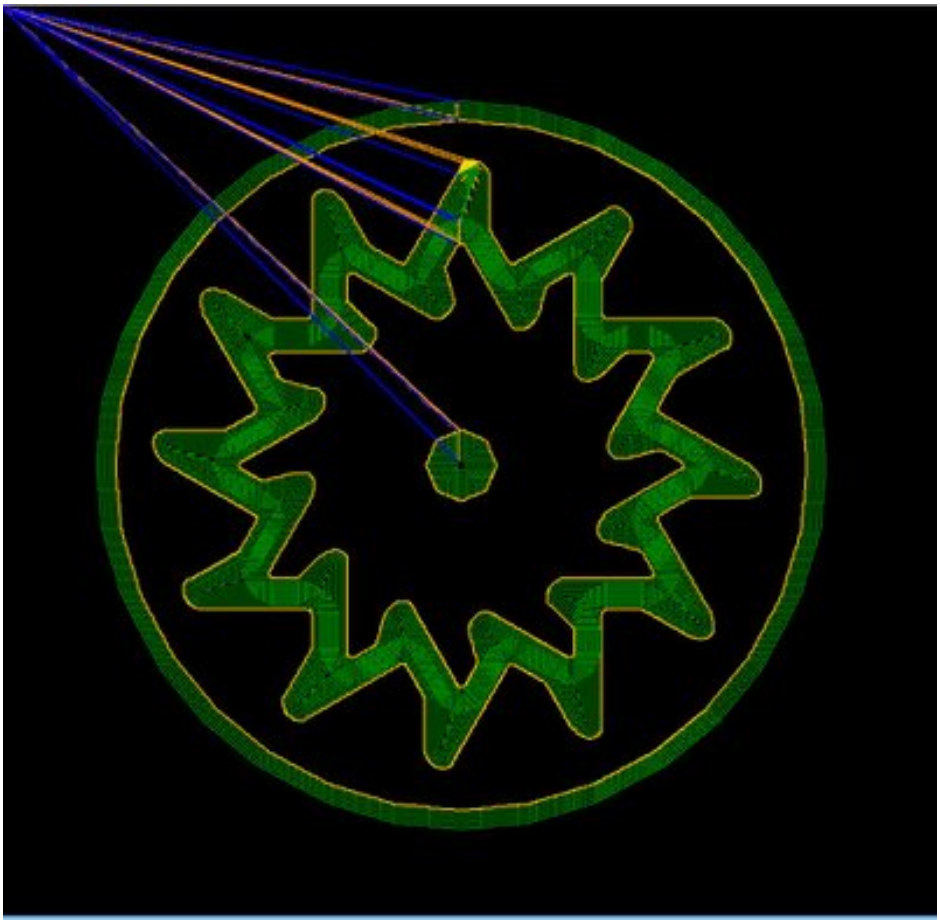
Friday, 6th February 2009 by Forrest Higgs

In which your narrator gets to have his cake and eat it, too.

I had had considerable trouble getting Art of Illusion's **boolean ops** feature to put together the solids that would enable the Slice and Dice app to do its job properly. Just before I went to bed I posted a description of my problem over at Art of Illusion's help forum and linked a copy of the aoi file that demonstrated it.

This morning, I found that Bob had diagnosed and solved my problem. It turned out that I was using the **simplify mesh** option at too coarse a resolution. I had wrongly assumed that if the simplify mesh option didn't reduce the number of vertices it wasn't improving matters. In fact, the mesh can be improved even if it keeps the same number of vertices.

I decided to give Bob's solution an acid test and designed a 30 degree step angle thrust plate at lunch. The boolean ops feature of AoI handled it without a hiccough.



I am going to try milling this tomorrow.

More steam punk

Tuesday, 10th February 2009 by Forrest Higgs

In which your narrator makes a first try at building a fluidic amplifier...

For various reasons I've wanted to get into pneumatic and hydraulic controls for quite some time. To make either of those happen you've got to have something like a solenoid actuator. I got a lot of experience with solenoid valves over twenty five years ago in both Sweden and South Africa. Not to put too fine a point on it, I'm not exactly fond of them. They tend to draw a lot of power, introduce vibration into your piping system and, in hydraulics, introduce water hammer into fluid columns.

Recently, I got interested in both air muscles and the notion of a printable stepper that could be run by either a solenoid or a pneumatic or hydraulic actuator. When I started looking at solenoid valves in the US all the problems with runaway budgets that I'd got into with Danfos in Sweden came back to haunt me.

The cheapest solenoid valve I could find ran about $15. As well, it needed 24 volt power (AC or DC wasn't specified). My ambition is to make a full telepresence 'bot using air muscles rather like Shadowrobot is doing.

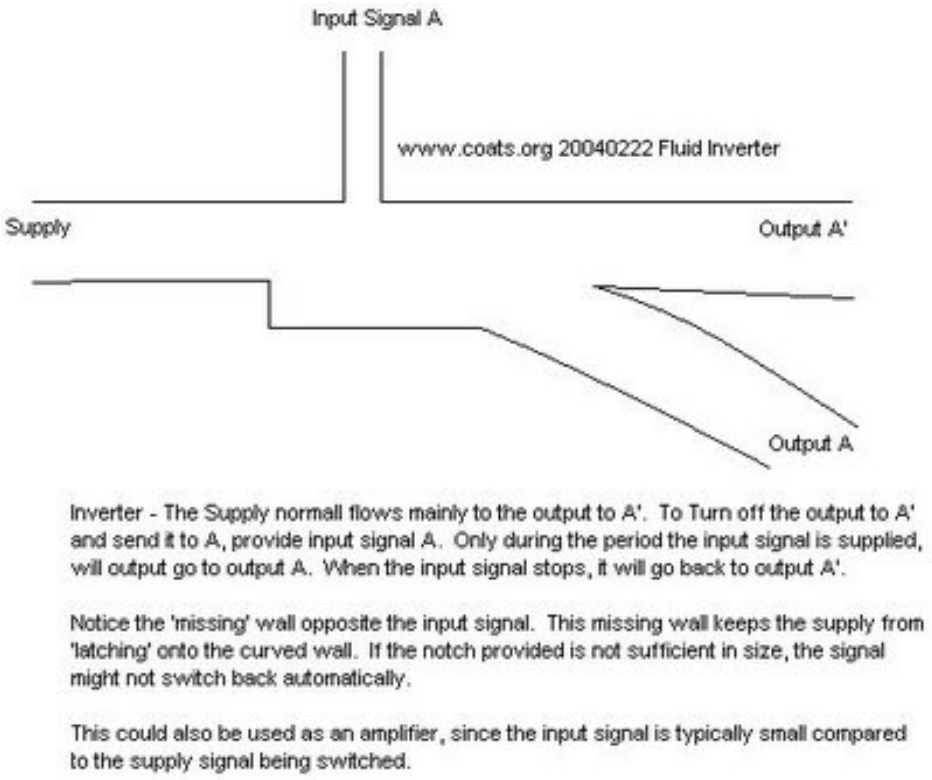
Unfortunately, just to run a hand takes 40 air muscles, each of which would need it's own solenoid valve. IIRC, ShadowRobot wants something like $70K for their hand. Just the solenoid valves would cost over $600, never mind everything else. Mind, air muscles are rather cheap to make. If I

can figure out how to control them economically much might be achieved.

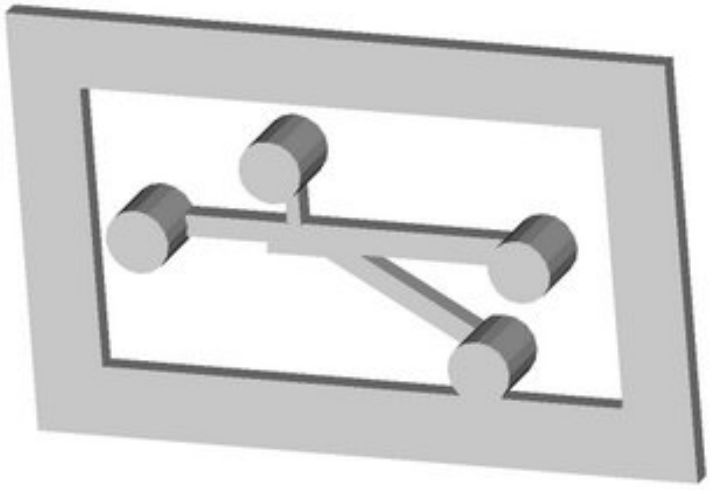
Obviously, we need a very small solenoid valve that can run off of 5 v or 12 v DC power. I'm working on a millable design for that. It would be nice, however, to be able to magnify the air or water flow. That's where fluidic logic comes in. The fluidic triode, the equivalent of the transistors that we use to shunt electricity around on the Reprap controller boards, was developed in the late 1950's. A small solenoid valve and a fluidic amplifier would be a very hot combination.

Thus, I started hacking fluidic amplifier designs. For a start, I milled out a copy of an illustration that I found on the web, viz...

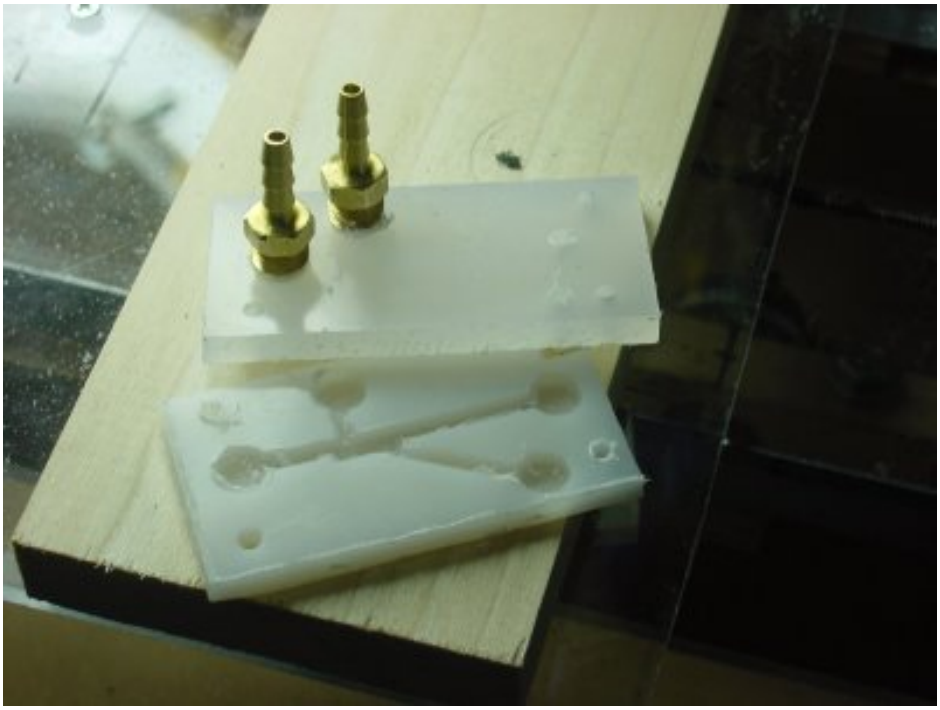
-



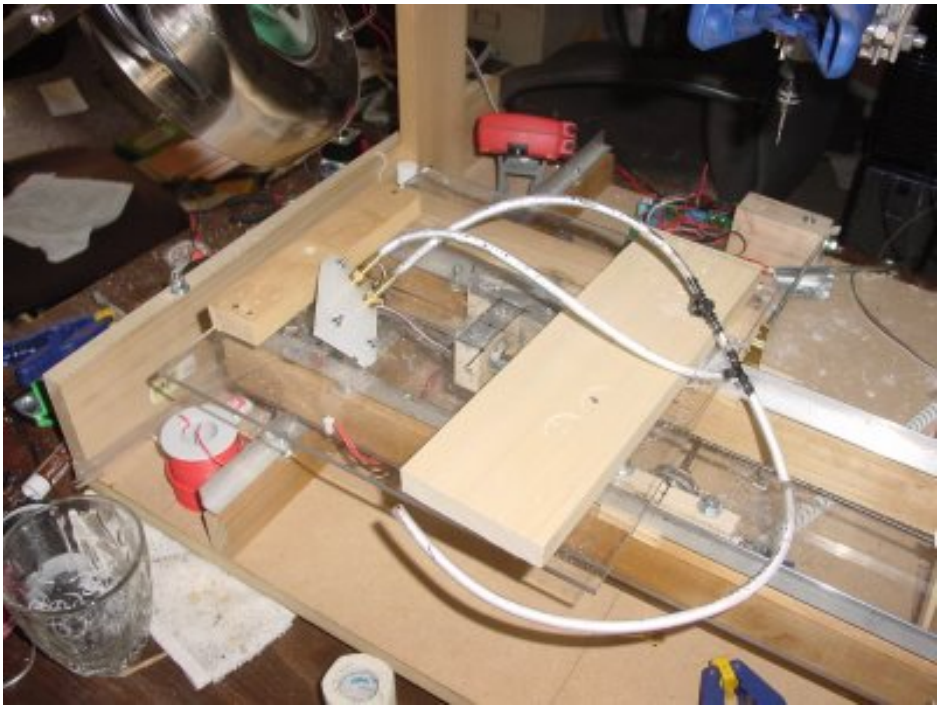
As shown, a fluidic inverter can be made to act like an amplifier. I mocked one up using the illustration as a starting point.



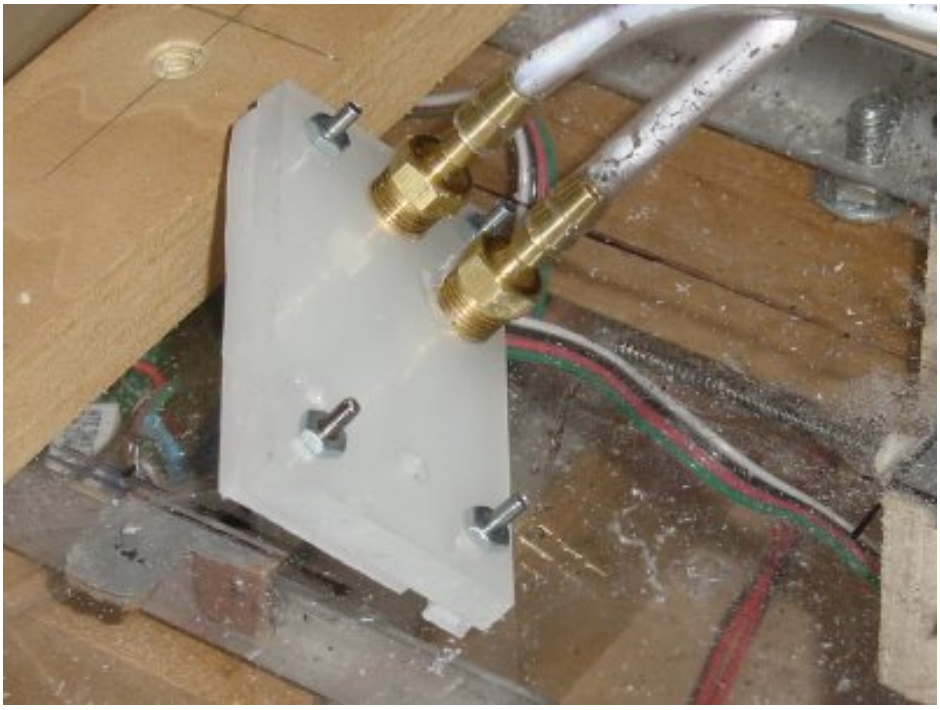
And a few hours later had a milled equivalent.



I rigged it for testing.



And checked it with compressed air first and then with pressurised water.



I used some brass pneumatic barbs to connect the compressed air and valves and t-joints ordiarily used for drip irrigation system plus quarter inch tubing of the sort you use with your aquarium. All were things that you could get at your local hardwares store.

Results were less than spectacular, but encouraging all the same. I had no manometer but it it definitely felt like I was getting a shift of pressure with the application of the control stream. I was unable to quantify that, though.

I next attached it to my Water Pic and tried pressurised water. With the control stream off the stream coming from the straight-through path was twice as large as the one from the diverted path. With the control stream on, the two emerging streams were roughly the same size.

It's a start.

A note about milling polypropylene

Tuesday, 10th February 2009 by Forrest Higgs

A few observations after several attempts to mill polypropylene over the past month...

I've about come to the conclusion that there is no practical way to mill polypropylene without some sort of coolant.

While it has a high melting temperature, it has a low glass transition temperature. what this means is that it goes somewhat fluid when you try to mill it. It is also quite sticky and binds to end mills and drill bits with depressing alacrity.

I've made half a dozen attempts to find anything resembling a sweet spot for this material on Tommelise 2.0 with disappointing results. It may be that you can mill it successfully at very low end mill speeds coupled with very slow transition speeds. Since my Dremel's minimum speed is 5,000 rpm, however, this doesn't appear to be possible for my machine.

Last night I went back over the literature and with all this in mind and discovered what I suspect is a very revealing paper that explores the problems in structural damage encountered with polypropylene when one cryogenically mills it, viz, sprays liquid nitrogen on it before using the end mill. That someone would resort to this sort of machining practice seems to me, at least, to confirm my observations.

At some point in the future milling polypropylene with a coolant might be a good idea. It is the cheapest plastic on offer that I know of. To give you an idea, compare the relative costs per cubic inch...

- ABS - $0.31
- HDPE - $0.12
- Polypropylene - $0.08

The economics are very compelling.

First thrust plate for a printable stepper motor milled

Tuesday, 10th February 2009 by Forrest Higgs

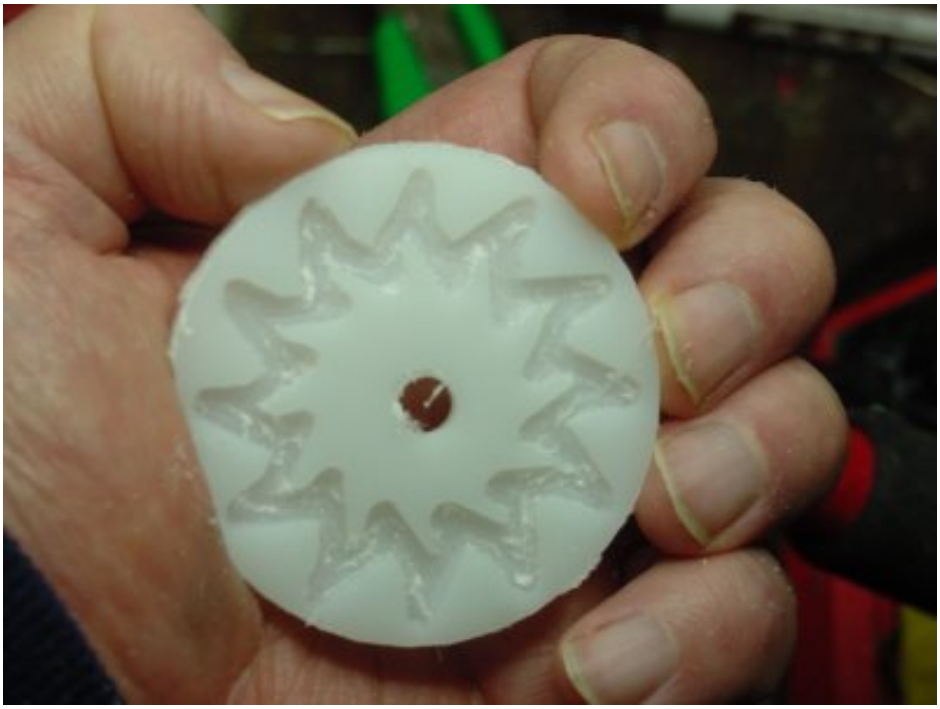
In which your narrator turns plans into product...

I'd like to say that milling things has gotten to be routine. It isn't quite yet, though it's a lot more routine than it was six months ago.

I was milling the thrust plate for my pneumatic stepper motor and had got about 75% of the way through when I decided that I needed to take a few video clips and snapshots of the thing taking shape. Things went well until while taking a last set of shots the strap on the camera got snagged on the corner of the xy milling table without me noticing. When I stood up for a final shot the strap pulled the milling table off it's tracks and the end mill started chewing through the thrust plate at an alarming rate.

I'm taking that damned strap off of the camera!

I set up another milling job during lunch and a few hours later it had finished out pretty much without incident.



I'd worried that milling was going to be so hands-on that I'd only be able to do it nights and weekends. Aside from getting up every ten minutes or so and blowing the swarf out of the cut, that is increasingly turning out not to be the case. Mind, if I am to do useful work, I have to put on a set of ear protectors that in earlier days were used at the pistol range or my PC headset and tune in some music, but other than that there is little distraction.

I gave the thrust plate a quick test with a crudely-made thrust piston consisting of a scrap of HDPE and a #4 bolt. When Vimeo finishes uploading a short video clip of that little adventure in a few hours, I'll include it in this blog entry.

A quick test of the printable stepper thrust plate from Forrest Higgs on Vimeo.
It misses a few steps. That was a result of my equipment, however, not the design per se. I may want to redesign this thing for a shorter thrust distance.

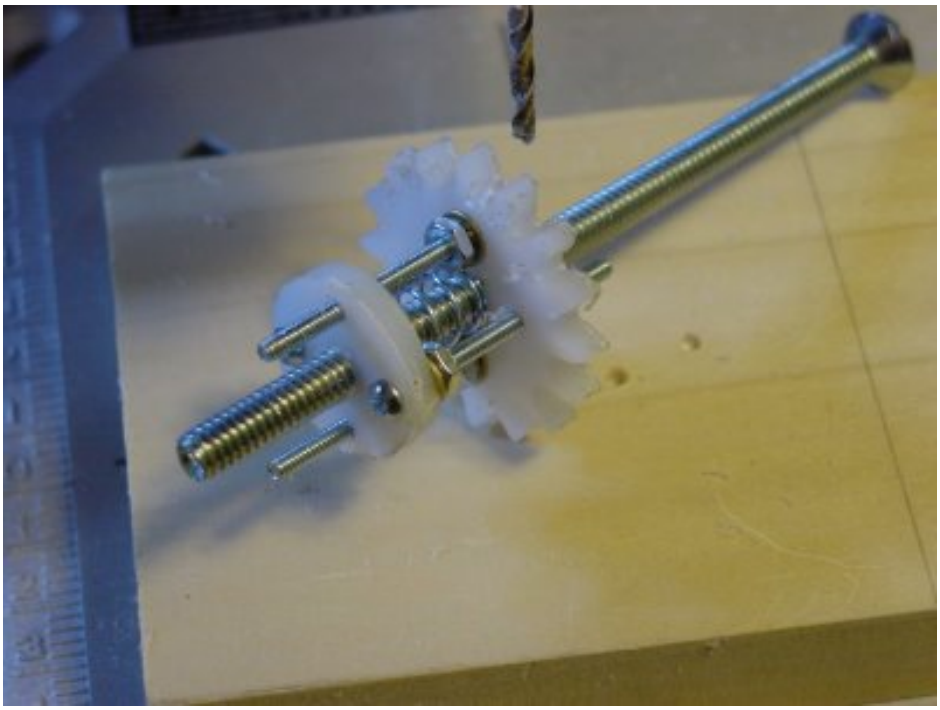
The next task will be to design and mill a proper piston and thrust shaft and get this whole prototype working on compressed air.

A much simpler approach to the backlash problem

Wednesday, 11th February 2009 by Forrest Higgs

In which your narrator has a "doh!" moment...

I had a classic Homer Simpson "DOH!" moment just as I was going to bed last evening. As you know, I spent about a week designing and milling a very credible anti-backlash thrust nut for lead screws using common threaded rod.



In its finished form it consisted of four milled pieces of HDPE sheet and a small handfull of hardware. It works fine. The concept is sound, primarily because it is a knockoff of an industry standard anti-backlash nut. Pretty, but not much creativity involved.

I already knew from Tommelise 1.0 that you don't need antibacklash nuts of any sort on the z-axis (vertical) because the weight of the xz tool mount provides the constant pressure that the springs would provide for the x and y axes.

A lot of Reprappers are building McWires and variations on that theme. They face both an antibacklash problem and the problem of dealing with the small, but significant amount of axial play that virtually any stepper motor has.

Why not just tilt the x and y axes about 5 degrees each so as to provide the slight gravitational pressure that the z-axis enjoys and eliminate the need for both anti-backlash nuts and thrust plates to counter axial play in the stepper motors?

DOH!

There is absolutely no reason for the xy working table on a lead screw design reprap machine to be absolutely level.

Oh well, I got a lot of experience in the ins and outs of milling things during the week I spend designing and building that anti-backlash nut.

Picking up speed

Sunday, 15th February 2009 by Forrest Higgs

In which your narrator reduces the time to mill a thrust collar gear by just over 88%...

Not to put too fine a point on it, I had no prior experience in running milling machines, never mind CNC ones. As a result, I've had to depend on what I read and what I've gleaned from conversations with much older hands at this game.

Heretofore, Nophead has been my mentor in this since he was the first on the Reprap team to have used his repstrap machine to mill parts for his subsequent work on reprap machines. The approach he'd taken was to use shallow fast cuts of maybe a few tenths of a millimeter depth done at high feed rates to minimize the potential for swarf melting and adhering to the end mill. I've been following that paradigm, more or less.

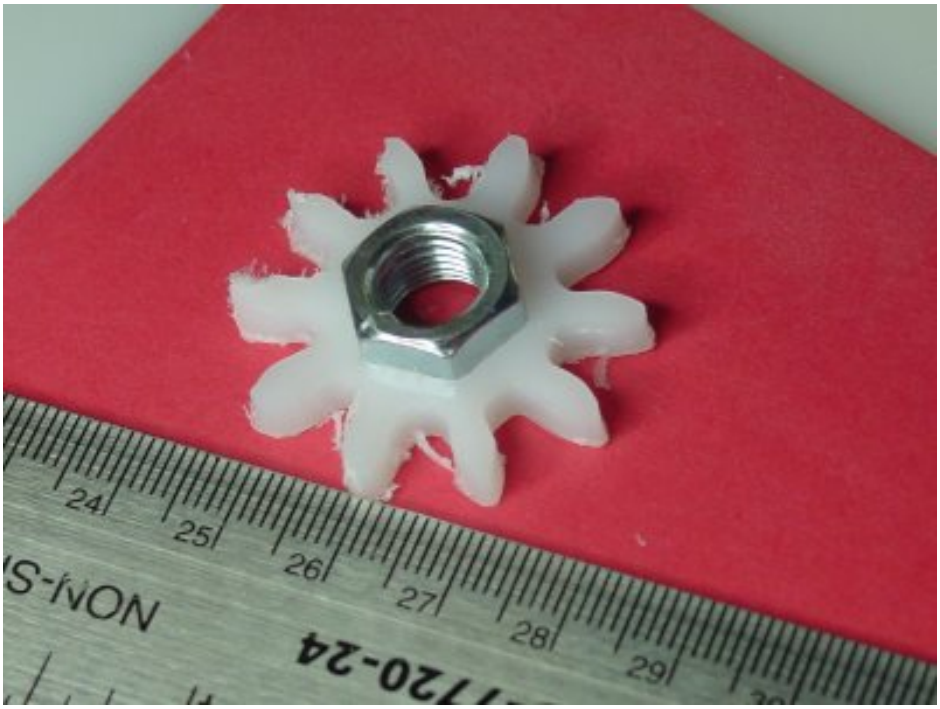
Fairly early on, however, I ran across this very disturbing video clip on YouTube...

Admittedly, these guys were using a heavily built, professional CNC milling machine. Note, however, how they just plunged the end mill into the HDPE and reamed out the hole they were making. At that time, I was using Dremel end mills which looked not at all like what the Tornach people were using. After I became acquainted with North Bay Technical, however, I began using, in a very small size, exactly the kind of end mill that the Tornach people were using.

Even so, I kept on using the light, shallow cuts paradigm.

I have Tommelise 2.0 set up where I can adjust the cut depth. A few times I made mistakes in this setting and did some unexpectedly deep cuts. Surprisingly, disasters did not ensue. About the only difference was that the deep cuts created huge windrows of swarf. I'd observed that I could take cuts as deep as a millimeter and not really stress T2.

This morning, I decided to see if I could cut a whole gear using 1 mm (actually 1.2 mm cuts). As you can see, it works.



I did this milling job with 4 cuts of 1.2 mm. Each cut took 8:45 minutes with the feed rate at a nominal 12.5 mm. I had done this piece before using 0.2 mm cuts at a feed rate of a nominal 8.3 mm/sec. That job took a touch over five hours to finish. Today I was able to do the same job in 35 minutes.

That's basically a bit over an 88% reduction in milling time.

The down side of this performance improvement is that you have to use either compressed air to blow away the swarf or a vacuum cleaner to suck it away. Since I already have a compressor, This isn't a problem for me aside from having to stand over the cut while it is underway blowing away the swarf. Fortunately, I found a source of cheap, 12v solenoid valves which will allow me to control this process from T2's controller board. They should arrive this coming Friday, if UPS is on-time and they usually are.

With that enhancement in place, I should be able to work with fewer milling passes. Right now I am using five cuts for this job. I might be able to reduce that to 3-4.

I am currently looking at adding the ability to peck that first plunge into the HDPE. If I can make that happen, it is entirely possible that I can cut one of these gears in about 10 minutes by doing the full depth cut in 1 pass.

That would be terrific.

IR Ranging

Wednesday, 18th February 2009 by Forrest Higgs

In which your narrator dodges the whole issue of computer vision...

As I mentioned some time ago, Tommelise 2.0 has gone into production mode as a CNC milling machine ... until I get bored and build an extruder. Mind, my plane is to use Tommelise 2.0 to make the parts for a much cheaper Tommlise 3.0. In that one, the expensive Haydon linear steppers will be replaced with reprapped ones driven by cheap tin can steppers.

One of my original motives for getting into Reprap was to get the technical background to fulfill a lifelong ambition, acquired at University, to print buildings. Printing large objects presumes that the product is much larger than the tools that make it. My inspiration stemmed from the recently deceased Professor Wolf Hilbertz, whose design studio I took as an undergraduate. Wolf conceived of the built environment as being both automatically constructed by extrusion machines. Similar machines would reclaim the materials of buildings no longer needed and use it in constructing others. His seminal thinking on the subject was published in Progressive Architecture in 1970.

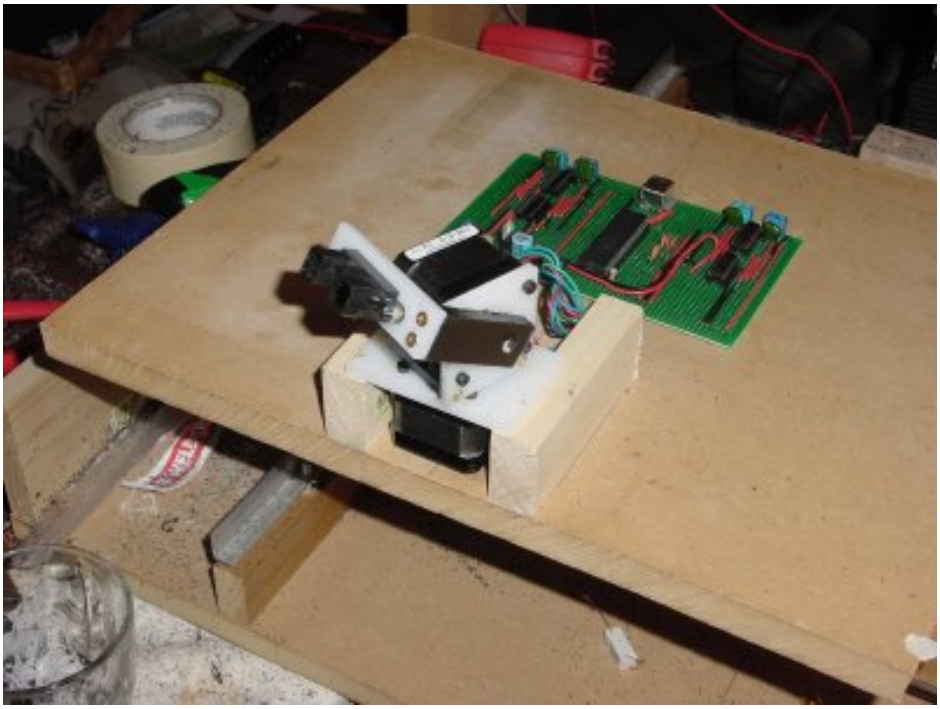
Wolf conceived of quite large machines building whole cities. My ambitions at the time were much more modest. My idea was for very personal spaces being built by a small swarm of machines no larger than shoe boxes. Interestingly, Dr. Bowyer entertained the same notion some years ago and went so far as to get his ideas off of paper and into reality after a fashion. He was kind enough to link me to his own work in this area recently.

One of the big issues when designing extruding 'bots is figuring out where one is. That's nearly so difficult is figuring out where other 'bots in your swarm are so as to avoid crashing into them. About a year ago I happened across a Sharp IR range finding assembly that could, depending on the model, calculate distances out to 5 meters. For starts, I bought a couple of more nearsighted ones that can spot things as far away as 1.5 meters. It struck me that if you put one of these on an alti-azimuth mount run by a MCU not all that different from what we use with our Reprap machines, you ought to be able to map one's surrounds directly in spherical coordinates with very little computing power.

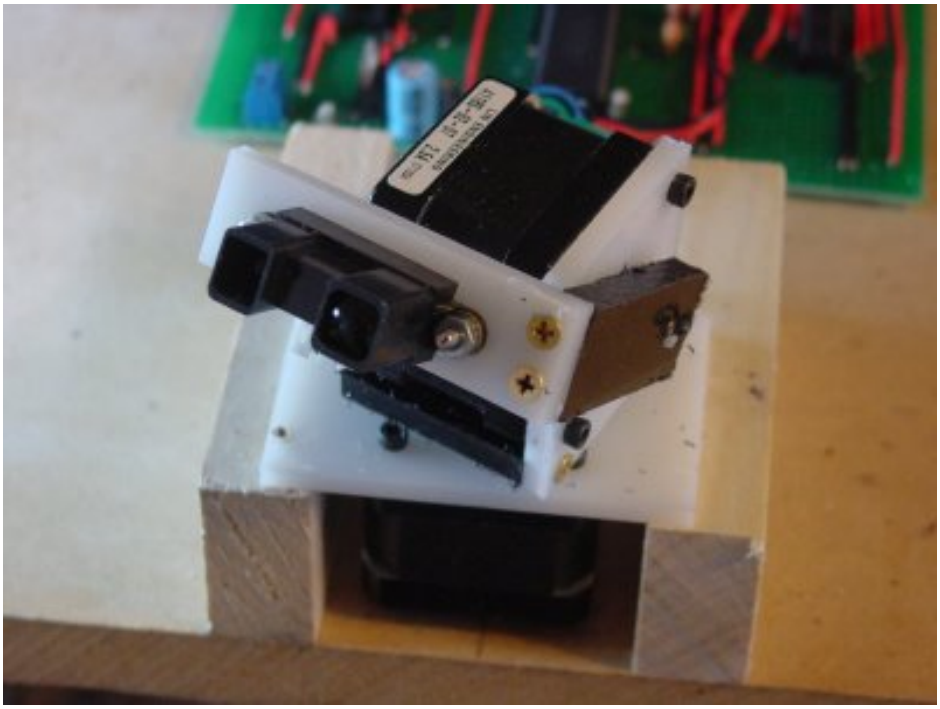
I was at the University of Texas in the 1960's when Professor Agarwal and his graduate student Steve Underwood began trying to reconstruct 3D environments using binocular vision. It was, computationally, horribly costly then. The years since haven't reduced it from being a very wicked problem. How much more simpleminded to simply do rangefinding!

All that had to wait, though, because my first priority was getting Tommelise 2.0 going. I reasoned that with a CNC capability constructing a workable alti-azimuth positioning system would be trivial. As you can see, it was indeed. I designed the parts for a simple system over the weekend and

milled them out in a matter of a few hours.



It is hard to describe the bliss one feels when one knocks out a stepper mounting plate in Art of Illusion using shop drawings of the stepper and after milling it discovering that all of the bolt holes are precisely aligned.
I've leveraged the prototype board that I designed to test the I2C slave chip's ability to run a stepper motor via a 754410 into handling two NEMA 17 stepper motors that I happened to have on hand.



About all that is required on the controller board now is to set a couple of I2C-friendly EEPROMS and finish writing the firmware.

Racks and pinions

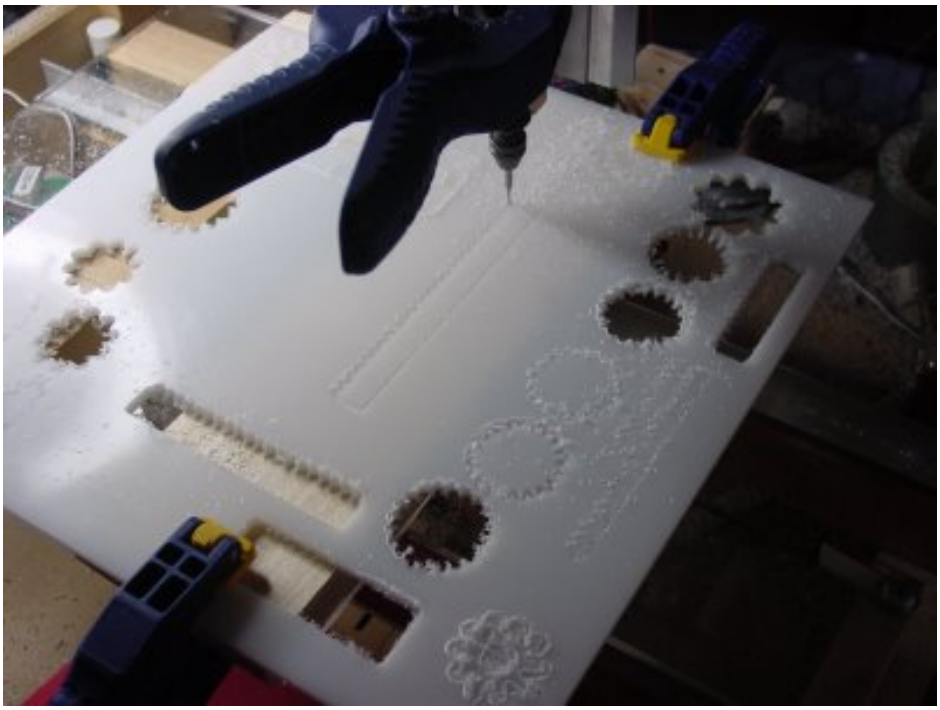
Sunday, 22nd February 2009 by Forrest Higgs

Wherein your narrator explores ways of upping the milled or printed content of future Reprap machines.

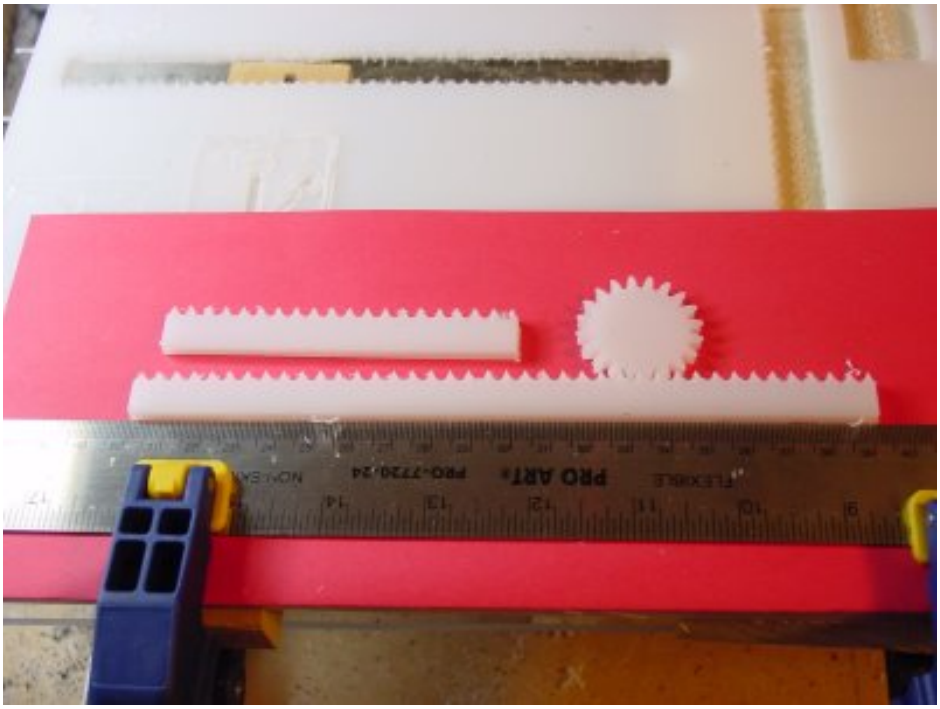
Some years ago I wrote a script in Java for designing involute profile gears on a dare. Having successfully done that, I went on to do another script for designing racks as a companion for the gear script. While I've tried out the gear script on a number of occasions in the past half-year, I'd yet to try the rack script.

Now I have.

I wanted to mill fairly fine toothed racks and pinion gears. Attempts to do that revealed some problems in my toolpath calculations which I've got some way towards solving. As well, I had a scaling problem in the Slice and Dice routine that I knew about but hadn't had an occasion to want to fix till now. I chased through the coding and parameterised the scaling factors. Right now I have it scaled to 240x240 mm at 0.102 mm resolution. That let me attempt a 190 mm rack {200 mm milling area}.



There is no practical reason why I can't do a diagonal cut on a 12x12" sheet of HDPE and get several 15" racks.



I could either tilt the axes slightly to get rid of backlash or offset and springload two parallel strips of rack to achieve the same effect.
Don't take the product you see here too seriously. I cut it with the brutal Airpax steppers that I just took delivery on in mind.

After doing this, however, I've reached the conclusion that the diminutive Jameco steppers would be better suited to this approach.

The Airpax steppers I intend to use for a geared linear stepper approach.

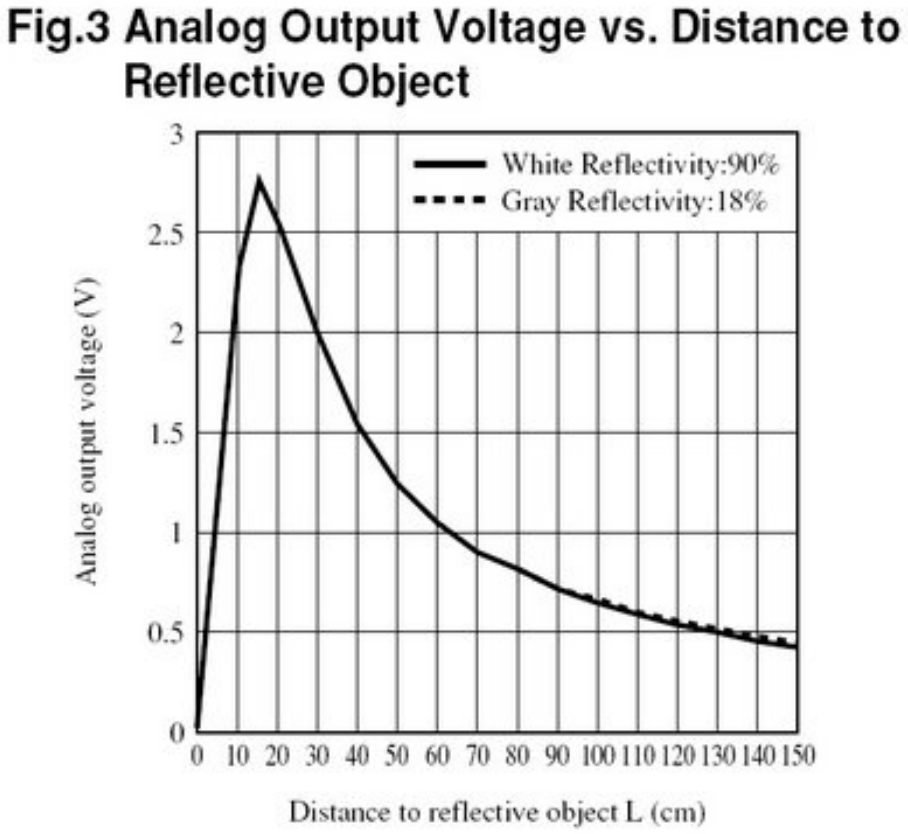
For now we see through a glass, darkly...

Monday, 2nd March 2009 by Forrest Higgs

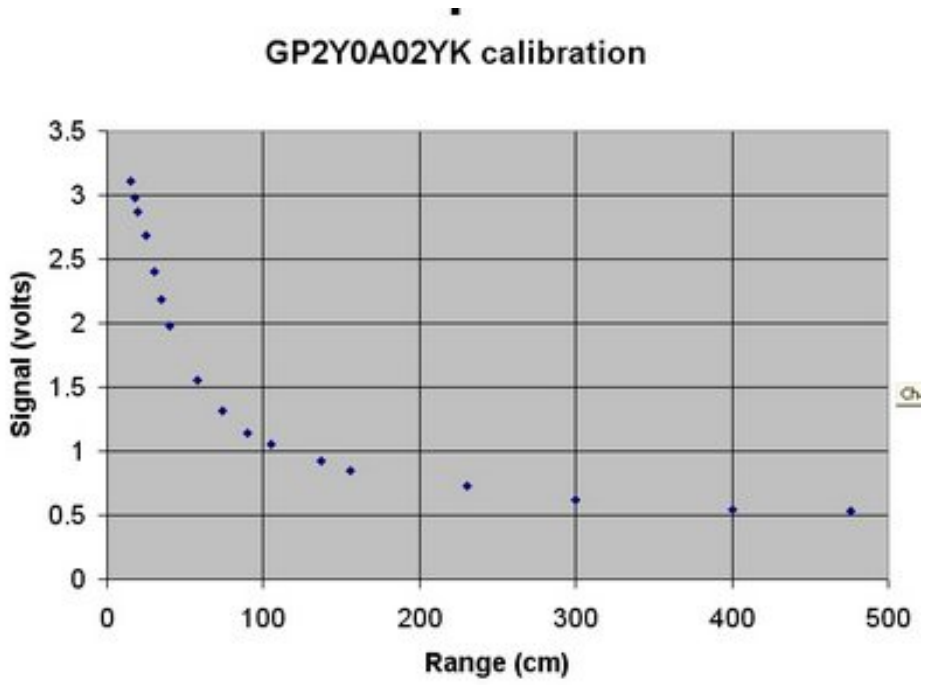
In which your narrator's short-range IR radar set begins to return images...

There are so many things that want doing and so little time available to do them.

This weekend I spent my spare time knocking the short-range IR radar set I'd built using milled fittings from Tommelise 2.0 into shape. The friction fits between the NEMA 17 stepper shafts and the HDPE bits weren't robust enough and slipped, so I tried JB Weld to solid that all up. It answered well.
I won't relate the trials and tribulations of getting the Sharp IR rangefinder integrated into the system. It was tedious, but I finally got things working relatively well and the firmware brought up to speed. It's not perfect, but it works well enough for testing.
The Sharp sensors came from Acroname. They have a little article about how to convert the voltage signal from the sensor into range data. I found it to be totally useless. Here is what the signal vs range profile sort of looks like.



I spent some time calibrating the sensor and derived this chart.



As you can see the one doesn't bear a whole lot of resemblance to the other. One bit I left out was the fact that when you drop below 15 cm in range the voltage drops off. I measured that, but left it off of the chart you see that I created.

The sensor is specified to be good out to about 1.5 meters. In fact, I was able to detect significant voltage differences out to four meters. What my calibration curve doesn't tell you, however, is that in practice, readings of objects further away than 1.5 meters with this sensor tend to be pretty noisy. One interesting phenomena I observed was that at those distances the sensor tends to return distances enormously closer than reality when the IR spot crosses a relative sharp corner of an object. Getting a scattered return tends to spoof the distance measurement electronics.

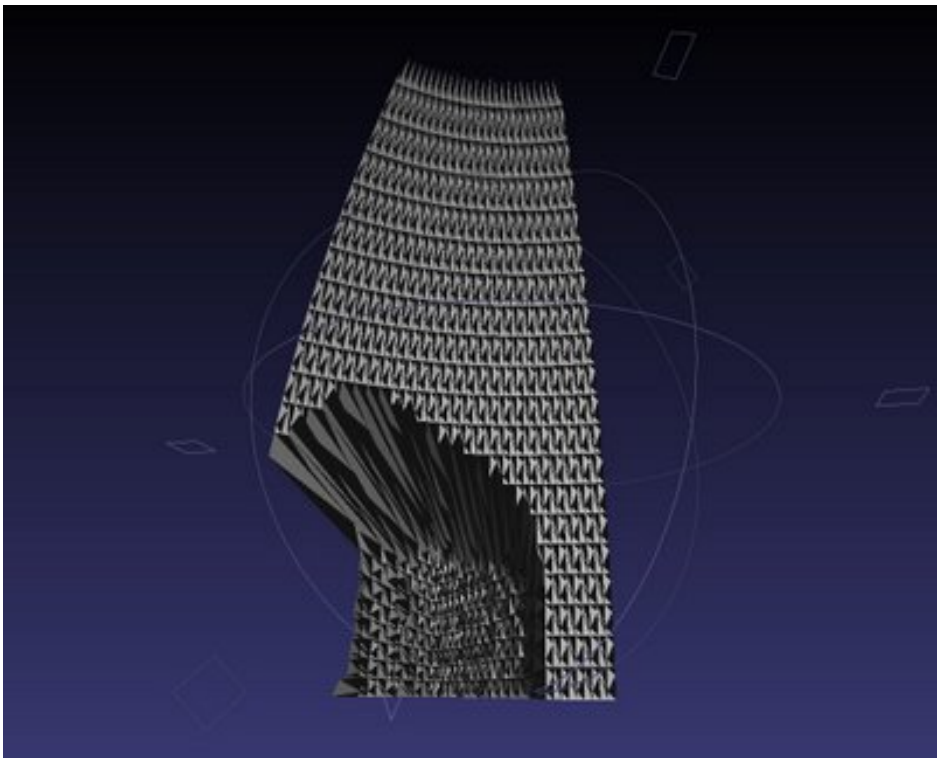
Once I figured that out I tried imaging nearby objects

Here you see a carton I used as an object to scan and here you can see a bit what it looked like to the radar set.

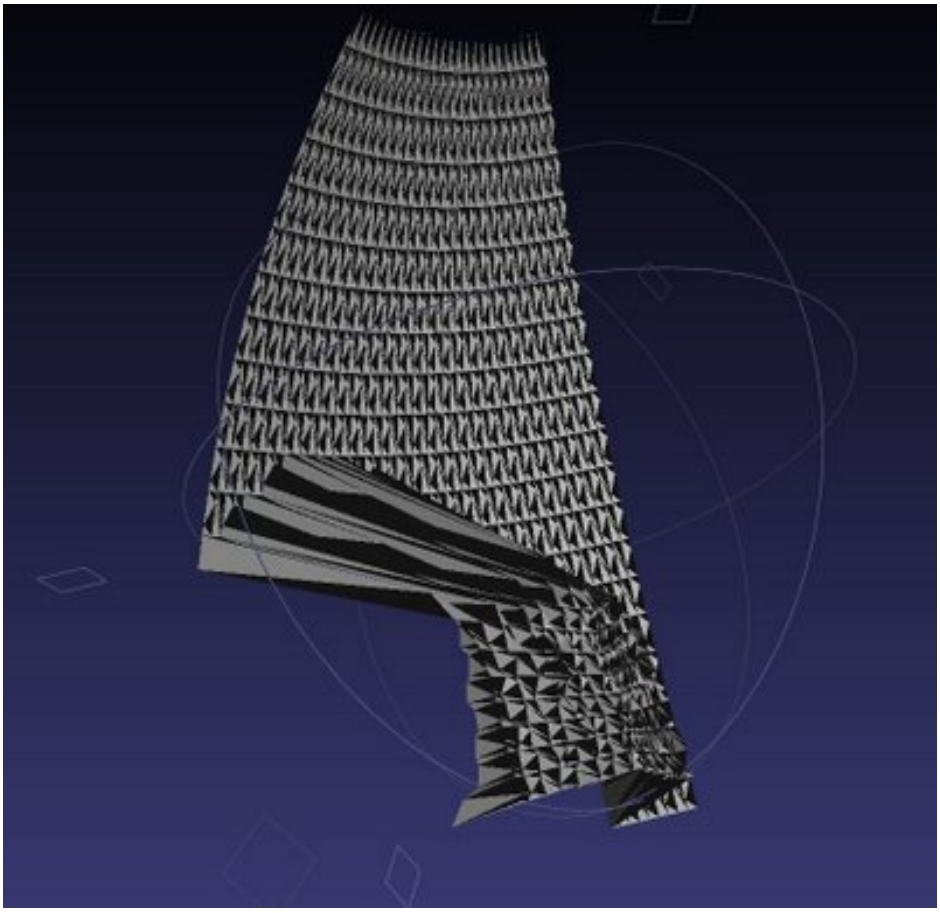
Once I had the data from the scan I realised that I had what is known as a "point cloud" to deal with. Looking around, I couldn't find any apps to turn it into a mesh that I could image, so I finally wrote my own routine to turn the point cloud into an STL. That wasn't as big a chore as I thought it

would be when I resolved to do it. In fact, having my own code I was able to put in a distance filter to remove noisy returns from features more than 1.5 meters away.

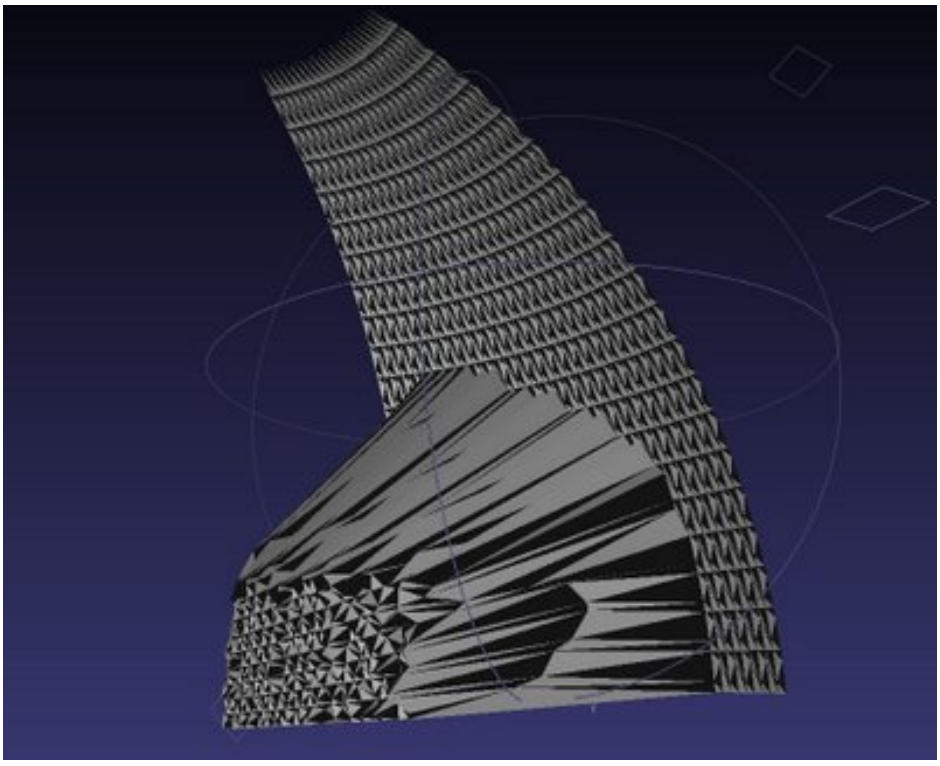
Here you can see the STLs presented in the open source Meshlab app. I was able to get AoI to do the job, but the Meshlab image is easier to look at.



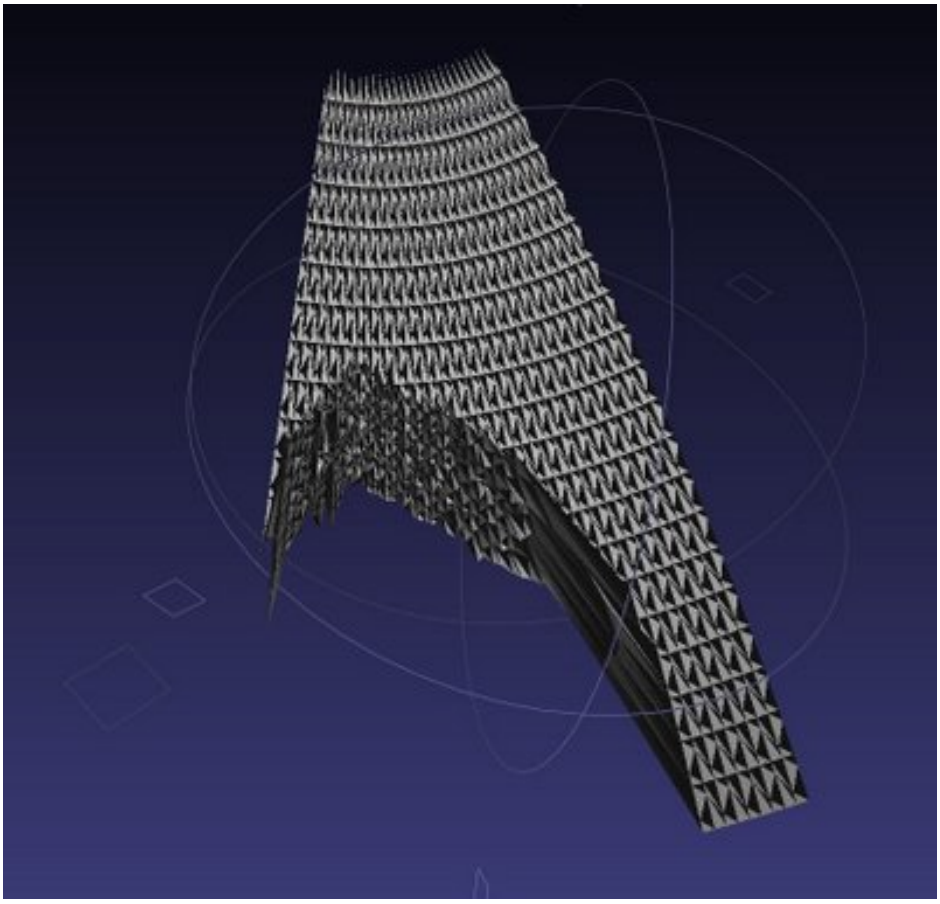
Front view of the carton.



Right side view.



Left side view.



...and the view from the underside.
The image was generated using half-degree steps in the scanner. The sensor appears to have an D:S ratio of about 30:1 which is pretty good. Matt at Acroname told me that you could image the IR spot with a digital camera. I wasn't able to make that happen with mine, however.
The next step I have to make is taking real-time measurements of the reference voltage that I am feeding into the sensor to see if some of the signal noise I am seeing results from my just assuming that it is five volts. Beyond that, I am probably going to buy a few Microchip MCP3425 16 bit A/D chips that are I2C interfaced. At just a few dollars that chip appears to be real value for money. It does reference voltage monitoring automatically as well as giving you 16 bits of accuracy.
Whether that complexification is going to be justified with this sensor, however, is not clear to me at this point.
I am tempted to get one of Sharp's GP2Y0A700K0F sensors which are supposed to really give you good measurements out to 5 meters. They only begin to give you a useful signal for objects further than a meter away, however. I would have to use one in concert with one of their shorter range sensors like the one that I am using now to get full coverage.

Vitamins and minerals

Thursday, 5th March 2009 by Forrest Higgs

In which your narrator takes a bit of a different direction vis a vis the question of replication...

This isn't one of those stories that you that has a beginning, a middle and and an ending. I didn't really start out to do what I'm blogging here. Actually, I wanted to design a printable linear stepper motor. Tommilise 2.0 uses excellent, but expensive (~US$75/axis for stepper and lead screw). They are rock solid and accurate. They do have a bit of an Achilles' heel, though. Their lead screw is a bit thin and flexible for serious milling.

I set out, therefore, to somehow put together a linear stepper with a heavier lead screw. That's where I started. It quickly became apparent that this was a pretty big job and so far I've come at the problem from several directions which I will not recount here. That said, I've been worried for some time about the direction that the mainstream Reprap project has taken in the past year. I commented on that a bit earlier.

When you scrape away all the nonsense, however, my worry has been that Darwin just isn't replicating. It's getting made with laser cutters in short production runs. Not to put too fine a point on it, it has become a disassembled industrial product, not unlike an Ikea sofa. Now that's fine, as far as it goes. It gets a lot of Darwins out in the field and gives a lot more people a chance to participate in the designing of a next generation Reprap machine that actually CAN replicate easily.

Recently, eD at Bath University has been designing something that I think falls somewhere between Darwin and Mendel. More recently still, Vik has been knocking together mockups of eD's design to work the bugs out of it. Vik and I began our respective systems using lead screws made from readily available studding {threaded rods}. Darwin took the direction of using belts. I won't get into the relative merits of the two approaches. That ground has been gone over any number of times in both the blogs and the forums.

What I will discuss is that both Vik and I have been rather... perturbed is probably far too strong a word, but lets leave it there for now, about the dependence of mainstream Repraps on NEMA 17 and NEMA 23 stepper motors. The bottom line is that while they are wonderful pieces of engineering, they're rather expensive and, more importantly to both of us, they're bloody difficult to lay hands on if you don't live in one of the G-7 {I'm not sure about Russia} countries or China.

Now Zach, at the RRRF, has moved Heaven and Earth to get the cost of the NEMAs on Darwin down to something reasonable. US$25 for a NEMA 23 is a brilliant deal ... if you're in the US and the RRRF has stock. If you're not, you find yourself twisting in the breeze. If you are serious distances from the US the shipping cost of NEMA 23's gets to be a real problem. Vik tells me that

you can lay hands on a NEMA 23 in New Zealand for about $100.

On the other hand, tin can steppers are pretty easy to lay hands on pretty much anywhere, either new or salvaged. Vik can buy a pretty nice tin can stepper new in Auckland for about US$14. I can do the same thing for about $12.50. These are typically unipolar puppies with either a 15 degree or 7.5 degree step. With a lead screw approach that kind of stepper can get you considerably better than 0.1 mm positioning accuracy. They are a touch slow, however.

One of the approaches that I took to designing a linear stepper was to take a 15 degree tin can stepper motor, tin can is a pejorative term for permanent magnet stepper, and slap on a gear pair onto it that makes sure that the positioning accuracy is about 0.1 mm instead of a lot less than that. I decided to use 3/8-16 studding which is extremely common here for the lead screw. The gear pair to get a 15 degree stepper motor to yield 0.1 mm positioning accuracy is about 1.6:1.

Vik and I began to informally pursue a tin can stepper/lead screw solution a few weeks ago. I snooped around on the web and scored half a dozen old Airpax steppers on the web at a true Scots-Irish price of $2.50/stepper.

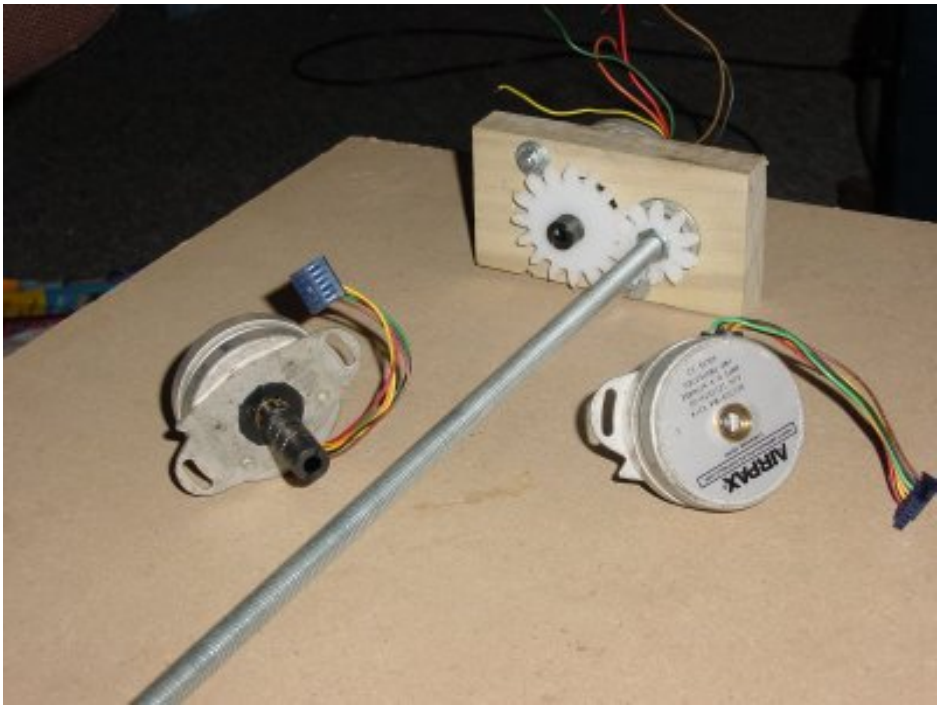
These Airpax steppers are heavy duty ones weighing a half pound each. They are so many of them around used because they were a part of the old DEC minicomputers twenty-odd years ago. You can get this stepper from that era new in the box for about $5. I've found listings for them in Spain and South America among other places. The same stepper is still made by Portescap {part of Thompson and Danaher Motion} for about $23.50. You can get knock-offs from China for less than half that.

This is not to make a case of this is the stepper that should be used. There are a bunch of different

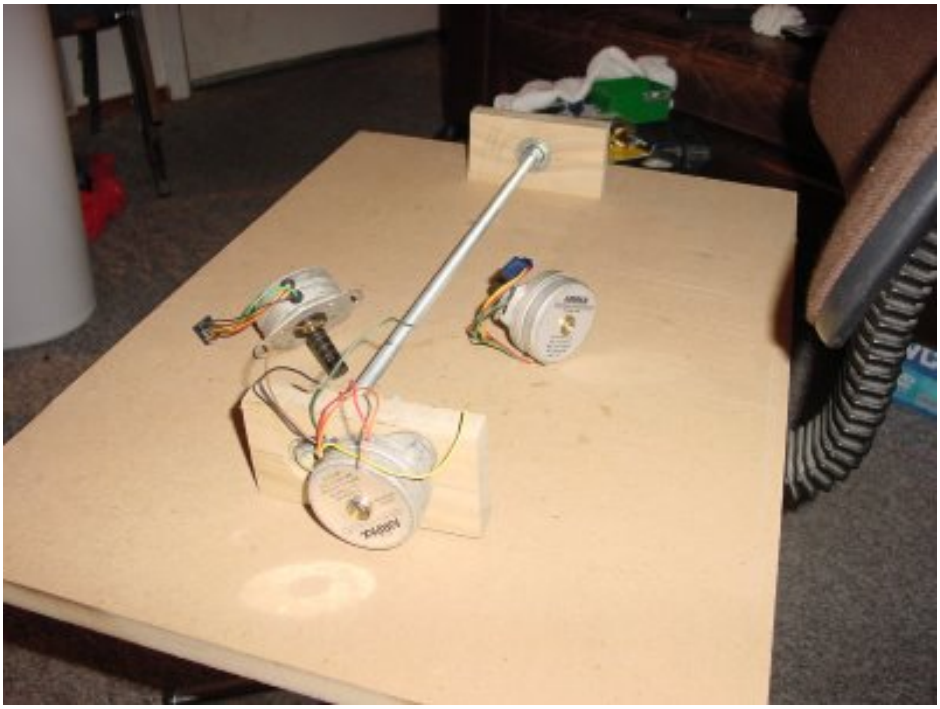
ones that could do the job easily. It just happens to be the one I chose.

Anyhow, I was busily trying to shoehorn one of these Airpaxes into a gearbox for a linear stepper drive when it suddenly occurred to me yesterday that I was focussing on entirely the wrong problem. What I should be focussing on was making something very conventional that was easy to make. Vik favoured using the tin can steppers to turn lead screws rather than turn a nut on a lead screw. A few calculations led me to the same conclusion. The Airpax weighs about half a pound. So does a foot and a few inches of 2/8-16 studding. It really doesn't matter whether the studding moves or the Airpax or the studding rotates. Probably rotating the studding takes the least energy.

With that in mind I took the gear pair that I'd milled and knocked together a lead screw mockup not all that unlike what Vik had done with eD's design.

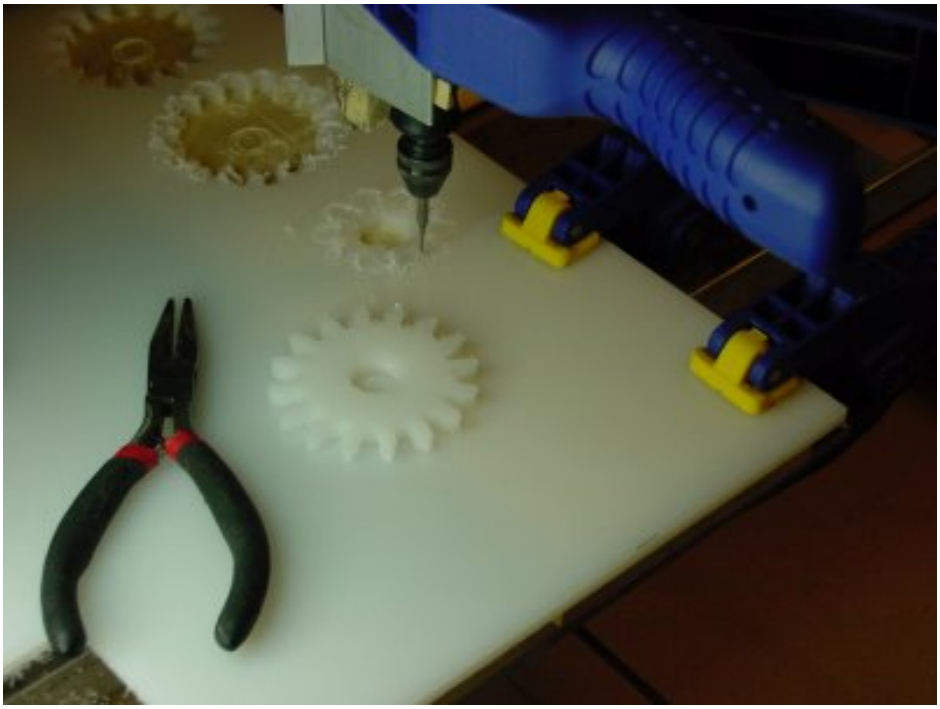


Here you can see the whole ensemble...



Here's where things get interesting. Traditionally, while Adrian has referred to "vitamins" as the bits that a Reprap can't print, viz, things like steppers, IC chips and the like, we've had an informal and unspoken definition of vitamins as being the bits you can't acquire locally. When I first joined the Reprap project at the beginning of 2006, Adriaan kindly forwarded me a set of printed parts for a Mark II Extruder plus a gearmotor to run it. The rest of the Mark II I was able to acquire locally. Eventually, I wound up with Tommelise 1.0 and 2.0 from that little box of vitamins.

Now, look at the first picture of my hacked lead screw drive. The vitamins are that gear pair.



Here you can see me milling the gear pair. It took me about 40 minutes to do the job and would

have gone a bit faster if I hadn't been fooling around. The gears have big teeth, which means that they could be printed on a Darwin by somebody MUCH less able than Nophead.

I'm tooling up to design a gear train that employs a cheap GM-17 gearmotor to run a pinch extruder using the latest Bowyer configuration. The point is that three pairs of gears for axes and a gear train set for a pinch extruder is all that anybody would need to make a next-generation Reprap machine anywhere assuming that they could get salvage or new tin can steppers wherever they happened to be.

The point is that the controller board kit that I use plus these simple set of parts which I could knock out in an afternoon can be put in a padded envelope and shipped anywhere in the world for maybe $10. The other bits could be acquired locally.

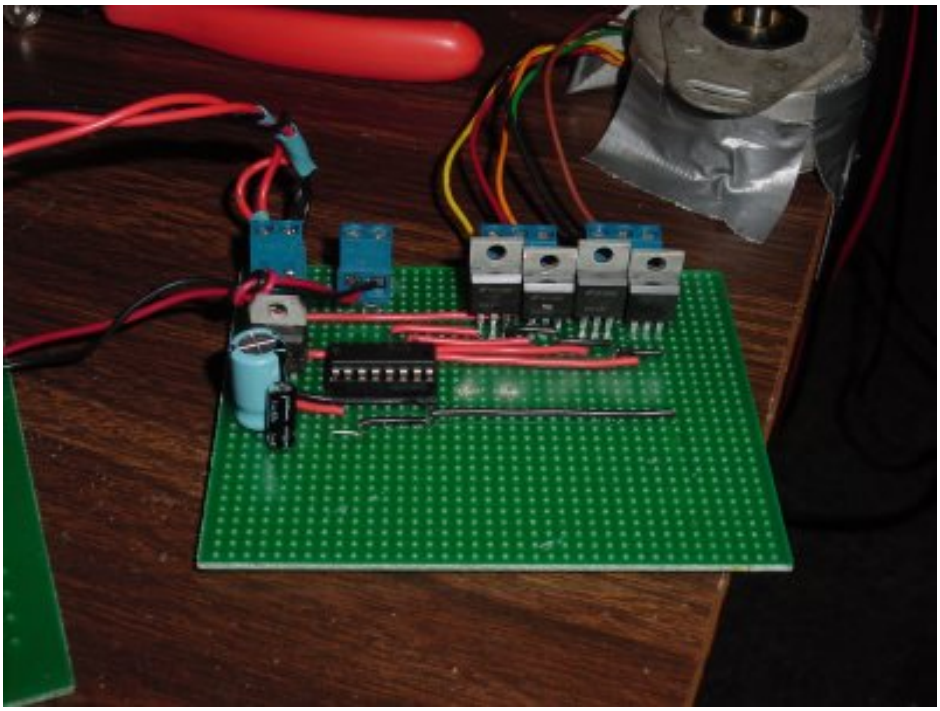
Now THAT's replication that will get Reprap everywhere.

Powering the tin can stepper lead screw

Saturday, 7th March 2009 by Forrest Higgs

In which your narrator knocks off a unipolar driver board using I2C comms with the MCU in a very few hours...

I put this little board together to explore the possibility that one could effectively drive cheap unipolar stepper motors via an I2C link with the MCU rather than tapping a bunch of I/O pins. I put it together by slaving four HUFA76419P3 Mosfets with a Texas Instruments PCF8574 chip slaved to a Microchip 18F4550 via a software driven I2C bus. The electronics are trivial and appear to be very reliable.



The board was trivial to make. The screw terminal and capacitors on the left side take in 12v and ground and turn part of it into 5v to power the Texas Instruments PCF8574 chip. The other 2 pole screw terminal just to the right of the power input screw terminal handles the SDA and SCL lines from the MCU's I2C bus and connect directly to the PCF8574 chip.

The last four bits of the byte transmitted to the board over the I2C bus are used to trigger the four Fairchild HUFA76419P3 Mosfets which charge the four phases of the Airpax unipolar stepper motor that you can see at the upper right of the. The four orange leads from the far side of the PCF8574 make those connections.

I used the HUFA76419P3 Mosfets mostly because I had them lying around from a project to build a bipolar driver board that I never got around to doing. They are monsters and can handle up to 27 amps continuously. They only cost a bit over $1 each from Mouser, though. I'd never contemplate putting anything near that kind of load on this prototype board, mind. It would vapourise the stripboard tracks, for one thing even though the Mosfets could handle it.

I've got the Airpax unipolar stepper cranking, but there is a bit more work to be done getting it into high torque mode.

To save comments on this point, understand that unipolar steppers are 30% less powerful than an equivalent bipolar stepper with the same weight of wire in it. The electronics, on the other hand, are trivial compared to what you have to do to run a bipolar stepper and can easily be built from discrete components.

This board will be driving my tin can {permanent magnet} stepper driven lead screw axis which I hope to use for Tommelise 3.0.

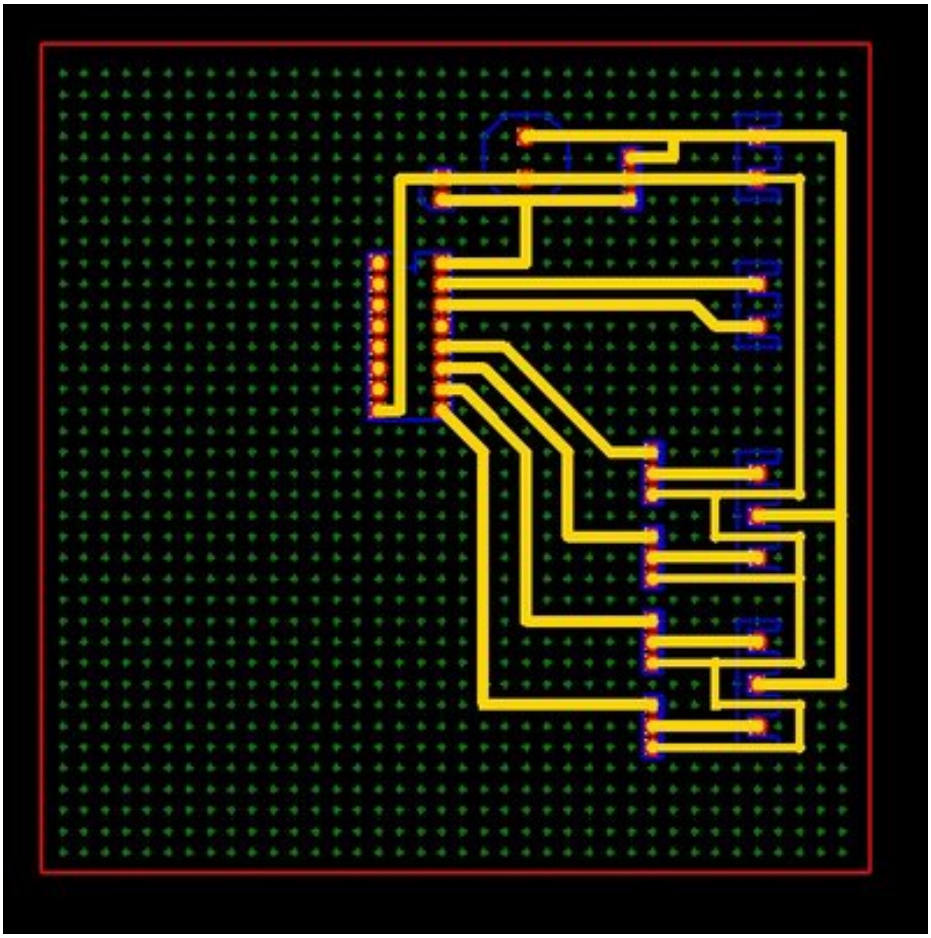
Late update: I found an old Airpax catalog and got the proper lead colour codes and stepping sequence. Putting that into the firmware I was able to test the limits of the big Airpax stepper motor. The one I have is the equivalent of the catalog's 57M024B2U.

The terminal velocity that I could get from a standing start was 228 pps which matched the pull-in torque line from the catalog very closely. When I used a half-speed startup for 24 steps I was able to push that reliably up to 311 pps. The best I could do with that startup regimen was 444 pps. There was virtually no available torque at that rate and precious little at 311. I suspect that we might be able to hit 250 pps and run the gear pair on the tin can stepper to power a lead screw.

What that means is that with the gear pair that I designed {1.6:1} I should be getting about 0.105 mm/pulse or a peak single axis velocity of about 26 mm/sec.

The nice part about this scheme is that I can change the resolution of the system simply by changing the gear pair.

Later update: Here is the basic board layout taken from my PCB milling software. The components aren't labeled, but from the picture they're not hard to figure out.



A small milestone

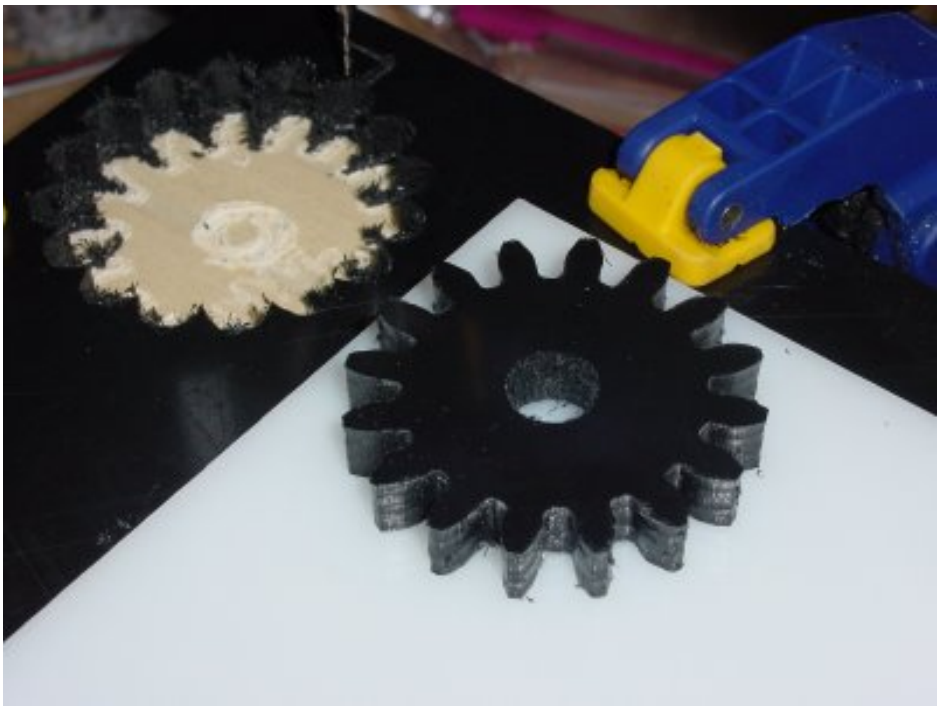
Monday, 9th March 2009 by Forrest Higgs

In which your narrator cuts a power transmission gear out of 3/8" inch HDPE, the thickest milling job yet...

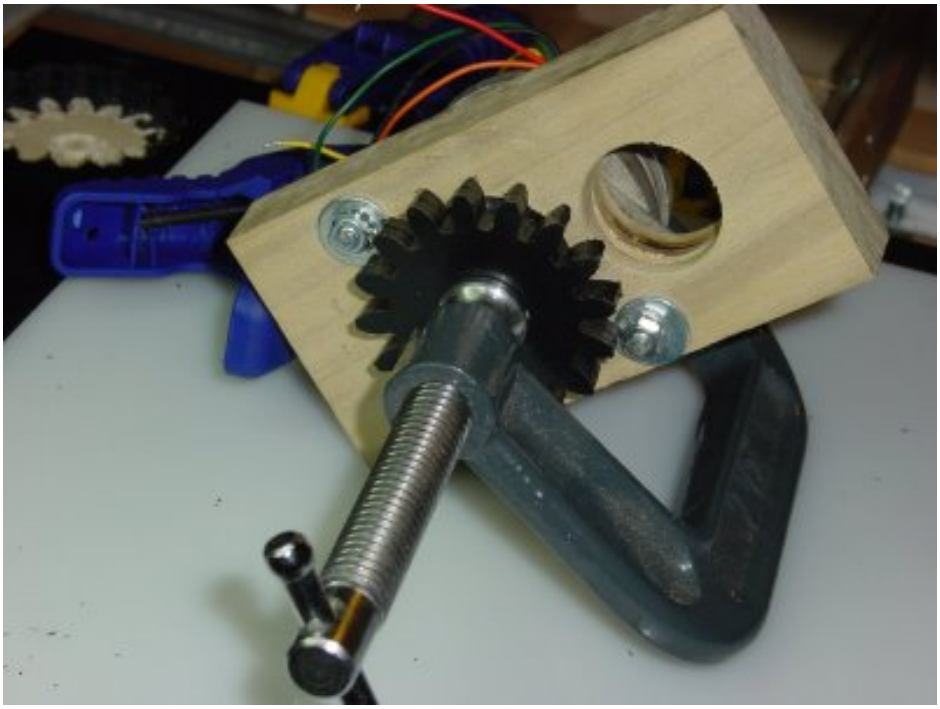
Nophead went over my I2C controller board layout and showed me how to deal with back-EMF with diodes and a zener diode this afternoon, so that I know now what to do with that. His electronics advice is priceless.

This evening I did several trial milling jobs to get the diameter right for the stepper gear for the tin can stepper axis for Tommelise 3.0 and, possibly, eD's new wedge Darwin 2.0. I wanted the gear to have a tight, compression fit with that weird half inch shaft sleeve on the old Airpax stepper. After two tries I got that right and then set out to see if I could cut the gear out of heavy, 3/8" HDPE. I wanted the stepper gear thick so that the lead screw gear, which I will cut tomorrow out of 1/4" HDPE won't have to be in perfect alighment with it.

In any case, it took me about an hour and a half to cut the gear. It came out looking pretty good.



Here you see me using an ordinary C-clamp to press-fit the gear onto the stepper shaft sleeve.



The nice thing about compression fitting is that you don't need set screws or glue or anything of the sort. You just put the gear on the shaft under compression and it's on to stay. I'd have to cut that wood mounting block apart to get the stepper out of there. That or cut the gear apart.

Abandoning the conventional approach

Monday, 9th March 2009 by Forrest Higgs

In which your narrator discovers that the convention way of running a lead screw drive isn't going to work properly...

I cut the lead screw gear before starting work this morning and knocked the drive together at lunch. It runs fine.

The problem is that it appears to me that you don't have enough torque to push the drive much beyond 12.5 mm/sec with anything like a useful load.

At that point I did a few calcualtions. My back-of-the-envelope calculations indicated that the kinetic energy in the spinning lead screws at 9 x 10^-3 Joules was more or less the same amount of energy as it took to move a 2 kg positioning table, that is, about 14 x 10^-3 Joules.

Now if I turn the thrust nut instead of the lead screw and move either the nut and the stepper motor assembly or the lead screw, both weighing about half a pound each, I only require about 1 x 10^-3 Joules instead of 9.

If I have time tomorrow, I will recut the lead screw to seat a thrust nut and see if I can't get more

speed and torque out of the drive.

Direct drive turns out to be the best

Saturday, 14th March 2009 by Forrest Higgs

In which your narrator finds that simplest is best...

After extensive tests I had decided that I had to freeze the lead screw and turn the thrust collar nut to get enough torque out of the tin can stepper to make a practical difference. Indeed, that might have worked but there was a simpler way.

Going from a 3/8-16 lead screw to a 1/4-20 dropped the inertia of the lead screw by 80%.

A bit of sticky tape on the end of the 1/4-20 lead screw gave me a very firm connection with the surplus Airpax stepper and enabled me to run meaningful thrust exercises with it.

I discovered that it was possible to get 250 pps out of the stepper in this configuration. Further, at that rotational rate I was getting 4-5 lbs of thrust, considerably more than the Haydon linear steppers are giving me now on Tommelise 2.0.

That translates into a touch over 13 mm/sec axis speed.

I'm done fooling with this. T3 is going to use a direct drive powered by old Airpax tin can steppers. $150 in parts, here I come. :-D

Having another go at milling polypropylene

Friday, 20th March 2009 by Forrest Higgs

In which your narrator mounts his noble steed, Rocinante, and has another tilt at milling polypropylene...

I broke one of my two 0.05 inch end mills a few days ago when my z-axis jammed. I had carelessly left a piece of electrical tape stuck to the z-axis work table and it had got wrapped around one of the spring-loaded skateboard bearings that I used to hold the z-axis worktable onto it's vertical guide rails. While demounting the z-axis by the simple expedient of loosening four wing nuts the Dremel milling head dropped to the MDF base of Tommelise 2 with the end mill still mounted. The end mill sheared off at where the cutting flutes meet the 1/8th inch shank.

I have a spare, so this was an annoying, but not debilitating incident. I was back in operation a few minutes later with my spare end mill. The next day, however, I called Fred at North Bay Technical to order several more of these useful little fish-tailed end mills. While chatting with Fred, the old issue of milling polypropylene came up again.

Heretofore, polypropylene has shown a distinct tendency to melt onto the end mill, rather like perspex, called lucite in the US and also known as acrylic, does. Fred thought that perhaps using a single fluted end mill instead of the double fluted end mill that works so well with HDPE might be the answer. Apparently, a single flute end mill generates much less heat while cutting plastic than end mills with more flutes.

In any case, Fred didn't have any single flute end mills in stock, so the question is moot, for now at least. I had been told by another long time milling pundit that the only way to successfully mill polypropylene was with coolant. When I mentioned this to Fred, he suggested that I build a little dam around the milling zone and squirt coolant onto the end mill to keep it cool. I'd been thinking of mounting the whole show in a tray and doing it that way. Till now, however, I'd searched in vain for a tray that would economically hold a one square foot piece of polyproplyene.

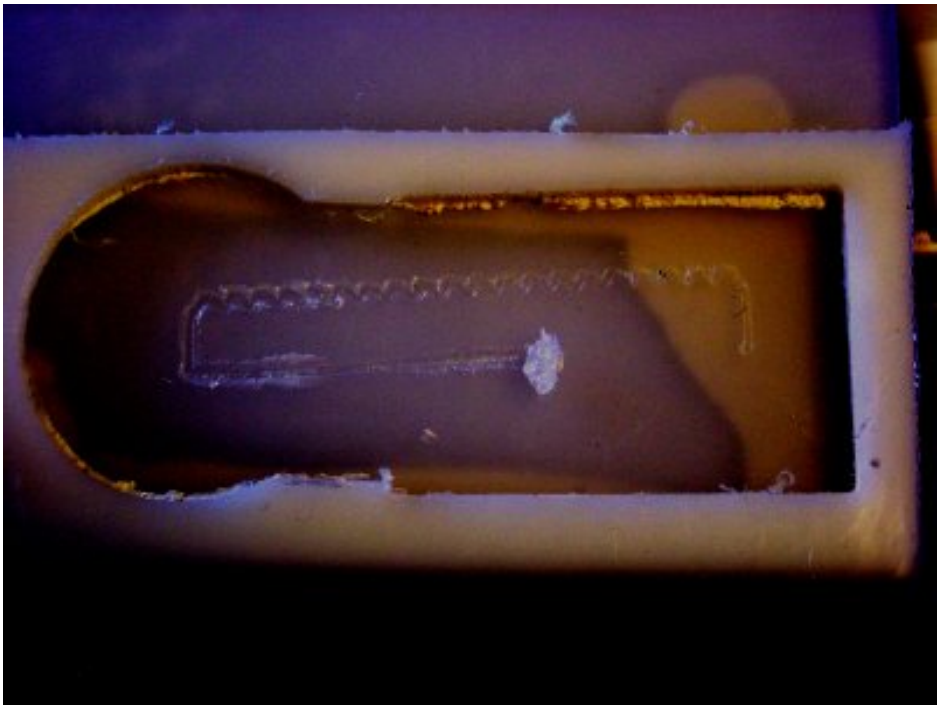
Fred's suggestion kept rattling around in my mind for the rest of the day, however. I got to thinking that I had a lot of HDPE scrap boards from which I might be able to extract the materials for such a dam. I found a 3/16ths inch board from which I'd milled several pieces for my IR scanner. I sawed out one piece from that and attached it to a quarter inch piece of polypropylene with bathroom silicone sealant.

Note how you can literally see through the quarter inch polypropylene. I had visions of actually being able to mill all the way through the polypropylene at that time, so I attached another piece of scrap HDPE beneath it with the sealant as well.

The sealant is particularly effective in making watertight seals as you might expect and also provides enough mechanical strength to keep the polypropylene and the dam in place. The nice part about this sealant is that it is both cheap and doesn't stick to the plastic. When you are done, you just pull it apart and wipe it off.

This sealant might well be the answer for sticking HDPE to the poplar I use for underlayment during milling operations. I'll have to try that out.

In any case, I had about an eighth of an inch of coolant on top of the polypropylene when I began milling. Here you can see an enormously contrast-enhanced picture of what happened.



I started the cut at the right and proceeded to the left. The cut began well enough and looks good. After a few centimeters, however, the end mill caught a few bits of swarf just long enough to act as a fan and blow the thin layer of coolant away from the plastic. Once the coolant was gone, the polypropylene began to melt on to the end mill and the cut got wider and wider.

After it turned the corner to return to its start point I shut down the cut as a partial failure.

It would appear that I've either got to establish a flow of clean coolant against the end mill or I've got to make the coolant pond deep enough to prevent the fan effect from doing what it did in this exercise.

I am going to try a deeper dam first.

Getting a handle on milling accuracy

Sunday, 12th April 2009 by Forrest Higgs

In which your narrator gets a handle on the everyday accuracy of the total tool chain from Art of Illusion to the milled product...

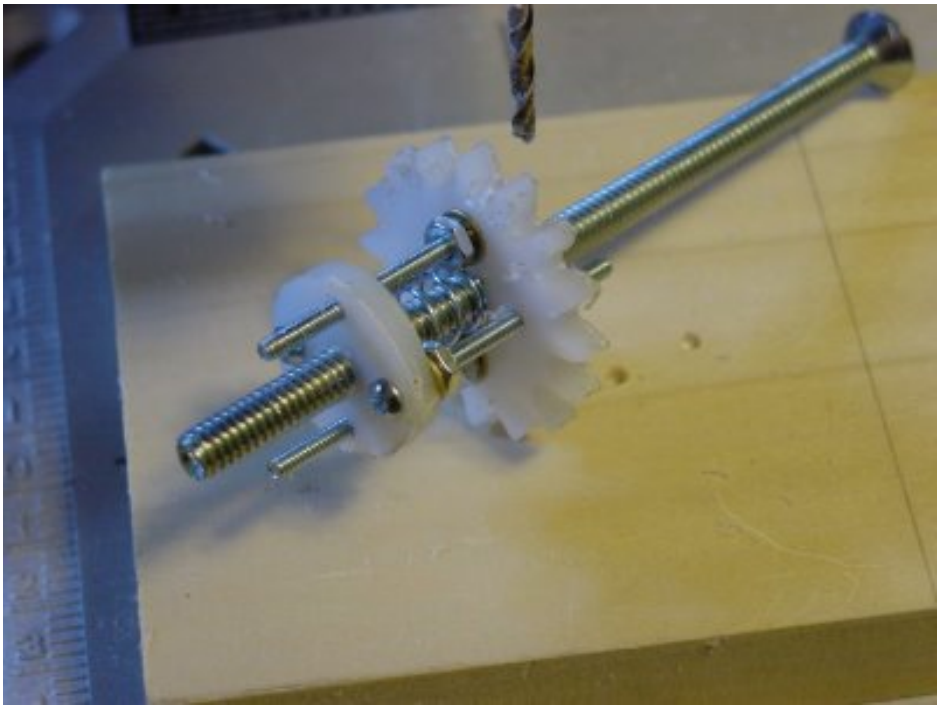
I've been milling HDPE regularly with Tommelise 2.0. Usually, I am designing parts to fit something made out of metal like a bolt, or a NEMA 17 mounting plate. In my more recent project, however, I have been designing parts that actually have to fit other milled parts.

Right now I'm on a quixotic quest to design an extruder that uses a threaded nut instead of a threaded bolt or pinch wheel to pump the filament. Milling is more limited than printing because you can't mill something thicker than the active length of your end mill. With my 1.27 mm diameter fish-tailed, two flute end mill I can handle about 10 mm thick material.

What that means as a practical matter is that if I want something thicker, I have to make it out of milled parts thinner than that. That can get tricky.

It's not like this is an unusual problem in Reprap. The laser cut variations on Darwin have the same problem. There are a number of ways that you can access the problem. The way the Ponoko and BitsFromBytes Darwins hand it is to use nuts, bolts and spacers to assemble parts made of relatively thin plastic.

I've done a bit of that and have been dismayed at how much you wind up paying for all of the nuts, bolts, washers, springs and lock washers you wind up paying for to make anything. A few months ago I knocked out a anti-backlash nut for a lead screw drive.



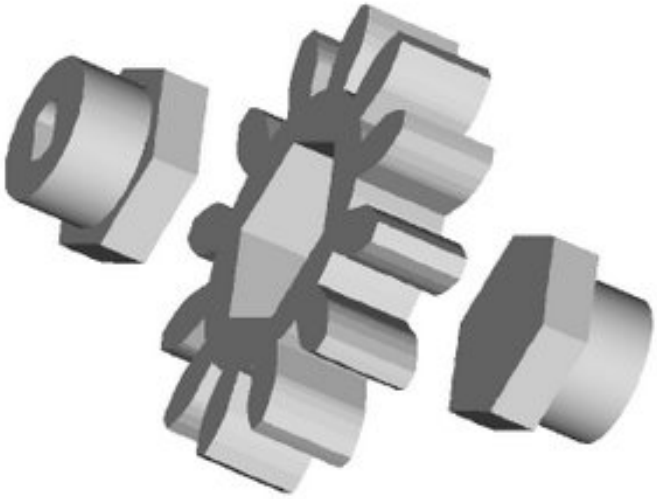
It was a beautiful part, but the bloody hardware to hold the milled bits together wound up costing about ten times what the HDPE parts did. There is something kind of crazy about that.

Since then I've been trying to design things that fit together with a minimum of fasteners. The particular problem that I've been trying to solve in the few minutes I've had away from my day job in the past few weeks has been how to connect the stubby little drive shaft on the Solarbotics GM-17 onto a spur gear to drive the extruder filament pump.

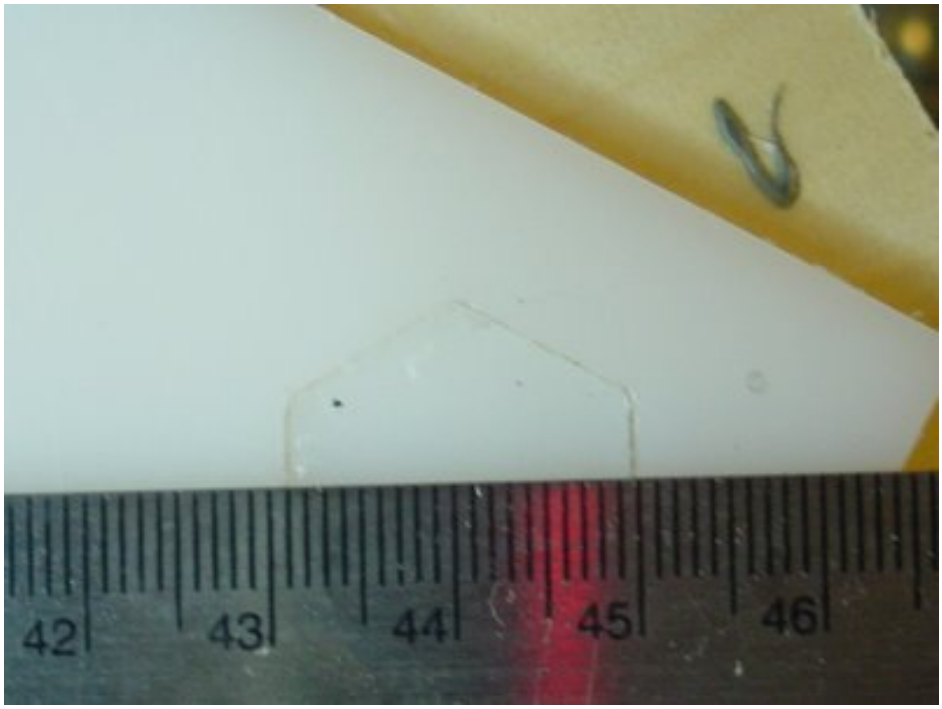
The GM-17 was designed to be bolted onto something like a thin metal plate with a few holes drilled in it. HDPE designed for the same purpose tends to be as thick as the stubby little drive

shaft of this very efficient little gearmotor. what that means is that you have to put a significant extention onto your spur gear and mill a big hole in your mounting plate around the drive shaft so that the extention can reach up through the mounting plate to engage the drive shaft. That's a pain in the neck, to say the least.

The approach I've been taking is to design plugs of 10 mm HDPE that receive the drive shaft and fit into the gear itself. What I have designed looks a bit like this.



When you do something like that, however, the tightness of the fit between the parts becomes important if you are to avoid a lot of wear. Because of that, I decided to see how close a fit I could get with such a plug using the same hexagonal prism in the open-source Art of Illusion 3D modeling app to make both the plug and the hole that it fits into.
I used the same milling settings for both pieces. Here is what the fit looks like.



It appears that I get around 0.2-0.3 mm gap between the plug and the hole. Mind, I know I could design tighter fits by upping the nominal size of the plug just a touch. That sort of thing takes a bit of trial and error, though. Now I know how much of a gap that I get when I just go straight for an design solution.

A different approach to extrusion

Thursday, 30th April 2009 by Forrest Higgs

In which your narrator goes nuts...

As you know I've been designing Tommelise 3.0. T3 will be able to be configured either as a printer or a milling machine and is looking to cost about $150 in parts for the printer option. Obviously, I've got to have an extruder if I'm going to do printing. Heretofore, I've been torn between building a "me, too" pinch wheel extruder or a more robust variation on the Mk II. With what seems like everybody and their Schnauzer building more flavours of pinch wheel extruders than Ben and Jerry's have flavours of ice cream and the formidible Nophead exploring variations on the Mk II, I was left to either wait for the dust to settle or somehow come up with some new wrinkle to those two initiatives that would add something to the state of the art. I frankly couldn't see what I had to offer in that regard.

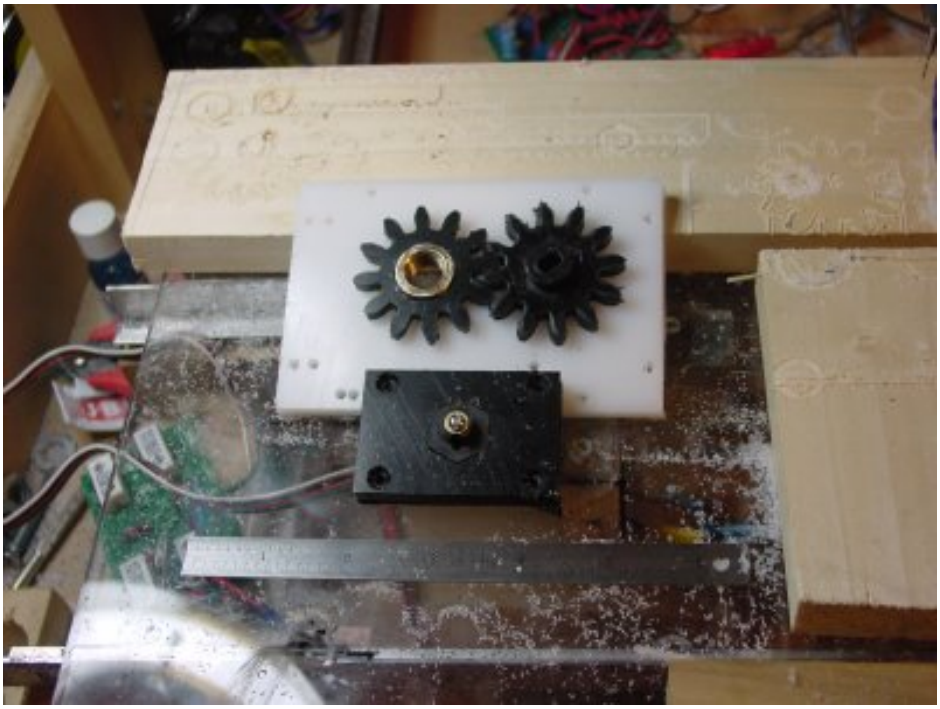
The dust hasn't settled yet on extruder designs. A clear favourite hasn't been clear to me at all. Most pinch wheel designs, while simple, require heavy NEMA steppers. I haven't been too happy with that. Nophead's, pinch wheel design and his variations on the Mk II, while light, require machined parts. I'm not happy with that, either.

That left me wondering whether there was a third way of designing a workable extruder that was both reliable, light and required no machined parts that couldn't be milled or printed on a Reprap machine.

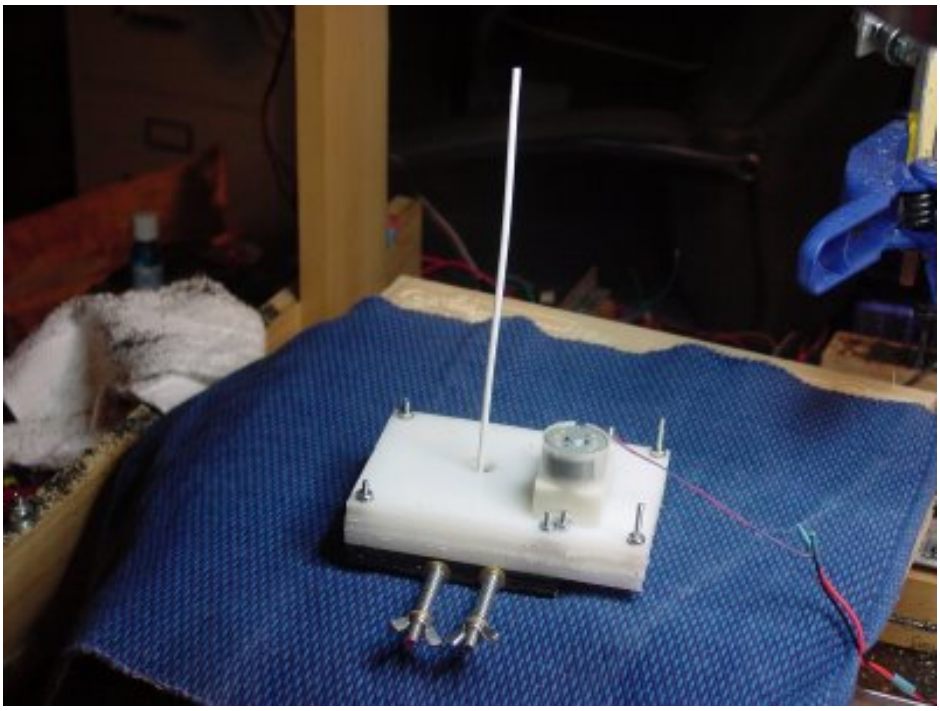
I got to thinking about threaded nuts, for some reason. The problem with threaded rods is twofold. First they have to have bushings. You have to mill those and the damned things wear and create metal dust which gets into your extruded plastic if you don't break the system down and clean it quite frequently. Second, the threads curve away from the filament, which means that the threads have to cut quite deeply into the filament to get a proper grip.

A nut, on the other hand, curves around the filament, giving a much larger contact area. In the past there was a notion that you ought to be able to thread a nut onto the filament. The problem with that approach is that the filament tended to turn with the nut. I came up with a different approach.

I used a nut considerably larger than the filament and then milled a filament guide {the black HDPE part just above the ruler} that pushes the filament against the threads.



The nut, I embedded in a gear and drove that with another gear connected to the drive shaft of the gear motor. The bushings and drive shafts are milled out of HDPE and are made with large diameters to keep the contact area between shaft and bushings large and the force per unit area small. Given that HDPE is about as slick as Teflon friction isn't a problem.



The black filament guide is connected to the Z-axis and the filament pump floats free. The two #8 threaded rods with wingnuts and springs provide the compression required to let the filament engage the threads inside the nut.

Tentatively, I am driving the system with a Solarbotics GM-17. Should that not be effective, I can

shift to a 12v GM-3 which has about 60% more torque. Should that not be adequate I can always use Nophead's tin can stepper driven GM-17 gearbox.

I've already demonstrated that this ensemble will pump ABS. I have not measured how much force it can apply to the filament, though I suspect that it is substantial. The design has a number of advantages.

- at 280 grams, it is light.
- it requires absolutely no especially machined metal parts.
- it can be milled, printed or laser cut {most of the parts for this last}.
- the feed rate can be increased by two-thirds simply by swapping the 1/2-20 fine pitch nut for a 1/2-12 coarse pitch nut.
- it can process both very stiff and highly flexible filament with equal ease.

If you had to you could probably cut these parts out of HDPE with a scroll saw.

I will be seeing if I can get this extruder running properly and assess it's robustness over the Summer.

Late addition: I was able to get down to the hardware store this morning and get the requisite #8 nuts, washers and lock washers that I needed to complete the extruder filament pump. The compression springs that you see in the picture are adequate and the extruder pumps ABS at 5v. I was also able to acquire a spring scale like Nophead uses at the hardware store as well, so I should be able to get an idea of how well it pulls very soon.

Milling polypropylene

Thursday, 30th April 2009 by Forrest Higgs

In which your narrator finally manages to mill polypropylene without melting it...

As you know, I've been trying to mill polypropylene for quite some time without success. Why I would want to do so is purely an economic matter.

- ABS - $0.26/cubic inch
- HDPE - $0.11/cubic inch
- Polypropylene - $0.08/cubic inch

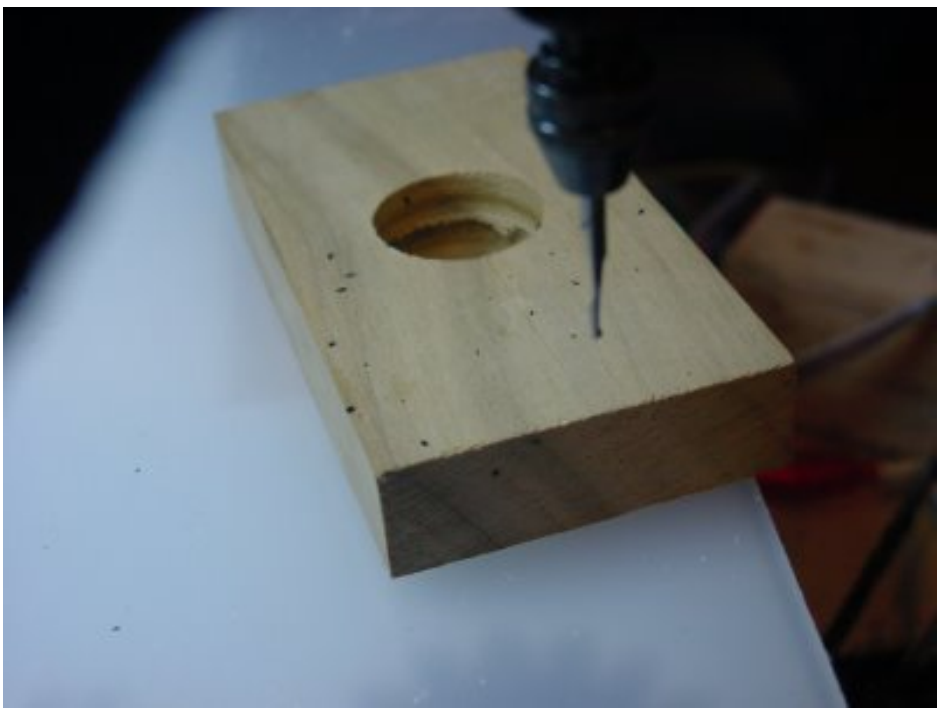
Polypropylene is far and away the cheapest commercial plastic on the market. It is also quite strong.

Anyway, today I milled it and I milled it without coolant, something I didn't think was possible.

I was talking with Fred Schultheis at North Bay Technical a few days ago about reordering some two flute, fish-tailed end mills. Fred and I had discussed milling polypropylene before and he had mentioned that you'd need to use coolant and/or a single flute end mill. As you reduce the number of flutes you reduce the amount of energy generated by the end mill.

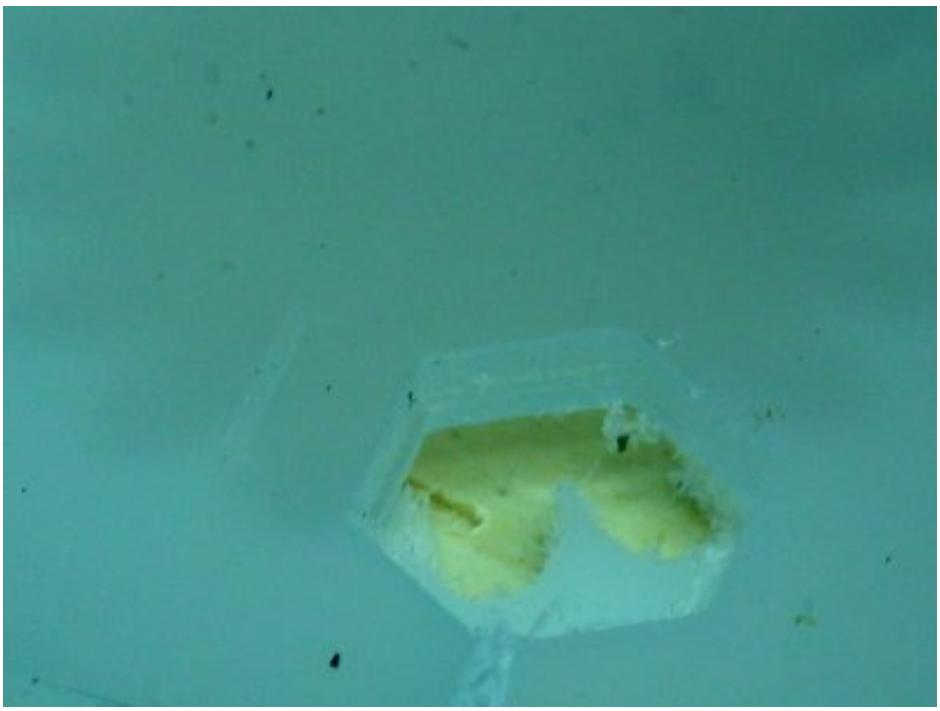
Fred checked his stock and for some reason mentioned that he had a 2 mm single flute end mill. I bought it immediately just to give it a try.

This morning Fred's order arrived and, since it was lunchtime and I'd already eaten, I thought I'd give the new end mill a try without coolant just to see how it behaved.



It turned out that I can reliably mill quarter inch polypropylene with the new end mill using the

same settings as I use with HDPE and two flute end mills, viz, 20K rpms/8.33 mm/sec except that I can, so far, only take 0.5 mm deep cuts rather than the 1.0 cuts that I can manage with HDPE using my regular, two flute end mills.



The new end mill makes VERY clean cuts with HDPE and I get no melting. The chips it throws are a bit bigger and tend to fly considerably higher, so safety glasses are in order.
I'll be doing more tests this weekend and blogging the results.

Documenting the Mk I Nutjob

Sunday, 3rd May 2009 by Forrest Higgs

Nophead: This looks really promising, but I am having trouble understanding the physical arrangement from the photos. Any chance of a view from underneath, or a diagram?

Forrest: Absolutely...

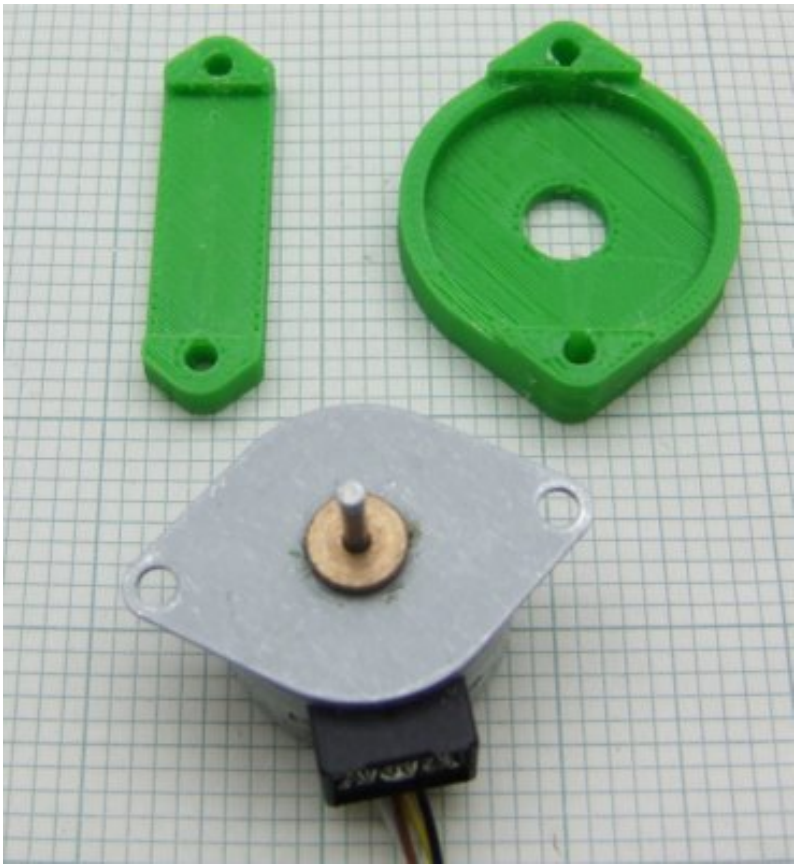
I had a few minutes Friday to run some tests on the Mk I Nutjob. Using a GM-17 it was able to generate several kilograms of thrust, which, from Nophead's research, should be enough to extrude ABS.

Frankly, though, I couldn't see that having enough torque to just extrude was going to be that much fun. Tommelise 1.0 taught me that operating on the edge of an extruder's capabilities is a frustrating exercise at the best of times and a waste of time at other times.

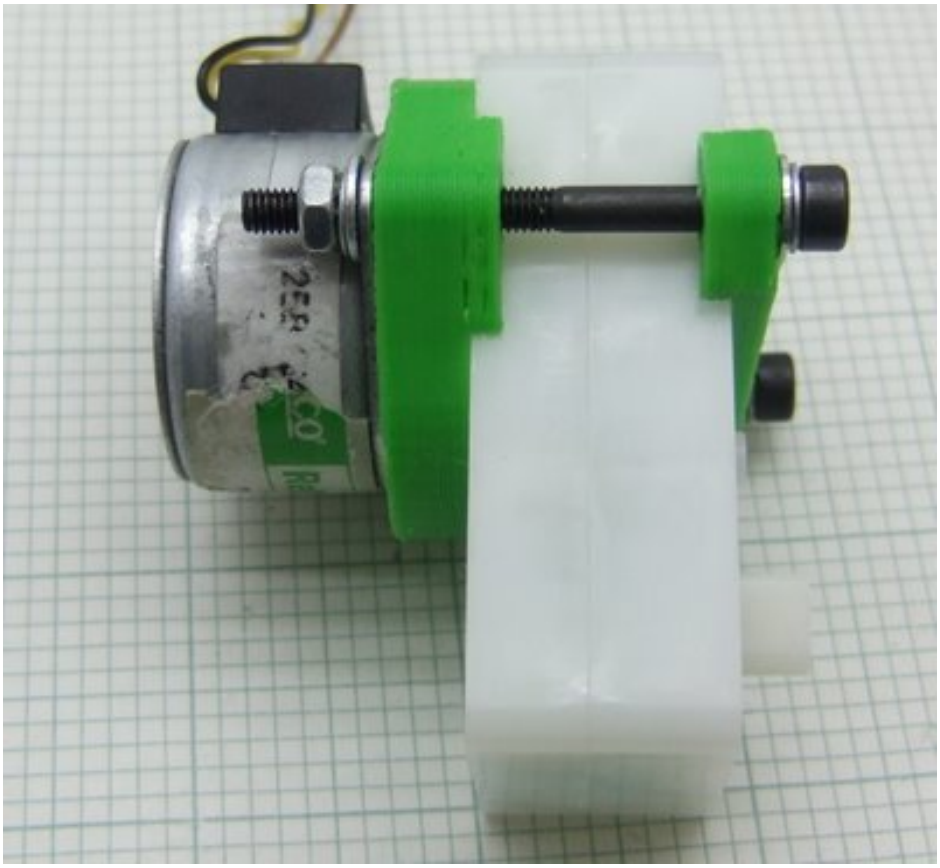
The diameter and shaft diameter of the little coreless DC motor in the GM-17 is a common one. A variety of more capable, read that longer, DC motors can easily be slotted into its place. Indeed, Solarbotics is looking for an upgrade motor for the GM-17 not unlike the 12v upgrade that they have for the GM-3.

At that point it occurred to me that I could easily do the GM-17 hack that Nophead demonstrated a few days ago. I certainly had exactly the Jameco tin can stepper that he used and I also had the ability to mill a special mount for it.

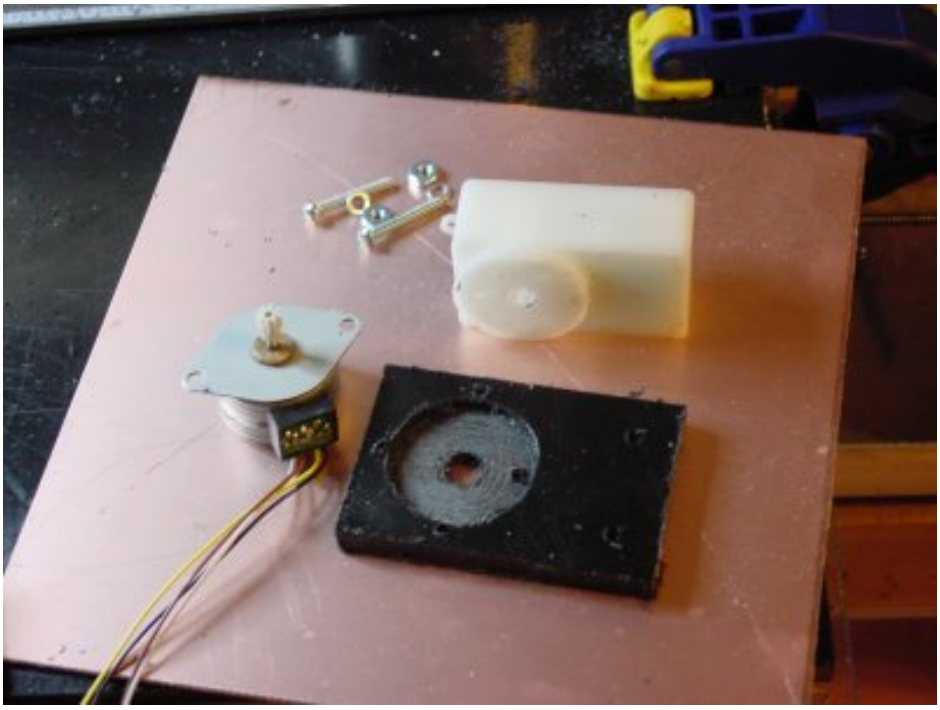
Nophead's mounting plate was, as usual, elegant.

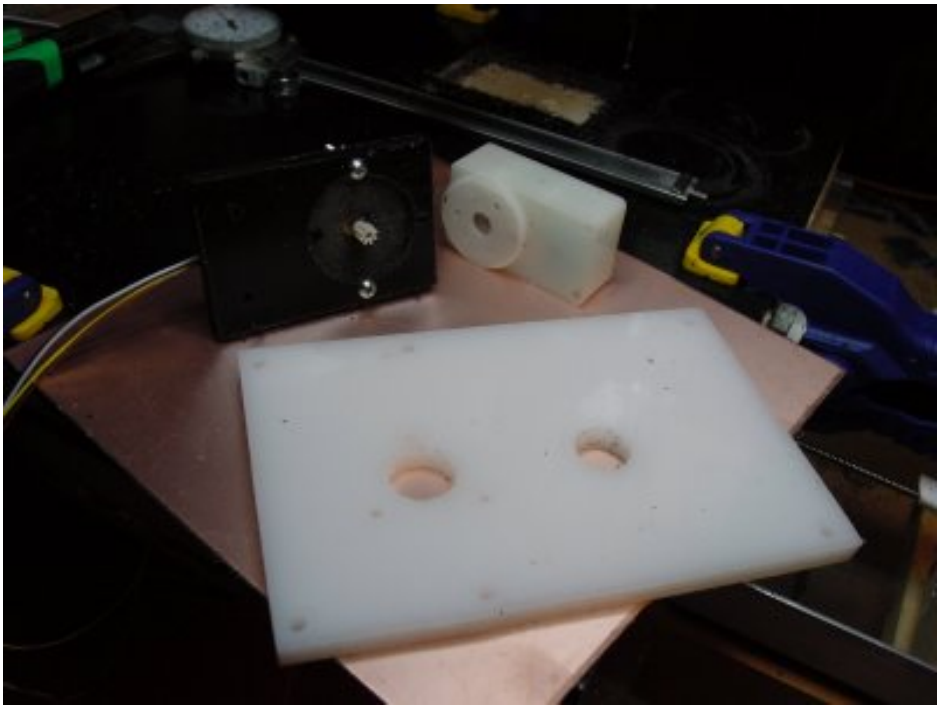


It would have been quite easy for me to mill it save one little issue.

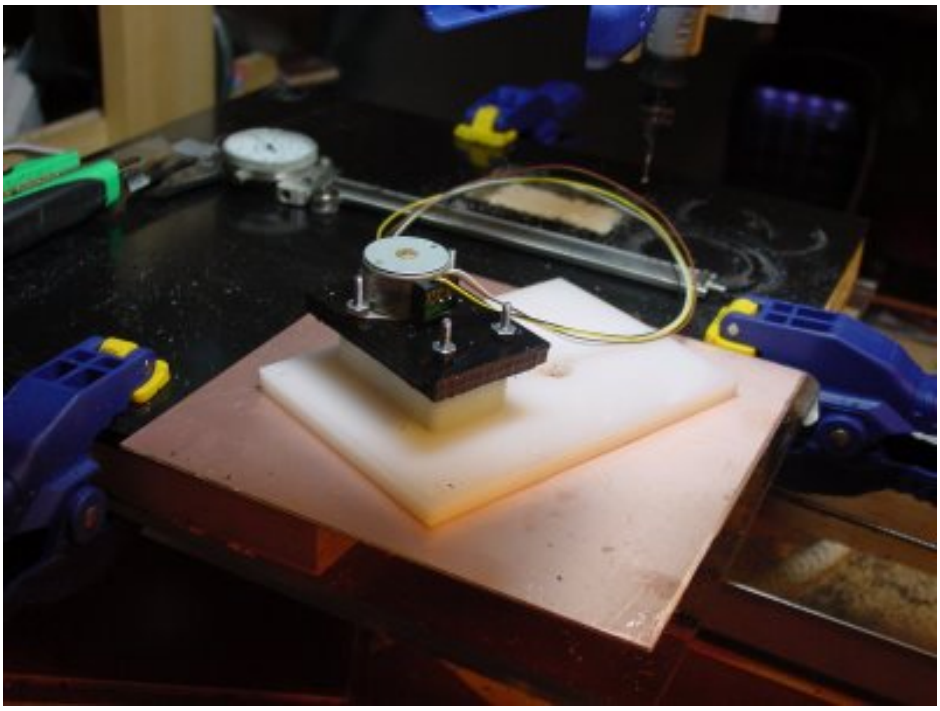


The ventral strap simply didn't work well for my extruder. As a result, I designed an alternative that made use of the extant mounting points on the GM-17.

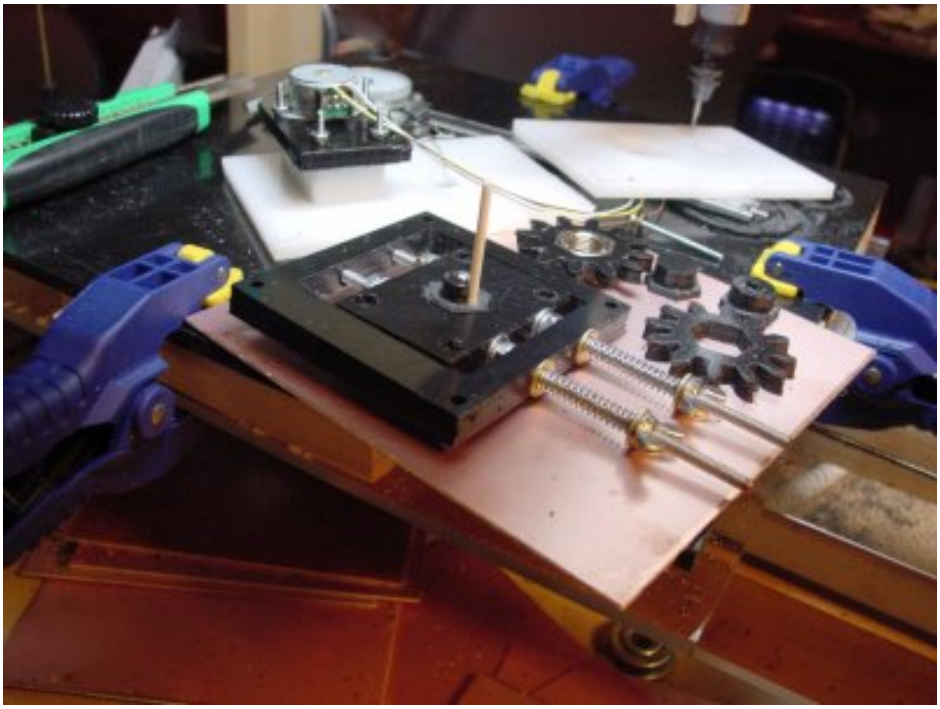




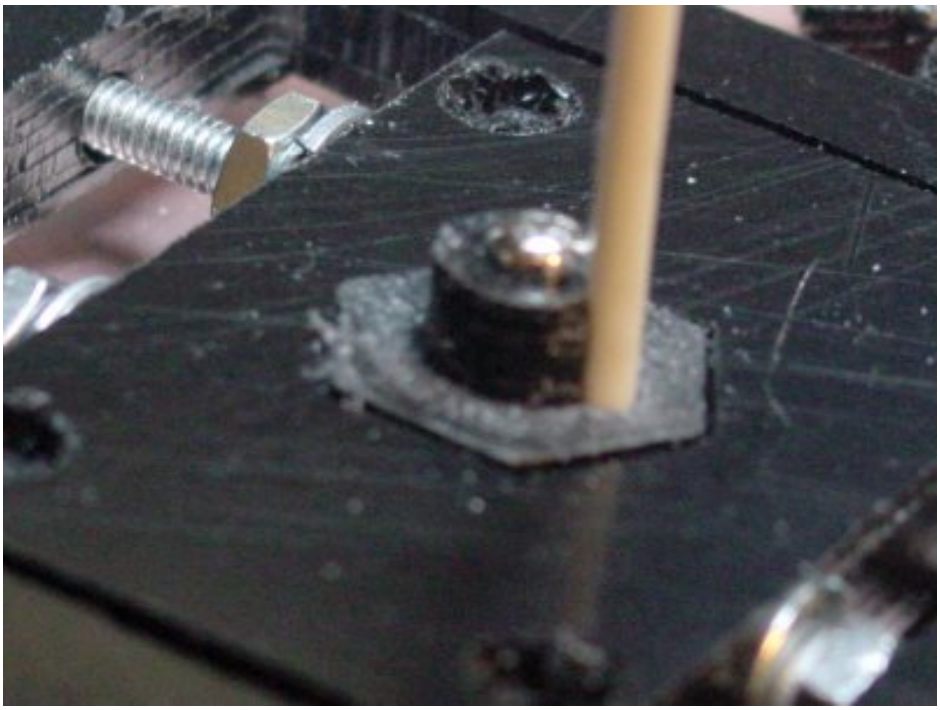
At that point I was ready to do the photodocumentation that I promised Nophead.



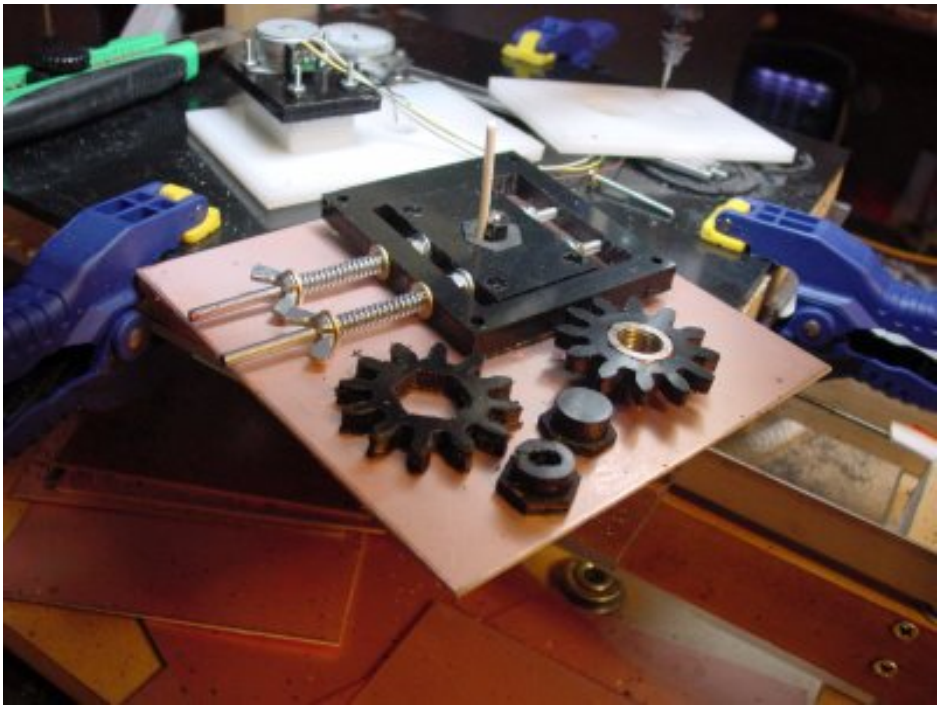
Here you can see my interpretation of Nophead's GM-17 hack mounted on the top plate of the extruder.



Starting from the bottom of the extruder, you first see the spring-loaded guillotine mechanism that holds the filament guide against the nut threads.

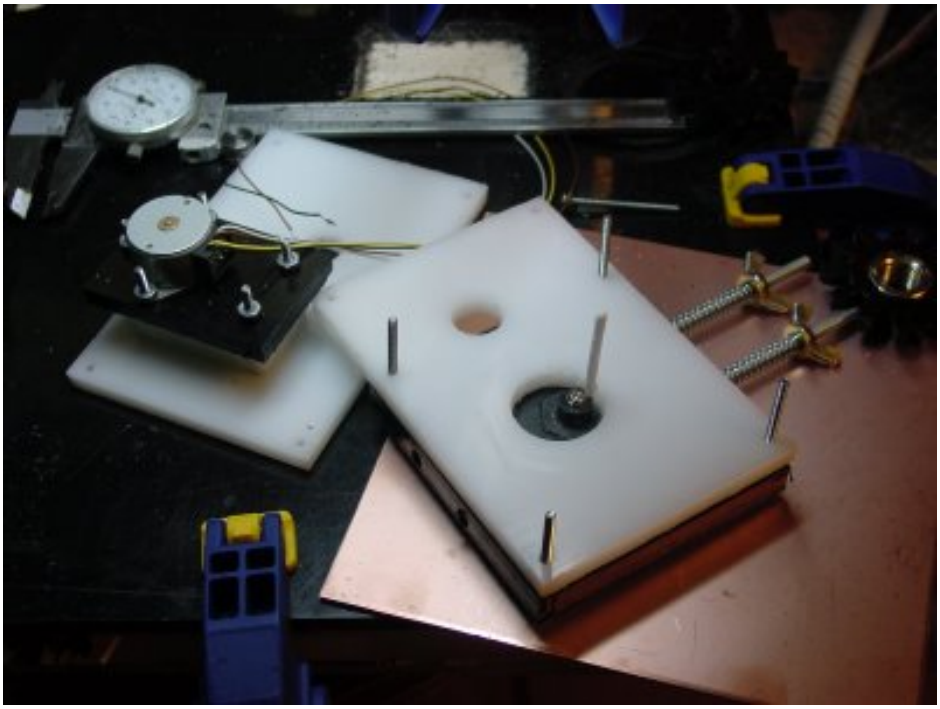


Here is a closeup of the filament guide. I designed it as an insert into the plate that is attached both to the extruder barrel and the z-axis. The mounting holes are plainly shown .

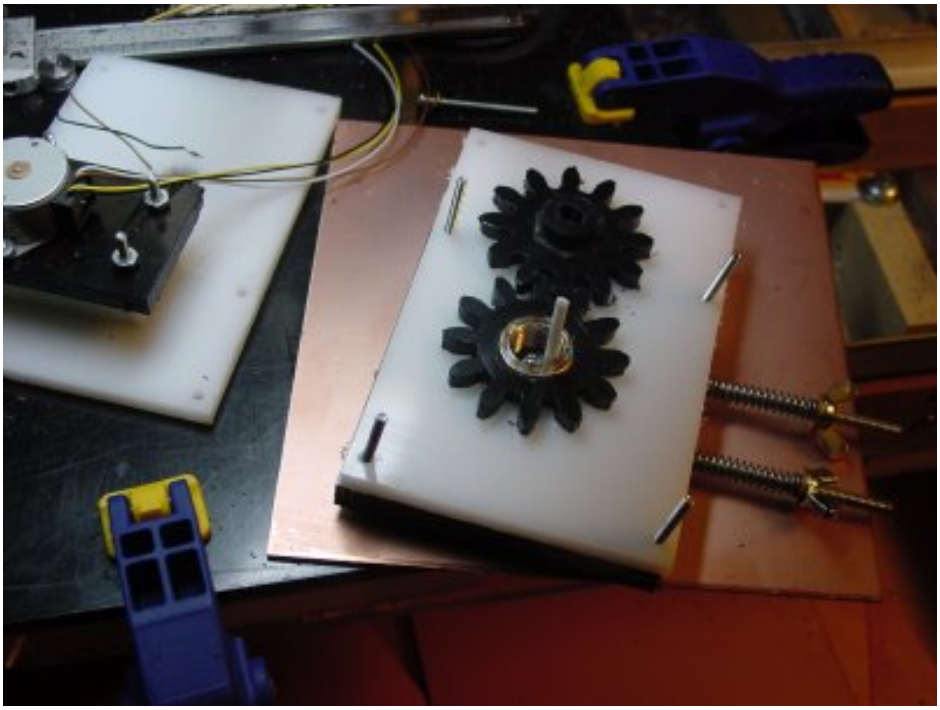


Power is carried from the GM-17 via a transfer gear. Here you can see that I milled it in three pieces to reduce machine time on Tommelise 2.0.

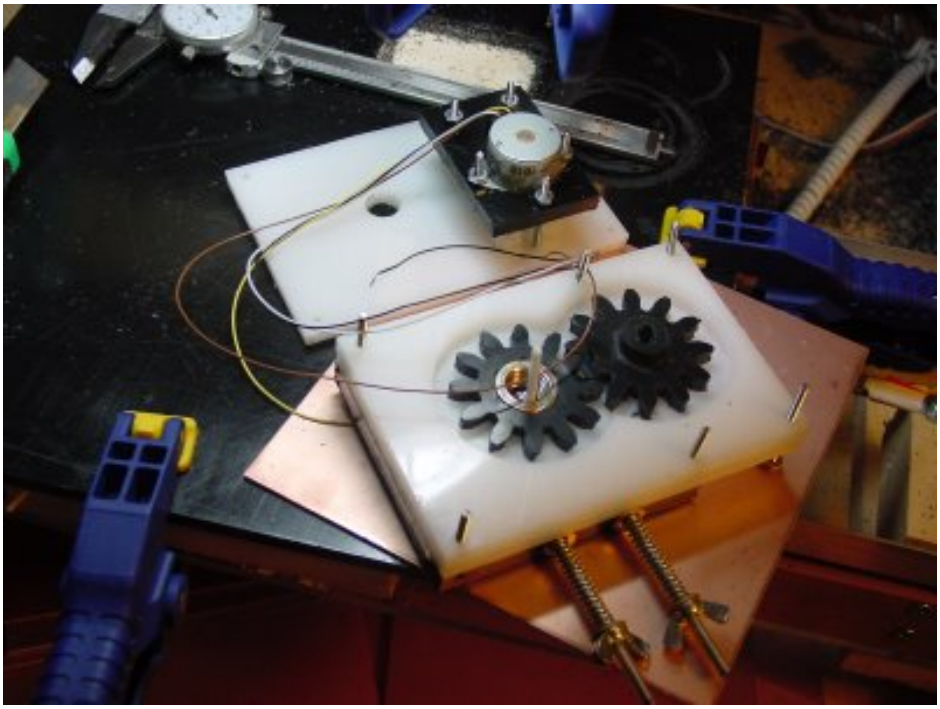
There has been considerable confusion about the embedded nut gear. Here you can see the bottom side of the gear from which a shaft protrudes which accommodates the full height of the nut, more or less. There is no shaft on the topside of the gear and the top of the nut simply buts up against the top housing plate, which serves as a thrust bearing.



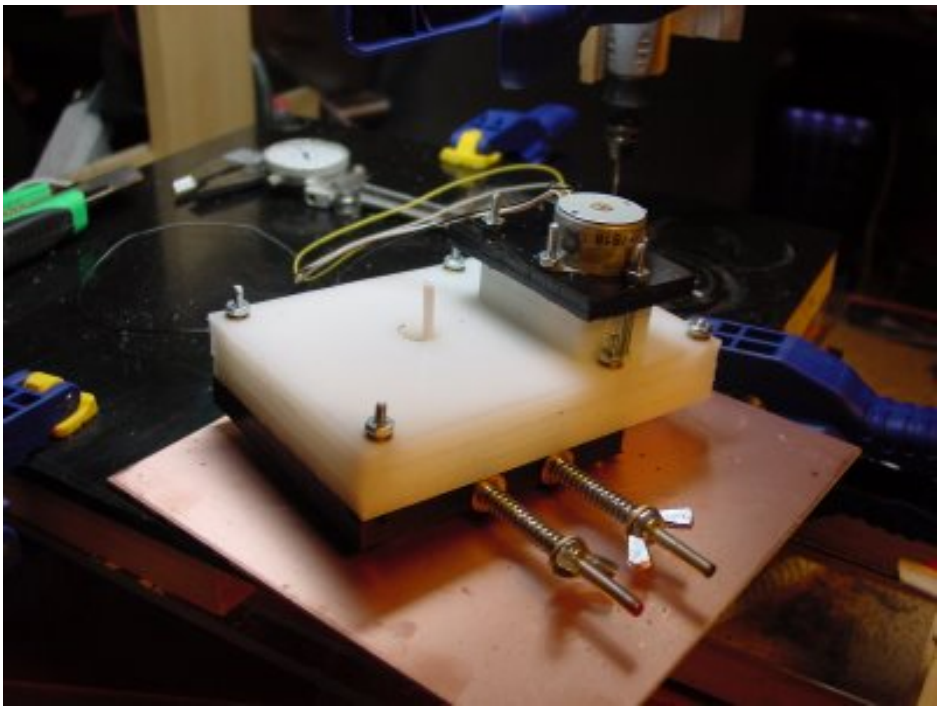
Now the bottom plate of the extruder slides onto the four #4 bolts in the filament guide frame. For you metric types, #4 bolts and M3 are pretty much the same diameter, though the pitch is, of course, different.



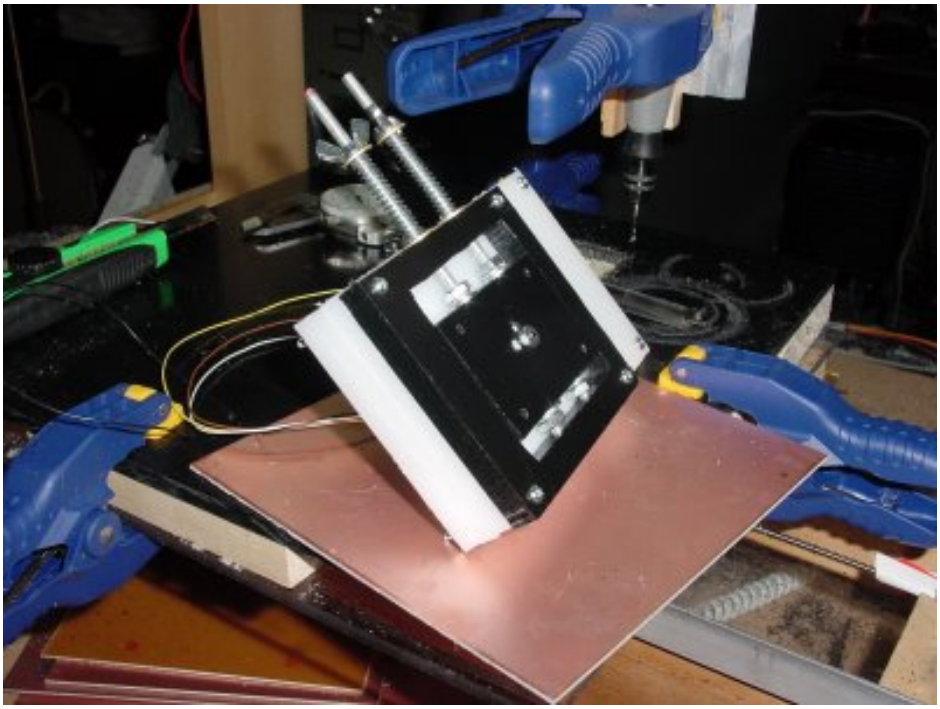
Once the bottom plate is mounted, fitting the drive gears is very straightforward.



Once the gears are in place the spacer plate slides onto the mounting bolts.



From there it is a simple matter of sliding the top plate with the GM-17 hack onto the bolts and applying the washers, lock washers and nuts to them to complete the assembly.



Finally, you can see the bottom of the assembly in this photo. with luck, I'll be able to test it within the next few days.

Stovetop recycling of HDPE swarf

Sunday, 3rd May 2009 by Forrest Higgs

In which your narrator solves the problem of recycling plastic scrap, for milling...

One of the great unsolved problems for Reprap has been that of either figuring out how to make plastic filament locally from scrap or how to design an extruder that can digest scrap. Reprap has successfully created a 3D printer for less than $1000 that replaces a commercial machine costing anywhere from $30-50,000. A plastic grinder and monofilament production machine runs anywhere from $75-100,000 and, unlike a commercial printer, can weigh several tons.

For milling, as opposed to printing, I've discovered that recycling can become much less challenging. Years ago in South Africa, I encountered a plastics fabrication technique called Rotary Moulding or Rotomoulding. Typically, you dump a measured amount of polymer powder into a big steel mould and secure it to a motorized gimbaled drive in a heated enclosure. The steel mold becomes slightly hotter than the melting point of the plastic while the tumbling action of the gimballed drive coats the inside of the mould with plastic powder.

The method is used to make everything from the petrol tank on your automobile to portable toilets and large water tanks.

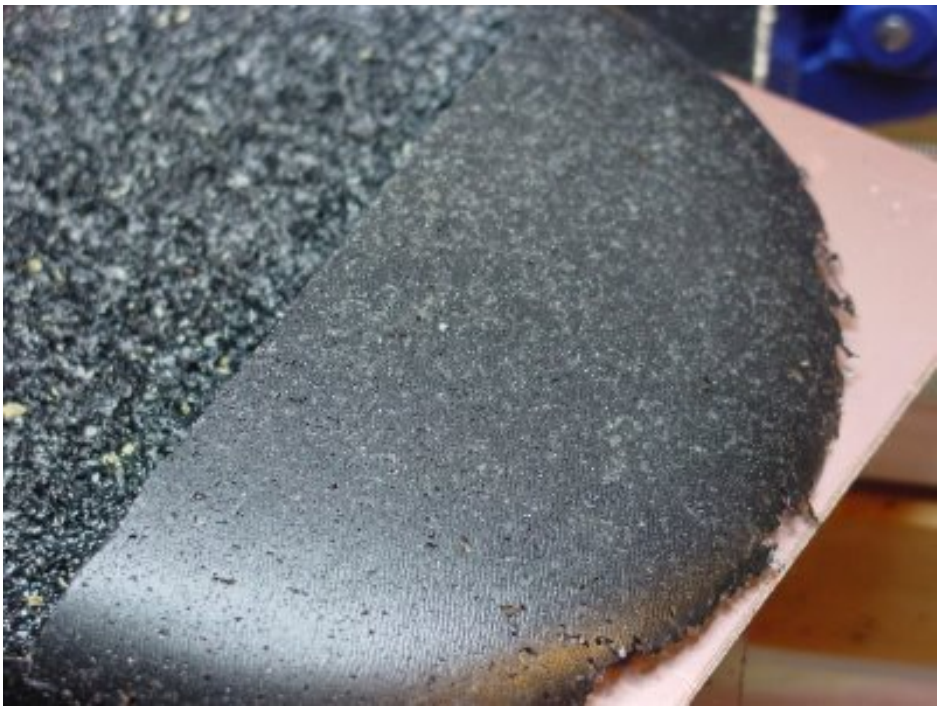
One guy I got to know in Pretoria back in the 1980's was using HDPE swarf to rotomold bins and water tanks.

Yesterday, I got to thinking. I'm buried in fine HDPE swarf and I need HDPE sheet. Could I modify the rotary moulding technology to create it. I took an old teflon coated frying pan that I didn't need any more and heated on my stove to about 220 C, checking the evenness of the surface temperature with my handy-dandy IR thermometer.

At that point I began sprinkling a couple of cups of milling swarf on the heated teflon surface. Interestingly, the swarf, being heated from below, tended to melt into the already melted surface without trapping large air bubbles like happened with my experiments some years ago to melt HDPE scrap in a toaster oven.

Once I ran out of swarf I cooled the pan and plastic down in cool water, then peeled the melted swarf out of the pan and cut it in half to see how thick it was.

The bottom surface is very slightly porous, though not enough to leak water. The top surface is rough with unmelted swarf and a few bits of poplar.



Here you can see the thickness of the melt, 2 mm. It would have been thicker except that I ran out of swarf. The slightly mottled colouring of the melt reflects the mix of white and black swarf that I had.

With thicker melts I could simply mill off a flat top surface. The need for that could be reduced by using a small aluminum rolling pin to flatten the top.

Now I have to either buy or mill a small hand vacuum cleaner to collect the swarf in a more orderly manner than I do presently by using compressed air.

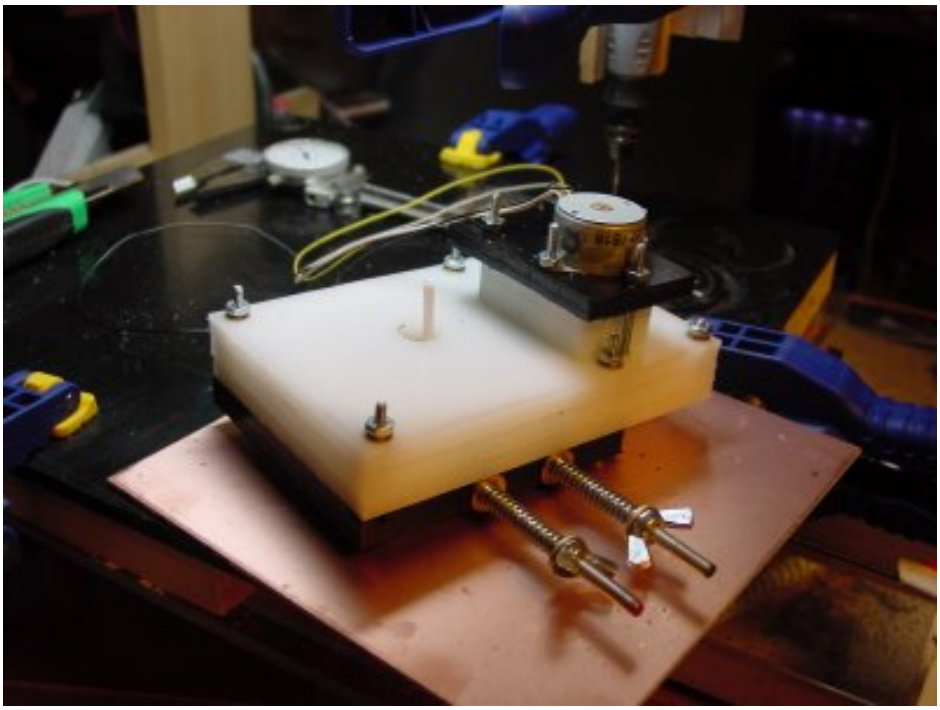
Taking the Gm-17 out of Nophead's GM-17/tin can stepper hack

Tuesday, 5th May 2009 by Forrest Higgs

Reducing the vitamin content of the Mk I Nutjob extruder...

These past few months have been very frustrating for me vis a vis Reprap. My day job has got intense to the point where there is little time for Reprap and worse still, the little time there is finds me mentally exhausted and incapable of working efficiently on Reprap work. In a less stressed time I could have designed and built the Mk I Nutjob over a couple of weekends. As it is, I've been working on it for since the New Year. Focussing has been hard.

Last weekend, things began to come together. I now have an assembled polymer pump using a nut as the threaded pump and it appears to work pretty well.



The pump has no problem biting into and moving ABS, HDPE and homopolypropylene. With the 228:1 standard gear ratio it moves a bit over 7.3 mm/minute. If I reduce that to 51:1 using the Dietl gear kits that Solarbotics sells it appears that I can extrude at a rate of 15-20 mm/sec.

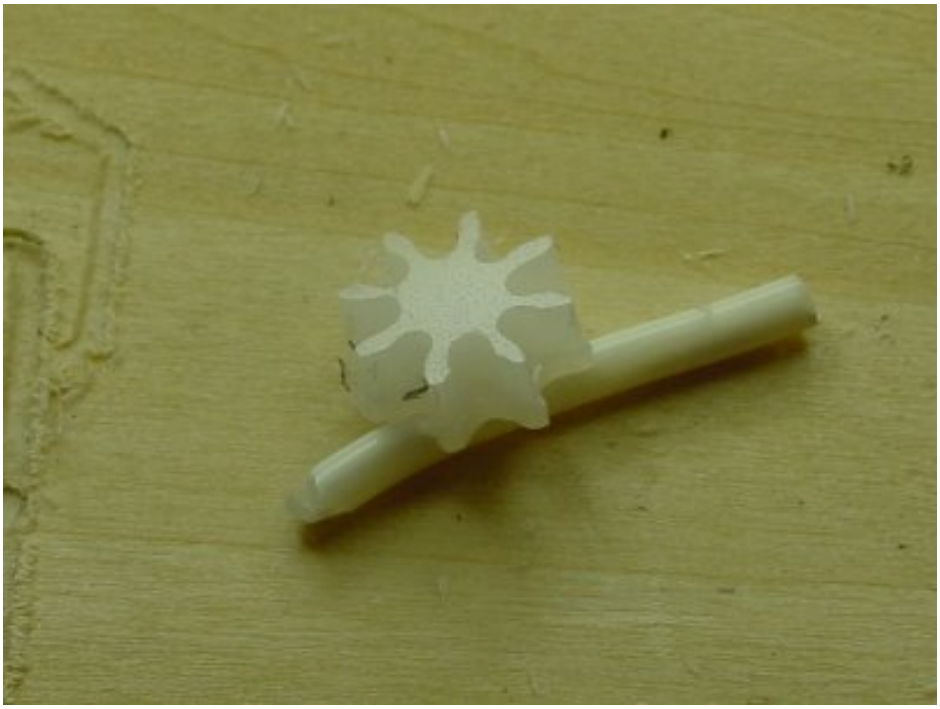
Nophead's GM-17/tin can stepper hack really delivers the power. I ordered four more GM-17s and the Dietl Rome gear sets to work with.

Having done that I looked at what Nophead's hack was actually costing. The GM-17 costs about $5-6, the Dietl gear set about $4 and the tin can stepper from Jameco about $7. Figure $17 dollars to get a lightweight, powerful, stepper-driven gearmotor. I can get a NEMA 17 that does the same job for that price without a lot of trouble. It's heavier, but hey?

That got me to thinking about the GM-17 gearbox. I don't use the little coreless electric motor. I'm paying $10 for a gearbox with about as much plastic in it as one joint of my little finger.

Quite some time ago, I was milling small gears and racks. They weren't very good. Since then, I've got a lot better at milling, but I haven't gone back to milling small things again. Before work this morning I decided to see if I could do a better job with my increased know-how.

In fact, I could.



What you see is a pinion gear suitable for pressing onto the 2 mm drive shaft of the tiny tin can stepper used in Nophead's hack. The cylinder under it is a piece of 3 mm filament. The pinion has a pressure radius of 5 mm.

Looking at the success of Nophead's hack, it is obvious that it is best not to have a lot of mass in the initial, high-speed gears in a gearbox. While my 10 mm pinion gear isn't as tiny as the GM-17's, it weighs a small fraction of a gram. It is also the hardest gear to mill.

I did some partial cuts of pinion gears with higher cut quality. My z-axis on T2 is a bit cranky this morning, however, and I wasn't able to successfully complete them before I had to quit for work.

Interestingly, gears of this size and thickness only take a few minutes to cut.

I will be using the GM-17/tin can stepper hack to work the bugs out of the Mk I Nutjob. Afterwards, however, I will be taking a very hard look at taking the GM-17 gearbox out of the vitamins column of Tommelise 3.0 Sampo.

Doh!

Friday, 8th May 2009 by Forrest Higgs

The additional Solarbotics GM-17s and the Didel Rome gear kits as well as the Didel GM-17 encoder pitched up today.

I popped the GM-17 gearbox off of the Nutjob extruder and put in the Didel 51:1 gears. On a really loose compression spring setting I could get Nutjob to pump filament at about 32 mm/minute, which is what I needed. The problem was that with that loose a spring setting the extruder really wasn't gripping the filament, just moving it.

After a great deal of fiddling with settings I finally detached the GM-17 from the extruder and discovered that the terrific torque that the tin can stepper was putting into the gear train was tripping the gear clutch. Whereas in the GM-3 the gear clutch made a huge racket when it slipped, I only got a very quiet buzz out of the the GM-17 gear clutch. This was keeping the torque in the GM-17 down to a very low level. Mind, it was high for a Solarbotics gearmotor, but nowhere near what I needed.

I used some epoxy to lock it down. It should have set by morning and I'll give it another try then.

More thinking about Nutjob

Saturday, 9th May 2009 by Forrest Higgs

I'm not so happy with the Mk I Nutjob. I finally got the GM-17 hack tweaked to give what Nophead got from his. It turned out that I I was running it on wave stepping instead of full stepping. Wave stepping only energises one coil at a time. As a result it gets half the torque. Once I reprogrammed it to do full stepping it began to behave properly.

Unfortunately, I seem to be losing a lot of energy running that big gear pair in Nutjob's housing. I can get Nutjob to do what I want but it seems to me to be too sassy for a bright twelve year old to keep running properly. The whole point of a 3D printer is to be able to walk off and leave the damned thing. Unless something magical happens with Nutjob, I can't see it being any less hands off than my milling rig with Tommelise 2.0. That's simply a non-starter.

What really annoys me, though, is the size of the Nutjob. Compared to those little dinky pinch wheel extruders that Adrian and Nophead are making and now that Nophead has figured out how to run one without a NEMA 17 stepper, Nutjob is HUGE!

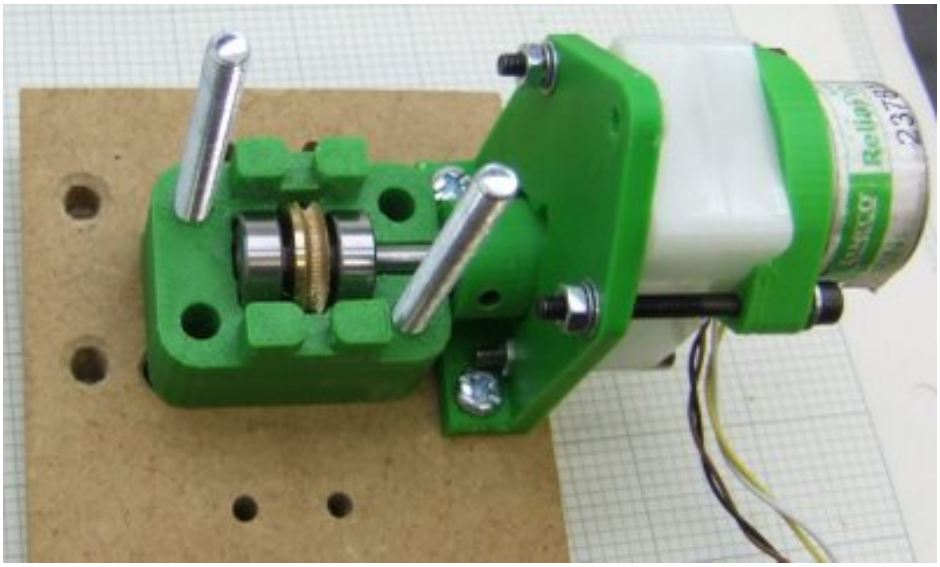
I am thinking of leveraging Adrian's method of putting splines on his NEMA 17 and Nophead's GM-17 hack, which I already have running, to make a splined shaft pinch wheel. I'm going to have a go at that this weekend.

Some notes on the way to an easy-to-make pinchwheel extruder

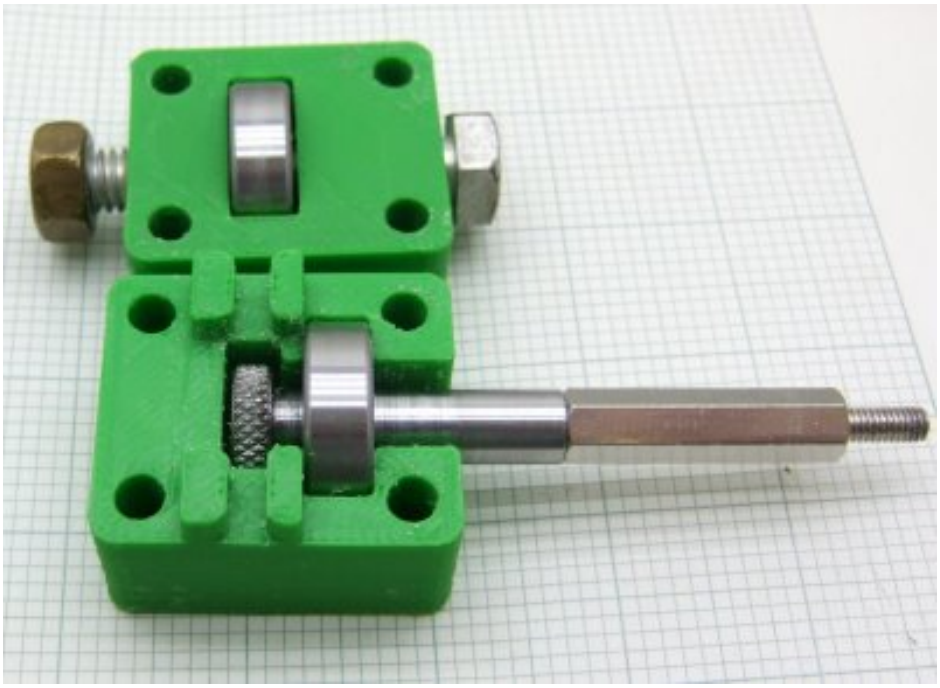
Friday, 15th May 2009 by Forrest Higgs

In which your narrator experiments with ways to make Nophead's GM-17/tin can stepper hack pinchwheel extruder without a lathe...

After working for a while with the Nutjob extruder, I came to the conclusion that it was too big, complicated and didn't perform well enough. I backtracked and decided to use my knock-off of Nophead's GM-17/tin can stepper hack and do a pinchwheel. I had a problem, though, in that I didn't really have a lathe that I could use to make a gripper like Nophead did.



Looking over his work I decided that I could make something like the knurled pinch wheel that he had also tried.



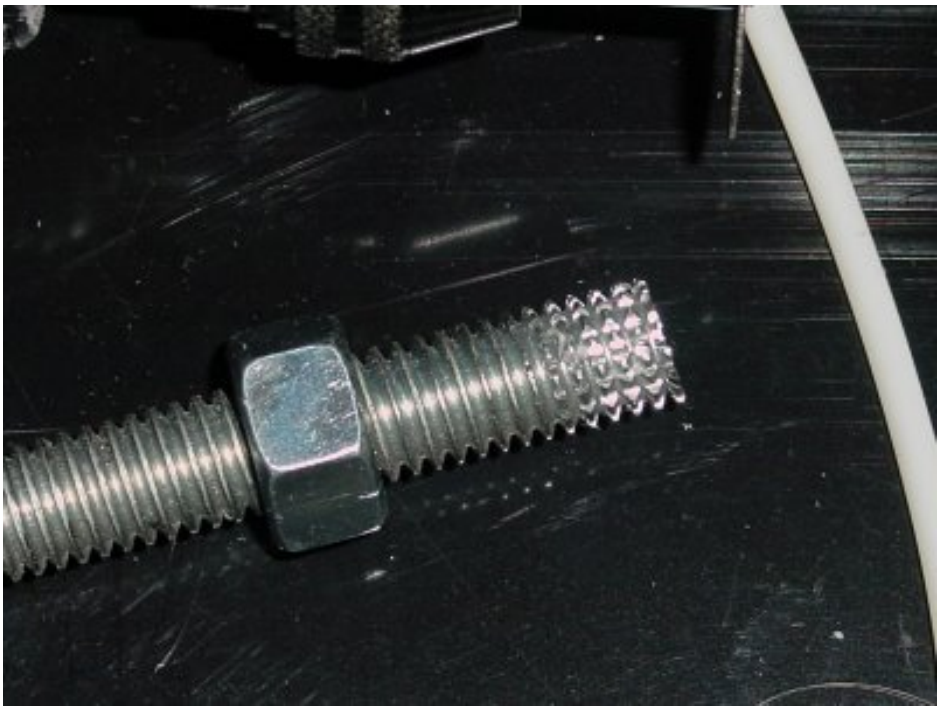
I took an hour off Sunday and spent time at my local hardware store seeing if there was something I could use as stock to do such a knurled concept pinch wheel. Sure enough, Ace had some nice stainless steel bolts made for hex wrenches.

I figured that I could use my little diamond cutting wheel to cut some knurling into the business end of one of these. Notice that I've gone for 3/8-16 bolts which go with the flanged, 3/8ths inch bearings that I have lying about from Tommelise 1.0 days. I'd have probably gone for an M8 bolt and skateboard bearings except they didn't have this kind of bolt in M8, this being the US, and the surf and skateboard shop where I can get M8 bearings is closed on Sunday. Pay attention! This is how critical design decisions are usually made, by me at least. :-)

I bought two of these stainless steel bolts so that I could afford some mistakes. Being Scots-Irish and the bolts costing $2.80 a pop, I wasn't excited about making a lot of mistakes. It wasn't long before I ran into design reality when I discovered that cutting knurling into what turned out to be a very hard stainless steel bolt was more than I could deal with.

Another reality I encountered was that for milling the thickest sheet stock I had was 3/8ths inch. Making a connector between the GM-17 gearbox and the bolt was going to require something a bit thicker than that. I either had to order thicker stock, which is expensive, or bolt two thicknesses together with a few #4 machine bolts. That was awkward and I had serious questions about how something like that would hold together with the torque loads that were going to be generated.

I started fooling around with some scrap that I had to see if I could solve these sorts of problems. The first thing that seemed promising was when I simply cut paths parallel to the bolt axis in the threaded part of a piece of 3/8-16 threaded rod.



Just for fun, I tried using the diamond wheel on the threads of the stainless steel bolt and discovered that I could actually cut those pretty quickly.

That cut held the filament as well as the threaded rod cut. It then occurred to me that if I took advantage of the hex pocket in the bolt I could easily make a 3/8ths inch thick connector to connect the bolt to the GM-17 gearbox.

This humble little gripper was able to handle 10+ kg before the connector between the filament and the spring scale failed.

You see only 4 kg of load here. That's all I could manage holding the scale in one hand and the camera in the other.

The moment arm on the gripper is about 4.5 mm. Combine that with a 10 kg load and you see that you are looking at a torque of...

torque = (10 kg)(1000 gm/kg) (4.5 mm)(.1 cm/mm) = 4500 gm-cm ~ 62.5 oz-in

...which is about what a low end NEMA 23 or a high end NEMA 17 can deliver.

Now, if we leave the GM-17 hack gearbox at its original 228:1 and calculate torque assuming 150 gm-cm which is the holding torque for the Jameco tin can stepper

torque = (150 gm-cm)(228) = 34200 gm-cm ~ 475 oz-in

Figuring that a 3 mm filament has a cross-sectional area of about 7.1 mm^2 and a 0.5 mm filament about 0.2 mm^2, you can extrude 35.5 mm of 0.5 mm thread for every mm of filament consumed.

Thus you'd need about 1.4 mm of filament/sec to extrude at a rate of 50 mm/sec. To do that you need to be turning the gripper at about 3 rpm. We should be able to do that pretty easily without overheating the tin can stepper. I already know that I can get 5 rpm out of that hack at that gear

ratio, so I should be okay.

A new controller board sooner than I'd thought...

Sunday, 17th May 2009 by Forrest Higgs

I was annoyed with the limited power I was getting out of my Haydon3600 series linear stepper that I am using on my z-axis. I'd plannedon shifting over the controller board for Tommelise 3.0 to I2Ccontrolled 754410's to drive them and I already had a prototype boardthat I'd used to run the NEMA 17's on the 3D scanner that I bloggedabout some time ago.

Because I'd used inverters on the T2 board I was limited to wavestepping. For T3 I wanted to see if I could get full stepping and halfstepping going so that I could double the resolution on T3 for milling.

Just for fun I hooked the x-axis up to one of the stepper controllerson that prototype board and to my shock I was not only able to run thattin can stepper on full stepping but I was also able to drive thez-axis up and down with plenty of power at 30 mm/sec.

I am going to have to do some serious thinking about the implications of all this.

I2C-based Reprap control

Tuesday, 26th May 2009 by Forrest Higgs

In which the control strategy for Tommelise 3.0 begins to emerge ... with implications for mainstream Arduino controllers...

I joined the Reprap project in 2006. At that time we were using Microchip 16F628A MCUs and the old SDCC open source compiler. I was Windows bound at the time and could never manage to get the cranky SDCC compiler to work on my system. I wanted to work with the firmware and finally acquired a cheap Serbian BASIC IDE for 16F family chips made by Oshonsoft that I could work with easily. From there I quickly migrated to the much more capable 16F877A and, when I discovered that stack space in the 16F family of MCUs was very limited, made the jump to 18Fs instead.

I began with the brilliant 18F4610 which I used to power Tommelise 1.0. I wanted to move over to USB comms from RS-232 and found the 18F4550 which had integral USB capability built-in when I began to build Tommelise 2.0 in 2008. By that time the mainstream of the Reprap project had moved over to the Arduino boards and the gcc C language compiler which they still use today. I was happy with what I had and stayed with Microchip processors.

About that time Reprap ran into a data transfer bottleneck. You could either transmit compressed control commands from your PC and have the MCU decode and expand them or you could send over less compressed data and buffer it. There has been a problem in that if you send compressed commands the decoding and expansion of them takes significant time which can cause pauses in printing operations. Given that extruders have a rather large lag time, that is not a good situation. The other option is to create a data buffer.

Most MCUs tend to be short of RAM memory. I took a rather heterodox approach to the problem. I decided that it would be nice if the PC could create pretty much completely decompressed data, viz, direct instructions to run steppers and extruders, and store them on a large buffer. The MCU would then only have to read the instructions off the buffer and execute them with no decoding to speak of.

I soon discovered that EEPROMS could be turned to this task. The 1 MegaBit 24FC1025 proved to be very well-suited to the task. You could address eight of them via an I2C bus from your MCU for a full megabyte of data buffering. While writing to them took a while you could read from them very quickly. I was able to build a half megabyte buffer with the 24F1025 which has proved very robust as a data buffer. Depending on the kind of milling or printing it is put to it can store anywhere up to 90 minutes worth of data.

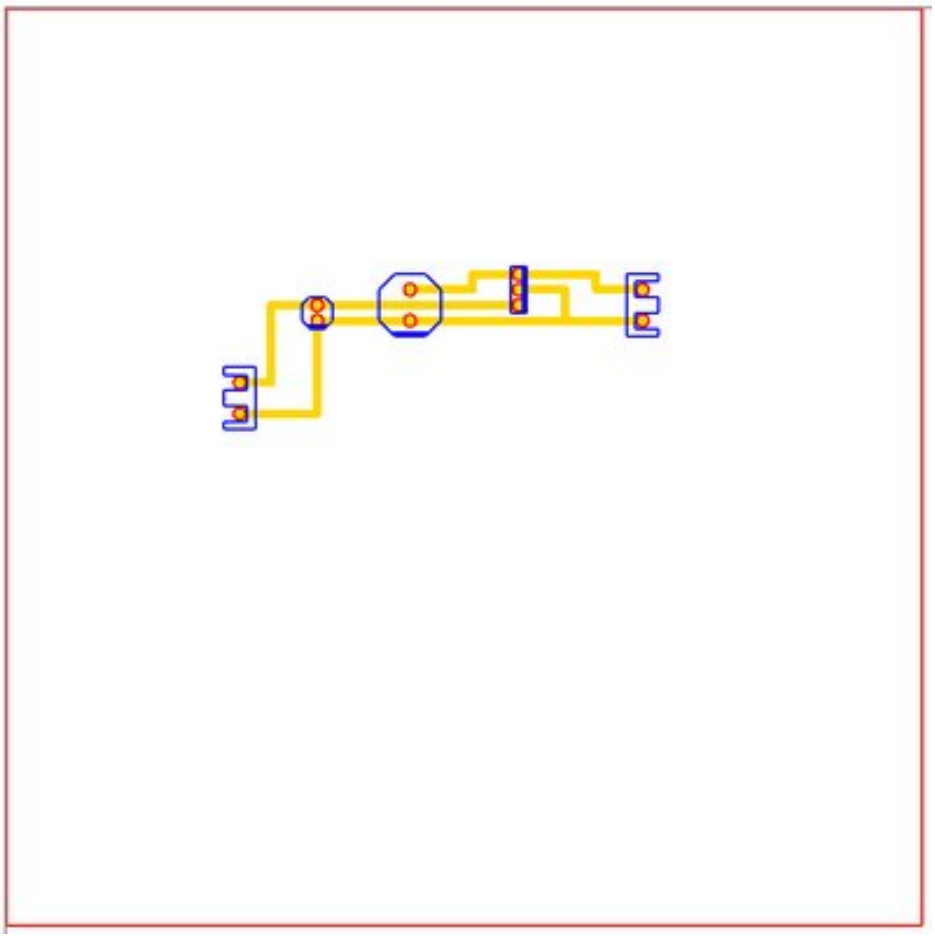
The I2C bus proved to be a very useful way of organising things on a board. Before long I discovered the PCF8574 I2C slave chip. You could use one of these with virtually any chip made

and reduce the connection to your MCU to a pair of traces. Instead of boards which were mare's nests of traces you could group a slave chip with a particular chip and then hang the slave on the I2C bus.

A few months ago I demonstrated that the PCF 8574 could successfully control an SN754410 half-H stepper driver chip with no trouble whatsoever. That success led me to commit to using an I2C bus architecture for Tommelise 3.0.

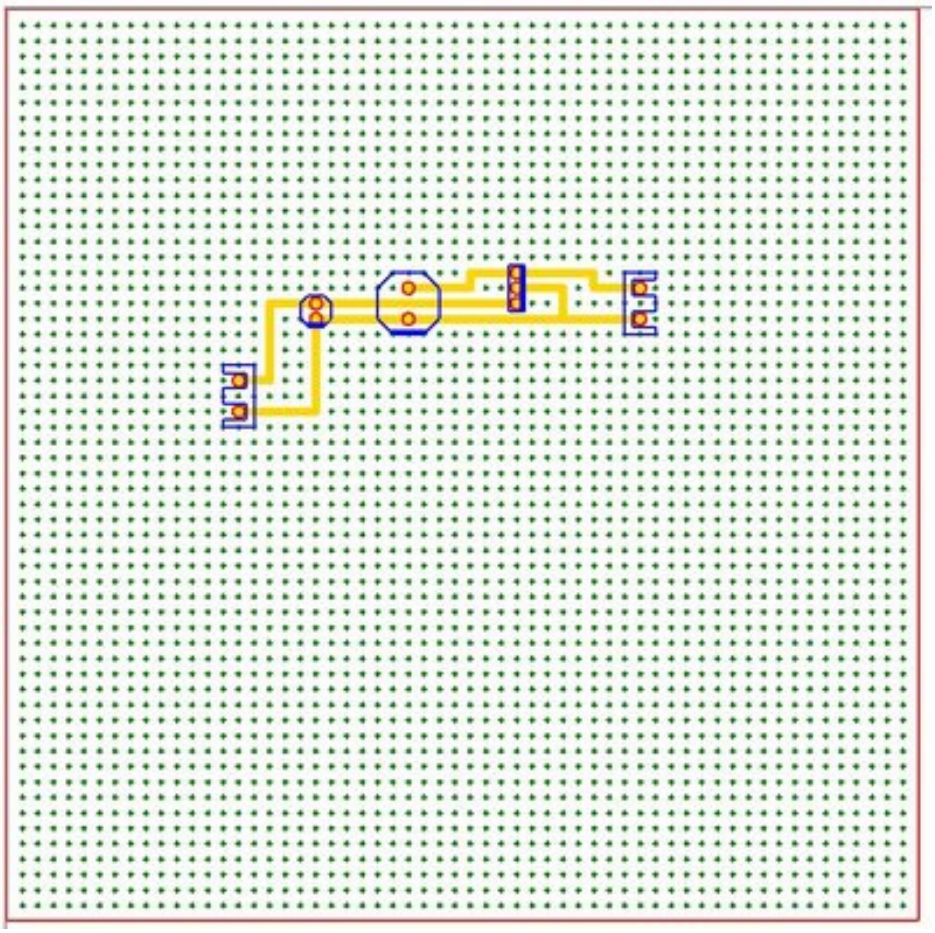
It's all very well to talk about such things, but quite another to actually see the possibilities. In order to let you see what is possible, I've undertaken to document the architecture of what I know already works. I've reduced the architecture to a series of modules connected to an I2C bus for simplicity. In reality it matters little whether you create a bunch of little module boards or park all of the circuitry on a single board. Modular boards are simple and flexible while big boards tend to be better if you have a firm idea of what your system needs.

I'll begin showing you the system with a basic power conditioning circuit that dates back to the 16F628A days.

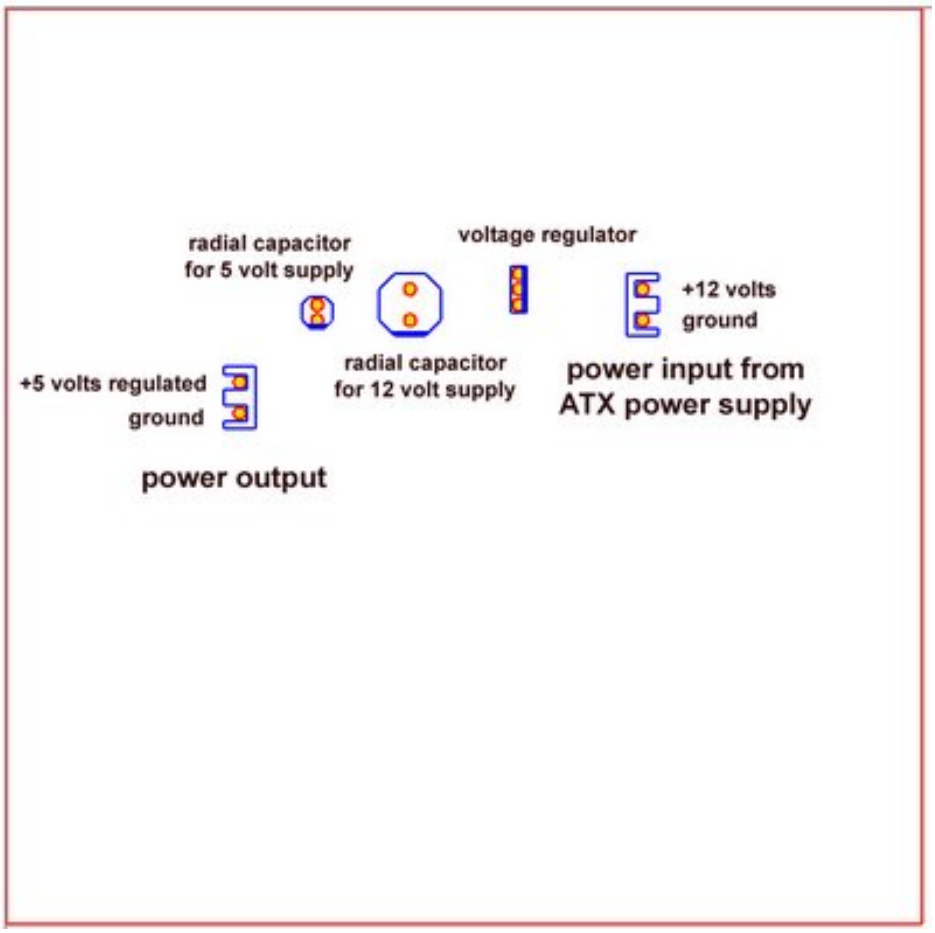


I've never found cause to change it because it simply works. You take in 12v DC power from an

ATX power unit that you can either buy or salvage from an old PC and put out +5v stabilised power for your Reprap. The circuit also has a nice big radial capacitor to smooth out any little wrinkles in your 12v power.

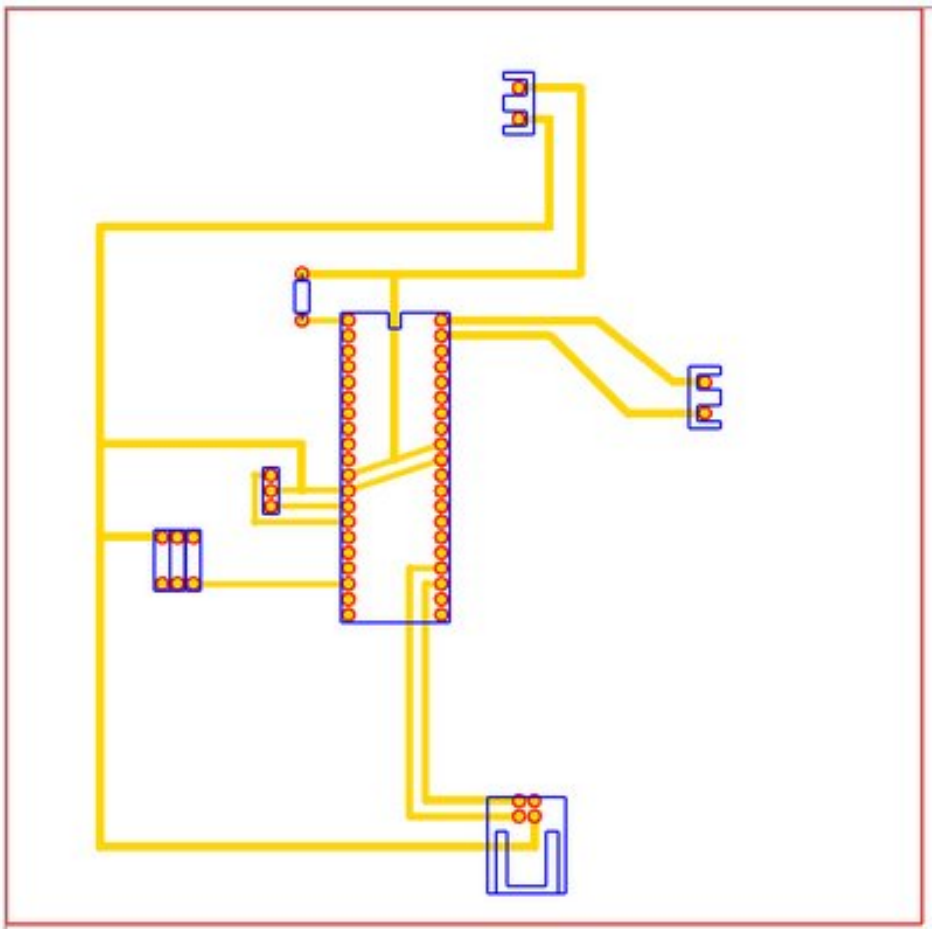


I've overlaid a standard 0.1 inch grid on the circuit so that you can get an idea of scale. Looking at just the components, you see them identified.

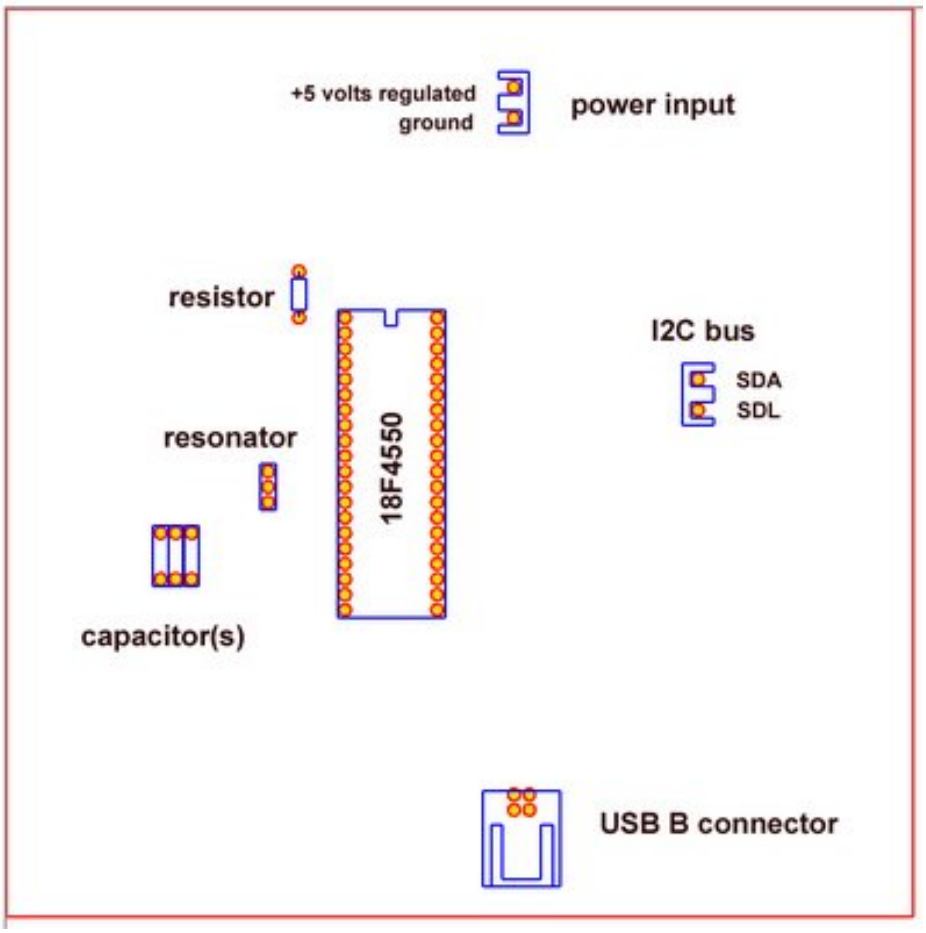


The blue stripe at the bottom of the capacitors corresponds to a stripe put on the capacitor cans which indicates which connector goes to ground. Get that backwards and you ruin the capacitor. Those connectors are standard 2-pole screw connectors that you can get from Radio Shack, Mouser or any of a hundred suppliers. There are cheaper and more task oriented connectors, but I've always liked connectors that I can just put a wire into without further ado.

Once you have power it is a simple matter to describe the MCU circuit.

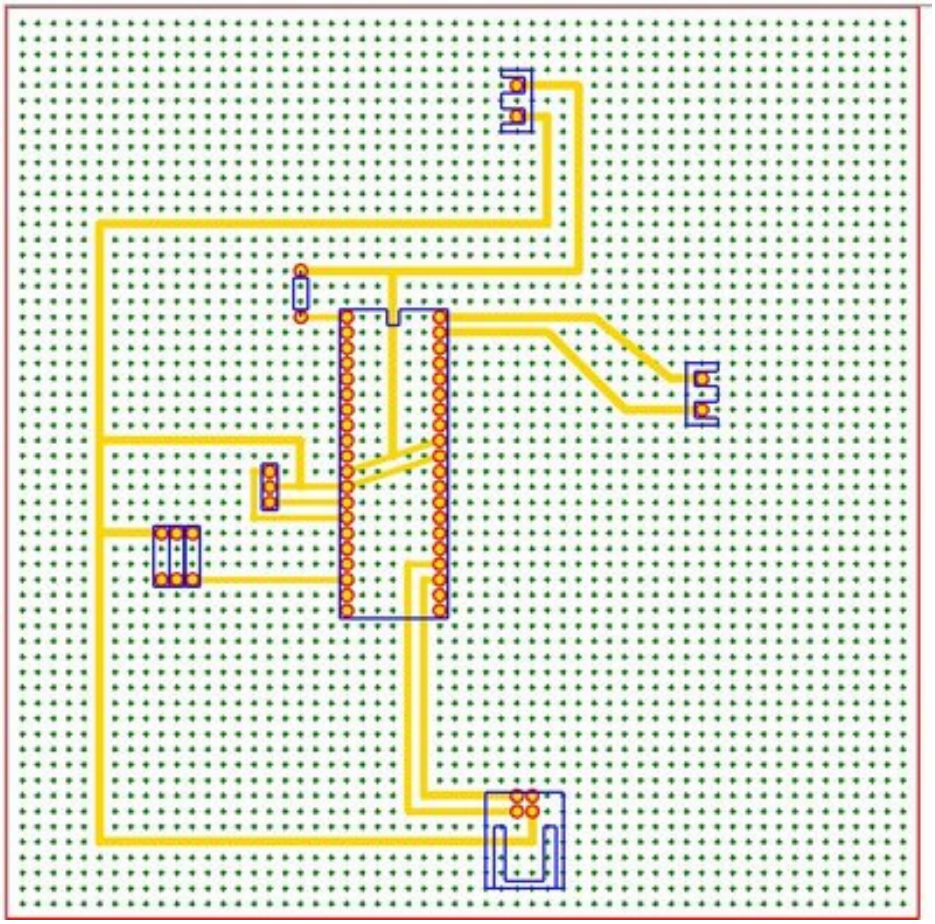


The Microchip 18F4550 is a 40 pin chip, rather big but not very expensive. Using the 4550 lets you dispense with silly RS-232 to USB chips and the whole mismatch between RS-232 as your PC understands it and RS-232 as your MCU understands it. You can see that there is just not a hell of a lot on this board. We also only use two port B pins (7 and 8) to drive the I2C bus. Nothing else is necessary. Looking at the components.

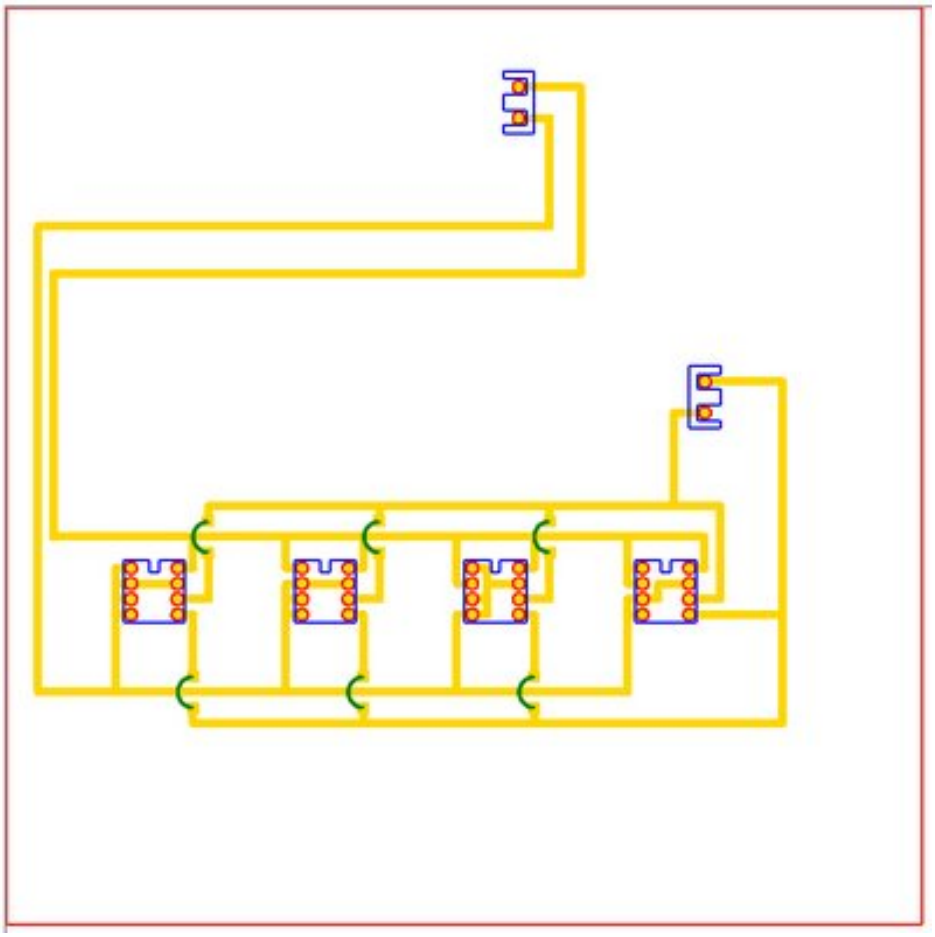


As you can see, there isn't much required on this board. There is a 10K ohm resistor between +5v and pin 1 on the 18F4550 and there are three (3) 104 nF disk capacitors that are required for the 3.3 v input to the USB power supply if you're not actually putting power into that pin. The MCU uses a 20 MHz resonator for clock timing. Aside from a couple of 2 pole connectors and the single USB type B connector, that's about it.

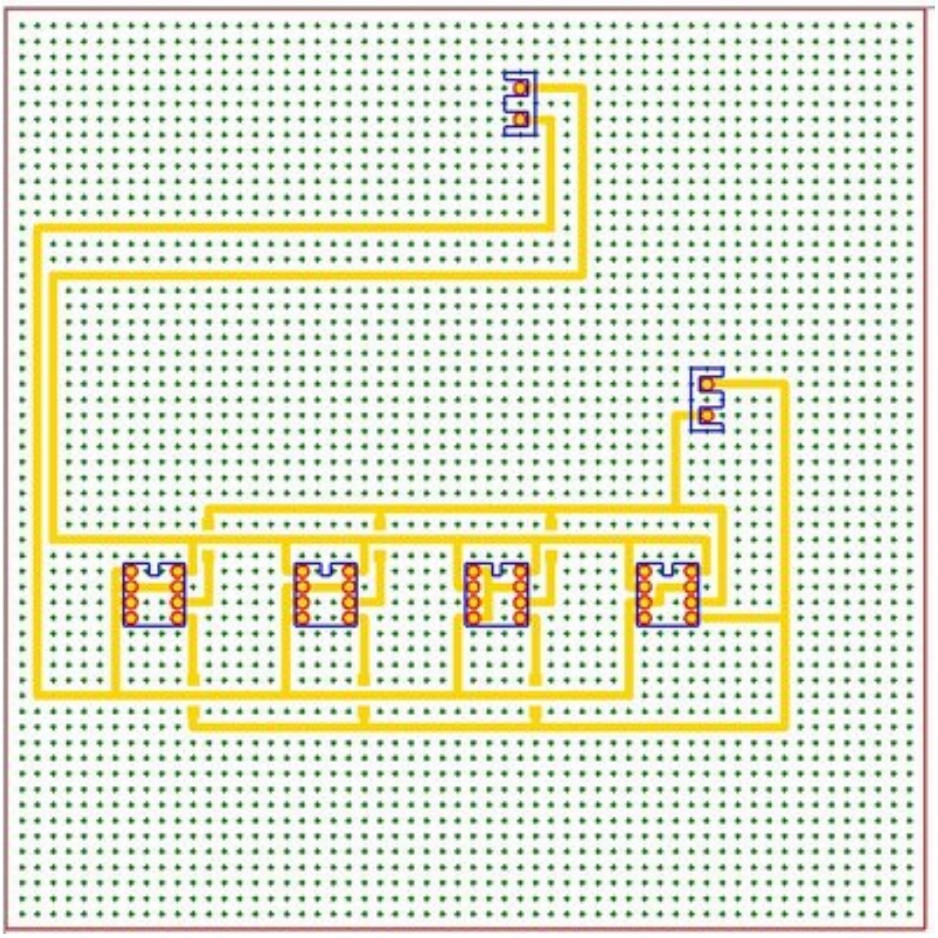
Here you can see the traces that connect everything. I plan on milling mine, but there is nothing to stop you from using stripboard or any of half a dozen other ways of connecting the components.



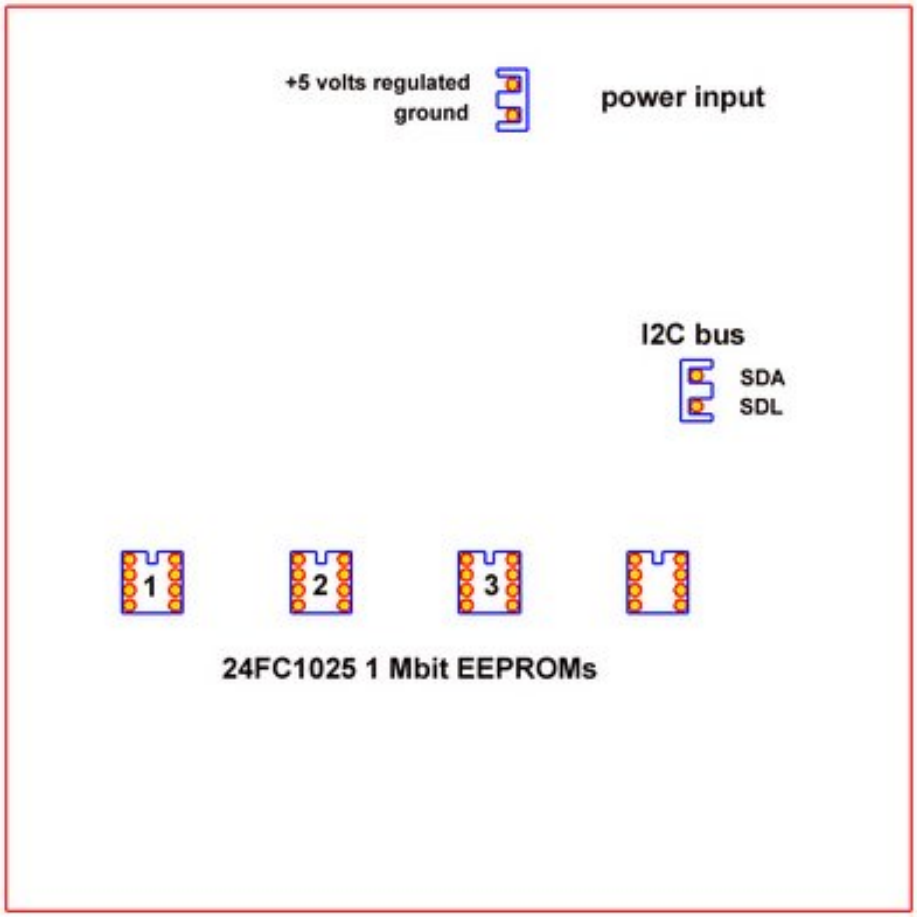
The EEPROM board is even simpler.



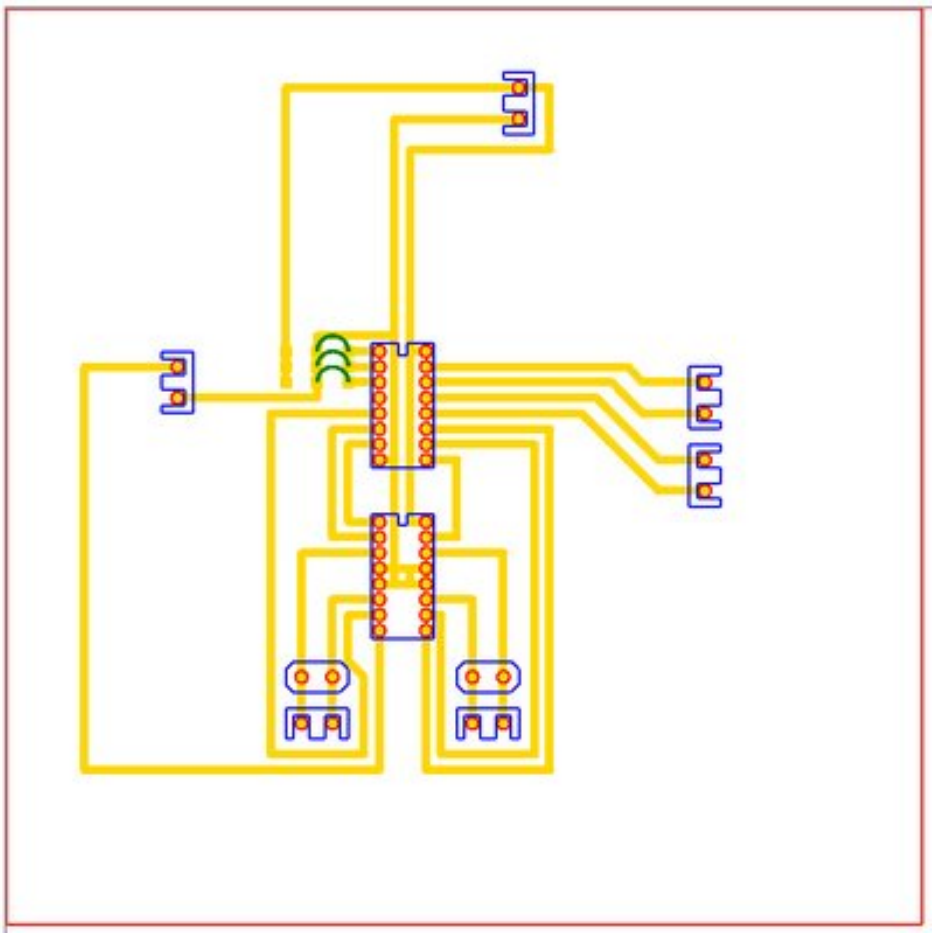
Those little green arcs are jumpers. I've designed the boards to be single sided, so jumpers are required from time to time. Not many, though.



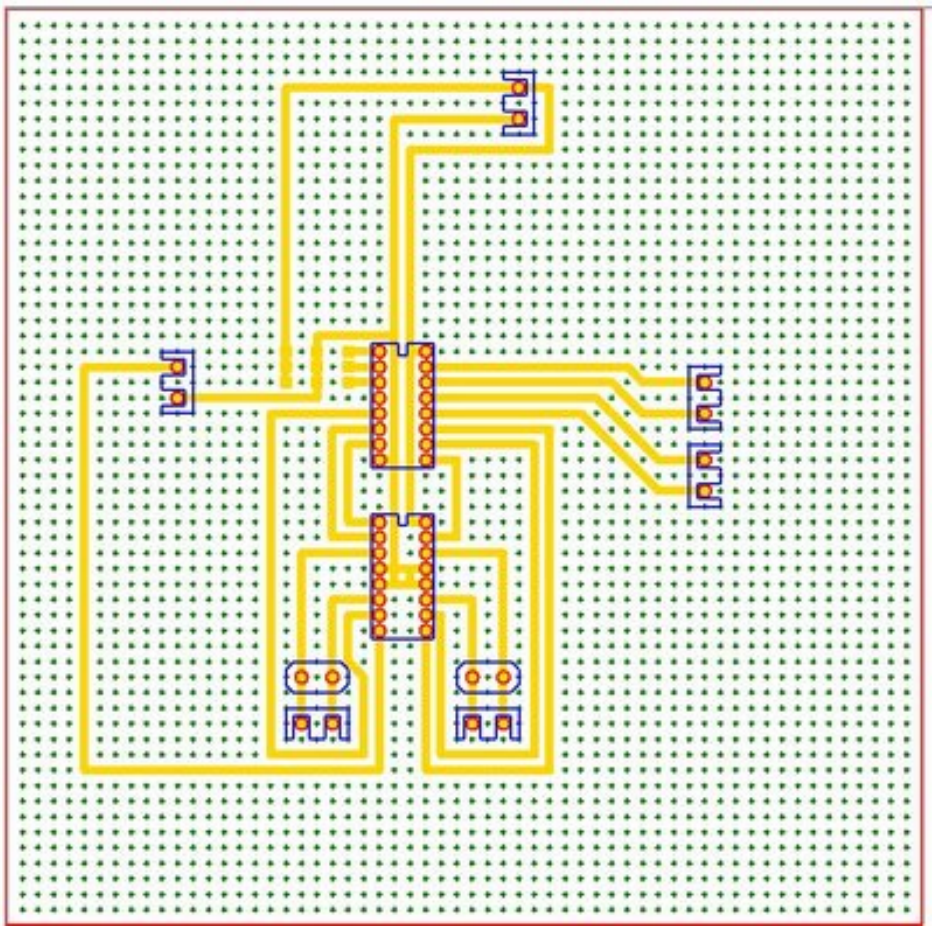
For components you've basically just got a few connectors, jumpers and the EEPROMs themselves. I've configured this board so that the EEPROMS fill 0-512 Kbytes. I could have made a single layout that you could use for both the bottom and top half of the megabyte that these EEPROMs will allow you, but that would have required four more jumpers which I felt would confuse matters for now.



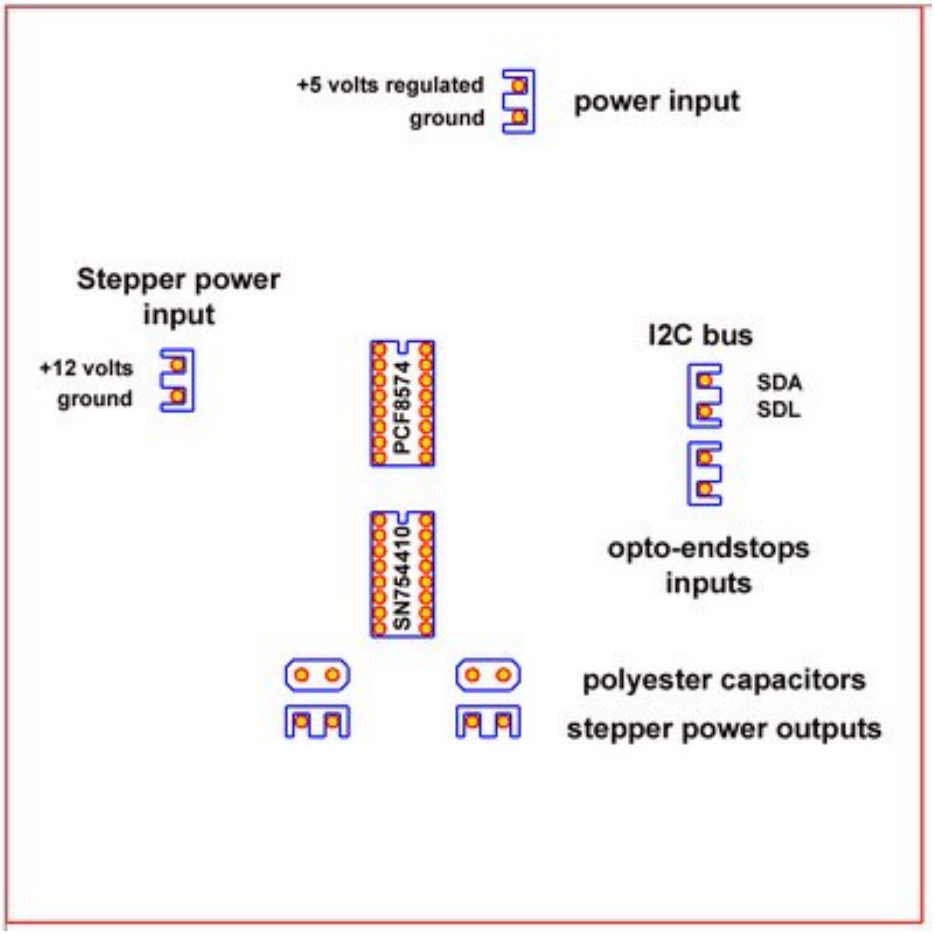
Moving on to the bipolar stepper controller board, you see that it is similarly rather trivial.



The three green jumpers on this board set the address for the I2C bus. You are going to want to control more than one stepper presumably, I couldn't just set the address to a single set of values.



Components are similarly skimpy.



That's basically it, except for a slave MCU circuit that uses an MCF8574 and another, smaller Microchip MCU chip to look after the extruder. There will be a little pushing and shoving, but don't expect much in the way of complexity.
I already have a unipolar stepper controller board designed. I would have included it here, except that I haven't incorporated the extra diodes into it that Nophead says I need to make it really robust.

Any of the I2C slave boards could be incorporated into the Arduino/Sanguino system without a hiccough. There is nothing sacred about the Microchip MCU that I'm using. You could replace my 18F4550 with an equivalent Atmel chip and program it with gcc. About the only obstacle you'd run into would be that Atmel only offers an MCU with integral USB circuitry in surface mount technology. The 4550 is through-the-hole DIP technology, which, imo, is easier for people with more than two thumbs to work with. :-)

I'm committed to designing Reprap machines that can be pretty much built from the ground up by ordinary people. I think the current arrangement goes a long way toward making that possible.

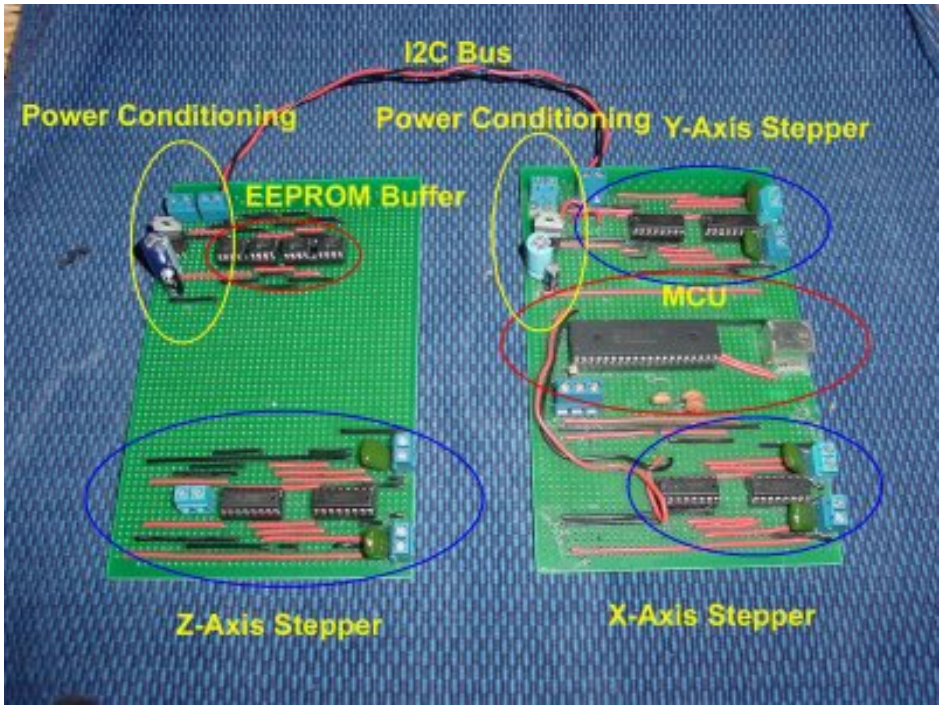
Tommelise 3.0 I2C board finally built

Saturday, 6th June 2009 by Forrest Higgs

In which your narrator turns designs into hardware with remarkably little trouble...

Some time ago, I built up an I2C stepper controller board which can control two steppers to run my IR ranging scanner. This project allowed me to test out the viability of using I2C comms with I2C slave chips and to run open loop steppers with them. The board was pretty clean and very effective.

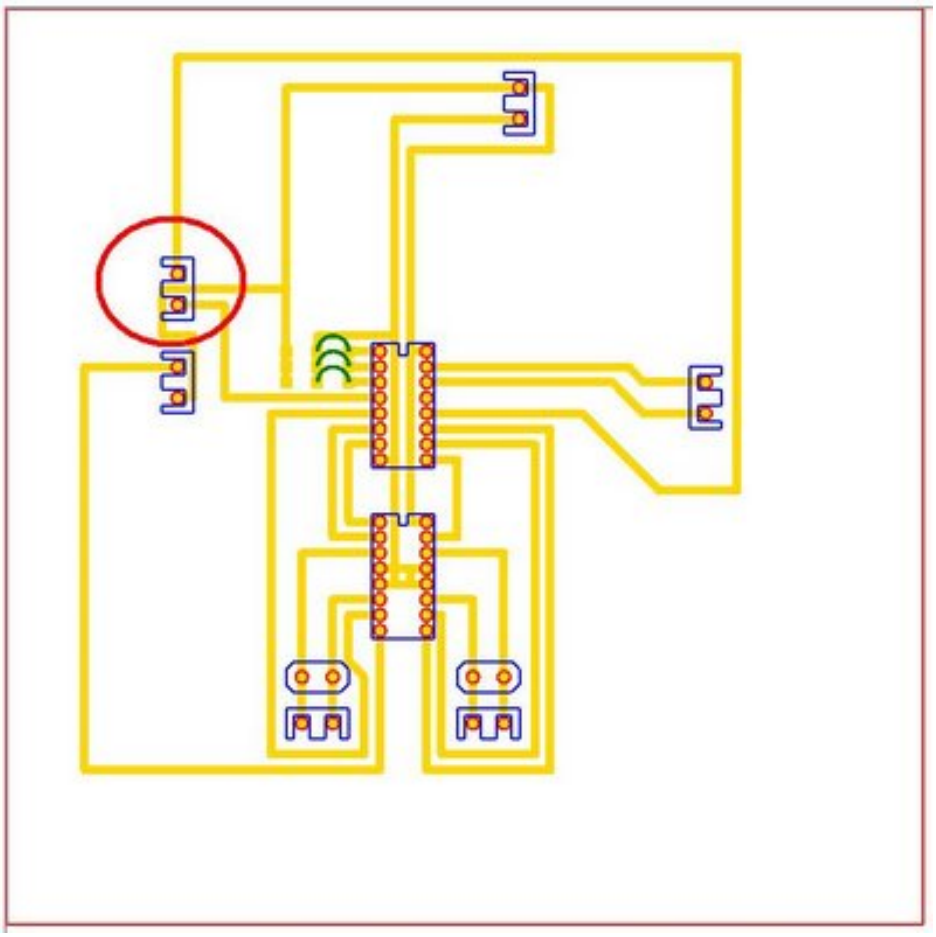
Rather than build a whole new board for Tommelise 3.0 I'm modifying that one and parking the third stepper controller and the eeprom memory buffer on a second board connected to the first by an I2C twisted pair. Here you can see the result.



The IR scanner board is on the right and the extender board connected to it by the I2C bus is on the left. You can see that there is considerable real estate in the middle of the board left free. I might be able to park the extruder controller there, but if not, I just create another extender board.

I included the possibility of handling opto-endstops on the z-axis stepper controller. I need to add that in to the x and y-axis steppers on the old board. That is not a big thing to do.

The circuit diagrams shown in my previous blog entry, I2C-based Reprap control, are exactly what was used in the new board with one exception. The stepper controller circuit diagram that I posted had a small bug in it that I uncovered when I actually went to make the board. Here is the corrected circuit layout.



The two pole connector that was on the right hand side of the board has been moved to the left (circled in red) above the 12 volt power feed and a mistaken pin assignment corrected. There is room there for endstops at both ends of an axis, though I've never used more than one and that for resetting the start point of the axis.
The EEPROM buffer layout is a bit different in look from my earlier diagrams. This is a result of two factors. The first is that I'm using what is known as full-line Euro Card stripboard rather than milled, single sided printed circuit board. On a single sided pcb one tries very hard to minimize the number of jumpers. On stripboard, jumpers are the name of the game, which means that your board, though topologically identical looks very different.
As well, I used twin, 18 pin sockets to house the EEPROMs rather than four, 8 pin sockets. I had the 18 pin sockets and didn't have any 8 pin sockets.

The next step will be to test out the card to make sure everything works and then to rewrite Tommelise 2.0's firmware for the new bus architecture. Using I2C should considerably simplify the firmware. As well, I will be able to do half and full stepping with the new cards instead of merely wave stepping. That has already shown me that I can get a LOT more power out of my z-axis linear stepper. Heretofore, the lack of thrust in the z-axis has been a continuing headache for me in terms of the reliability of positioning in the z-axis. It appears that that will be a thing of the past with the new controller cards if the tests are any indicator.

More printing large objects with HDPE

Sunday, 15th November 2009 by Forrest Higgs

After I achieved a 145x5x20 mm beam in HDPE, I decided to go for the limits of what my Rapman 3.0 could achieve. Rapman 3.0 has a printing area of about 200x200 mm. My thought was that if I rotated my beam around the z-axis by 45 degrees I ought to be able to push my largest dimension out to 280 mm or so.

If I were printing a line that would be true. Unfortunately, by the time I took the width of the raft that was necessary to keep the beam tacked down during printing was taken into account, I was only able to achieve a touch less than 180 mm, a length that was about the same as I could have managed if I'd printed the beam either along the x or y axis.

If you look at the tension pads you can see that they began to separate just as the print finished. This put about 1-1.5 mm of bowing into the beam over its length.

The printing parameters for the print were:

- printer: Rapman 3.0
- firmware version: 1.0.6
- slice and dice app: Skeinforge, 2009-10-31 build
- material: HDPE {no additives}
- print speed: 16 mm/sec
- extrusion speed: 65 rpm
- fill: 40% using hexagonal pattern
- raft temperature: 225 C
- first layer temperature: 225 C
- beam print temperature: 230 C
- lab temperature: 18.3 C
- lab humidity: 50%

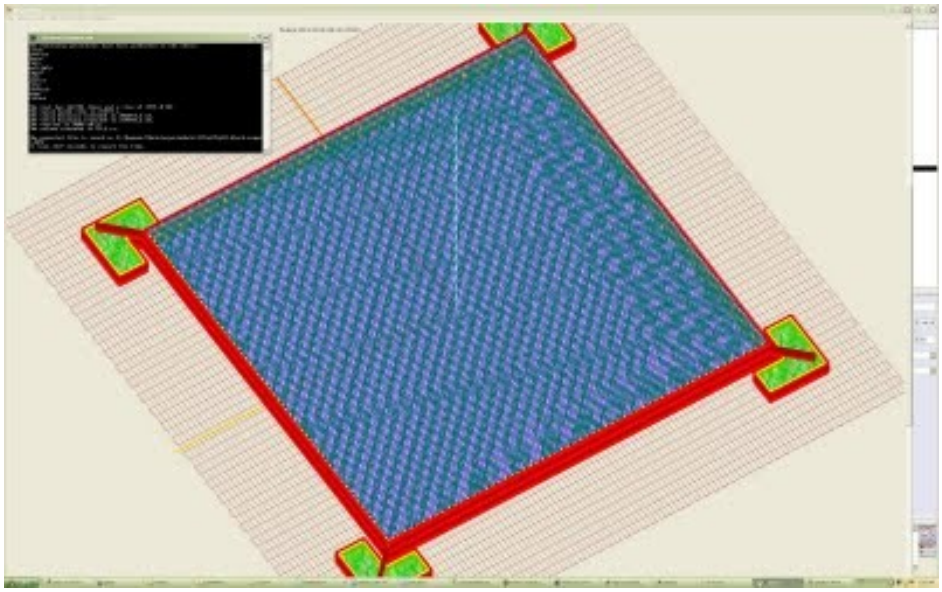
• raft perimeter width: 12 mm
• beam dimensions: 5x20x180 mm
• print time: ~100 minutes
• bowing: 1-1.5 mm over length

I am going to attempt this print again with larger tension pads to see if I can get rid of the bowing observed in this print.

Bogdan's acid test

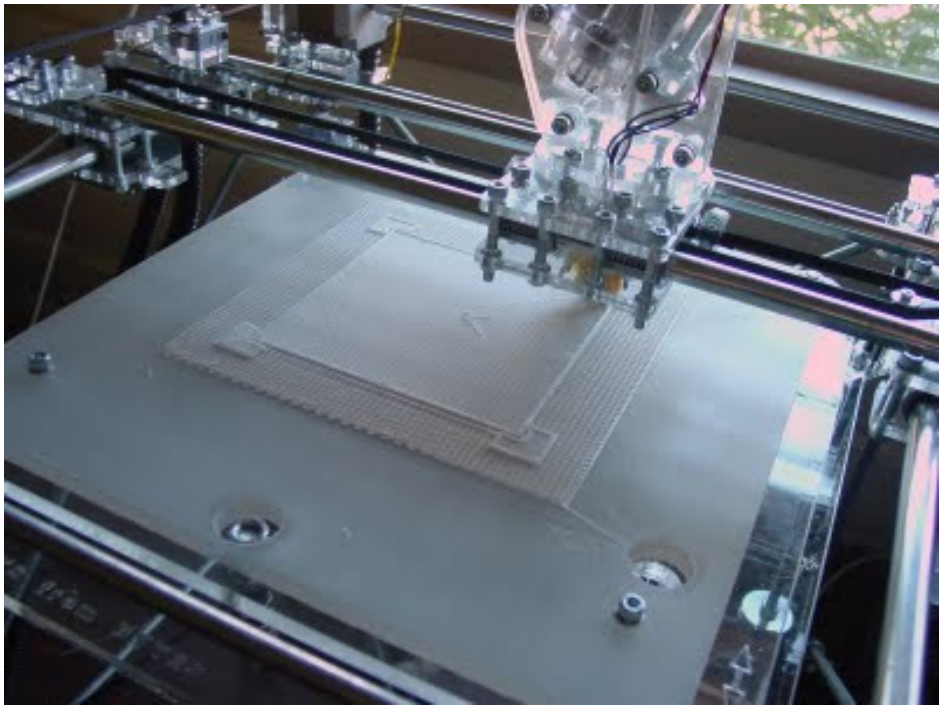
Sunday, 15th November 2009 by Forrest Higgs

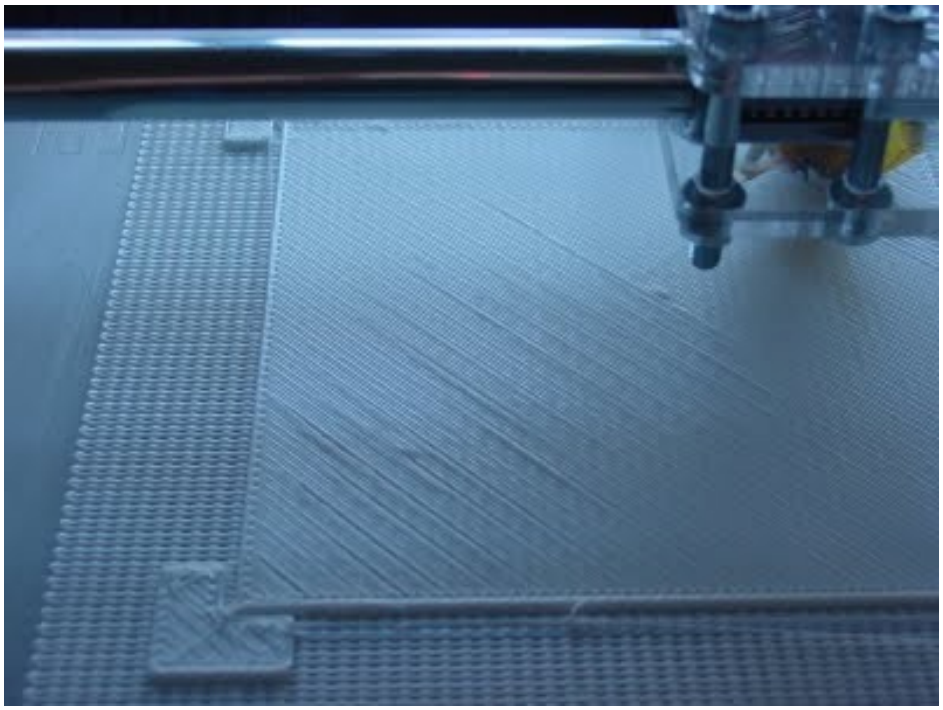
Bogdan suggested that we give the method an acid test and print a 125x125x10 block. I made up a test block in Art of Illusion and gave it a try. I ran the STL file through Skeinforge.



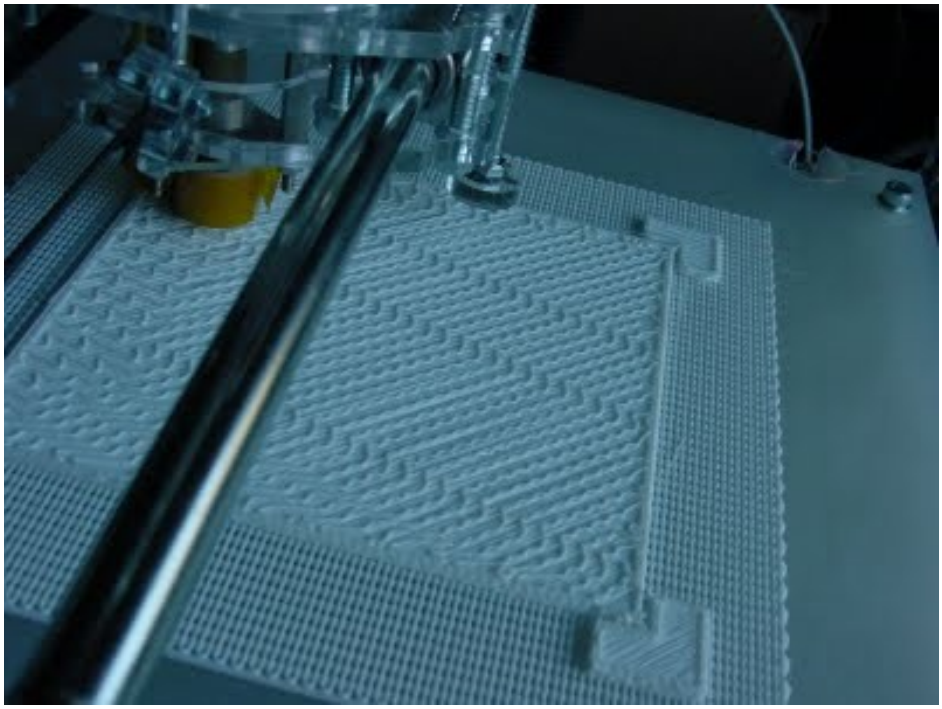
• object: cube {125x125x10 mm}
• printer: Rapman 3.0
• firmware version: 1.0.6
• slice and dice app: Skeinforge, 2009-10-31 build
• material: HDPE {no additives}
• print speed: 16 mm/sec
• extrusion speed: 65 rpm
• fill: 40% using hexagonal pattern
• raft temperature {first layer}: 225 C
• raft temperature {second layer}: 230 C
• first layer temperature: 225 C
• beam print temperature: 230 C
• lab temperature: 18.3 C
• lab humidity: 50%
• raft perimeter width: 12 mm
• beam dimensions: 5x20x180 mm
• print time: ~360 minutes
• print material ~53 cm^3
• bowing: 2-2.5 mm over one diagonal {175-180 mm}

I used "L" shaped tension pads. Here you can see the raft printed and the transition layer for the block being laid down.

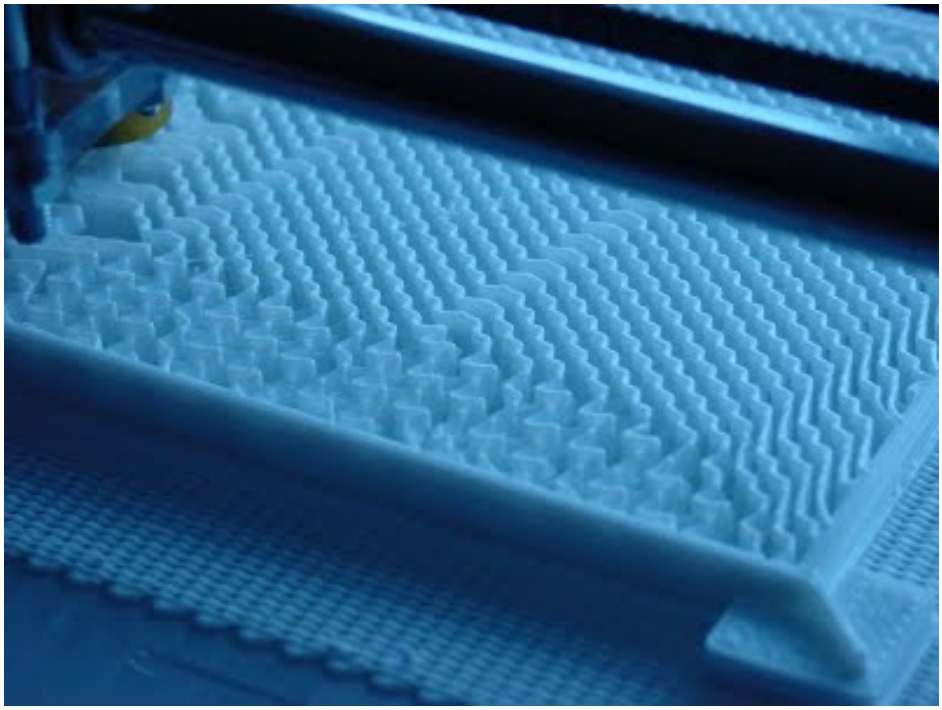


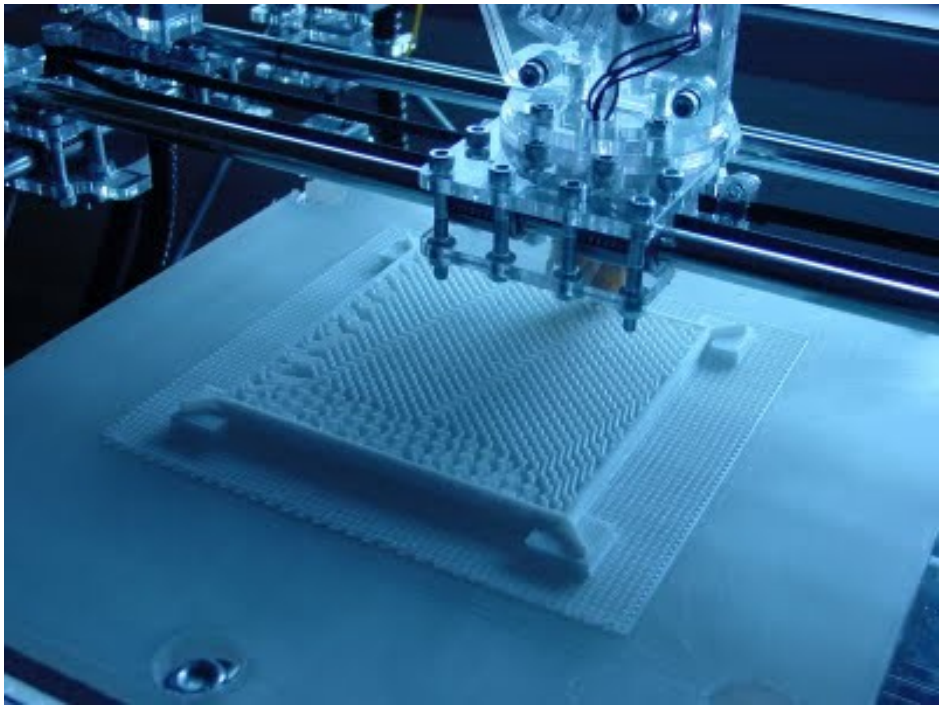


Enrique characterises this as a "hexagonal" fill for reasons that I do not fully understand.

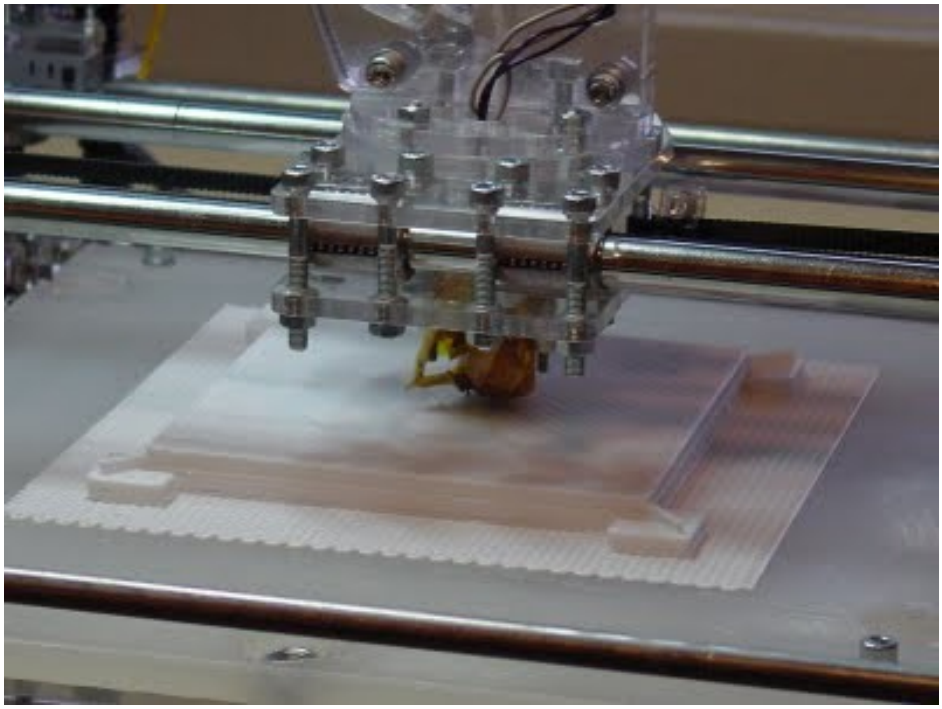


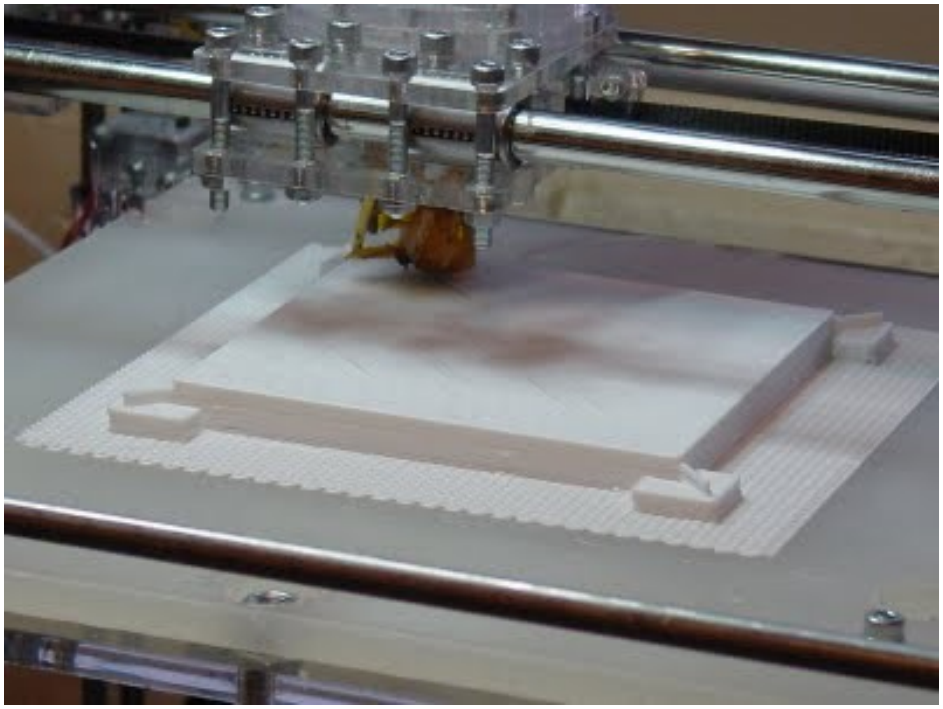
All the same, in the right light the patterns his fill option makes are ravishing.



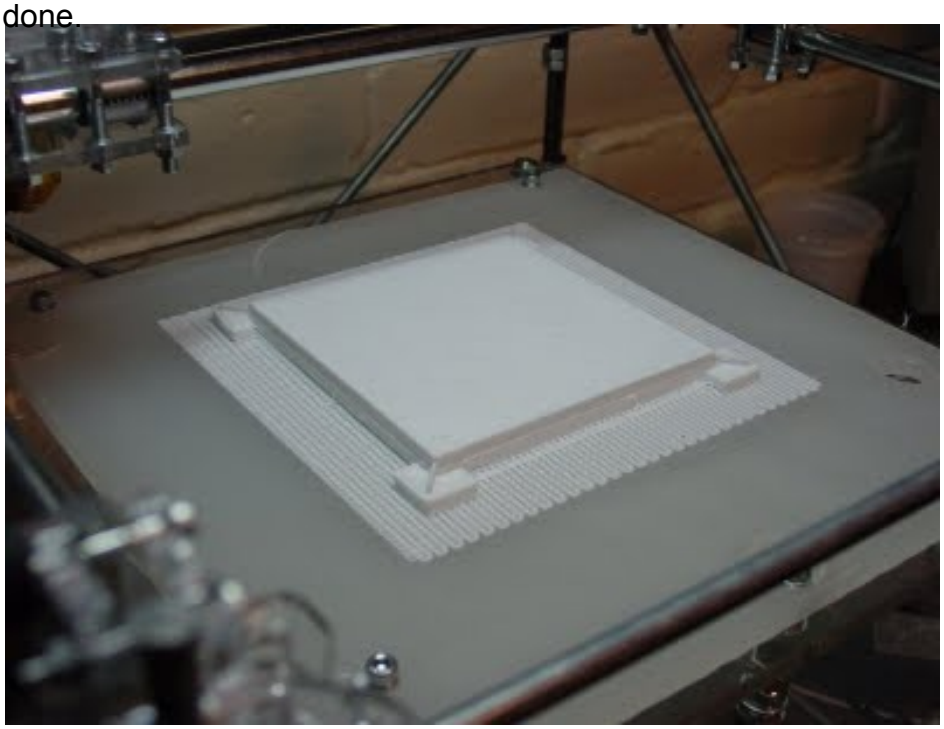


After about five hours the Rapman began to print the top on the block.
Bogdan's acid test 01 from Forrest Higgs on Vimeo.

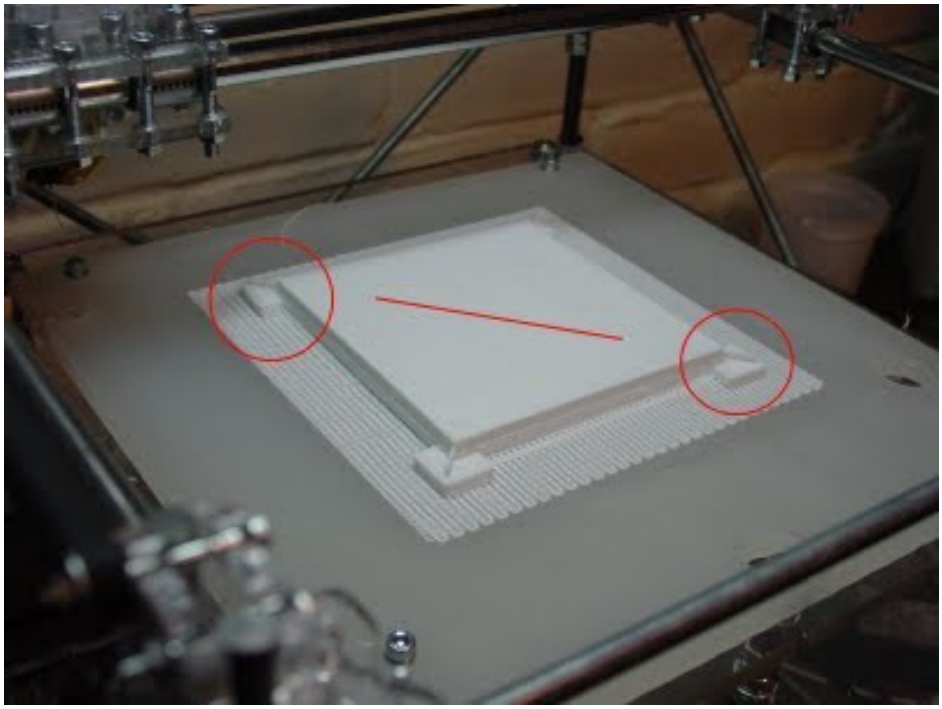




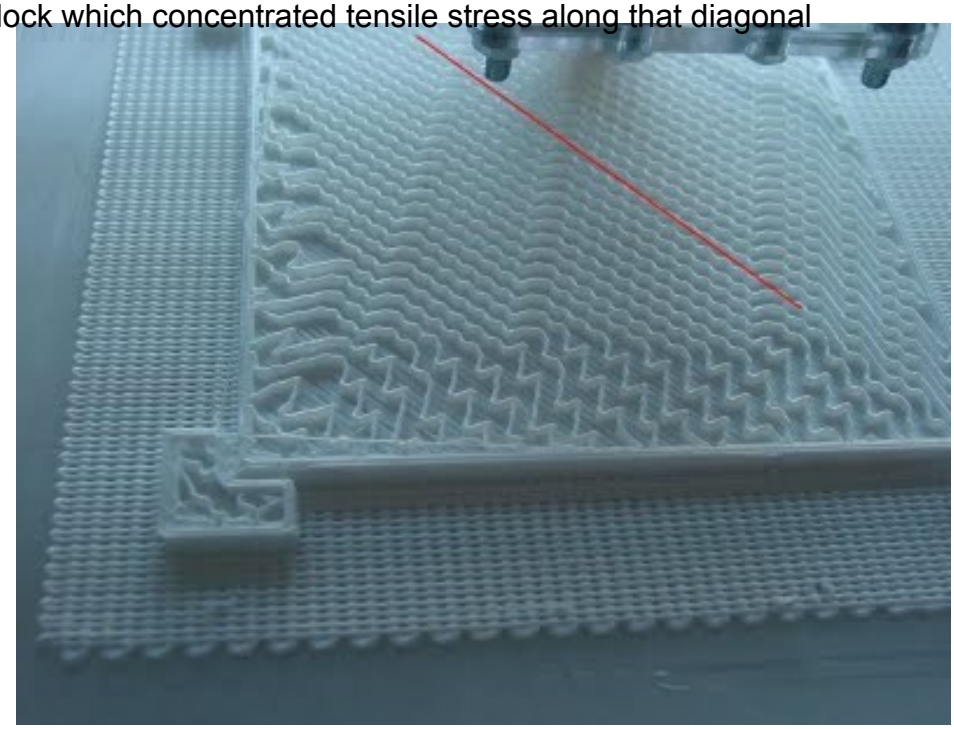
And then it was done.



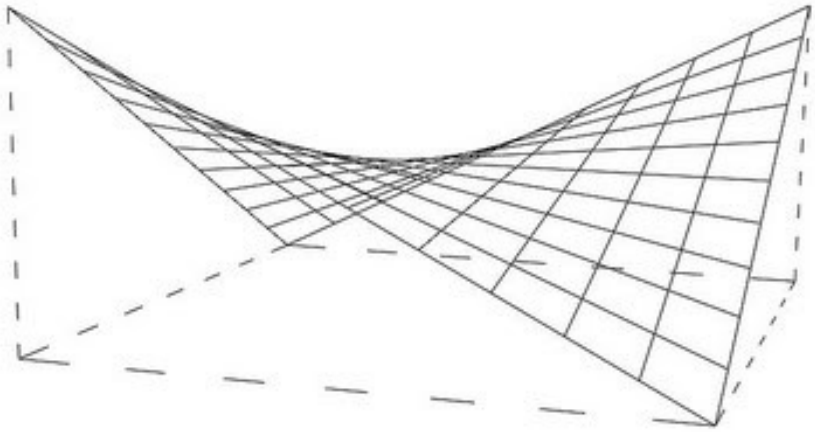
I didn't see the small amount of separation that had occurred across one diagonal until after I began to look closely at the pictures.



Looking back at the fill pattern what had happened is that the fill created a series of diagonal beams in the block which concentrated tensile stress along that diagonal



A similar shape is encountered in hyperbolic parabaloid roofs where beams are laid parallel to the perimeter boundary of the roof as in this illustration.



The difference was in our case the beams were laid on a diagonal.
The distortion along the diagonal amounted to about 2-2.5 mm over its 170 mm length. I suspect that I can cure this problem by using a larger tension pad. In the longer run, though, we've got to get a proper hexagonal fill instead of this pretty, but anisotropic pattern that we currently have.

I can't get over how nicely the polypropylene printing surface for HDPE that Bogdan suggested works. I took a short videoclip to show you just how easy it is to get the object off of the print surface.
Bogdan's acid test 02 from Forrest Higgs on Vimeo.

Please pardon my elbow.
There was no distortion along the perimeter, only along the one diagonal.

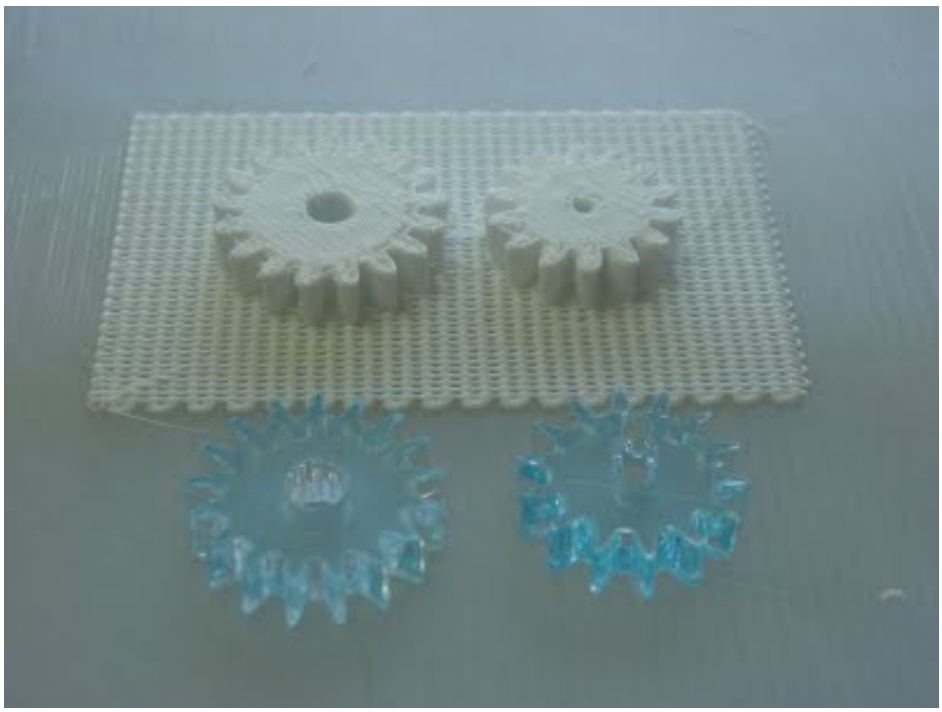
Making spares for Rapman

Monday, 16th November 2009 by Forrest Higgs

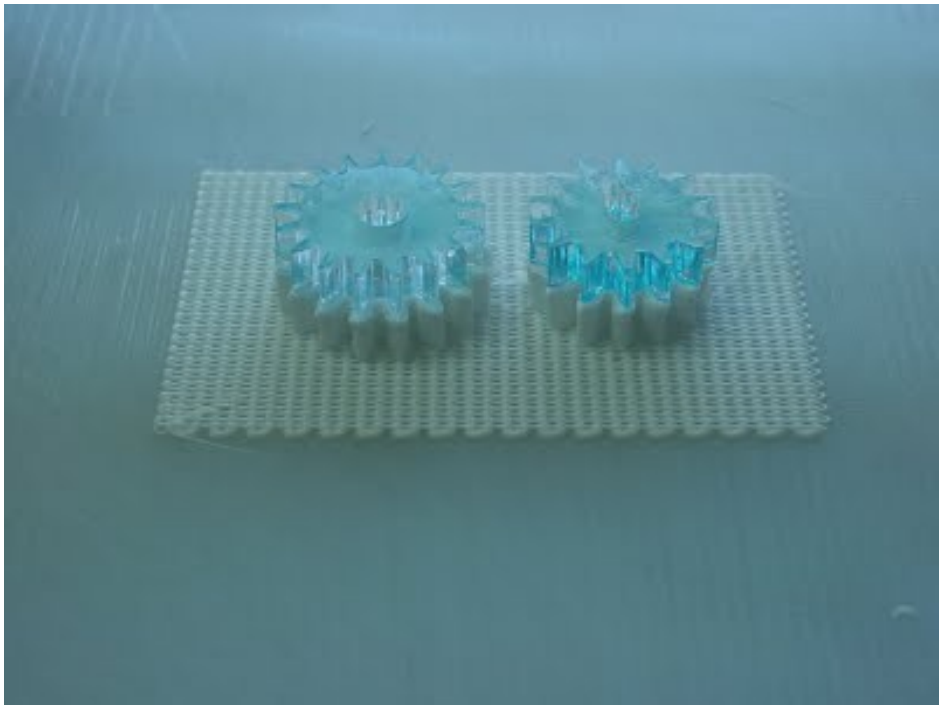
Yesterday, klaszlo, a member of the Rapman users' group wondered how he could get spares for the gear pair on the Rapman 3.0. For many of us, the whole point of Reprap machines is to print our own spares rather than tormenting Ian at BitsFromBytes every time we break something.
Did somebody already designed (ready to print) replacement gears for the extruder?
I think the gears are big enough, that we are able to print them out...

As it developed, nobody had. I wondered how hard an exercise it could be and decided to have a go at it.

I designed the gears as involute profile using a script I wrote some years ago for Art of Illusion rather than attempt to duplicate the spiky ones that come with the Rapman. The pair had a rather odd 14:17 ratio. That done I printed them out in HDPE.



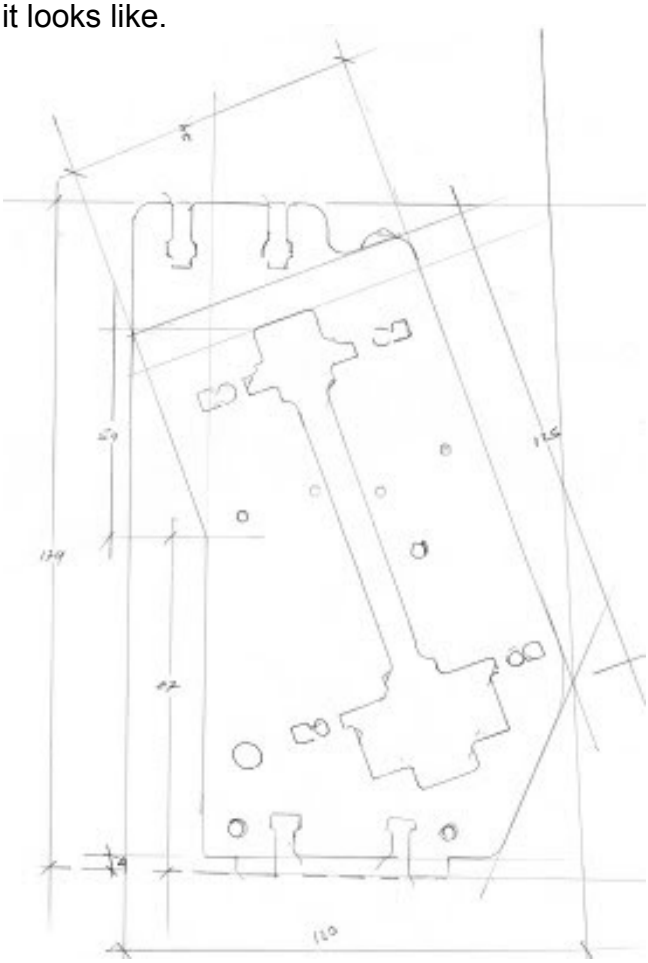
As you can see, the dimensions and tooth counts are the same. In the 14 toothed gear I used depended on a friction fit with a flat side on the 4 mm stepper shaft rather than the grub screw arrangement used in the acrylic original.



I printed the raft base at 225 C and the second layer of the raft at 230 so that it would stick properly. I also printed the transition layer between the raft and gear at 215 C. I've not had any trouble getting the raft off of the polypropylene print surface. Getting the raft off of the printed part when the transition layer is 225 is nearly impossible, however. The 215 C setting made separating the gear from the raft barely possible. I will be trying lower temperatures to find a sweet spot that makes getting the raft detached workable without special tools being required.

This would probably work better with an 0.3 mm extruder head. I've ordered one, but I am not sure yet how much drama will be entailed in getting it to work with Skeinforge.
That exercise got me to wondering how much drama would be involved in getting some of the other Rapman parts printed. Looking at the parts, the extruder seemed to have the most massive pieces. AK47 commented yesterday that the 125x125x10 mm text block that I printed was possibly the largest single piece printed so far. The block is big, but I don't know if it sets any sort

of record. Certainly, Mendel is bound to have pieces requiring more than 59 cm^3 of plastic.
I grabbed the biggest part of the Rapman extruder and did a tracing of it to get an idea of its dimensions. Here is what it looks like.



At 120x174 mm it is going to be a bit bigger than the test block we did yesterday and, once we put 15 mm of raft perimeter around it, it will pretty much occupy the whole of Rapman's print surface.
Addendum:
Bogdan suggested that I up the Infill Overlap Ratio from 0.3 to 0.7 to get a better fill into the teeth of the gear.

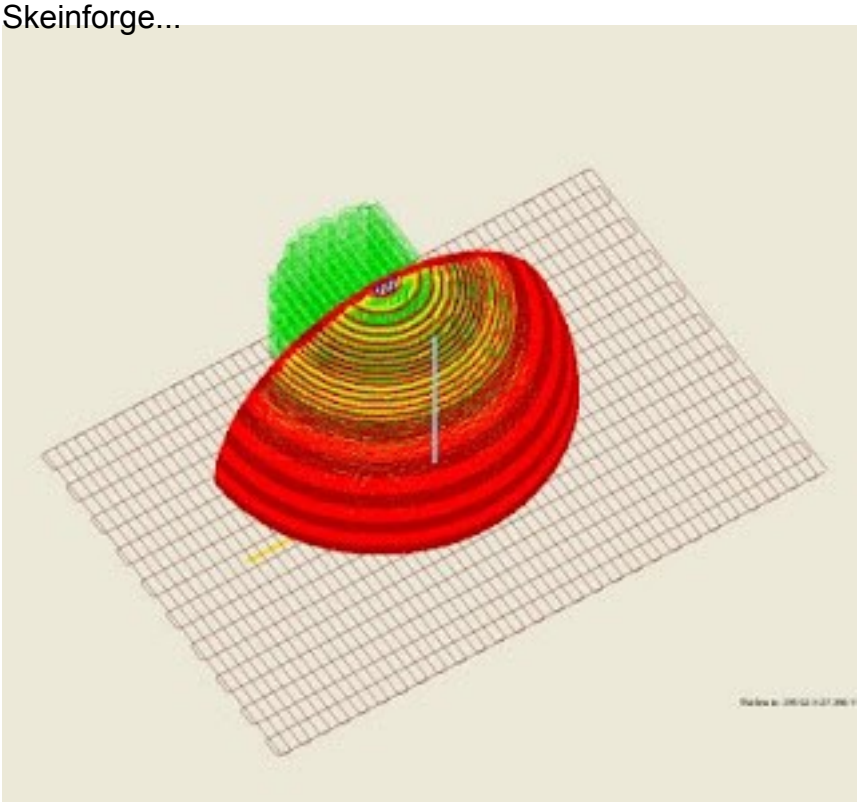
It does help. You can see the pair with the 0.7 infill overlap at the top.

Exploring support material in Skeinforge

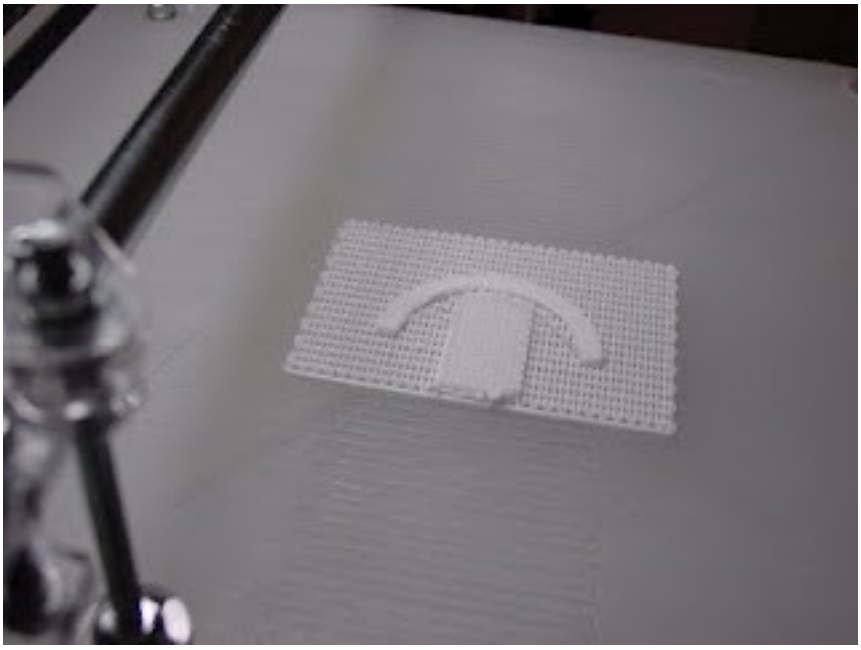
Monday, 16th November 2009 by Forrest Higgs

Skeinforge has the possibility of generating support materials for overhangs for single extruder Reprap 3D printers. I decided to see how it worked. To do that, I created a dome in Art of Illusion and then cut away half of it.

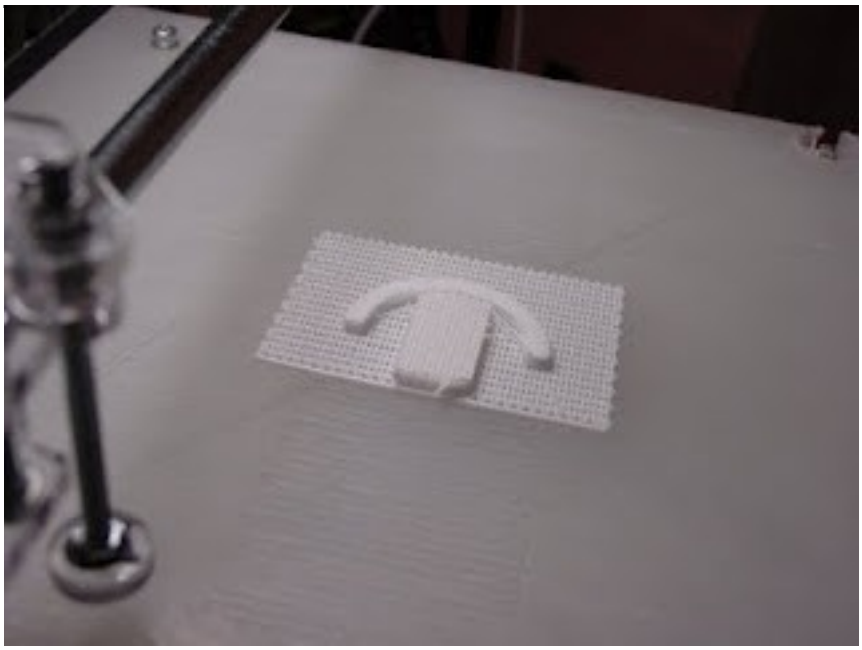
Running it through Skeinforge...



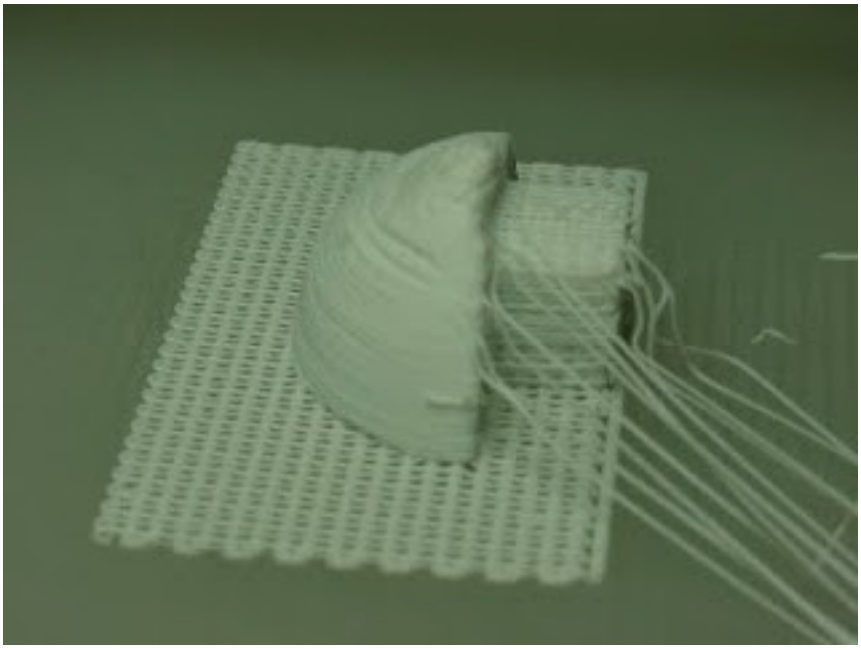
The print began well...



...and progressed..

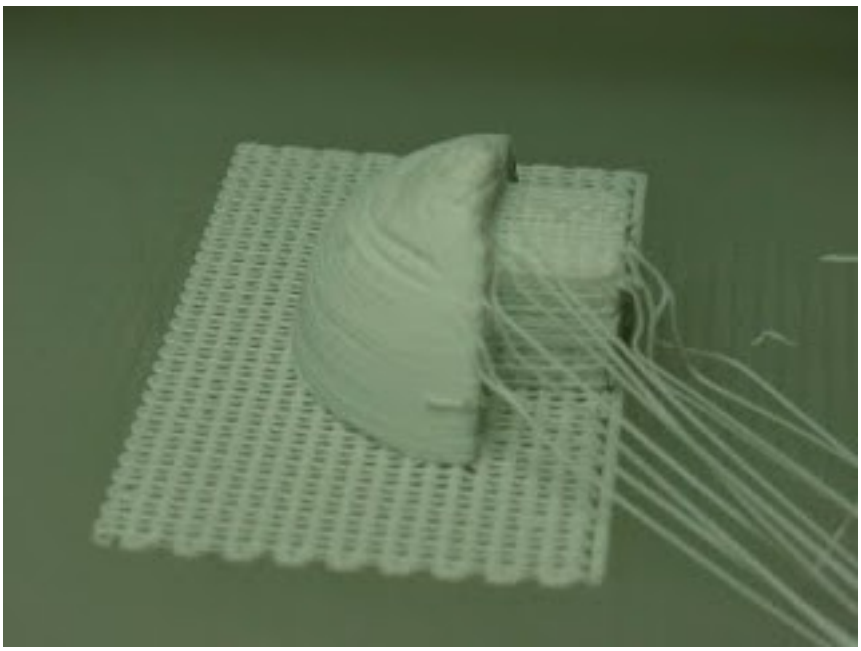


It was obvious that the extruder head was spending more and more time on a progressively small area as the dome print progressed.

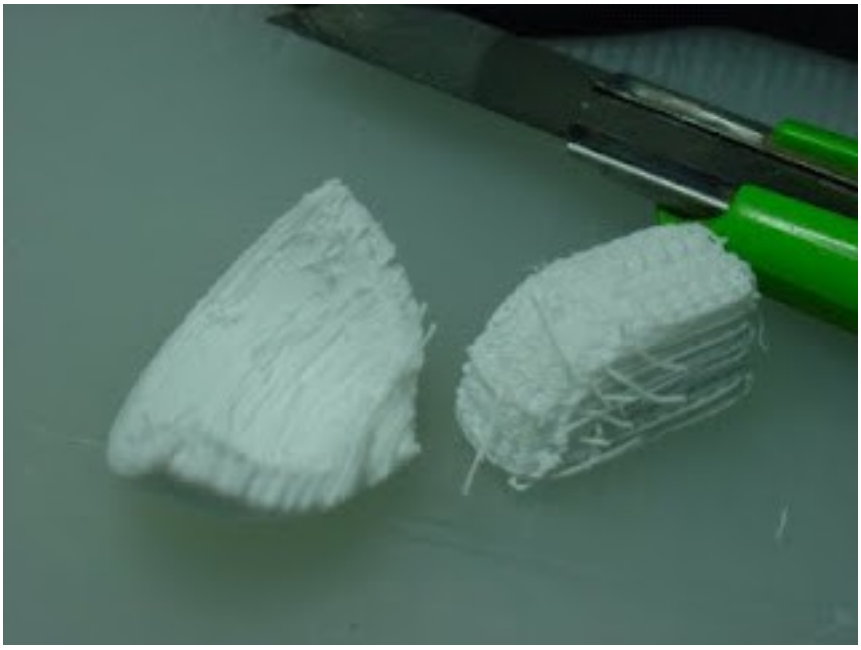


You can see as well that waiting for temperature exchanges to take effect was causing a great deal of extruder dribble. I discovered that you can get a raft off of your polypropylene print board if you print the first layer of the raft at 225 C and the second at 230 C.

If you then drop the temperature of the layer between raft and dome to 205 C it is fairly easy to get the print off of the raft. This had been a problem at higher print temperatures. The dome itself was printed at 230 C and the support material at 205 C. While Skeinforge estimated the print time of this half dome at an hour it didn't estimate the time temperature changes in the extruder would take. Those added another 1-2 hours to the print time.

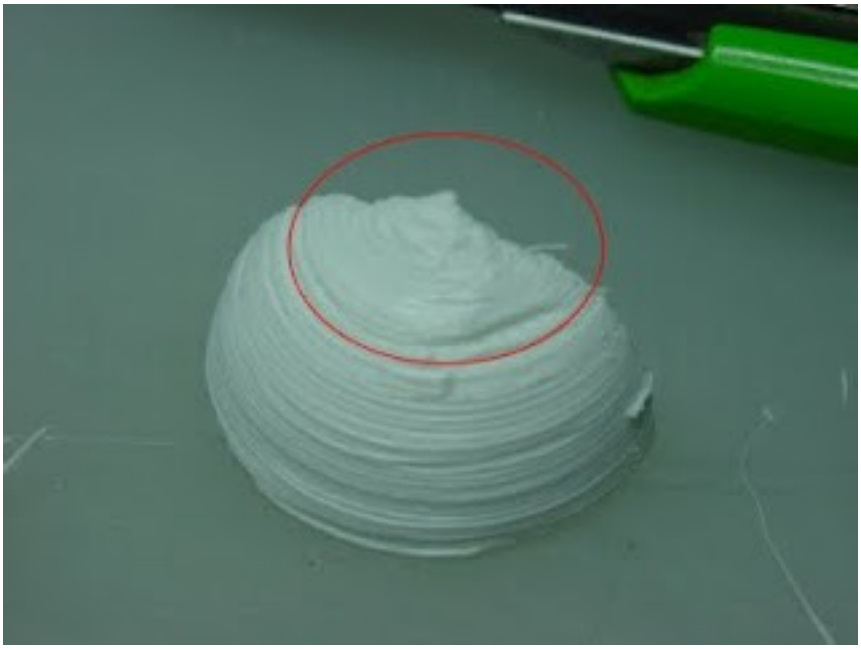


Support material separated easily from the half dome.



Looking closely at the support structure it was well consolidated. That left me to wonder what would have happened had I printed the whole half-dome at 205 C.

I wonder as well if a bit of moving air would have retarded the melting at the top of the dome.



It would seem as well that programming in significant wait times between layers would have given me a finer print.

I have quite a few parameters to twiddle in coming days.

Slicing and dicing with image processing

Sunday, 29th November 2009 by Forrest Higgs

Originally posted in ATechnicalFix.com on August 9th, 2009

In which your narrator rebuilds Tommelise's Slice and Dice app.

Some years ago with Tommelise 1.0, I built a routine to slice STL files and lay out tracks for printing plastic on each slice. It worked reasonably well. After that, I completely rebuilt the app to do milling instead. As well, I developed routines to do milling of printed circuit boards. A few months ago, I resolved to go ahead and develop Tommelise 3.0. This entailed a completely new controller design centered around the I2C bus. That's pretty much done. I hit a dry point in my creativity and spent a few months developing a stripboard design routine which I recently published.

Since then, it has become obvious from the complaints of several other Reprap team members that the open source Art of Illusion 3D modeling package leaves a bit to be desired. I've always had good luck with AoI and have, until recently. been puzzled by the bad press it's got. Nophead (Chris Palmer) in Manchester has been very critical of AoI. I always take Nopheads criticisms VERY seriously in that he is a very thorough person and doesn't say negative things without considerable justice. Nophead and I did a cycle of parallel tests on AoI and discovered that you can run it in what are, for practical purposes, virtually identically configured PC's and get either excellent performance (me) or miserable performance (Nophead). That wants some serious looking into by the people on the Art of Illusion team.

Mind, my experience with AoI hasn't been without it's frustrations, though nothing as bad as what Nophead reports. The major problem I had was with my own slicing routine. If AoI sent over a STL which was not Euler Valid, that is, with all of the triangles making up the surface pointing outwards, my slicing routine couldget a little unstable. Not to put too fine a point on it, that means that I'd get a very bizarrely printed layer.

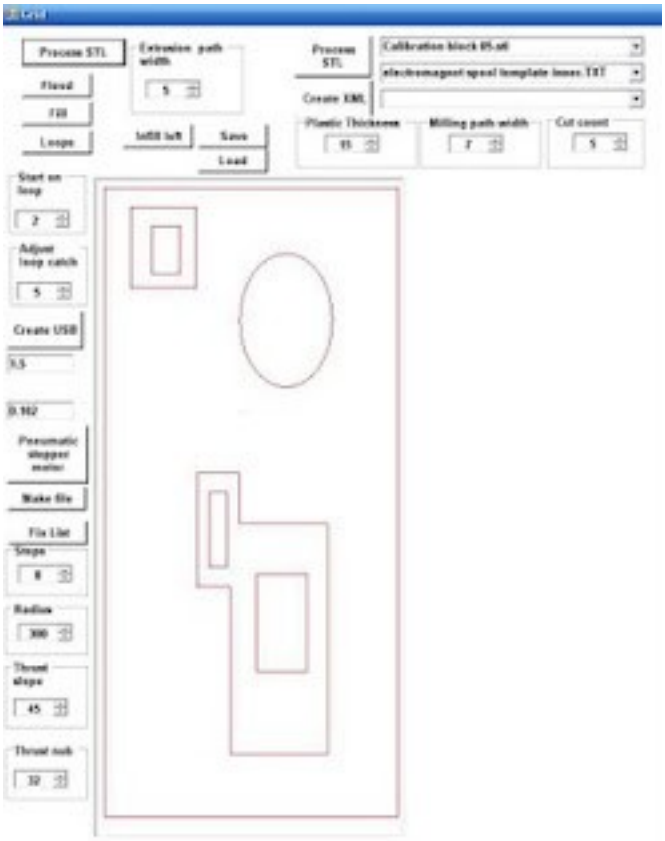
I've liked to boast that I use a grid approach to dealing with slicing and dicing as opposed to the geometric approach that Adrian's august background in 3D graphics lets him do. Grid processing is slower but all the same tends to result in much simpler and easier to maintain, by me anyway, code. The problem historically with my slice and dice routine is that it wasn't quite completely grid oriented, but rather a "neither fish nor fowl nor good red beef" mix of grid and geometric techniques. I decided to fix that.

Doing the Slice

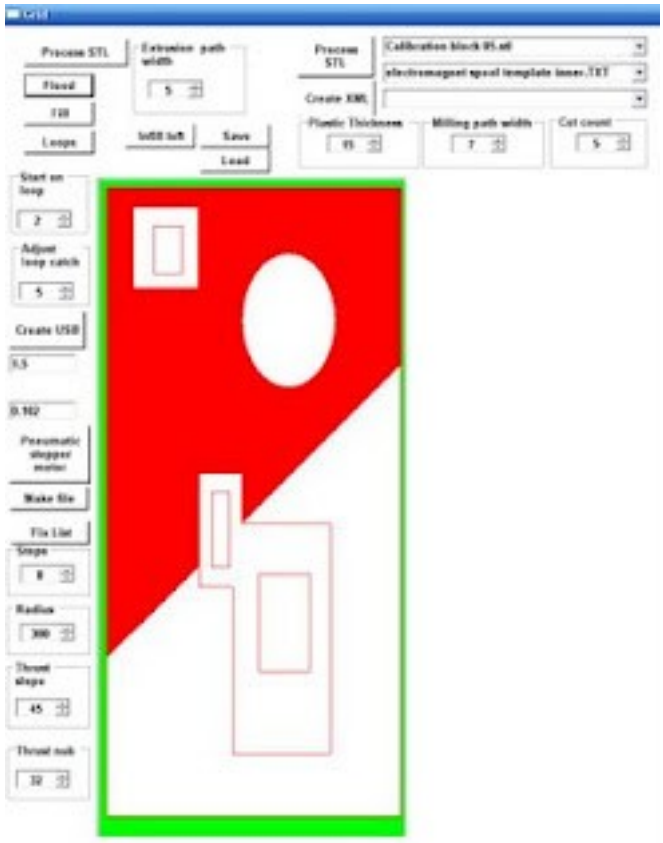
Using a grid is very simple. Basically, I just create a picture box big enough to hold the object's xy dimensions and set the pixel size to the step size of the xy steppers. I then take a slice of the STL and map it onto the picture.

The lines making up the slice are automatically represented as pixels on the picturebox, so which way a vector is pointing and other minutae that give geometrical approaches headaches simply aren't there.

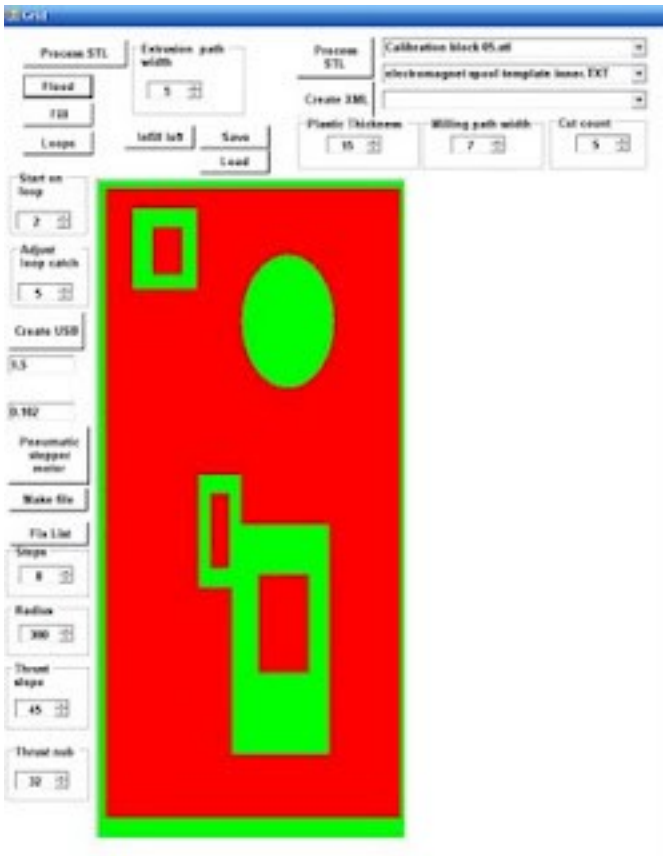
You get something that looks like this…



Once you have your slice boundaries laid down you have to fill in the parts of the slice which are to be printed with plastic. Those areas are shown in red while empty places are shown in green. The routine that does this is very, very short and very robust. You can see the process underway here…



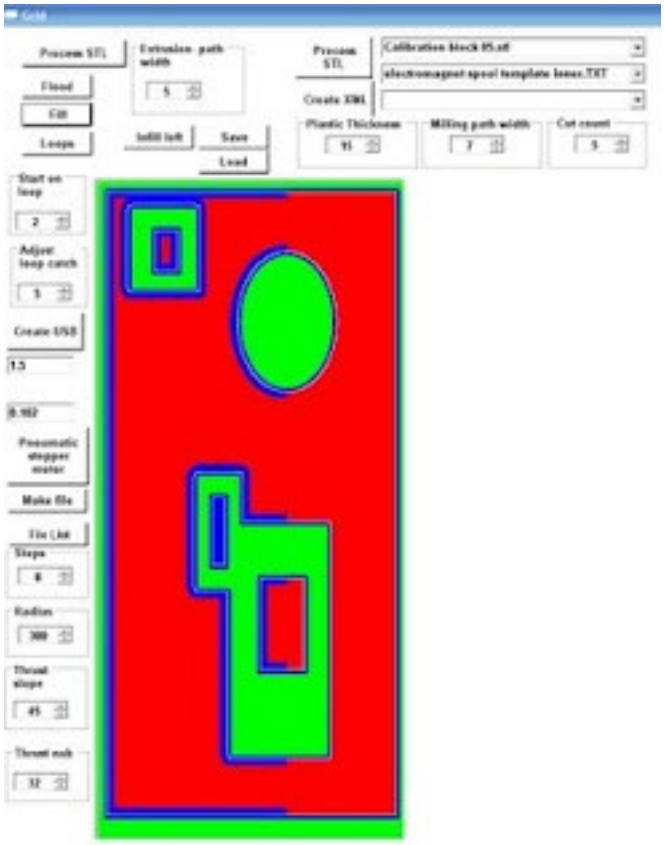
…and complete here…



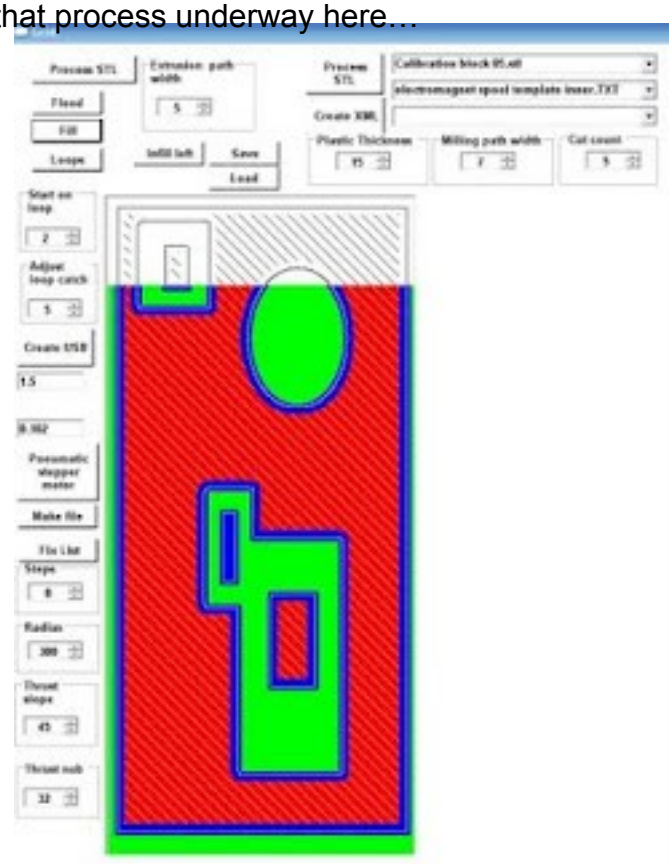
Once you have your slice painted onto your picture box you need to figure out where the roads your extruder is going to have to travel. This is accomplished by…

- Painting a border around the inside of the perimeter of the slice that is one-half of the width of your extrusion road less one step resolution distance {done in blue}
- Paint a white midline for the extrusion road on the inside of that blue stripe
- Paint another blue stripe next to that which is one-half of the width of your extrusion road less one step resolution distance plus the distance to the ends of your infill roads.
- Paint your infill road midlines

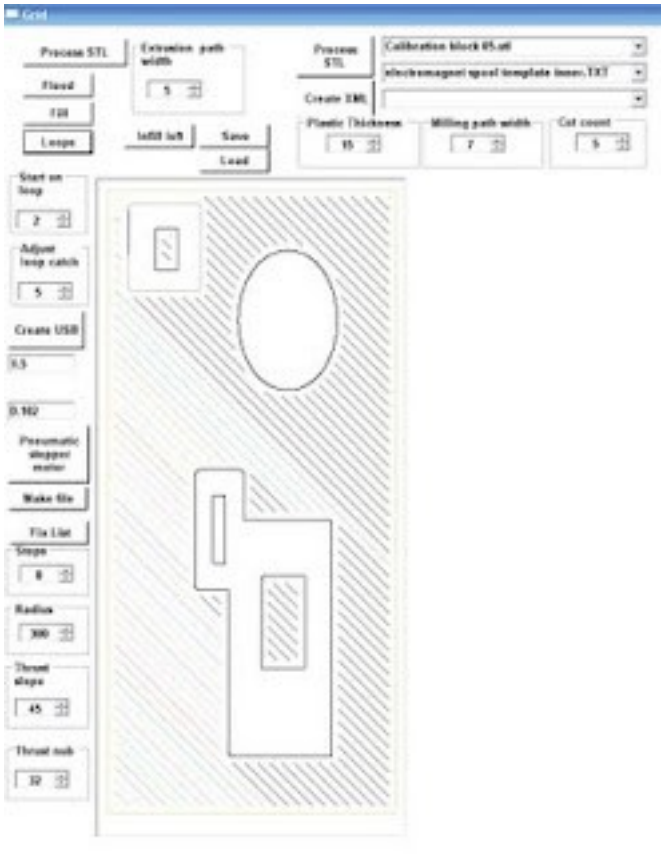
You can see this process underway here…



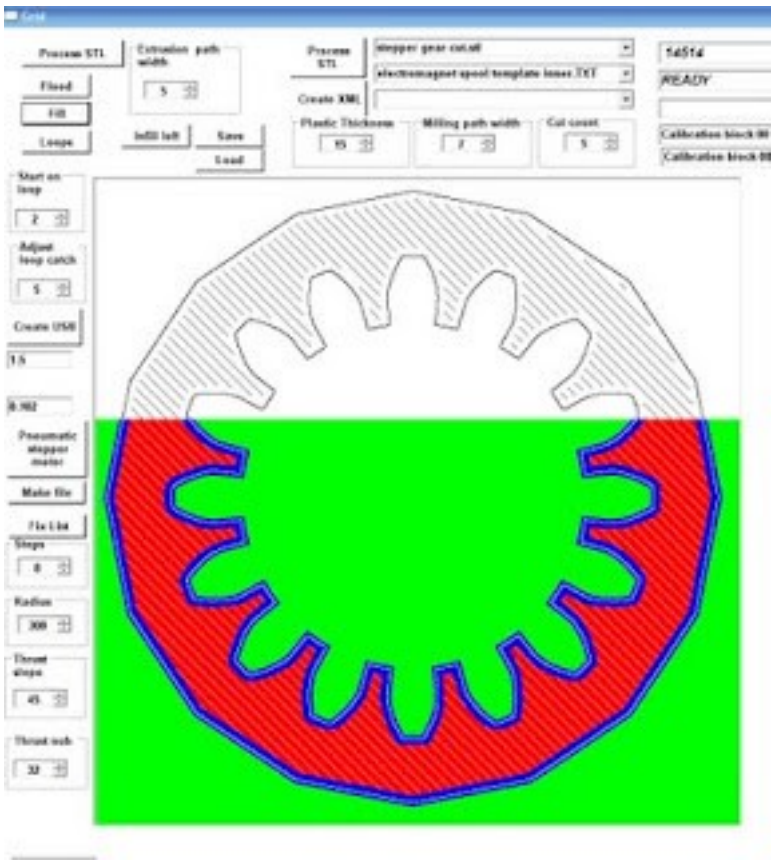
Once that process is done, you simply wash the colours away leaving only the pixels delineating the printroads. You see that process underway here…



Once that is done, you collect the roads. I paint each road a different colour so that I can make sure that there are no breaks. That process is underway here.



Ultimately, I am going to do this in two steps, the first to collect the perimeter roads and the second to collect the infill roads. That is only a bookkeeping issue.
The only part of the code I have not yet written goes through and arranges a print order which minimizes the distance that the print head must move between infill roads so that the Tommelise 3.0 print head does not spend a lot of time moving around while not printing a layer. I've written that sort of code for the old slice and dice app, so I don't see it as being any big deal.
Years ago in China, I had the unique opportunity to spend an evening with Benoit Mandelbrot, who brought the fractal into the general public awareness. He related a story that, I think, has some important things to say for Reprap design. When he originally went to work for IBM he was given the task of working with a particular device that was very sensitive to electrical noise. Any time repairmen in the building used an electrical drill the device malfunctioned. His supervisors at IBM wanted Benoit to track down and eliminate the sources of the noise. Benoit, being the sort of man he is, disagreed and insisted that there would always be noise caused by one thing or another and it would far better to figure out how to somehow shield the device from such noise or somehow make the circuitry itself in the device noise tolerant.
That is what I am trying to achieve with this new incarnation of Slice and Dice. It seems to me that we are going to be confronted with how to process bad STL files for a long time, given the state of the art with open source 3D modeling packages' boolean ops routines. The flooding process that I use in Slice and Dice shows you very quickly where such faults manifest in a particular slice.
 Rather than having to fall back on fooling with code, a Slice and Dice user can simply use something like Windows Paint or Photoship to fix the the and image and then go on from there.
Just so you are reassured that this is only robust for an over simple case…



Keeping that in mind, I've arranged to save the developing slices as images. This requires substantial hard disk space, but disk space is a trivial expense these days. Doing that gives me a number of very powerful capabilities.
I can review a print job visually, slice by slice and assure myself that there are no problems therein.
Should there be problems that can't easily be cured by working with the 3D modeling app {AoI, for instance}, I can easily just pull the marred images into something like Photoshop and fix them, then reprocess the road building.
I haven't got a full tool-path from AoI to Tommelise 3.0 yet, but it is looking very strong. It would be a very small matter to filter myslice and dice output into G-Code and use it in mainstream Reprap machines. I want to convince myself that I've really got something worthwhile before I suggest that sort of thing, though.

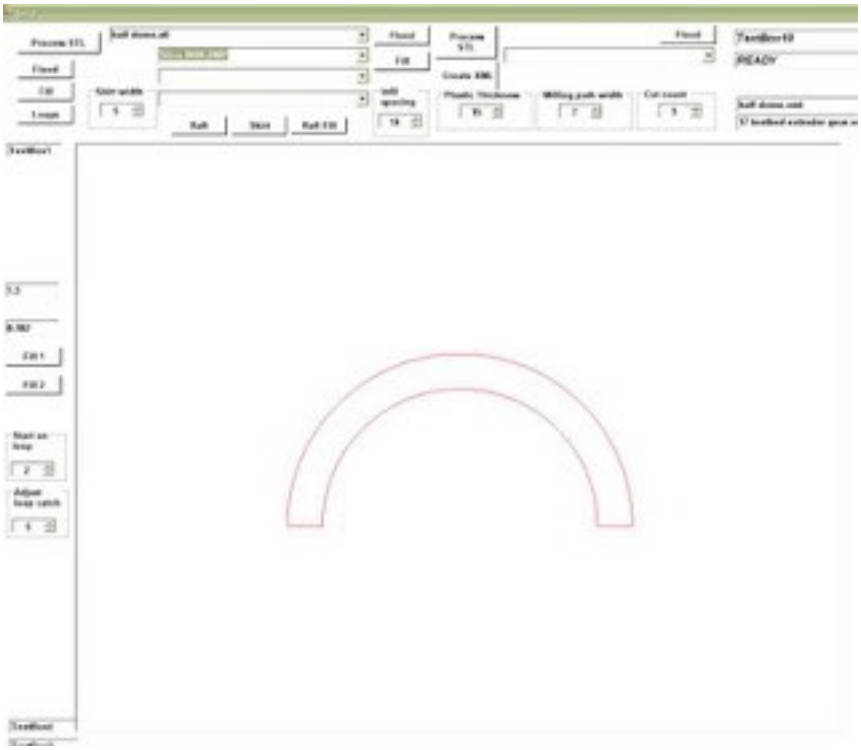
Patching slices and building rafts

Sunday, 29th November 2009 by Forrest Higgs

Slice and Dice works a bit differently than Skeinforge in places. To demonstrate a few of the differences, let me use the half-dome that I processed in Skeinforge a few days ago as a point of departure. This is what it looked like in AoI.

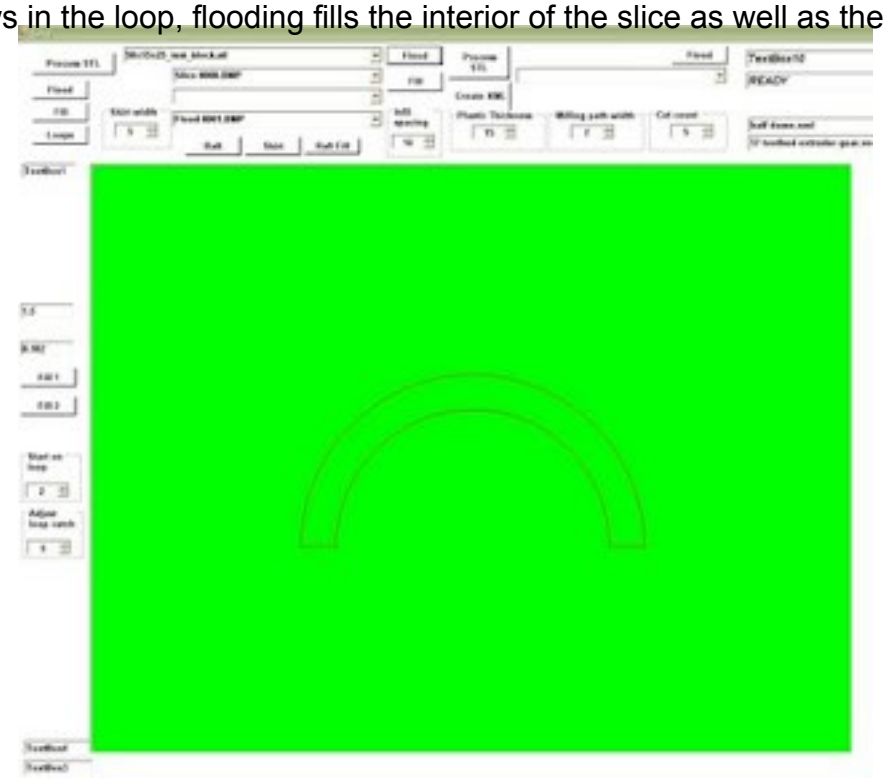
Fixing flawed slices

Slice and Dice cuts the half-dome's STL file in much the same way that Skeinforge does. Here is an example slice.

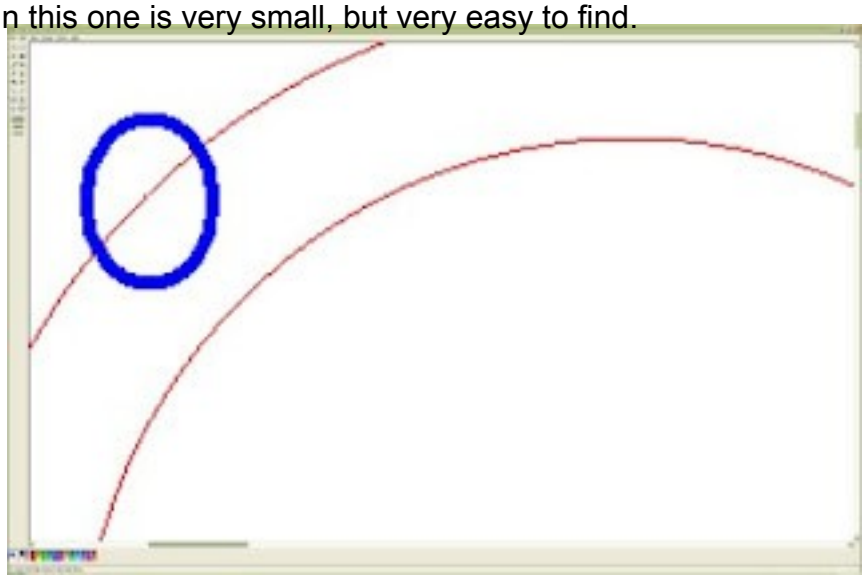


When I do image processing, however, having a tight loop around the sliced area is mandatory. If

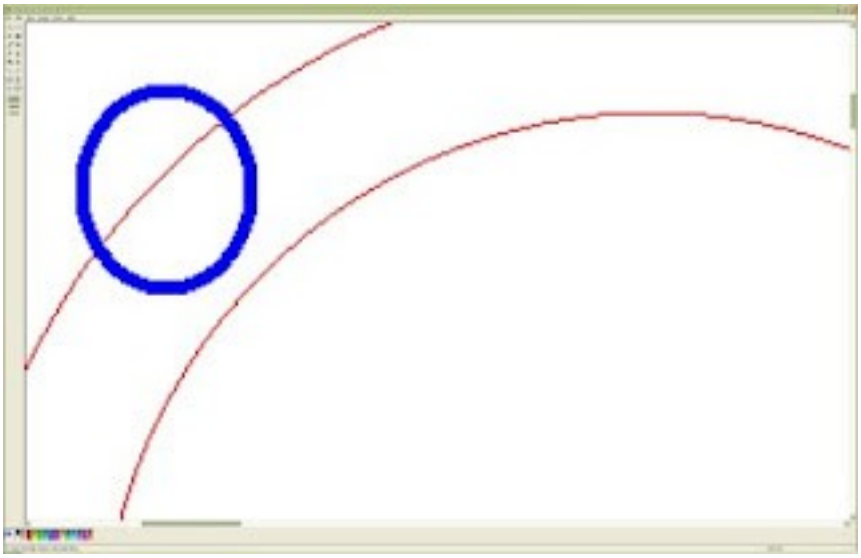
there are any flaws in the loop, flooding fills the interior of the slice as well as the exterior.



Where Slice and Dice starts getting different to Skeinforge is that when this sort of thing happens, you just pull the relevant image into something like Windows Paint and see if you can find the hole(s). The hole in this one is very small, but very easy to find.



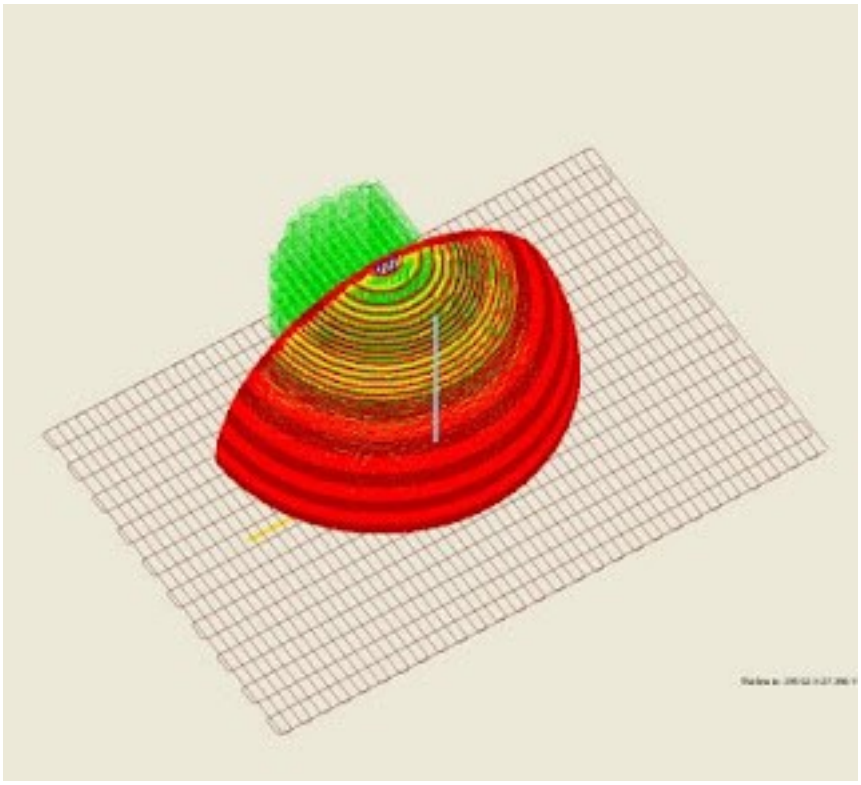
It was a matter of a moment to plug the leak using Paint.



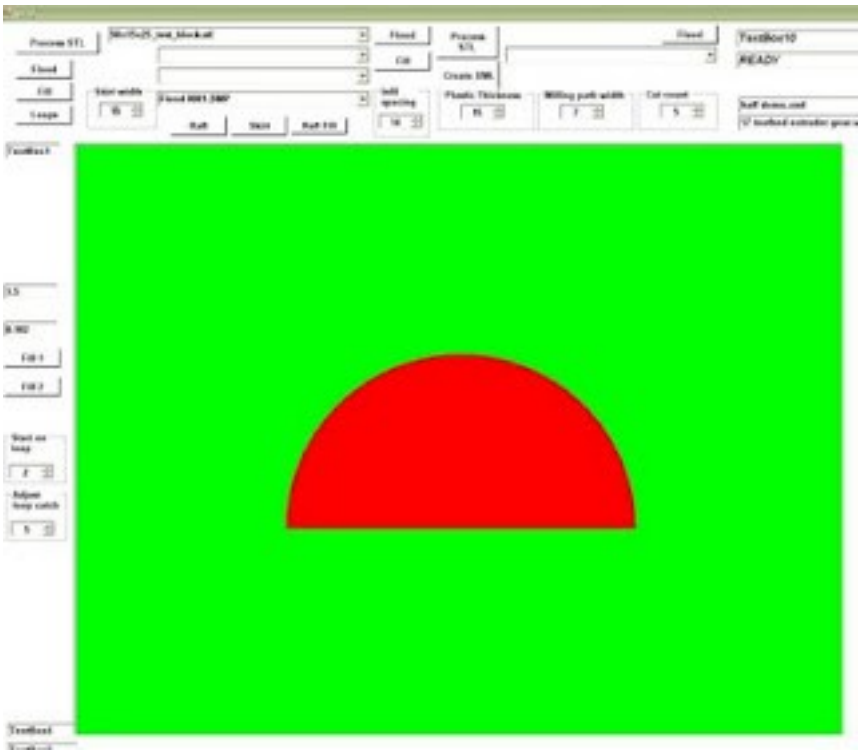
I then just reflood that slice and it now works perfectly.

Building the raft

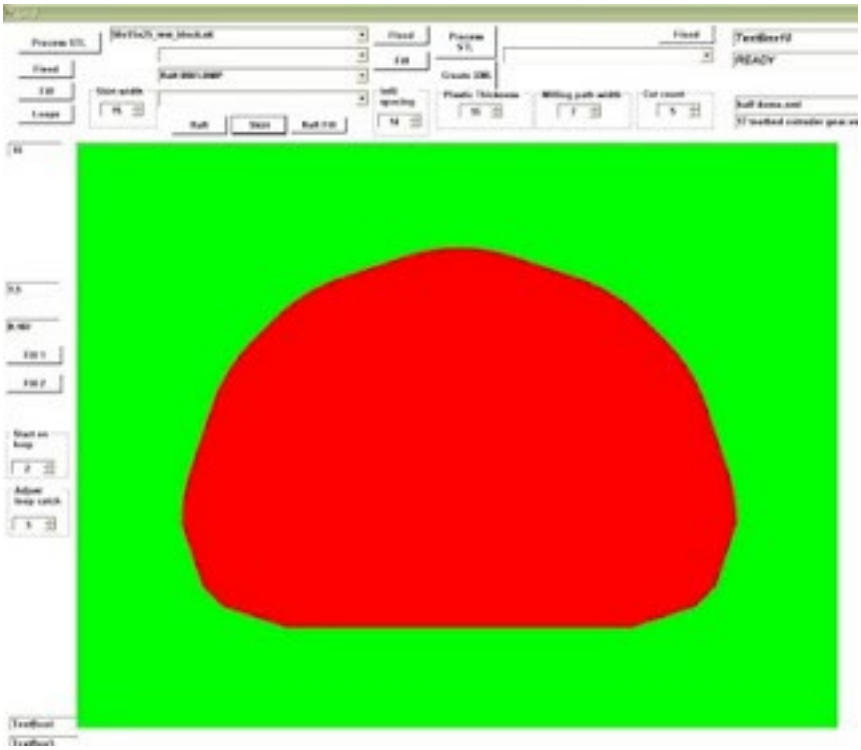
I had run into trouble with the Skeinforge raft. If you have a long object and rotate it 45 degrees to take advantage of the diagonal Skeinforge still designs a raft using the largest and smallest x and y coordinates. That wastes a lot of filament and print time. As well, when you have a non-rectilinear object such as my half-dome you still get more raft than you really need.



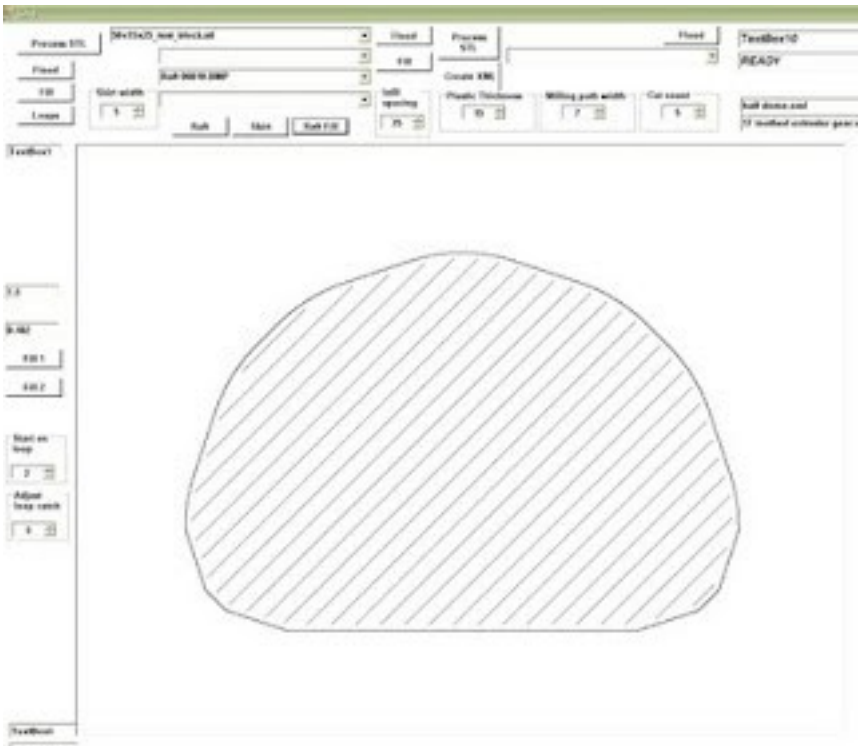
I had two problems to address. First, I had to have a footprint which included overhangs. I achieved this by the simple expedient of just summing the flooded slices which yielded a composite.



I then created the skirt by wrapping half millimeter bands around the footprint till I had the skirt I wanted {15 mm}.



I won't illustrate, visually, for generating the print roads for the extruder given the area to be printed. You can see that in a previous blog entry. The resulting roads, however, are somewhat different than what Skeinforge generates. Slice and Dice treats the raft as just another slice and uses a perimeter and infill arrangement. Here you see the first layer.



I am trying out the notion that having perimeter support for the raft might let it need a smaller footprint than Skeinforge's openwork raft arrangement.

First Rapman raft with Slice and Dice

Monday, 30th November 2009 by Forrest Higgs

I got things hot-wired together to produce a gcode file for the first layer of the raft.

It looks a little rough, mostly because I am not that comfortable with gcode yet. It's a fitted raft as opposed to Skeinforge's largest dimensions plus two times the skirt width method for creating them, however.
It's a start.
Oh yeah, still no resets.

Can we print belts?

Saturday, 5th December 2009 by Forrest Higgs

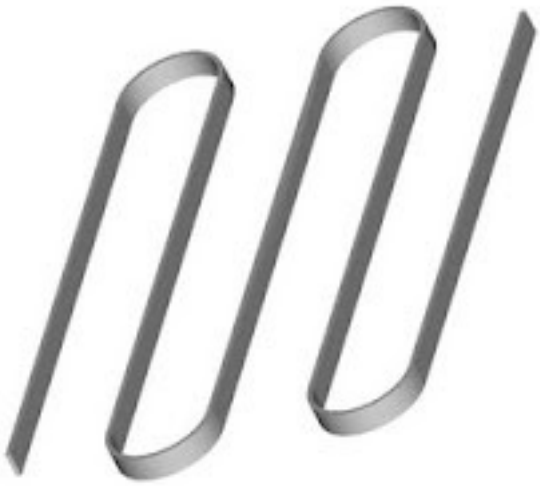
Yes, if we step back and look at what a belt does instead of simply trying to replicate existing ones.

Belt driven Reprap machines have always bothered me. They're fast and now that the price of NEMA steppers is getting reasonable, they're even becoming economical. It bothers me, however, that the belts and sprockets can't be printed very easily.

Last night I was cleaning up a corner block that I'd printed and was cleaning out a bearing seat with my Dremel tool fitted with a cylindrical abrasive head. These are a nice little fitting that lets you slip what is a cylindrical piece of sandpaper onto your Dremel. They are cheap and very effective.

I was looking at it when I realised that the toothed belts are toothed to prevent slippage and that there are other ways to prevent slippage than teeth. An abrasive cylinder, for instance. With an abrasive sprocket you wouldn't need teeth and the belt wouldn't need teeth either.

It was a matter of a few moments to design a 600 belt in Art of Illusion that would easily fit on my Rapman print stage.



A few minutes more and I had it run through Skeinforge and printed it in HDPE.

Cleaning the belt took a few minutes, not no big deal.

The resulting belt is very flexible and quite strong.

I've got a pile of NEMA 17s. I might just print myself out a Mendel now. Mind, I will be using my OWN electronics.

Killing flocks of birds with one stone

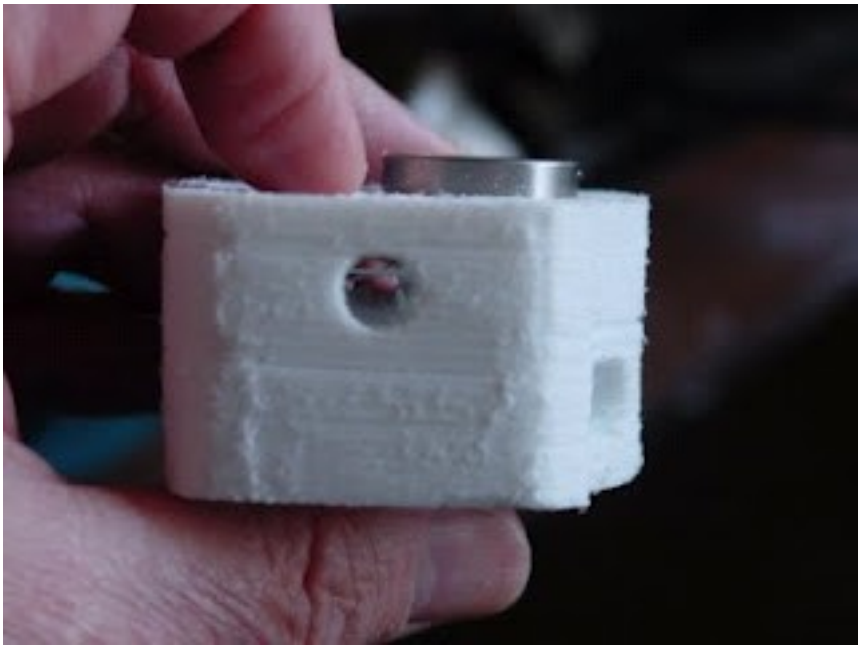
Saturday, 5th December 2009 by Forrest Higgs

I've got the curling on HDPE prints down to something quite manageable. Last night I decided that I could dress down the remainder if I only had a belt sander. A little while ago, I went into town and bought a nice little Makita belt sander. It wasn't the cheapest, but it would lay, belt-side-up, quite firmly, which is what I needed.
It took about 15 seconds to grind the raft off of Bogdan's corner block for Rapman and dress off a few other rough places.

I bought a new little drum sander for my Dremel {~$4.50} and cleaned up the seating for the z-axis bearing.

It also did a nice job cleaning up the hole for the z-axis threaded rod.

One added benefit which might be the Rapman or might be the HDPE or might be the Skeinforge settings is the roundness of the horizontal holes in the print.



None of this teardrop nonsense.

I'm beginning to think that HDPE is a very serious contender for printing Mendel parts. It's dirt cheap, readily available, strong, doesn't make nasty fumes AND now we know how to work with it. I expect that polypropylene, which is cheaper still is going to be just about as good.

Now here is the serendipitous Christmas present. That Makita belt sander grinds HDPE a treat and puts it in a little bag, or a big one if you want to sew one.

Guess what? The problem of grinding plastic so that it can be recycled into filament has just gone

away. Virtually any kind of extruder can eat plastic powder. It takes a heftier one to use 3 mm pellets or shreds of a similar size.

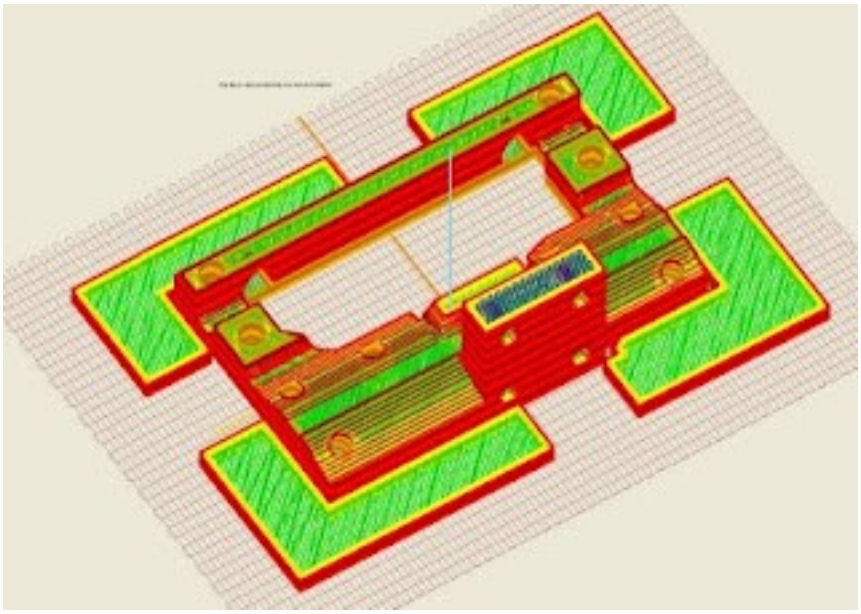
Build yourself a slope-sided hopper on top of that Makita and your grinding problems are solved. When you wear out your belt, go down to your hardware store and buy another. The Makita uses a 3" x 18" belt and it costs a bit over a dollar. No difficult-to-sharpen, never mind dangerous, macerating blades.

God Jul, everybody! :-D

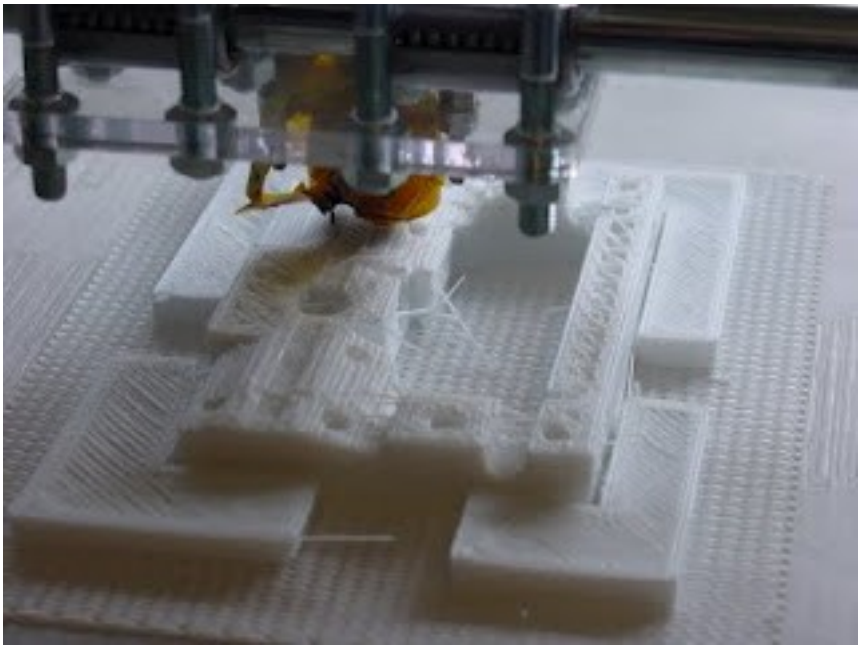
Printing the big bits of Mendel

Monday, 7th December 2009 by Forrest Higgs

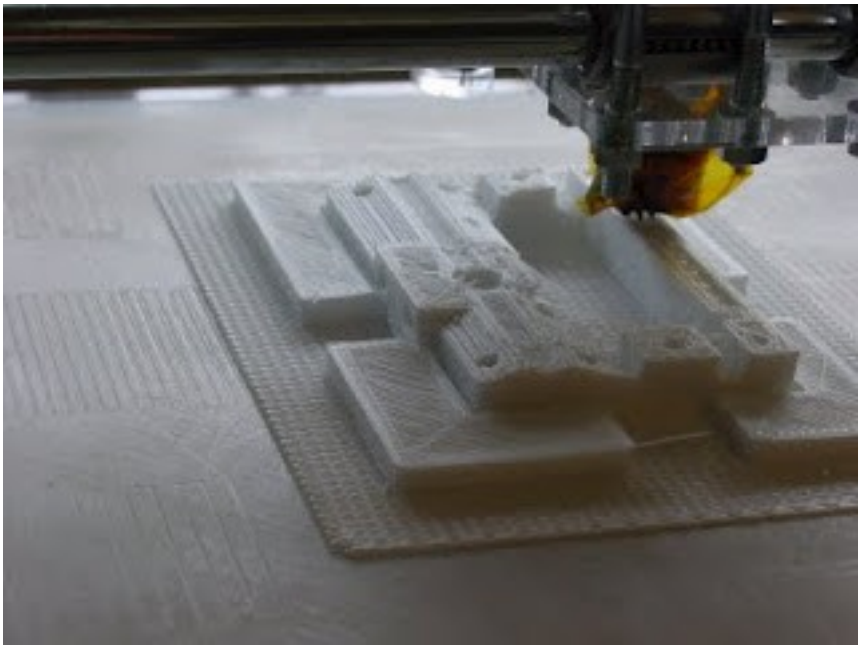
I've been hearing that people are struggling to print the largest parts of Mendel in ABS, so I've decided to see if it can be managed in HDPE using variations on my "no-sweat" method.
I tried Sunday with a full apron around the "x-carriage-lower_1off " part. The print failed for several reasons, the most important of which was that the apron in shrinking mechanically connected with the print in the y direction As well, having a full apron around such a big object added to the contracting force put on the raft by the printed object. That force caused the raft to break free from the polypropylene print surface.
This morning I broke the apron up into four parts, not unlike what I was doing when I was doing guying and flanges a few weeks ago. You can see what I've done with a screen grab of Skeinfold's Viewer.



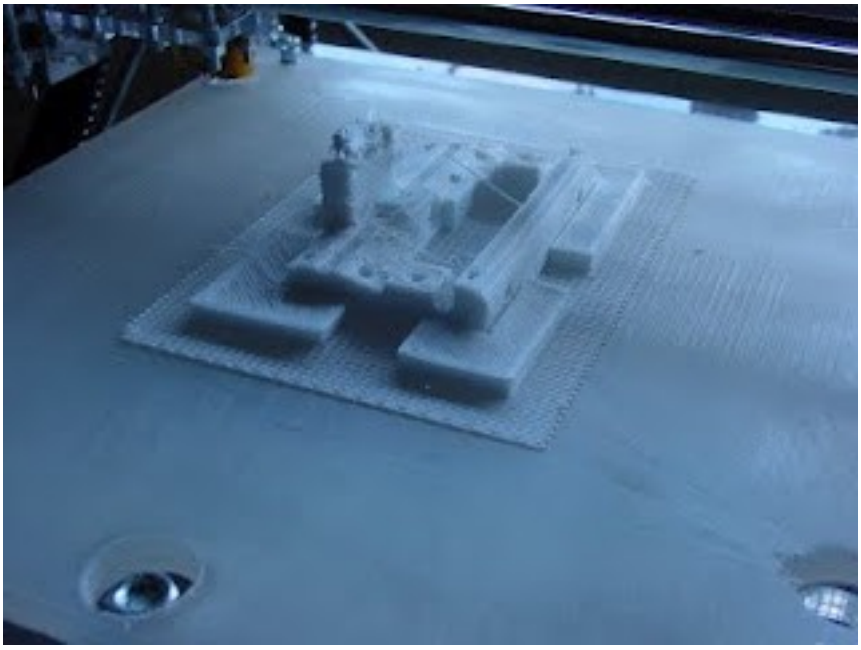
Just some interim pictures.



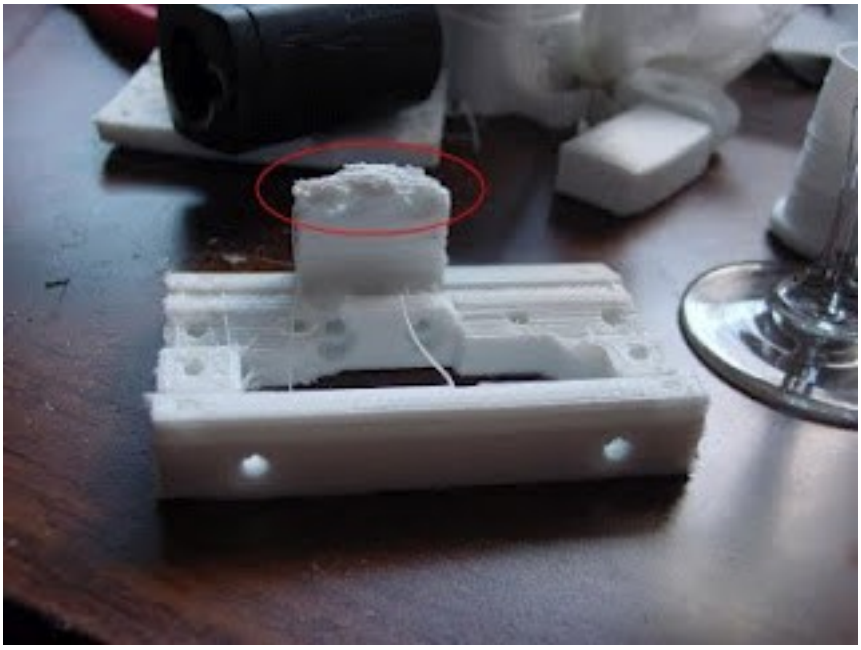
Breaking up the apron so that I didn't have long stretches of apron adding to the stress that the print was putting on the raft/polypropylene interface seems to be helping.



I've got way past where I was when the print failed last night and so far I haven't heard a single one of those little crackly noises that signal that the raft is beginning to pull away from the print surface.
Okay, I stopped the print. We have a qualified success here. I could see a few things going wrong so I stopped the print. Here is what it looked like in the printer.

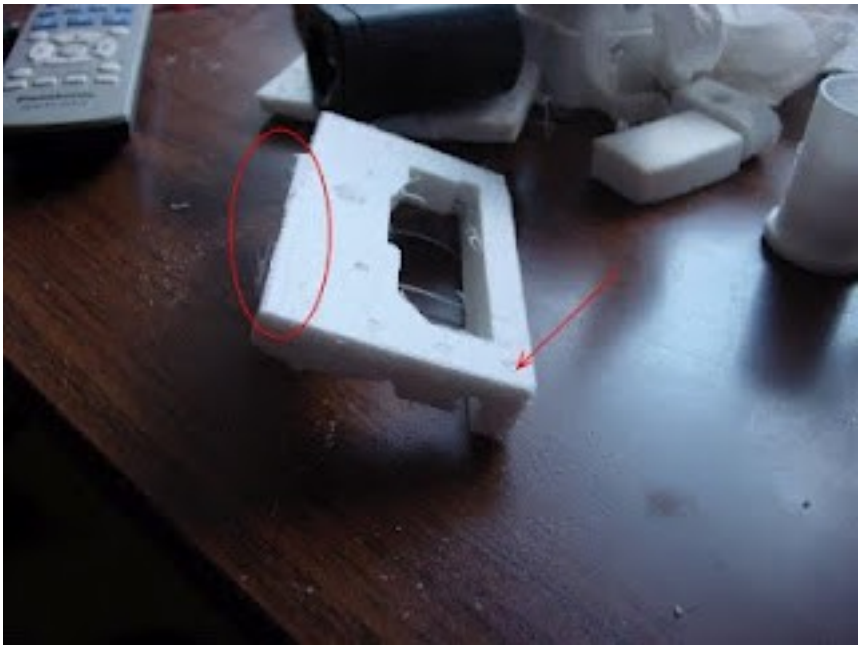


The major problem, as you can clearly see, is that the final little flange on the left was melting down.



I didn't set up Skeinforge for a cooling cycle. I am still getting used to how to do that, so I am going to just print the flange and test settings till I get something satisfactory.
The major problem I had was that when I moved the apron segments away from one end of the print I got them a little too far away.

The lift on this corner was 1-2 mm on the near side {circled}and ~1 mm on the other. Looking at the other end.



You can see that the left side {circled} lifted about 0.5 mm while the right side was perfectly flat {arrow}. I got a little over-ambitious about moving the apron segments away from the print. I can fix that with zero trouble.

This is going to work. It's down to a matter of just tweaking the apron position a bit and sorting out some Skeinforge settings.

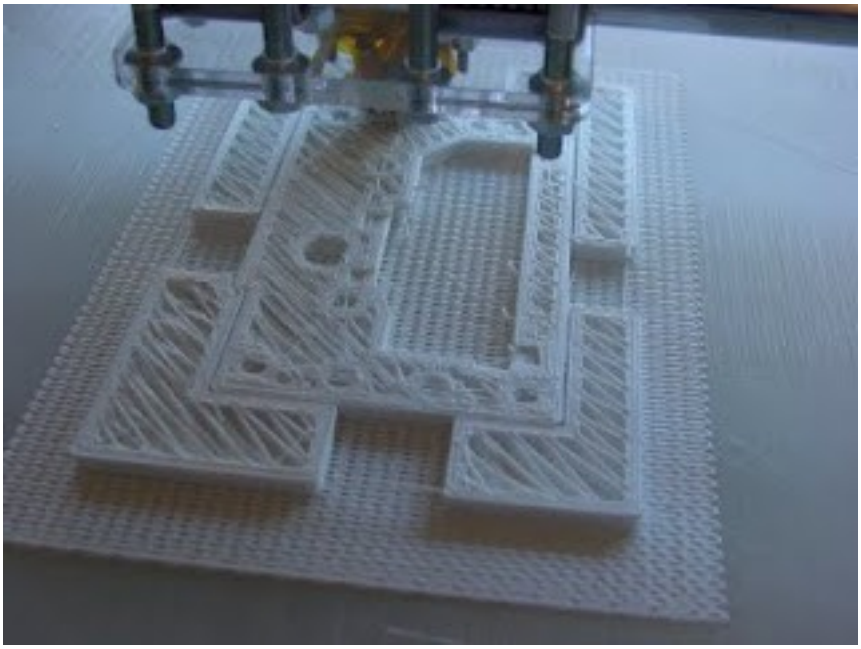
I expect that the combined STL for this can be used with good effect in ABS, since ABS is reputed

to warp much less than HDPE.

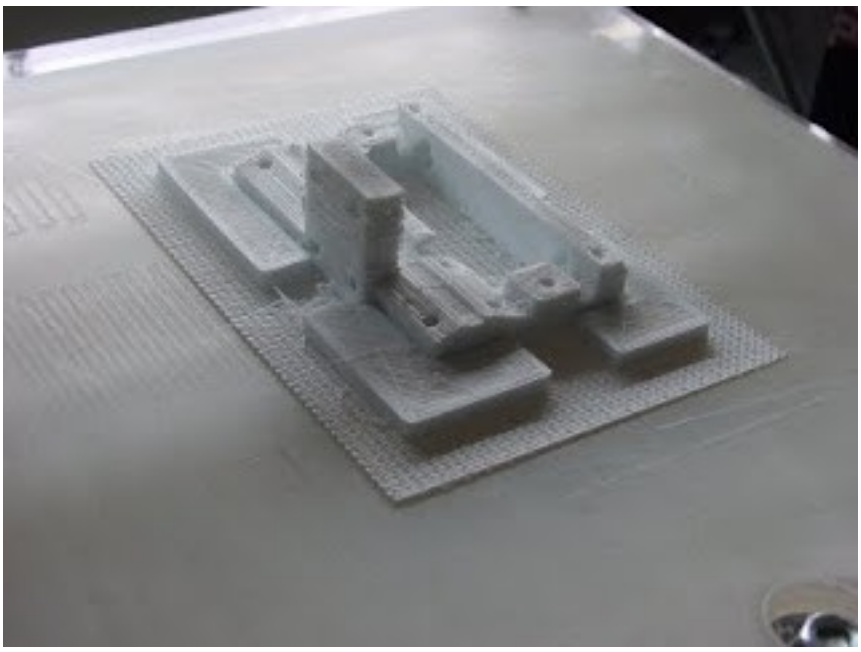
Got it!

Tuesday, 8th December 2009 by Forrest Higgs

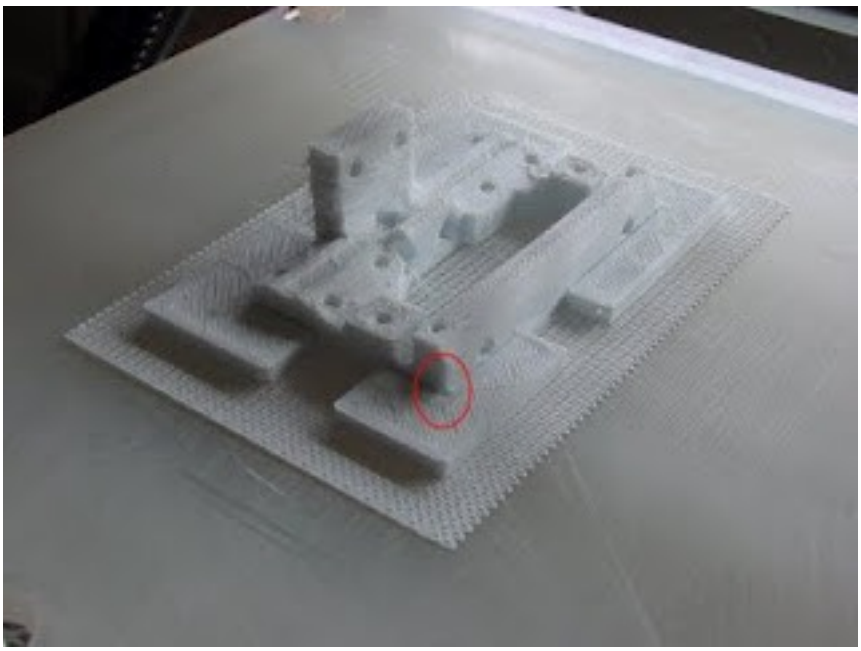
I'd heard that people printing Mendel were having trouble printing the largest pieces of Mendel in anything but PLA. Since PLA filament is hard to acquire in the US at this point I thought it would be nice if those hard-to-print parts could be printed in some other plastic. Since I am working with HDPE right now, I decided to have a go with HDPE.
The trial part I've begun with is x-carriage-lower_1off. I've discovered that a segmented apron positioned 0.5 mm away from the perimeter of the part does the job of successfully suppressing warping. Here you can see the spacing between apron and part.



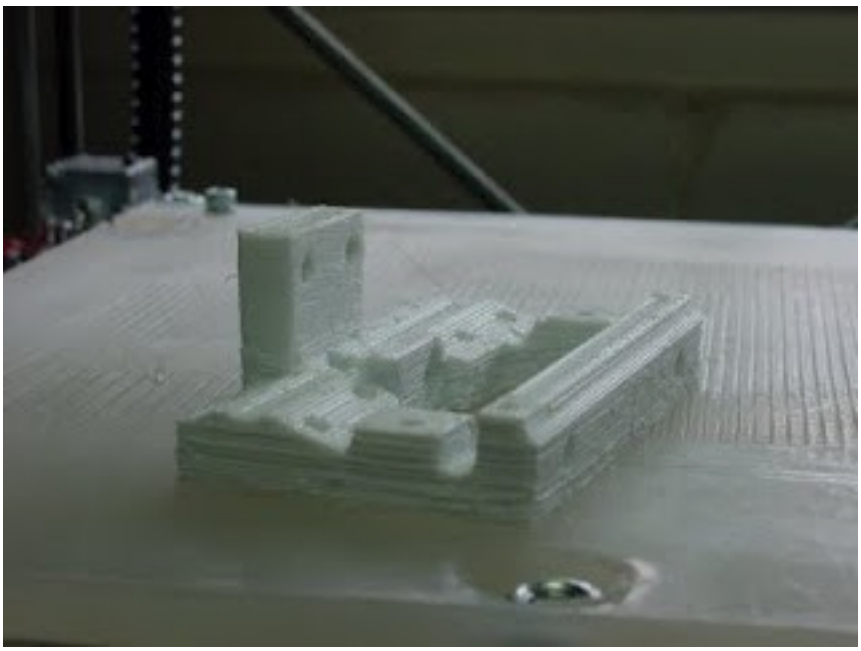
Here is the completed print.

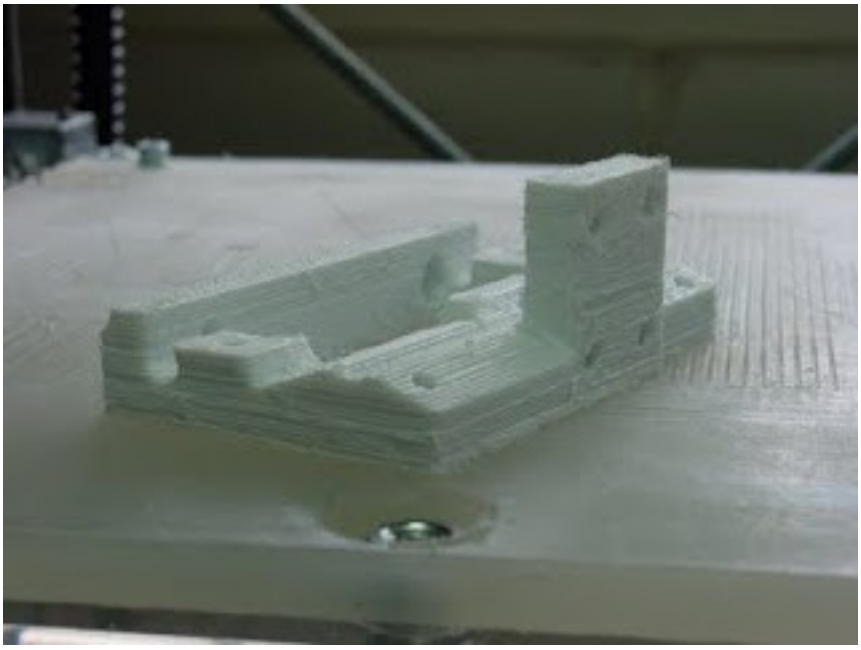


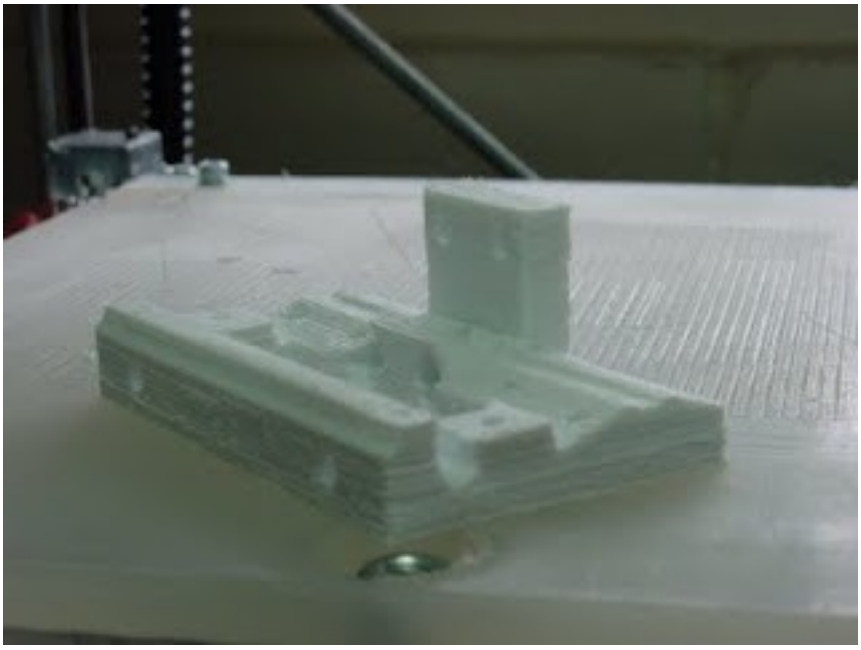
I've cleaned off a bit of the strings. I understand that the newest Rapman firmware release pretty much gets rid of strings. I haven't taken time to install it yet, though. There was only one instance of corner curling.



I've circled the corner in red. The curl amounted to about 1 mm. Belt sanding leaves you with a flat bottom to your part. The fill was raft material. I got the part out of the the apron and off the raft with my belt sander.





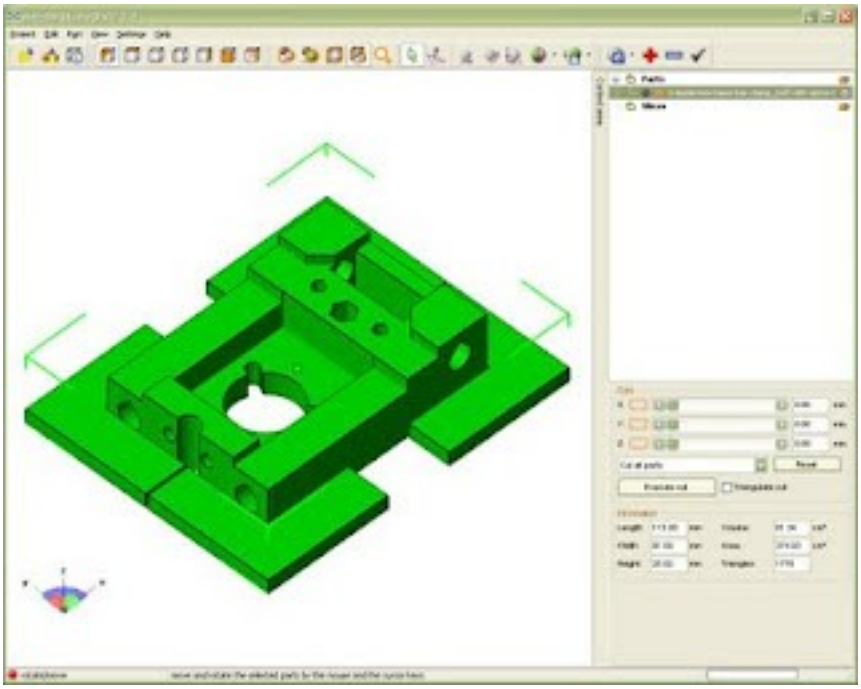


If you want to print this on your on machine you can download the STL for the part combined with the segmented apron here. I made no changes to the part save adding the segmented apron.
I had to rotate the original part to work on my Rapman 3.0. You many have to rotate it again to get it to work on other Reprap machines. I would appreciate hearing if this approach works for ABS. I have ABS here and could try it, but I want to see if I can apply the same technique to do what I understand to be the other big Mendel part, viz, z-leadscrew-base-bar-clamp_2off.

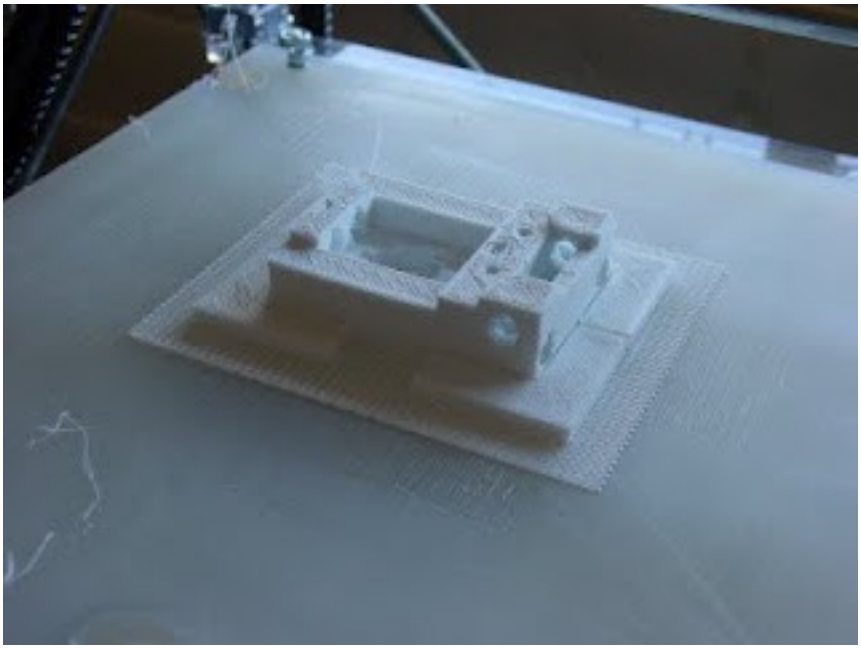
z-leadscrew-base-bar-clamp_2off

Wednesday, 9th December 2009 by Forrest Higgs

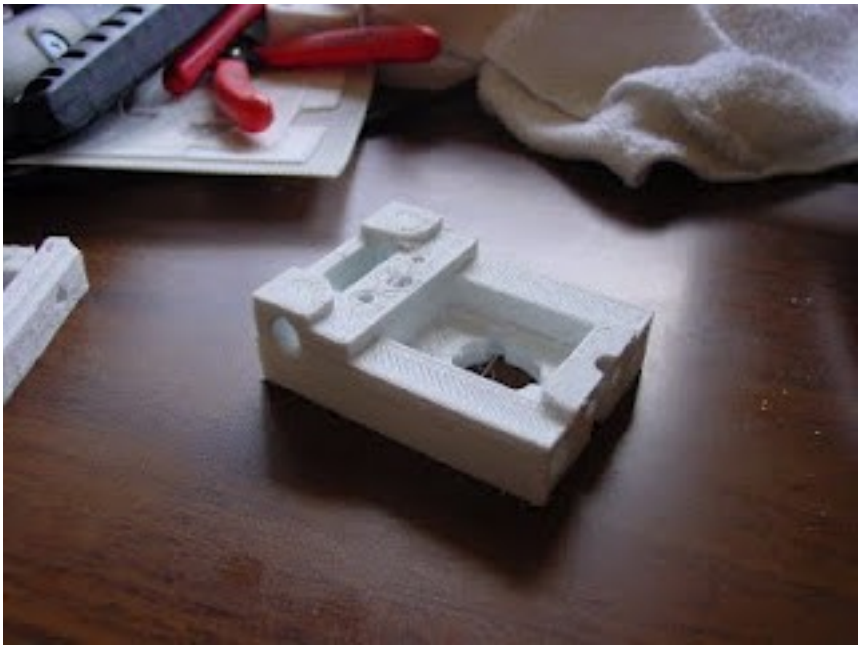
I treated Mendel part z-leadscrew-base-bar-clamp_2off in the same way, viz, a segmented apron. You can see what this looks like in Netfabb.



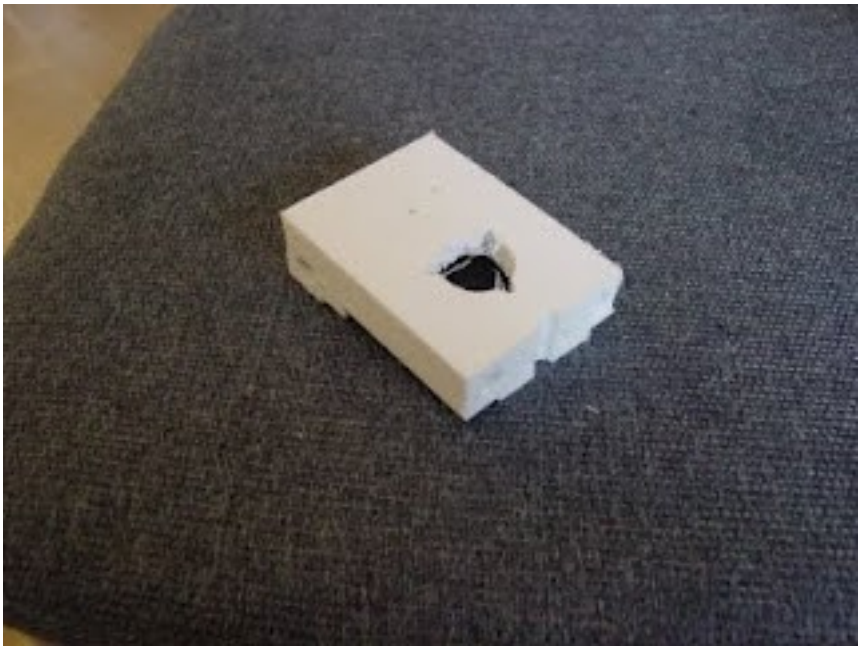
As printed, the part looked like this.



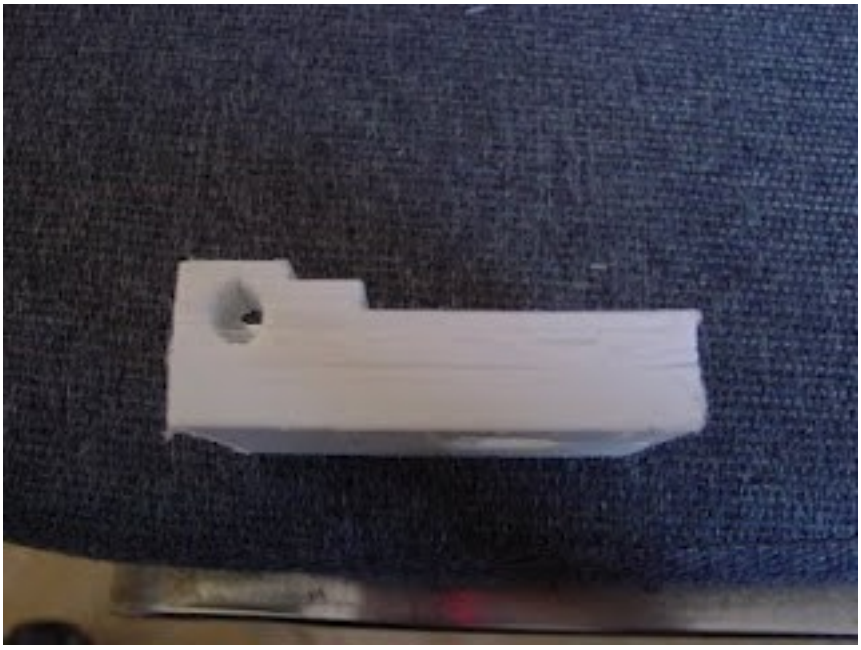
Cleaned, it looked like this.



There was a very mild bit of curl on the right-hand edge {~0.5 mm}.



There was a small problem in that the near side of the part tried to delaminate. The part has two horizontal holes that run the length of the part.



I think, perhaps, that the settings that I am using with Skeinforge didn't get me a proper fill there. More to the point, as I learned eons ago as an architect, it's much easier to design a part than it is to design a part that you can build. For HDPE, it seems to me that if this part was maybe 4-5 mm wider leaving the longitudinal holes in the same place, this delamination wouldn't have occurred. Mind, it may not be a problem with PLA and ABS.

Other than that, no problems.

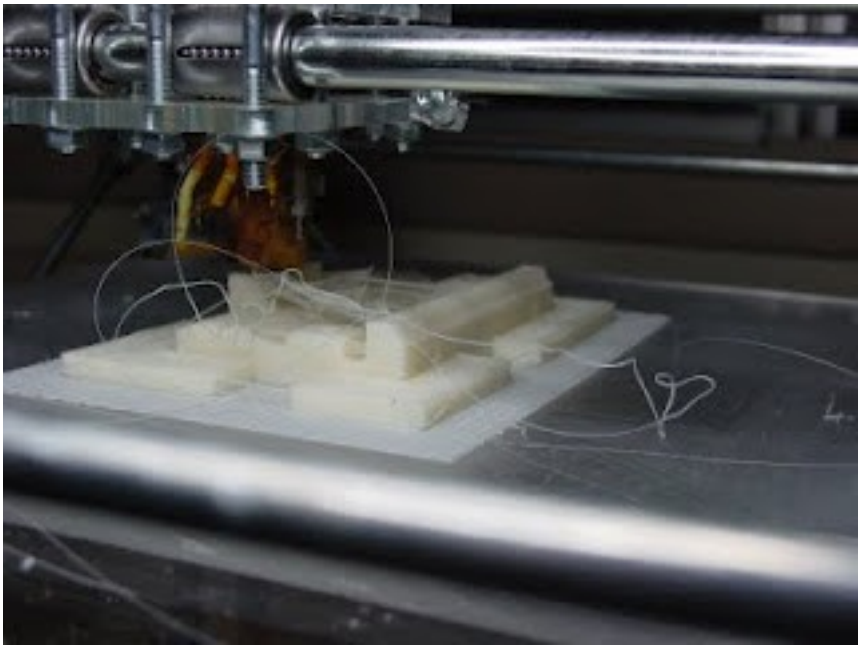
If you want to play with this part associated with segmented aprons, you can get the STL file here.

Several people have complained that they can not get the x-carriage-lower_1off part to print in ABS. I've rerigged my Rapman 3.0 for ABS and am attempting to print it now. I hate the stink of ABS.

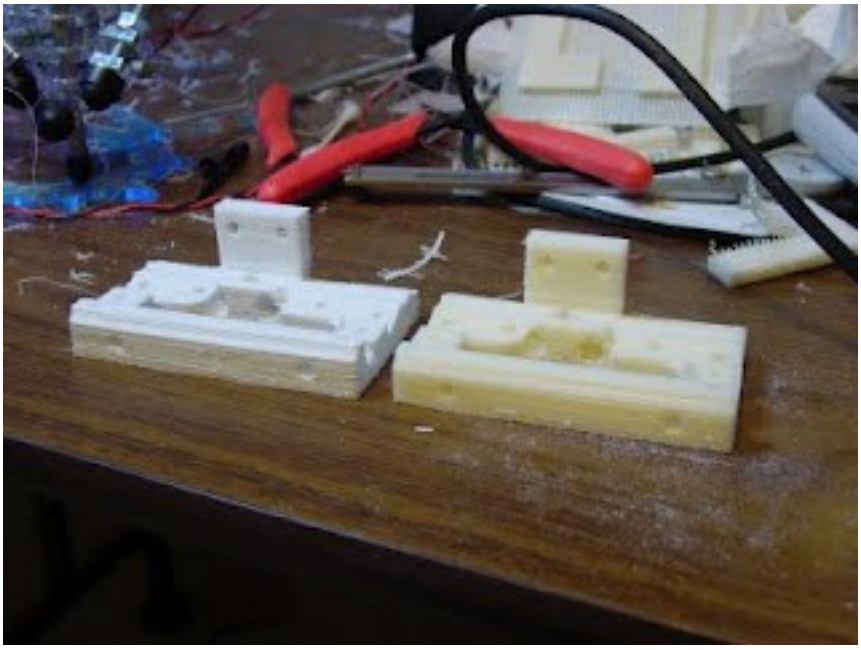
Segmented aprons work with ABS

Wednesday, 9th December 2009 by Forrest Higgs

I was lazy. I didn't even bother regenerating the gcode for x-carriage-lower_1off using ABS assumptions instead of HDPE assumptions. The first time I ran it just after resetting the Rapman 3 for ABS I didn't get the extruder orifice close enough to the print surface and the raft peeled away after a while.



I reset the z-axis and got it right the second time. Hint: The first level of raft extrusions have to be 2 mm wide or it will peel.
I'm too tired to do anything more than wait for this print to finish. I'll finish the blog entry in the morning.



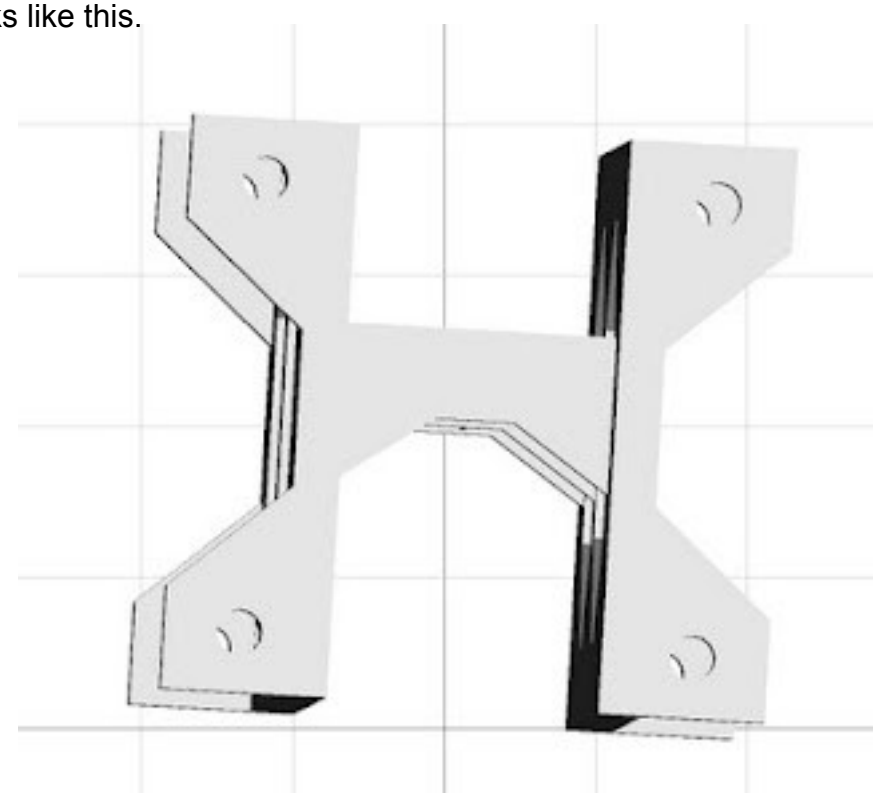
Okay, so I wasn't too tired. HDPE on the left, ABS on the right.
The point is, however, that segmented aprons pretty much completely suppress warping in the Mendel x-carriage-lower_1off part. Using ABS gives you much finer detail than HDPE does for a 0.5 mm orifice.

A problem with overhangs

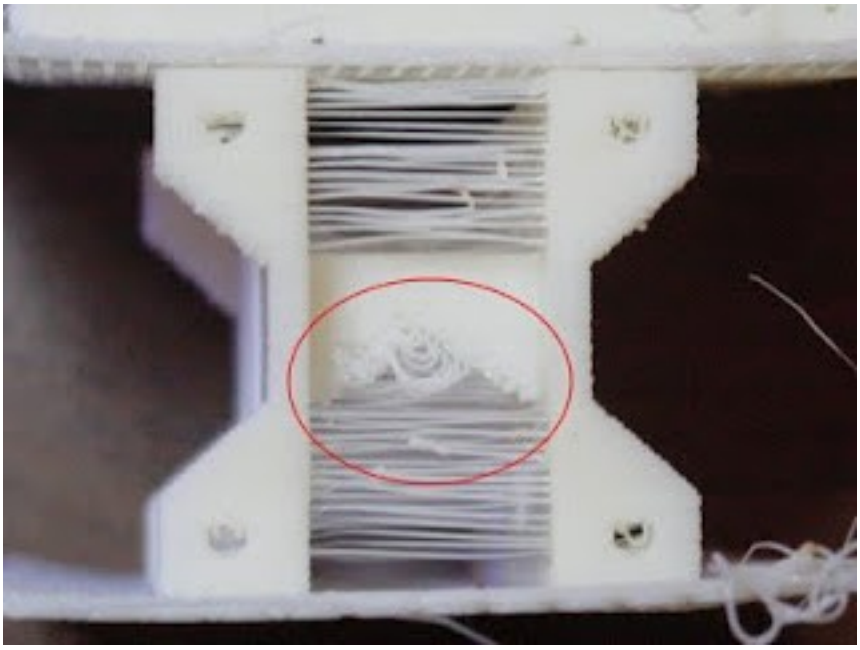
Monday, 14th December 2009 by Forrest Higgs

I don't know what is going on. When my Rapman/Skeinforge system prints horizontal holes in objects they typically come out looking quite good. When I try to do an overhang, however, things start to go wrong.

Right now I am printing out the parts for jdavis' Sarrus linkage. There is a spine segment amongst the parts that looks like this.



The center span has what looks like a relatively easy span. When I go to print it, however, it comes out looking like this.

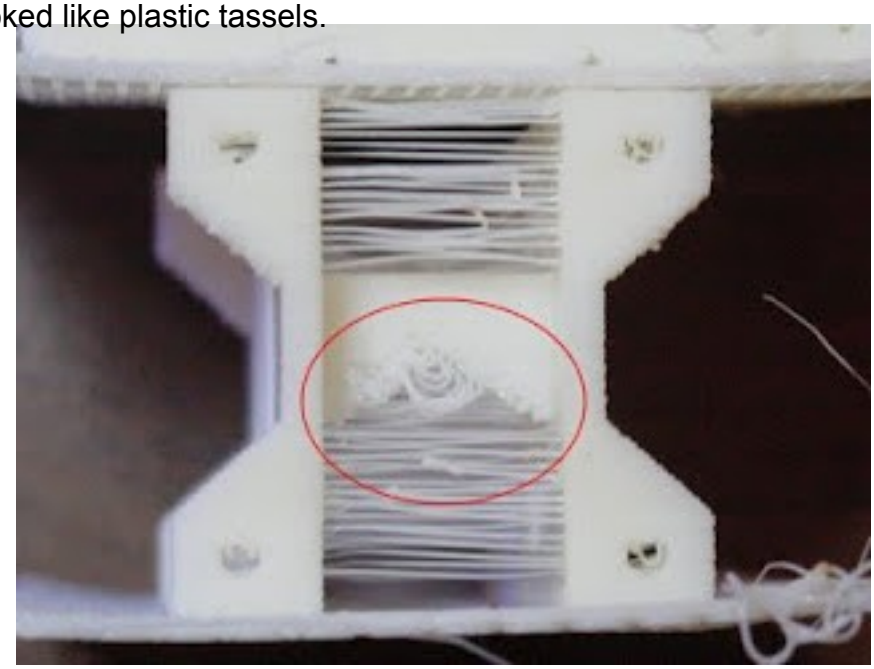


I am pretty sure that there are some settings in Skeinforge that cure this up. Does anybody know what they are? Thanks!

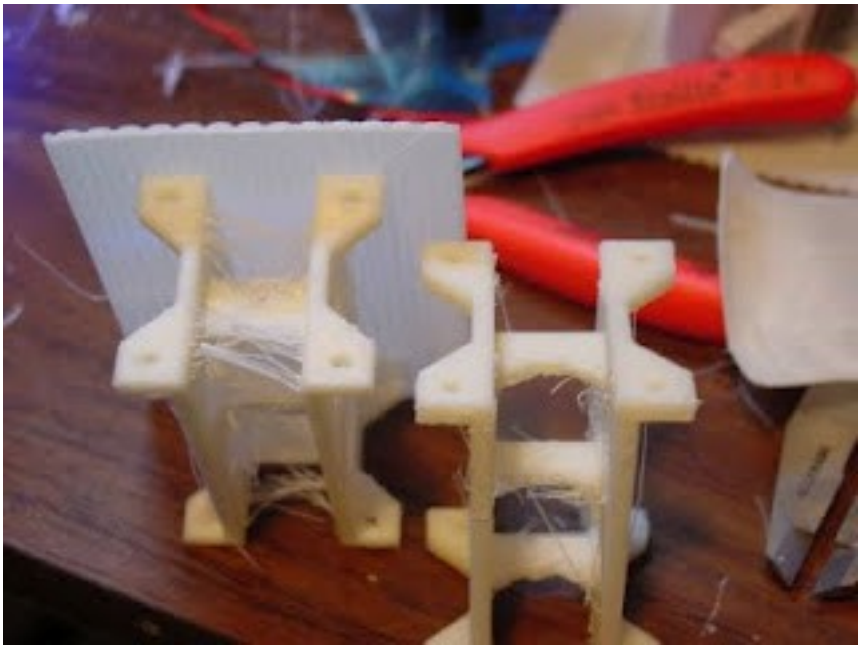
Getting overhangs right

Monday, 14th December 2009 by Forrest Higgs

When I bought my Rapman 3 I ordered two extruders and an extra extruder barrel drilled to 0.3 mm. When I assembled the extruder, I thought I had picked up one of the two 0.5 mm extruders. Instead, I had got the 0.3 mm one.
Oddly, I hadn't noticed the difference. The Rapman 3 extruder's filament pump was more than powerful enough to force 0.5 mm flow rates through my 0.3 mm extruder orifice without complaint. Thus, I was able to print what appeared to be 0.5 mm objects ... until we got to overhangs. When the extruder got to the overhangs it literally squirted the hot plastic into the voids festooning the print with what looked like plastic tassels.



I'd have probably never figured it out what was wrong save that my extruder orifice jammed. When I cleared it with a 0.25 mm strand off of a multistrand wire, I noticed for the first time that the ABS being extruded looked a bit thin. In fact, when I measured it, what was coming out was 0.38 mm in diameter. There was no way a hot filament that thin was coming out of a 0.5 mm orifice. That was when I realised that I had put in the 0.3 mm extruder barrel in error.
Knowing that, I reduced the flow rates and layer height to 150 and 0.25 mm and the overhang problem magically went away.



The new print is, of course on the right.

God jul!

Wednesday, 16th December 2009 by Forrest Higgs

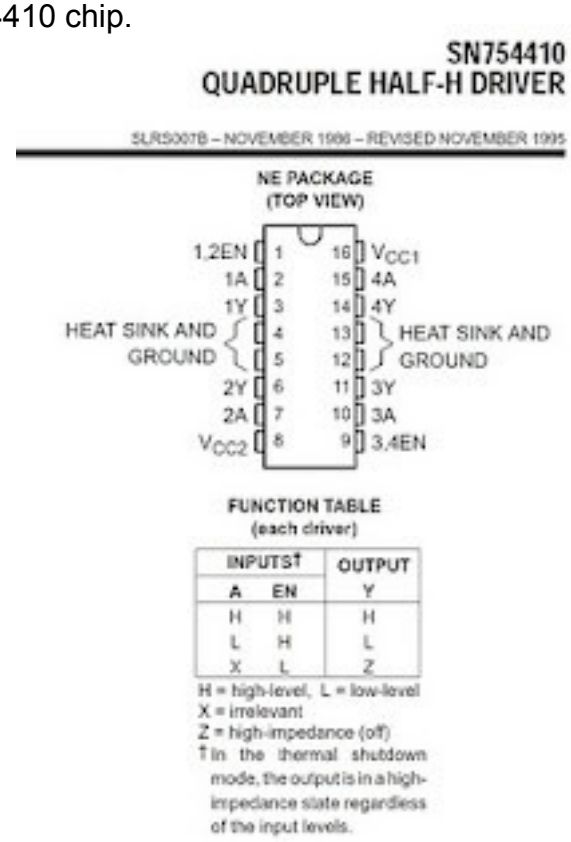
Jul p laboratoriet

Jag tror att Gud inte tvingar oss att tnka p honom, bara att vi tnka. nd tror jag att vi aldrig fr glmma sin son som dog fr oss.

Brute force stepper controller

Friday, 18th December 2009 by Forrest Higgs

Back in 2005-6 Reprap was using the venerable Pic 16F628A and was controlling steppers with the old, but reliable SN754410 chip.



SN754410
QUADRUPLE HALF-H DRIVER

SLRS007B – NOVEMBER 1986 – REVISED NOVEMBER 1995

NE PACKAGE
(TOP VIEW)

FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

The SN754410 is a very nice little chip. It is DIP, easy to use and very reliable. The only real problem with it was that you couldn't run more than 1.1 amps on one of them continuously. Simon McAuliffe, a Reprap core team member from its modest beginnings in 2004 solved the capacity problem by stacking one on top of another. While this worked, it created a rather nasty soldering problem for people like me who lack good fine motor control.

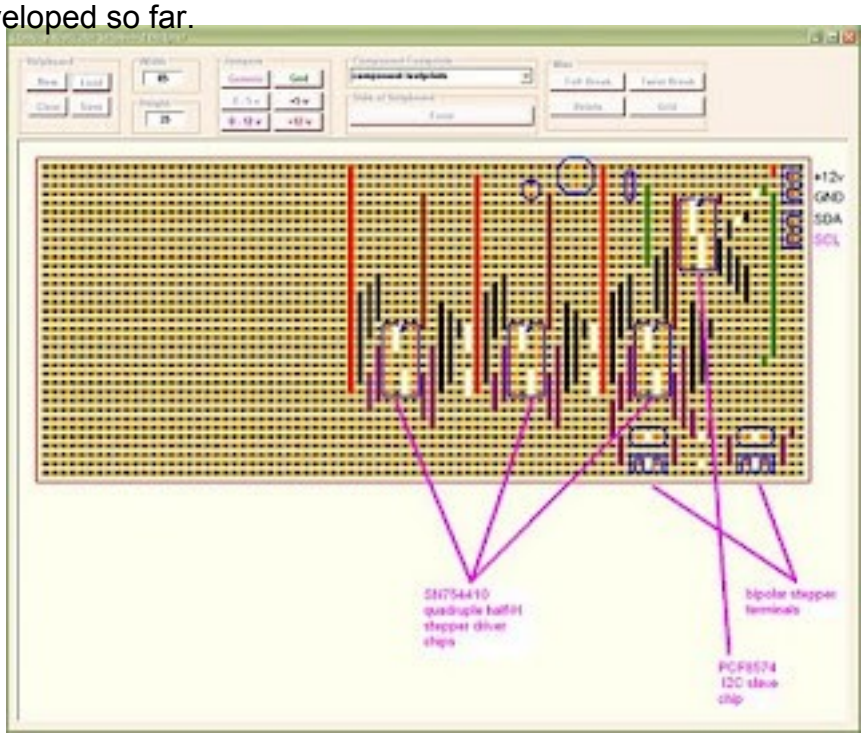
In those days we were using stripboard, known in the UK as Veroboard after it's original source. Stripboard is a wonderful development system for DIP chips. Unfortunately, it was hard to get in the US, a Velleman Euroboard costing anywhere from $6-15. As time went on, Reprap went more and more over to designing boards with Eagle and sending them off to lithography shops. Once there, it was only a matter of time before surface mounted chips began to creep onto the boards. Surface mount is a whole other chip packaging technology which caters to automated circuit board production. It is a bit difficult to make surface mounted chip boards, especially for clumsy oafs like me. As a result, I stayed with DIP technology, suffering considerable derision by other Reprappers as a result. My point was simple, however. I felt and still feel that if a bright 12 year old can't master the technologies and techniques needed to make a Reprap machine, we're not going to see viral diffusion of the Reprap technology.

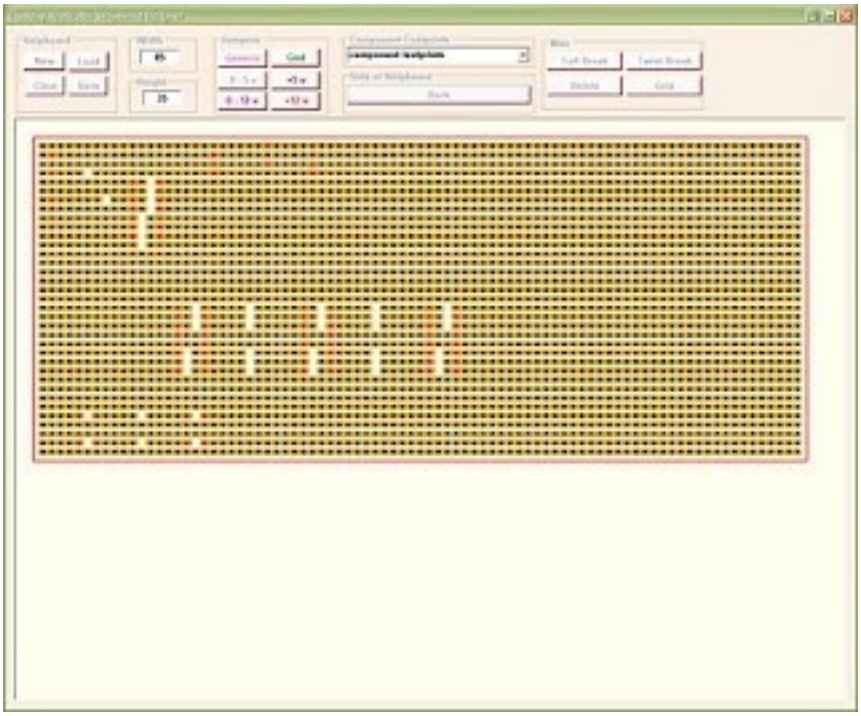
Last year, I found a cheap supply of stripboard in the US. Once I had a cheap supply of stripboard, I looked around for a nice, open source design tool for putting circuits on one. Unsatisfied with what was available, I developed my own.

Last week, having got a grip on how to run my Rapman 3.0, I decided to look at developing the controls for a next generation stepper controller. I ordered a bunch of the cheap stripboards...

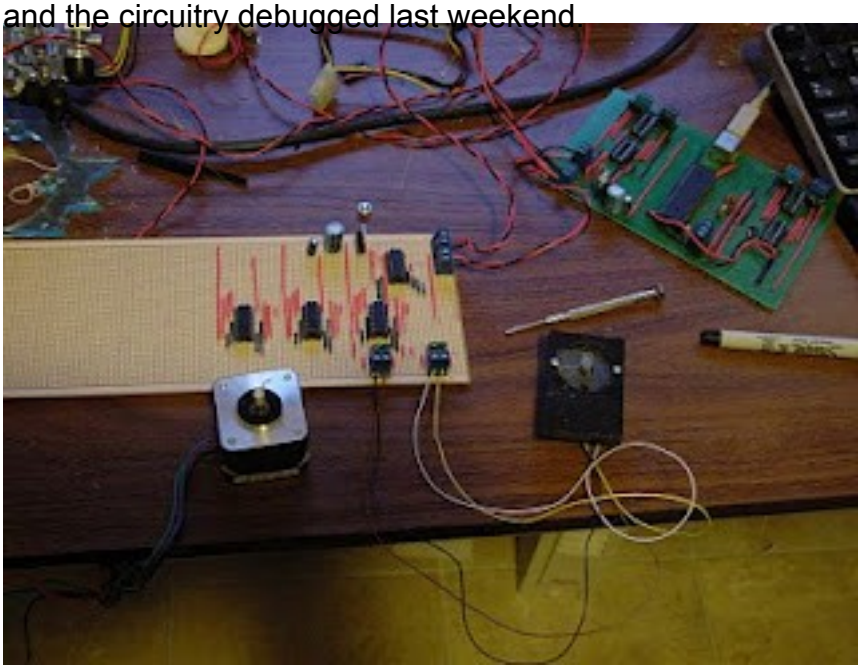
...to make sure that the web advertisement wasn't a fluke. The boards are quite high quality, made in Taiwan.

Once I had those in hand, I decided to leverage the work I'd done with I2C coms last year and design a high capacity stepper controller board. To keep things simple I took a page from Simon's book and decided to simply keep adding SN754110 chips till I had the amperage capacity I needed. I wanted the design to run a NEMA 17 {I own about a dozen} that I'd bought at the beginning of my involvement in Reprap. It draws 2.4 amps, so I needed three SN754410. Here is the design I've developed so far.





While I've configured this board for 3.3 amps capacity, it will easily seat another 2.2 amps of capacity. The circuit design is extremely repetitive.
I got the thing built and the circuitry debugged last weekend.



It has been a while since I built up a board, so I blew up a few chips until the drill of checking every circuit trace for continuity and checking again and again for shorts caused by sloppy soldering came back to me. I found that putting a washrag over the board when I powered up contained the fragments of exploding SN754410 chips.
In testing, I have been able to get a small, half-amp tin can stepper to run at speeds of up to 830 Hz. I seem to be able to run the NEMA 17 at about 4 KHz at full step. Mind, that is an estimate from my timing cycle and is probably on the overconfident side. I still haven't found the bug in the firmware that is causing problems with directional control.

In passing, the three SN754410 controller chips stabilise at about 90 C when running the NEMA 17. I haven't decided whether I will glue heat sinks on top of the controllers or bring the temperature down by adding another controller chip.

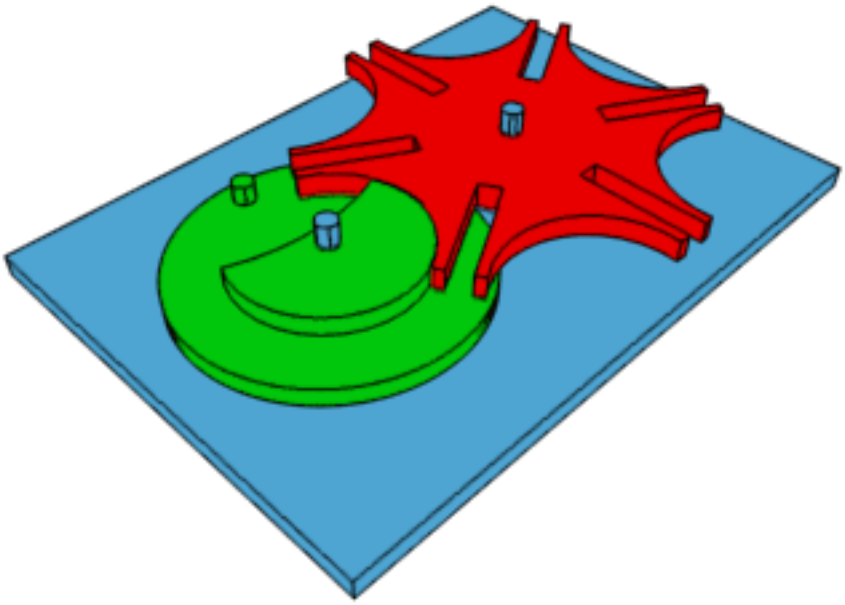
Printing Geneva Wheels

Monday, 28th December 2009 by Forrest Higgs

For some time I have been working on a magnet coil winder project. One of the things that I wanted was a tally counter to keep track of how much wire I was putting on the coil and another to keep track of how many turns that I made on my electromagnet spool.

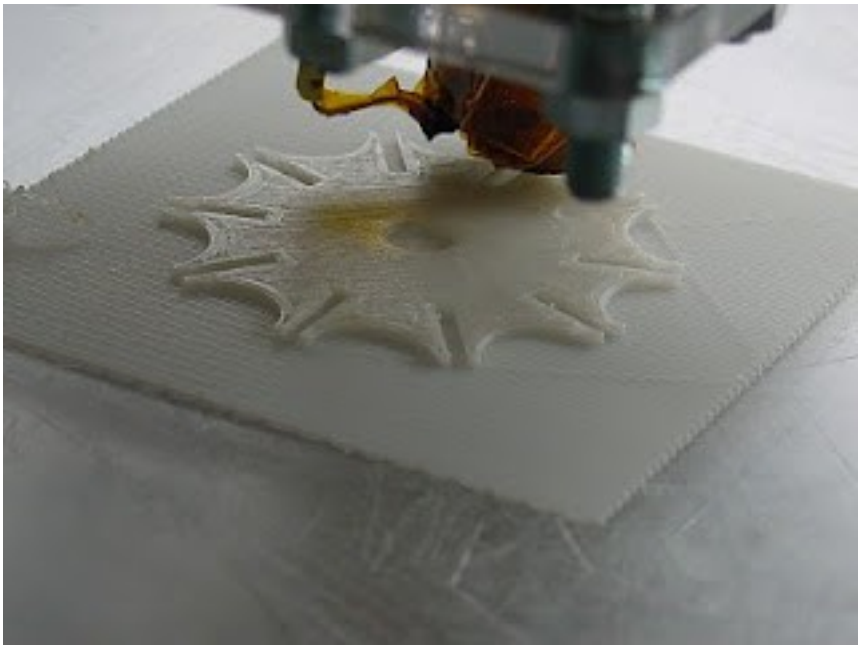
For a long time I tried to simply tried to replicate a classic tally counter. Unfortunately, that design was optimized for the use of springs and disks and a pawl of spring steel. I wanted to do the whole thing in plastic, or as much in plastic I could, so I turned to the venerable Geneva Wheel to do the job.

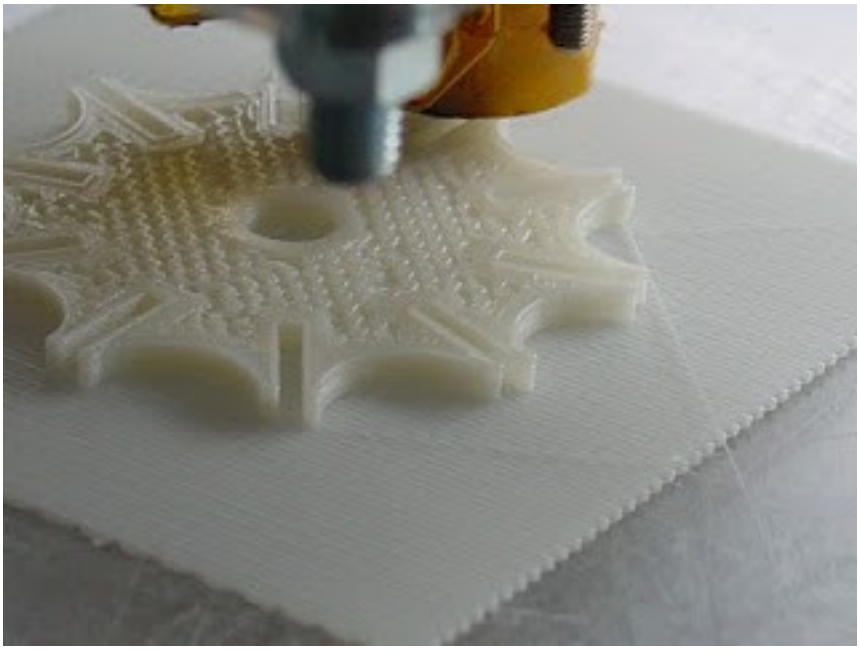
Here, complements of Wikipedia, you can see how a Geneva wheel mechanism actually works.

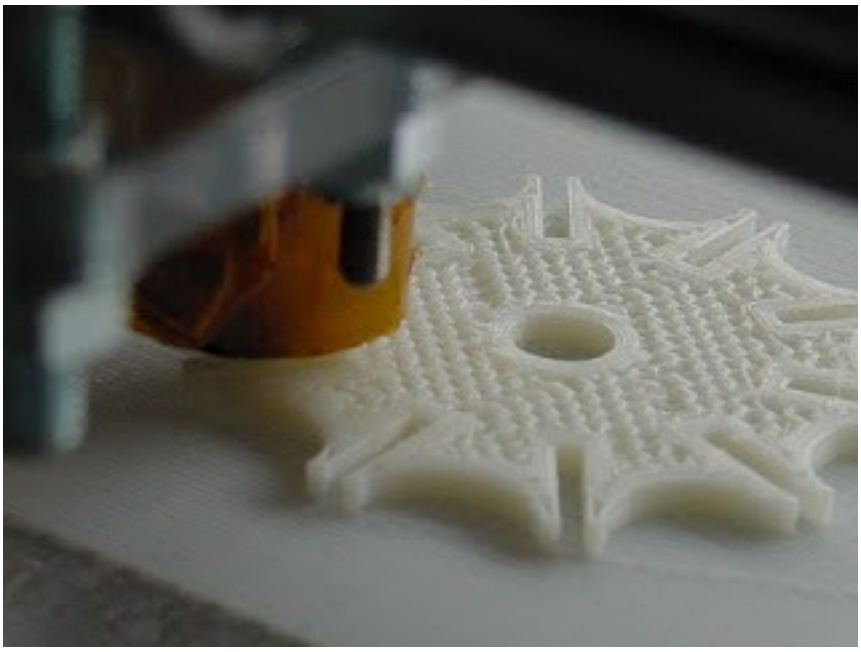


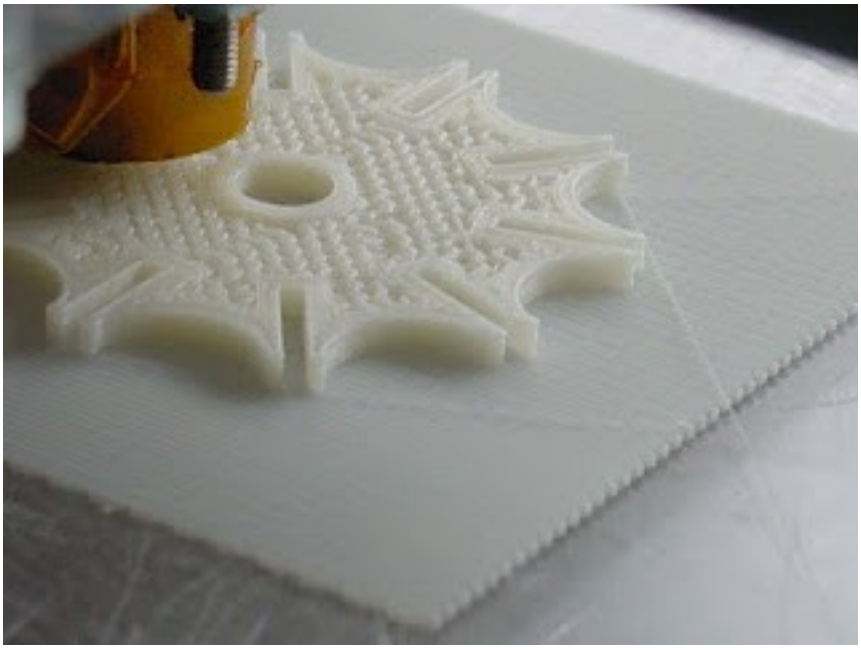
I wrote a script in Visual Basic .NET 2008 to produce the outline of a ten step Geneva Wheel instead of the four step one shown in the Wikipedia graphic. I read a file generated from that script using a script in Art of Illusion and created a solid object of the outline.

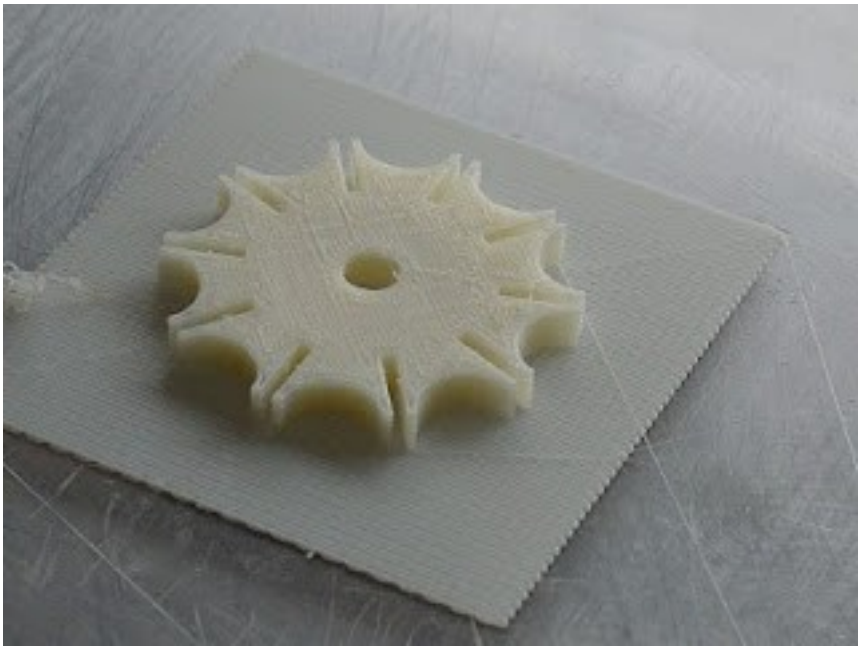
From there I generated the resulting gcode using Skeinforge.









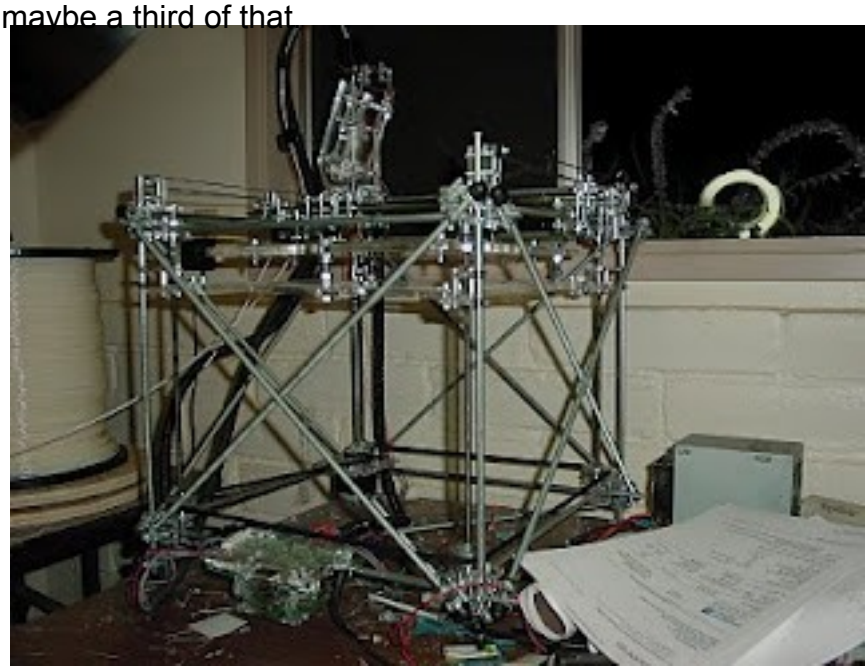


I will post the rest of the mechanism to this blog entry as I get it done.

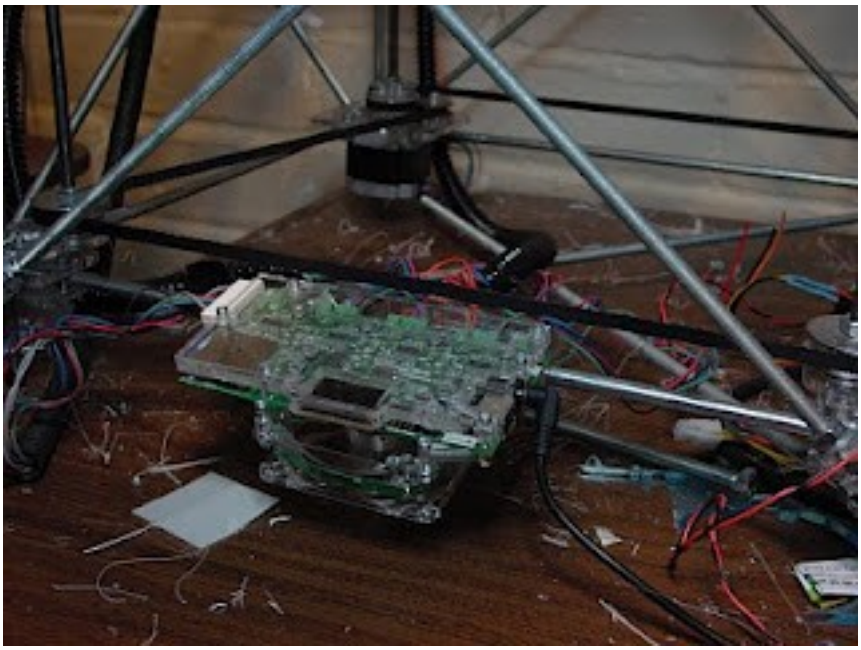
A note to my old friend, Hitech.

Sunday, 3rd January 2010 by Forrest Higgs

Think about buying or building your kid one of these. It's an open source 3D printer. I bought mine from BitsfromBytes in the UK for about $1,200, mostly because I was in a hurry. You can build your own for maybe a third of that.



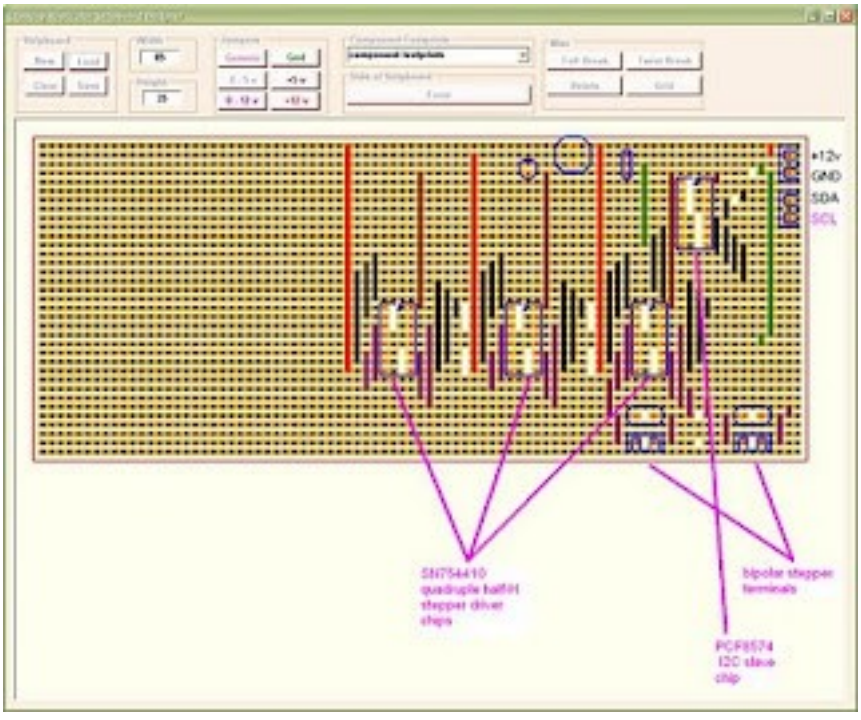
You can either buy the controlling electronics off-the-shelf.



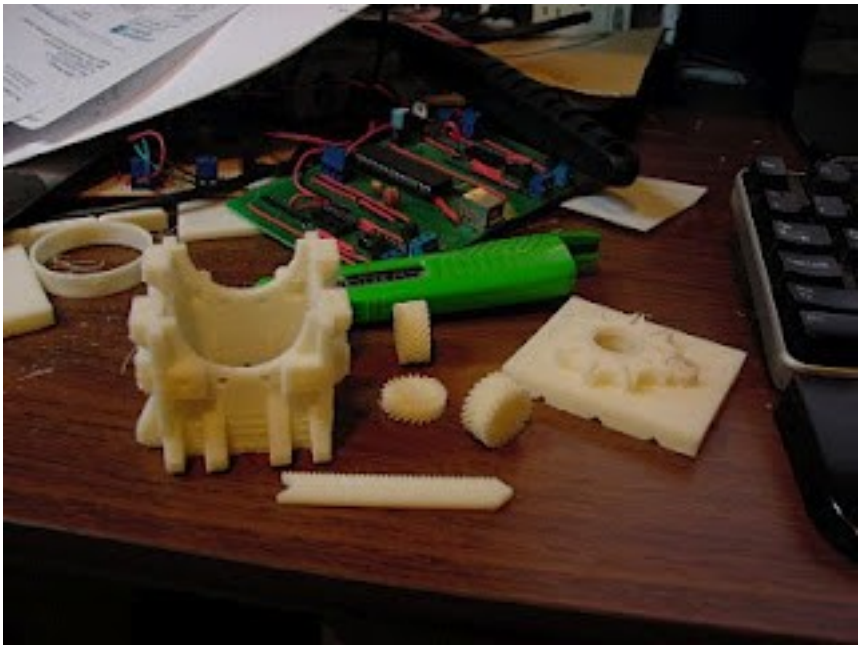
... or ... if you want to get your kids into electronics, you can build up your own like I do at times. I

tend to use stripboard for circuitry. You can get stripboards cheap out of New Jersey.

I use DIP though-the-hole chips because I'm old and clumsy. Little fingers can handle them a lot easier than you can with surface mount chips and components. I put together a layout app for design.



That makes getting from concept to board a lot easier.
Once you have your 3D printer working you can get your kids going designing and building THINGS instead of sketching things on paper or on some 3D CAD program.

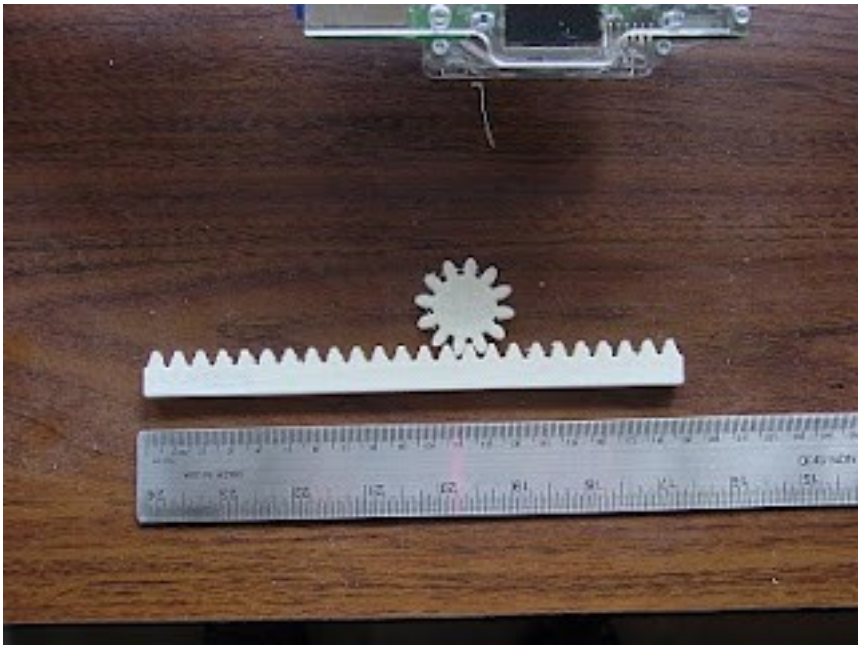


I think there is a lot of potential for getting kids used to thinking in terms of solid objects and how things go together. One of the immediate applications is for kids to design and build their own robots. Robots tend to be team efforts at the kid level largely because there are parts that are expensive to make because you either have to have a machine shop or access to one. With a 3D printer you can simply print the fiddly bits.

A printable, high speed alternative to belts?

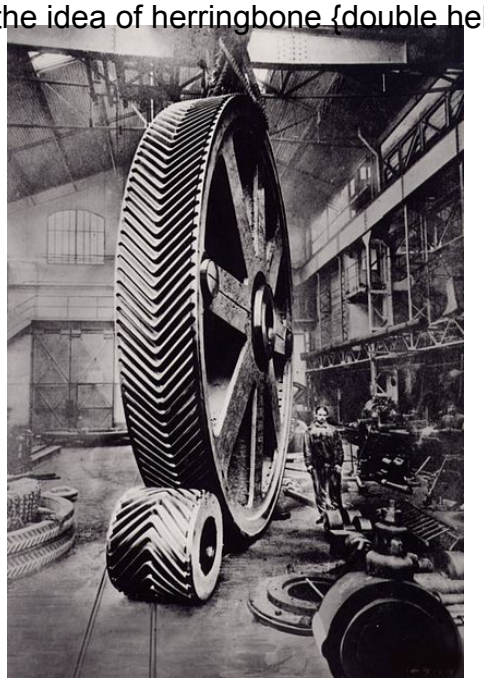
Thursday, 14th January 2010 by Forrest Higgs

Years ago, I created AoI scripts to design involute profile gears and racks. Recently, I've been exploring the notion of using rack and pinion drives instead of belts. While it is relatively easy to design and print a rack and pinion gear set, conventional ones have a problem with lateral stability.



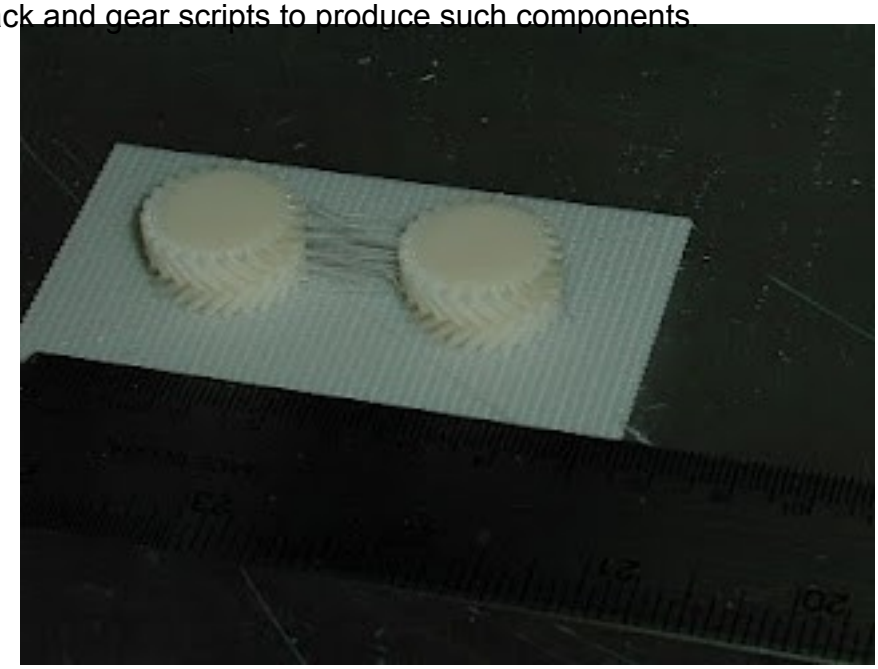
I soon found that I was buying far too many skateboard bearings to make up for this problem than was sensible.

A month or so ago, I ran across the idea of herringbone {double helical} racks and gears.



You don't hear too much about this kind of gear and rack mostly because it can't be machined with

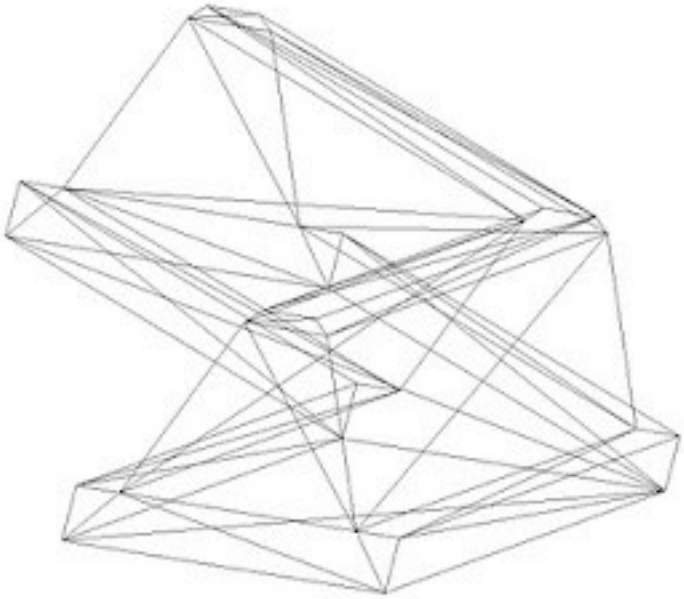
conventional hobbing machines. It can, however, be printed relatively easily. I found that I could coax my extant rack and gear scripts to produce such components.



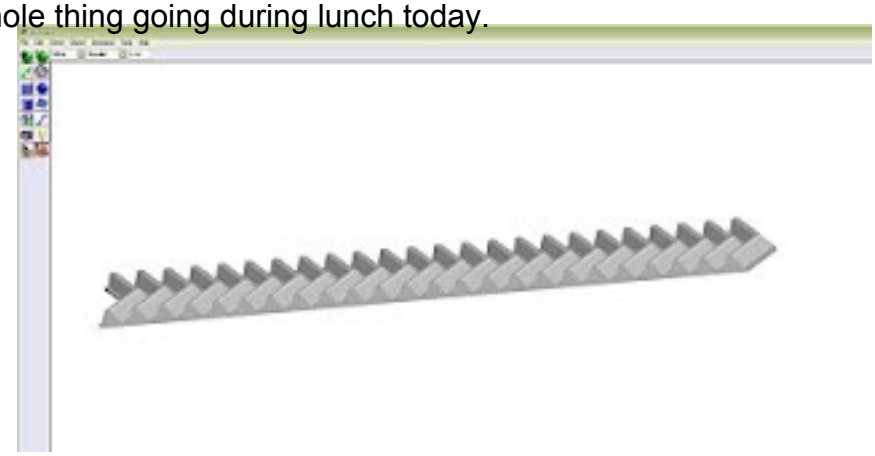
I was at, however, pretty much the limit of what Art of Illusion could handle. By the time I converted an involute gear profile or a rack to a triangle mesh then extruded and did a few boolean ops on it even the improved power of AoI 2.8 was barely up to the job.
While I could typically make herringbone gears happen the racks were a real trial. This had a bit to do with the fact that a rack profile does some really strange things when you apply AoIs triangle mesh routine to it.

You could extrude that and only get webbing on the outside surfaces. As you can see, however, you get lots and lots of triangles from AoI. I soon found that I could not make herringbone racks with more than about 12 teeth. It was easy to see with a rack that you ought to be able to describe it with relatively few triangles, so I gritted my teeth and decided to go directly to a solid description.

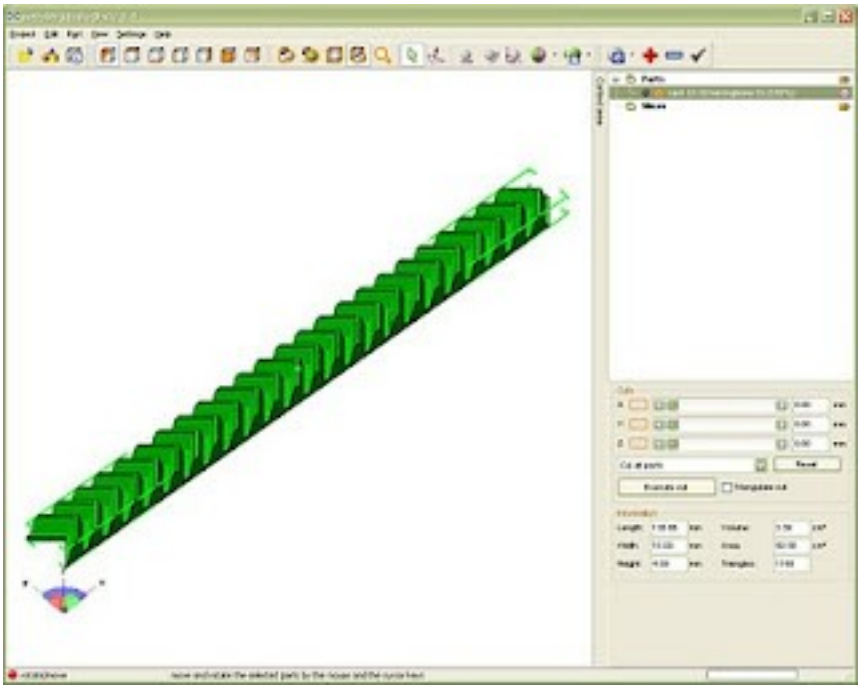
Using the Platonic Solids script as a point of departure, I began to develop a herringbone rack script.



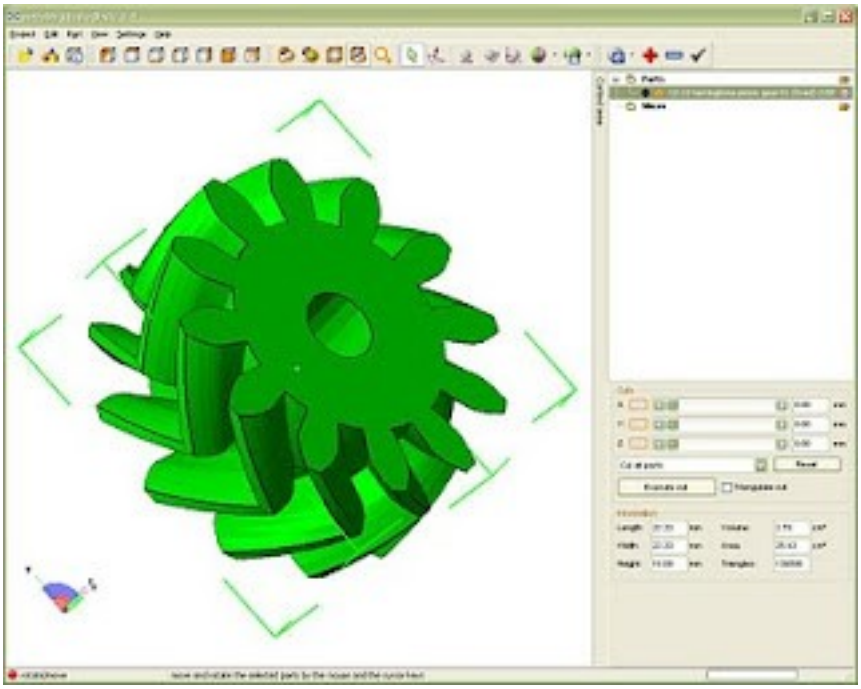
I finally got the whole thing going during lunch today.



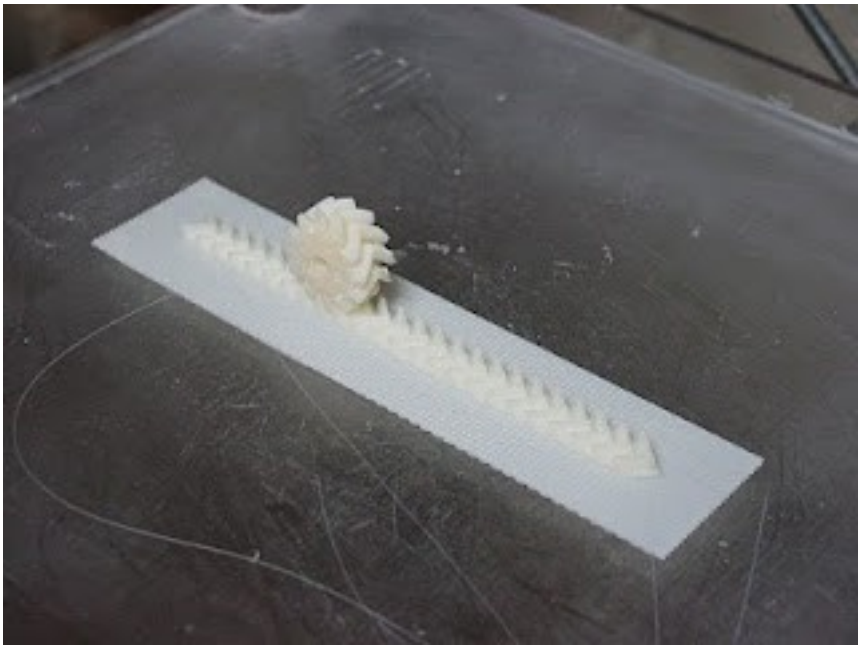
I checked the resulting STL in Netfabb and determined that it was perfect.



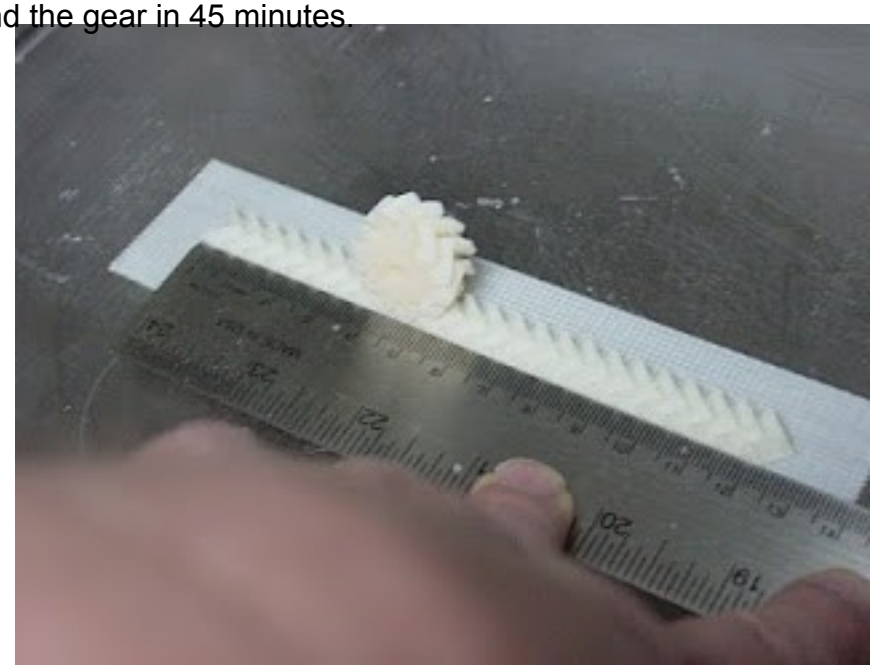
I developed a 12 toothed 10 mm radius herringbone rack and pinion pair.

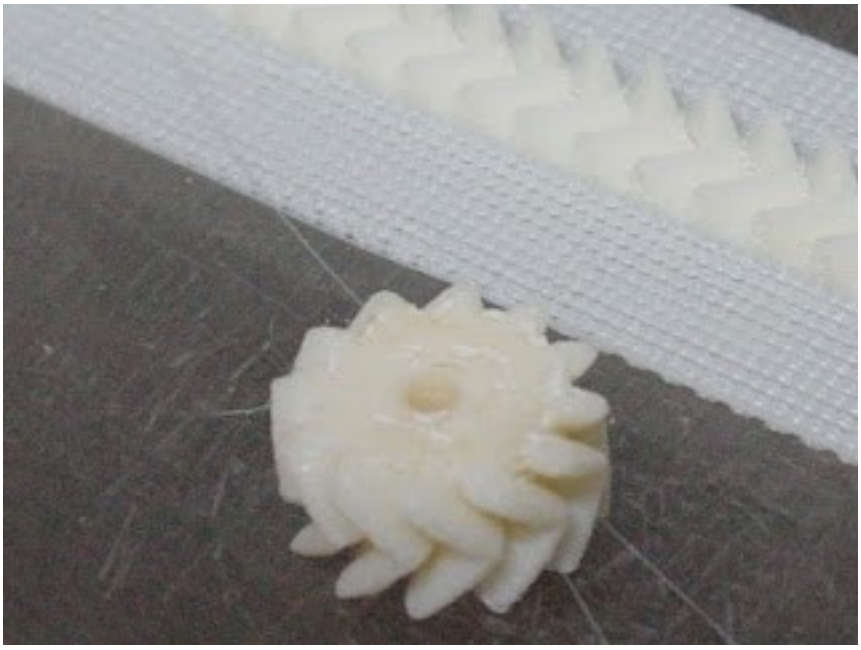


This configuration gives you about 0.3 mm/step when you attach a 1.8 degree stepper to it. Gear that down by a factor of 3 using another herringbone gear pair and you've got the 0.1 mm/step standard Reprap resolution.
Herringbone gears are pretty much naturally anti-backlash. They seat firmly, are quiet and have strong lateral resistance. They're printable and there is no reason whatsoever why you shouldn't be able to use them instead of a belt. I printed up a 130 mm rack and pinion set this evening.



The racks can also, since they are very thin, be printed in long lengths without warping. Using a 0.3 mm extruder orifice and printing at about 16 mm/sec I completed the rack in about an hour and twenty minutes and the gear in 45 minutes.





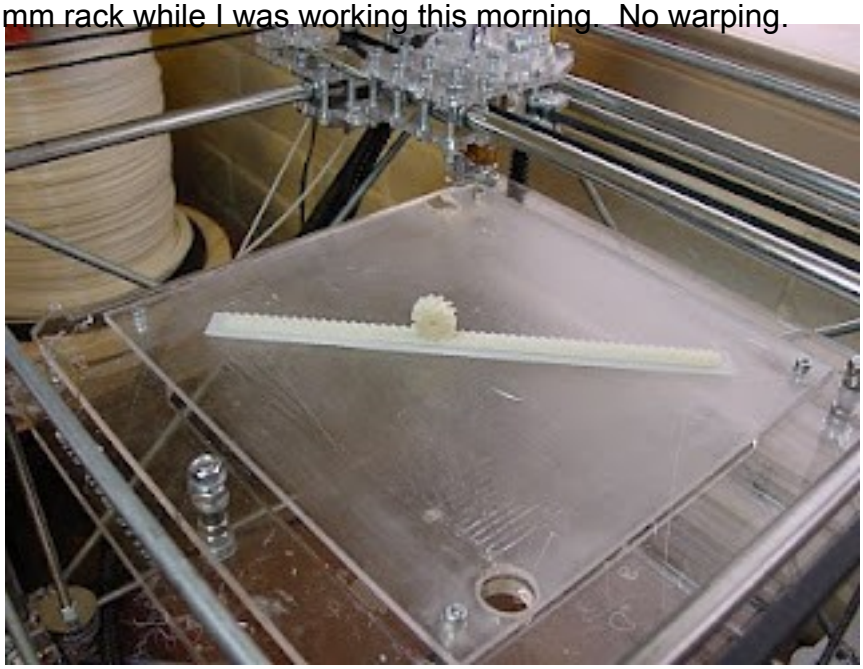
I've been able to make rack and pinion sets with gear finenesses of 24 teeth for a 10 mm radius. That's really pushing the envelope, though. It appears that 20 teeth for 10 mm radius is a practical limit.

I intend to clean up the rack script and write one for the herringbone gears now that I know that they work well. I will make the scripts available as soon as I have them cleaned up a bit.

This experience has brought home what I consider a very important point for me. We need to be looking for technologies that aren't necessarily cheap or usual in metals. Such components are dependent on purpose built milling machines. Our printers are much more flexible than that. We need to choose technologies to print with that in mind.

Friday update:

I queued up a 267 mm rack while I was working this morning. No warping.



Not bad.

Getting there with herringbone rack and pinion

Monday, 25th January 2010 by Forrest Higgs

I finally hit a break in my day job and after sleeping the clock around beginning Friday afternoon was able to get back to my Reprap work. I'd been working with herringbone rack and pinion design and had begun writing Art of Illusion scripts to generate this kind of technology. I'd done most of the rack script a few weeks ago, but I needed to be able to design herringbone gears a bit more efficiently.

Finally, on Sunday the scripts for the racks and gears began to come together. After a considerable amount of feeling around I found that I could reliably print an 8 mm radius, 12 toothed gear. Connecting such a pinion directly to a 1.8 degree step NEMA 17 gives me a 0.25 mm/ step on the axis without microstepping. I then designed a 32 toothed gear which let me get that resolution down to 0.094 mm/step. You can see the layout here...

The NEMA 17 turns the 12 toothed pinion at the top of the picture. It turns the 32 toothed gear which shares an axle with a second 12 toothed pinion which engages the rack. It's simple, easy and quick to print and doesn't backlash if you apply just a slight bit of compression to the gear train. My next task is to design a printable axis assembly to house it. My goal is to get rid of the skateboard bearings, too.

The scripts are in a lot better shape, but they're still not really ready for prime time.

Many useful little things...

Sunday, 31st January 2010 by Forrest Higgs

Back in the late 1960s when I was very young and worked for IBM for a few years there was a magnetic tape that always lay beside the operator's console on the ancient IBM 360-40 computers labeled MULT. One day I worked up the temerity to ask the operator what it was and he said MULT stood for "many useful little things", viz, MULT. It was a compendium of utilities programmes that enabled the operator to maintain the old 360 and, given how reliable mainframe computers were in those days, was always kept very close at hand.

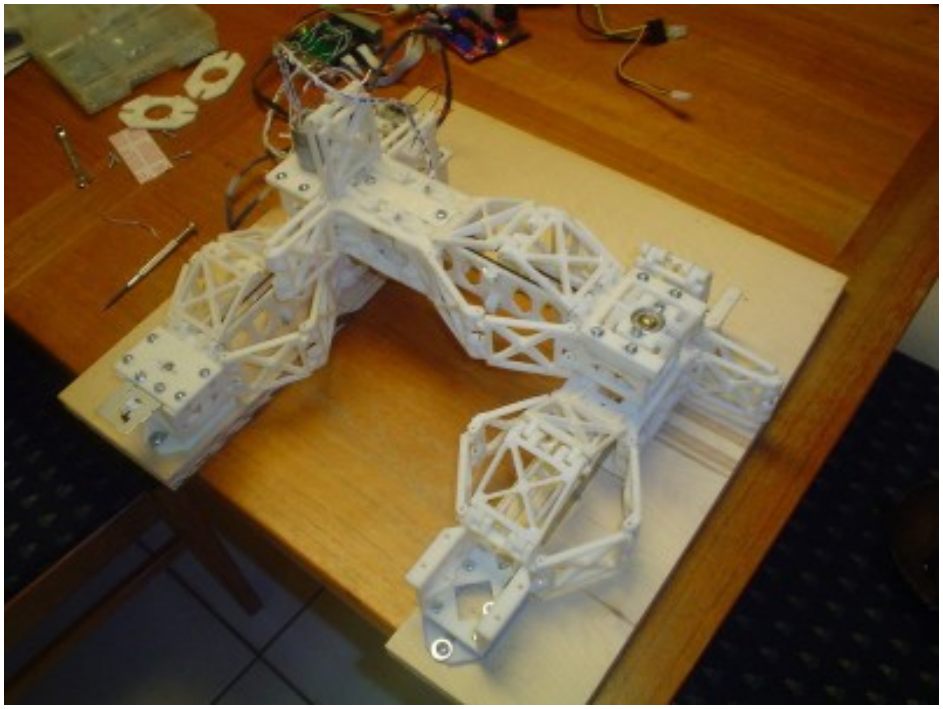
The projects that I've begun to undertake vis a vis Reprap reminded me of that old mag tape. Basically, I've been undertaking to explore technologies that might get the Reprap community into the next generation of Reprap machines. More to the point, I'm trying to crack some of the technical challenges posed by the Kartik M. Gada Personal Manufacturing Prize. Mind, I'm not looking to compete for the prize. The technical challenges, however, are very interesting.

Before the Gada Prize was announced, I was developing herringbone racks and pinions as a printable alternative to belts. Once the prize was announced, the 90% printed by volume and the 60 watt power limit specifications drew my interest. Prior to the prize announcement, I'd bought all the pieces to build up a heated bed for my Rapman 3.0 printer. It was obvious, however, that there was no way that a 3D printer with a heated bed was going to be possible using less than 60 watts.

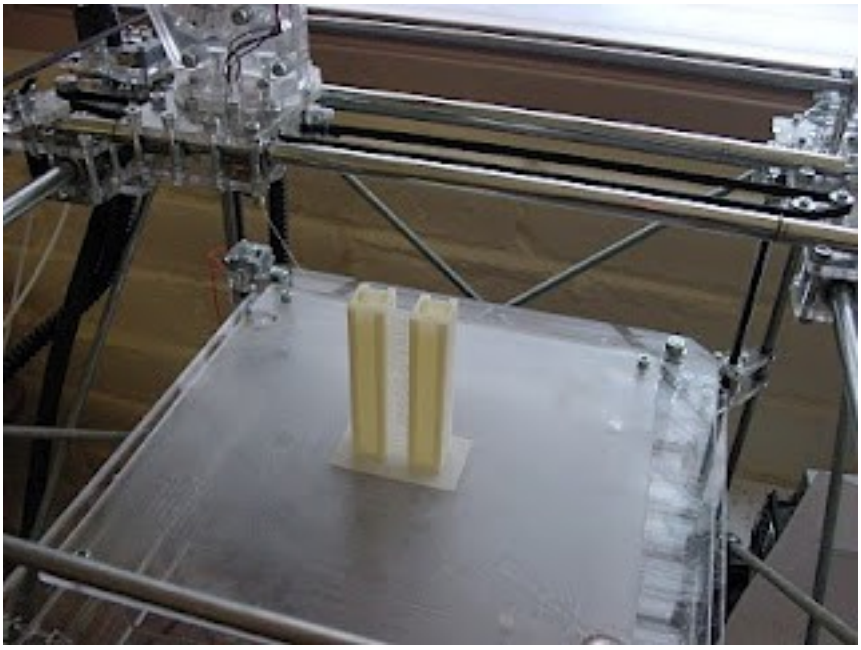
At the time that I was thinking about all of this I was trying to develop a rail system to contain my herringbone rack so that I could use it to drive axes. I wanted to build a Delta Robot something along the lines that Festo had done.

The Festo Delta Robot uses lead screws in the three columns that seat the arms that move the extruder. I wanted to replace those with herringbone racks. I also wanted to make the columns printable. In that a usuable Delta 'bot is about a meter high, it was obvious that I wasn't going to be able to print a column in one piece. That put me face to face with the question of how to make a large piece out of a bunch of little pieces.

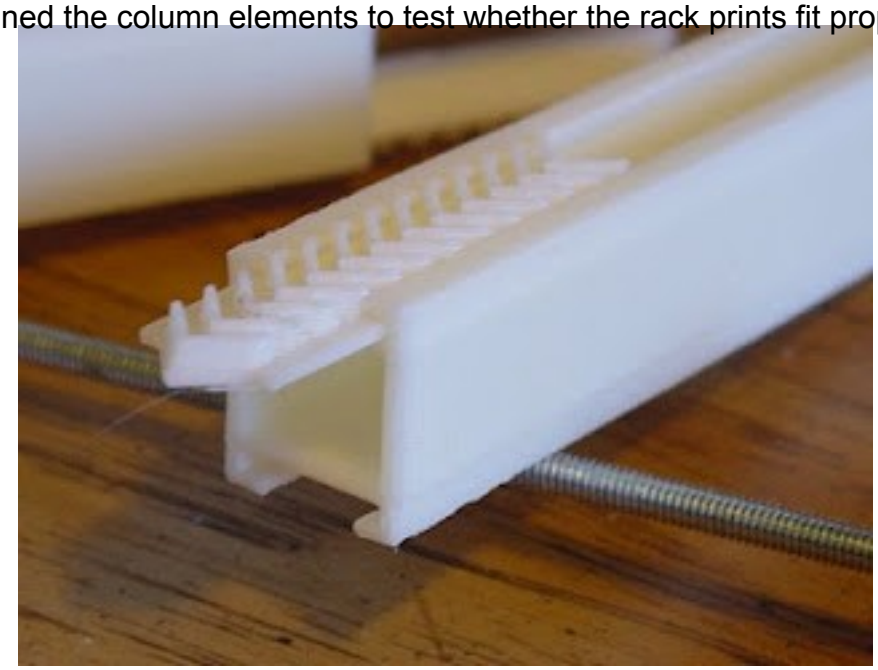
The conventional approach is to simply bolt the small pieces together. Frank Davies took this approach with his brilliant Sarrus Linkage positioning system.



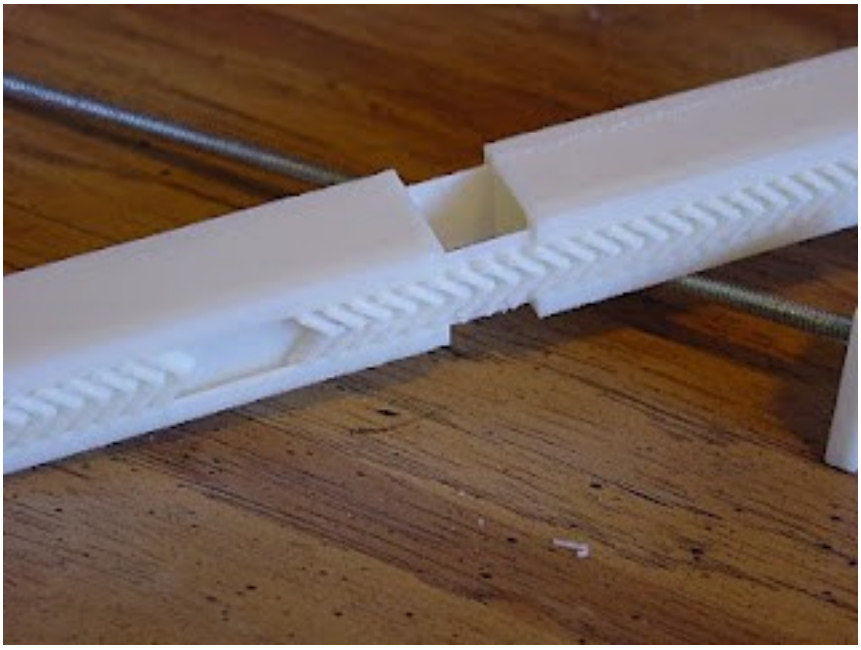
Frank had avoided a lot of problems with printing larger parts by using relatively thin-walled, open structures. I shamelessly stole a lot of his techniques after having printed part of his Sarrus system. While thinking about that it occurred to me that you get relatively little warping if your part's biggest xy dimension is less than about 50 mm. Most of the parts I wanted to print had a cross section much smaller than that but were long and I needed them to be VERY accurate.
Then came the little epiphany. Why not rotate the long dimension to the vertical and leave the small cross-section on the xy print surface? Nophead {Chris Palmer} quite rightly pointed out that for a beam the extreme fibres on the upper and lower surfaces would be no stronger than the bond between two layers of printed plastic whereas if you printed the long dimension flat on the xy plane the bond between layers would only be subject to shear stress. In spite of that I began designing a columnar rail system for my herringbone racks and began printing it vertically, reasoning that a bending stresses would not be as severe in a column as in a beam.
After half a dozen false starts I finally got a design I liked. I rewrote the rack generation script for Art of Illusion so that it would put a flange on either side of the rack and then designed a cross-section that would seat racks on front and back sides. One of the racks would carry the drive pinion while the other would seat a pair of unpowered bogies to stabilise the positioning assembly.
Here you can see a printing of a 100 mm long pair of these columnar rails.



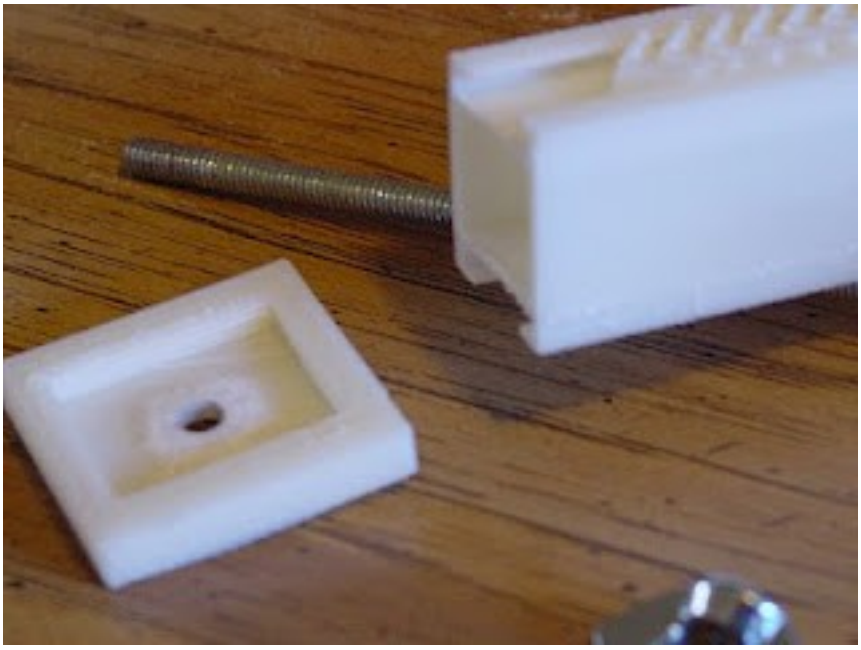
Note that printing vertically allows you to create hollow structures, something much more difficult to do with present technology if you print them on their side. Also note how much like an extruded plastic or aluminum section this columnar segment looks. That was intentional. If some enterprising small Chinese factory starts paying close attention, it may be that this sort of section can become a very cheap vitamin instead of something that Reprappers in well served market areas have to print. That cuts down replication time dramatically.
I'd originally designed the column elements to test whether the rack prints fit properly. They did.



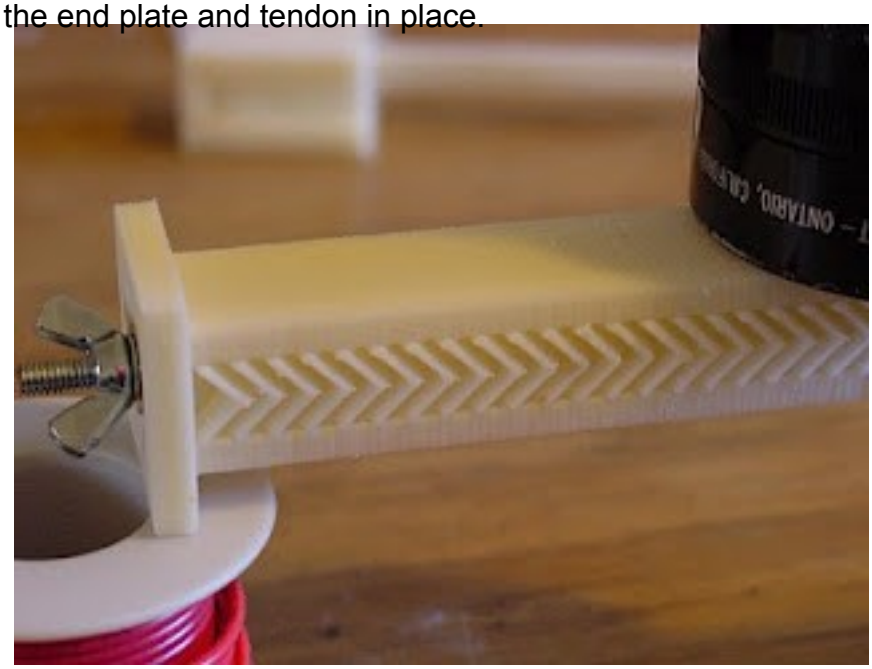
Before you start thinking that I'm some sort of design wizard, let me say that it took me four tries to get the fit right. This sort of design thing is very hard work for me.
That accomplished I was beginning to design the connection between the columnar segments when it occurred to me that the rack prints already did that.



What is nice about this approach is that you can simply slide rack segments into the column until it is full. I found as a practical matter that the segments dovetailed very accurately with just a touch of very fine grit sandpaper to knock off tiny bits plastic flash on the ends.
Just looking at the system one immediately realises that the join between segments is not what you would call a particularly strong. That is where an old trick I learned in architectural and structural engineering design eons ago came into play, viz, post-tensioning. In construction concrete is well known for being able to resist huge compressive loads but can resist virtually no tensile loading at all. Plastic, the way I was printing it vertically, as Nophead rightly pointed out, doesn't have good tensile strength and will, given sufficient load, fail in bending. In that way it is very much like concrete.
In building in concrete post-tensioning allows you to overcome this problem. The method is quite simple. You cast your concrete structure leaving channels through it. After it has hardened {cured} you thread tendons through these channels and then use hydraulic jacks to put the tendons in tension. The tendons are connected to either end of the beam you are building by face plates. The tensioned tendons, usually made of either high strength steel rod or cable, compress the concrete beam strongly via the end plates to which they are attached. Any bending forces put on the beam thereafter have to overcome this compressive force on the concrete before the beam can fail. The method is very widely used.
I printed up some end plates and cut a piece of #8 studding {4.2 mm threaded rod} for the tendon.



Here you can see the end plate and tendon in place.



Finally, I point loaded the resulting post-tensioned structure with a 750 gram Mag-Lite. The 200 mm column section weighs about 45 grams of which the steel and fixings account for 12-15 grams.

No visible deflection was observed. This beam in this orientation is 18 mm deep, mind, and the top and bottom membranes are 1.75 mm thick ABS. I will use this same approach for both columns and beams in the Delta Robot I am designing.
I think that by deepening the beam to 24-30 mm you could probably replace Rapman's 12 mm milled steel guide rods {and probably Darwin/Mendel's ... I haven't checked the exact specification} with a post-tensioned, virtually entirely printed equivalent using about 6% of the steel {#8} as is presently used.
Keep in mind that the #8 studding tendon is massively overdesigned. I bought #8 simply because, for some odd reason, it cost about on-third as much as #4 {2.8 mm equivalent}. Using perfectly adequate #4 studding would bring that steel fraction down to 4%. Heavens, even #2 would do the job!
One issue with this kind of development is creep, the tendency of plastic under stress to deform over long time frames. It may be that we are simply not loading plastic to anywhere near the stress levels where this becomes a problem. It warrants a hard look, however. Unfortunately, the reference manuals on plastics creep are quite expensive, viz, hundreds of dollars and are rather spotty in the coverage of plastic types that they discuss.

A stepper controller for the Delta Robot

Monday, 1st February 2010 by Forrest Higgs

Some time ago, Nophead {Chris Palmer} suggested that I use microstepping instead of gearing to get the resolution that I need in using the herringbone rack and pinion with the Delta Robot kinematics. I had hesitated because doing so got me into the surface mount Allegro controller chips. I could have used the standard Reprap controller board except that it is incompatable with my Pic-based controller.

A happy solution presented itself yesterday. Pololu offers a breakout board equipped with an Allegro A4983 controller.

The A4983 is somewhat different than the A3977 that is standard in Reprap electronics in two ways. Foremost, the A4983 can handle a maximum current of only 2 amps instead of the A3977's 2.5. I don't find this compelling because in practice Repraps rarely require more than 1 amp in any case.

The winner for me is that the A4983 controller allows for microstepping down to 1/16 compared to the A3977's 1/8. While the A3977's microstepping is more than adequate for conventional Reprap machines I can use the extra resolution with the Delta Robot.

Part for Prusajr

Tuesday, 2nd February 2010 by Forrest Higgs

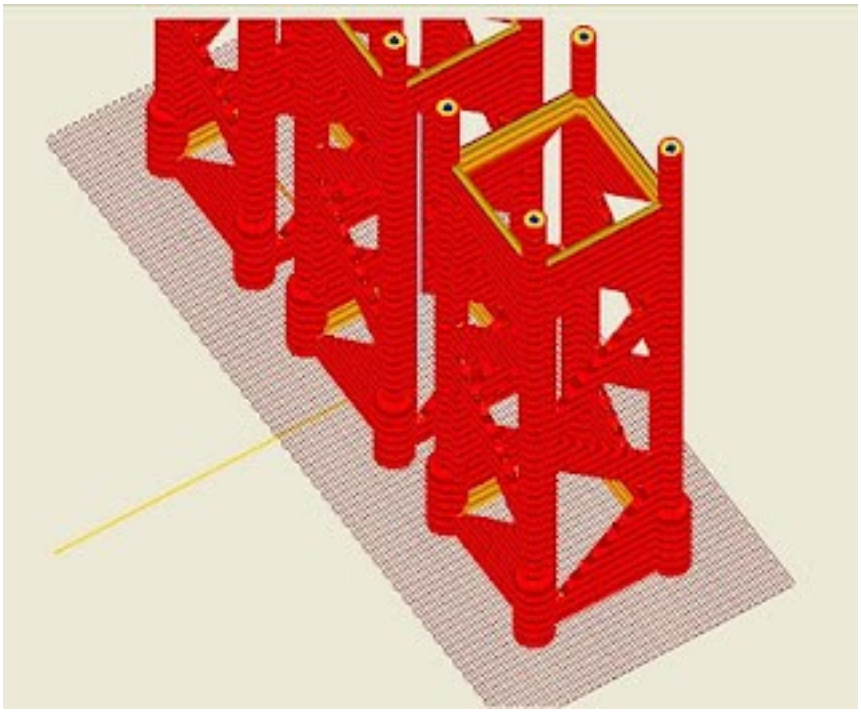
Skeinforge on a tear

Thursday, 4th February 2010 by Forrest Higgs

Has anybody seen anything like this before?

Skeinforge processes the STL properly but crashes when it tries to export the BfB file. I've had this happen a bunch of times now. Oddly, if I lay the beams on their sides they process and export the BfB properly. When I stand them on end, however, it crashes.

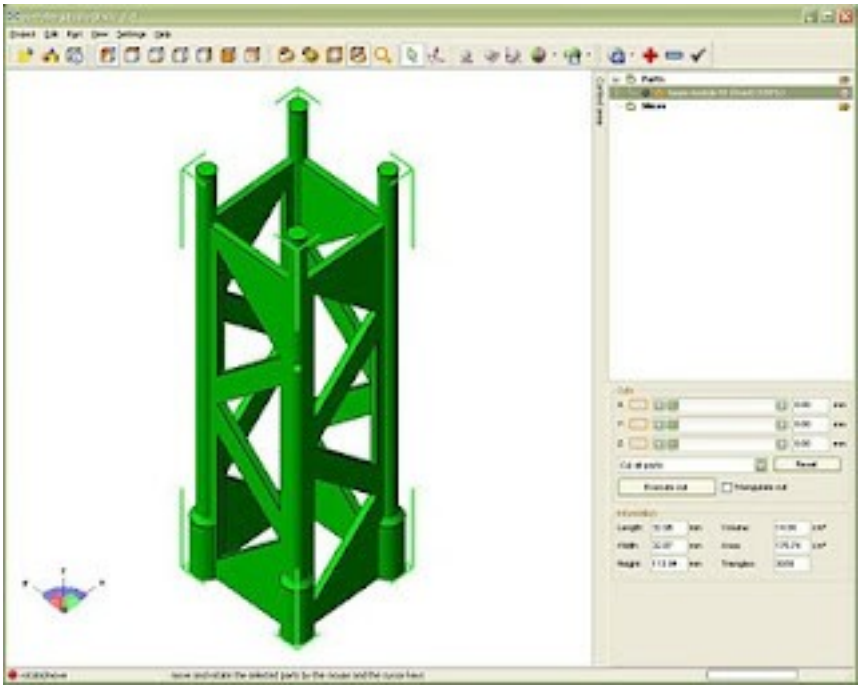
I've watched the job's progress half a dozen times on the Task Manager and haven't been able to see any consistent set of conditions which might be causing the crash.



Any insight into what might be happening would be greatly appreciated.

Latest column system project

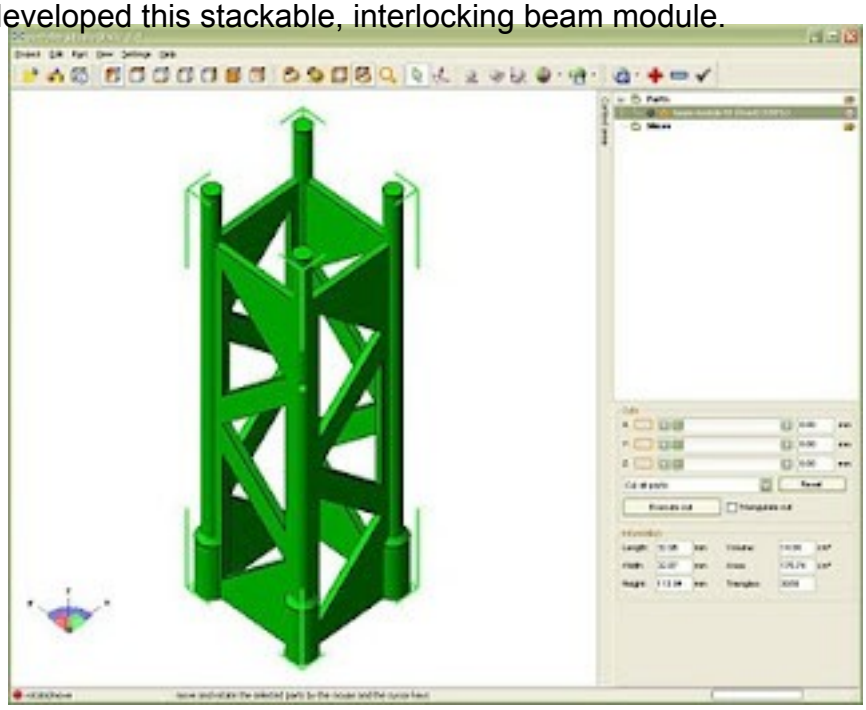
Friday, 5th February 2010 by Forrest Higgs



Testing the envelope

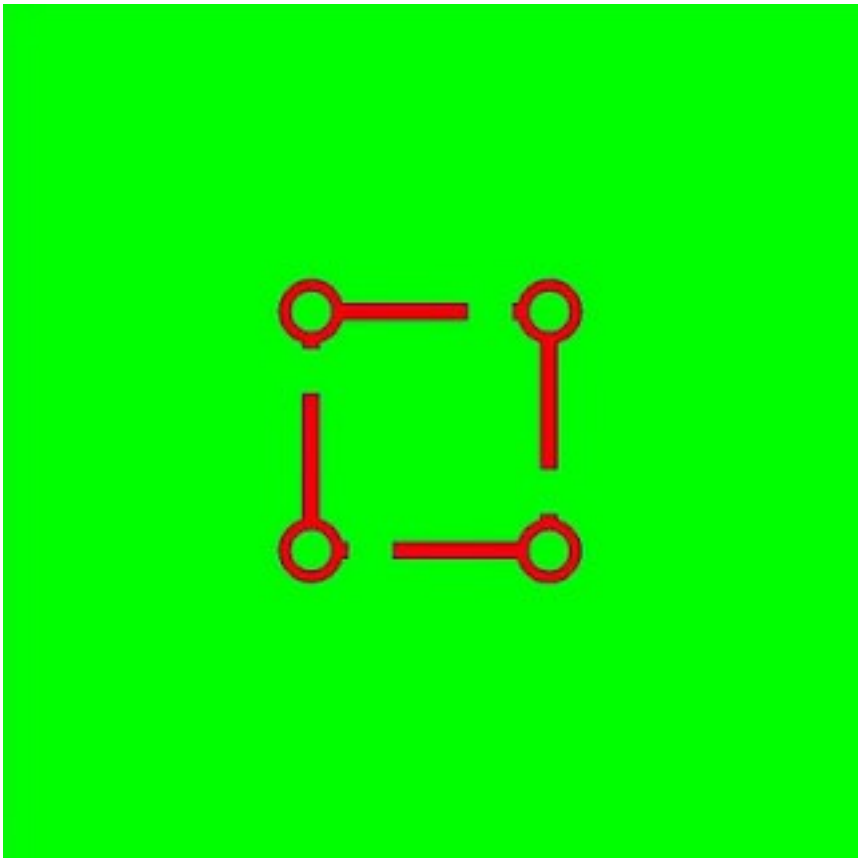
Saturday, 6th February 2010 by Forrest Higgs

This has been a pretty frustrating week. Last week I got a long way towards evolving a thin-walled approach to making post-tensioned, composite structures which could support herringbone rack and pinion systems. This week I wanted to push the post-tensioned theme a bit further and decided to see what issues were involved in printing an open structure beam/column system.
For a first pass, I developed this stackable, interlocking beam module.



It was a bear to create with Art of Illusion, but with Netfabb to clean up AoI's nasty STL files I managed, finally, to make it happen.
I did some preliminary partial prints to see what the issues were and discovered that I could either print one segment using the Sleinforge cooling option and wind up many hours later with a column segment that was festooned with strings of ABS ooze OR I could print several in the same time which required much less after-the-print cleanup.
That is when the trouble started. In the several days prior to this exercise I'd discovered that I could print the column section for the herringbone rack at 32 mm/sec without a serious degradation in print quality. This made printing these beam segments much less daunting a task.
When I began, however, I ran into a nasty concatenation of disasters that pretty much ruined my printing week. The first one occurred when I started a print and shortly after was called away for about 45 minutes to help my sister install a new wireless printer she'd bought.
When I returned to the lab I discovered that my Rapman 3 had partially reset whilst printing the raft for the segment leaving the extruder running. This unfortunate situation burned a small pit in my print table and buried my Kapton tape extruder head in a big blob of molten ABS. The Kapton tape extruder head had been happily running since October, I believe, without complaint. Being buried in ABS, however, pretty much put an end to it.
Fortunately, I'd bought several of BitsFromBytes new pre-made, silicone covered extruder heads.
After about an hour swapping out the old, ruined head for the new one I was printing again ...
almost. There was, of course, the usual running in of a new part. The new head was about 3 mm shorter than the old one, so I had to go through the whole adjusting for the new print height thingy,

complete with printing new trial rafts and the like.
That done, I went back to trying to generate a set of four of the beam segments with Skeinforge. Months ago, I'd downloaded the latest release, August's. Skeinforge these days is a busy, overcomplicated piece of software which has far too many bells and whistles hung off of it. In concept and execution, however, it is a brilliant piece of work.
That said, my column segment exercise pushed right though Skeinforge's performance envelope. It appears that when you create a BFB file with more than a million lines of gcode in it, that the August version of Skeinforge simply runs out of memory and blows up in the Export routine. After several false starts I finally got the most recent release of Skeinforge down and sort of working. The memory problem was gone, but in its place was a total buffet of new bugs relating to both the Multiply option and the Speed option. The two options appear to interact to produce some really bizarre gcode.
I'd hoped by now to have the new Netfabb gcode generator. Sadly, the Netfabb people have discovered that the developing of that app was a bit bigger task than they'd originally thought, so the release date got pushed back from 1 February to 1 March. That left me with either going back to my old copy of Skeinforge and limiting myself to the sorts of objects that it could process or sitting on my thumb for four weeks and hoping that Netfabb's revised release date didn't slip again.
There had been a prior bit of frustration back in November that pushed me into reviving my old Slice and Dice software app from the Tommelise project, I had actually got pretty far along with that before I was able to get past Skeinforge's nasty learning curve and was able to print acceptably using it. I went back and looked at that. Recently, Adrian confirmed that I was on the right track using a brute-force, pixel-oriented analysis instead of the more conventional geometric approach. Of course, my code is hideously slower than Adrian's, an understandable situation considering that he has been doing this kind of thing for his whole career. :-)
Still, when I sat down and thought about my situation, I realised that it is in the nature of who I am that if I'm given something I will inevitably test it to its limits. I always try to make things do things they weren't intended to. That's just the Scots-Irishman in me, I suppose. Given that situation, it's very dangerous in a way, to be dependent on equipment and methods that one can't get into and tinker with. When Skeinforge crashed on me it was suggested that I turn in a problem report on how I crashed it and wait for Enrique to fix it.
Enrique is fast at responding, but I am still too obsessive, now that I am trying to work with post-tensioned, printed structures to happily wait to see if he can easily fix the problem. Indeed, I find myself not much wanting to even report the problem.
So ... back to working on Slice and Dice.



The latest and greatest from Skeinforge

Monday, 8th February 2010 by Forrest Higgs

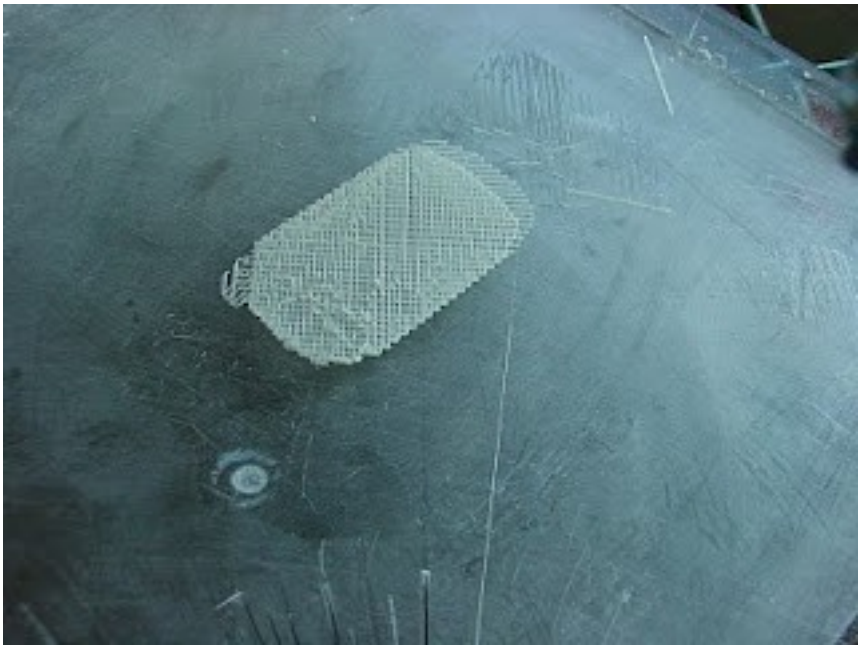
Skeinforge continues to exhibit bizarre behaviour when you use the multiply feature.
All that was changed between these two runs was the number of the same item being printed.
The one on the left printed 2 beam segments while the one on the right printed four.

It's rude, it's crude ...

Saturday, 13th February 2010 by Forrest Higgs

... it's socially unacceptable. Slice and Dice is my code, however, and when I want to try something new I can just dive into the code without having to ask anybody anything.

I've got to the point where I'm testing parts of the gcode generation on the Rapman.



Right now, I'm doing test prints for a fitted raft for a small, rectangular block. As you can see, I have a problem with the alignment between the two layers of the raft. The roughness of the second layer is happening because I haven't got around to arranging the print roads end-to-end yet. Sorting out the alignment comes first.
I've also tried printing out a few layers of the block itself. I've got a bit of trouble with the perimeter which I hope to get sorted out later today.
Having a printer sitting next to the PC I'm developing the Slice and Dice code on that can test code changes in a matter of a few minutes is a real blessing.

The raft {infill module} works

Saturday, 13th February 2010 by Forrest Higgs

Luckily, I was able to isolate the alignment problem to a loose y-axis belt pulley

You can see that the alignment is perfect this time. The missing bit of print path happened because that path was the first one printed and the printed filament broke loose for about a centimeter from the point of first contact.
Now, on to checking out the module that does the perimeter.

Got the perimeters going

Sunday, 14th February 2010 by Forrest Higgs

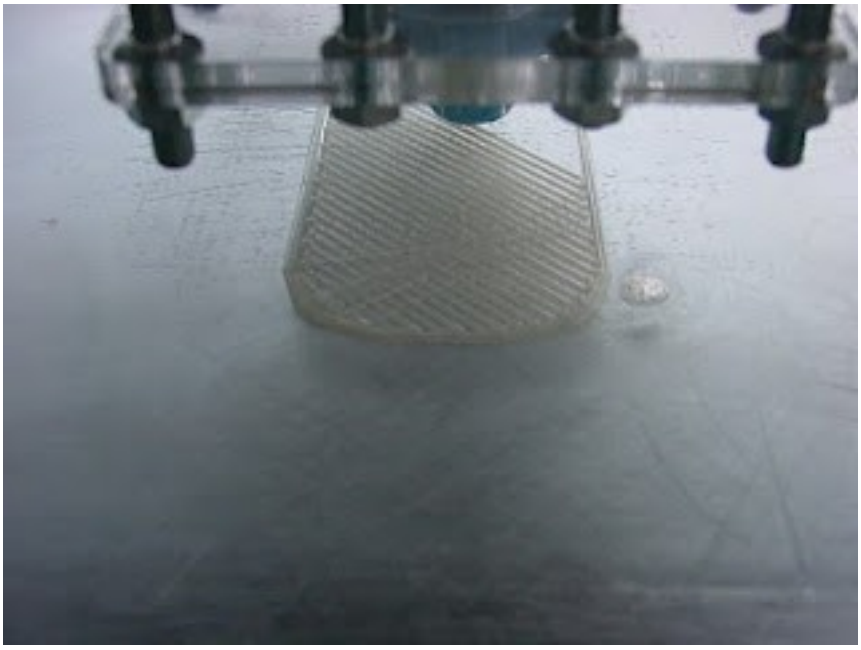
I wanted to do a direct translation of my XML format to gcode instead of breaking it into perimeters and infill as I had it before. The way I did it previously worked fine as long as you only wanted the possibility for one kind of infill. I'd like to a little better than that this time.

I've got a bit more work to do getting the ends of the infill to match a little better with the perimeter, but for now it is working pretty well. The brown bits are fried ABS that fell off the extruder head. I should have cleacked to see that it was clean before starting the print.
You can see a little misalignment of the perimeter of the raft. It turned out that the grub screw on the stepper side of the y-axis had shaken loose. Fifteen minutes of fishing around in the pocket in the acrylic assembly that it had fallen into with a hemostat recovered it. Once reinstalled, the misalignment disappeared. :-)

End to end

Thursday, 18th February 2010 by Forrest Higgs

I wrote a routine that serially checks paths and reverses them, if necessary, to minimize the travel distance between one path and the next. Basically, I look at the end points of each path, then print the first path and compare its trailing end with the end points of the next path. I then choose the nearest one. If it is the leading point then I print the path as is. If it is the trailing point, then I reverse the path and then print it.



As you might imagine, it greatly reduces the print time and makes for a much cleaner printed raft.

I now have all of the routines running more or less properly to print objects. That comes next.

Stumbling across the biopolymer zein

Sunday, 21st February 2010 by Forrest Higgs

It is hard to find time to hare after all of the potential research directions implicit in the Reprap undertaking. As a result of that, I've tried to concentrate on just a few areas dealing with the increase of the printed percentage of Reprap machines. Yesterday, however, I got off-track for a few hours.

For some reason I found myself looking for the Wikipedia entry for bioplastics and serendipitously keyed in biopolymers instead. There, I ran across a reference I hadn't seen before about a biopolymer called zein. The article referenced an incredibly detailed review of the material concentrating on its extraction written up a few years ago by John W Lawton at the USDA laboratory at Peoria in Illinois. Apparently, there is a large collection of zein related articles there which served as the basis of Lawton's extensive article. A few things about zein jumped out at me.

The biggest was that it was a protein, biopolymer which used corn gluten meal, a rather useless byproduct of the corn milling process. Corn gluten meal is a non-nutritive waste product that is typically used as a bulk agent for cattle fattening. It also has developed a small reputation as an "organic" herbicide for home gardens. This last is important because it means that 50 lb sacks of it can be had by ordinary people in one-off quantities at prices of under $1/lb. What that implies is that corn gluten meal is basically free and that what you are looking at is mostly the packaging, warehousing and transport cost when you purchase it.

The second was that reading over the Lawton article that the most successful extraction methods were achieved with kitchen chemicals {rubbing alcohol with a touch of lye} and chemistry. As well, zein has a long history of being used as a plastic, coating and, interestingly a fibre. This last I will talk about a bit later in this blogging.

Zein was used extensively before being displaced by petroleum-based substitutes during the 1950s and 1960s. Currently, ready-to-use zein is quite expensive {~$10-25/lb} not because of its intrinsic cost but rather because its very limited market, viz, food grade coatings for pills and food products commands such prices due to its highly regulated nature.

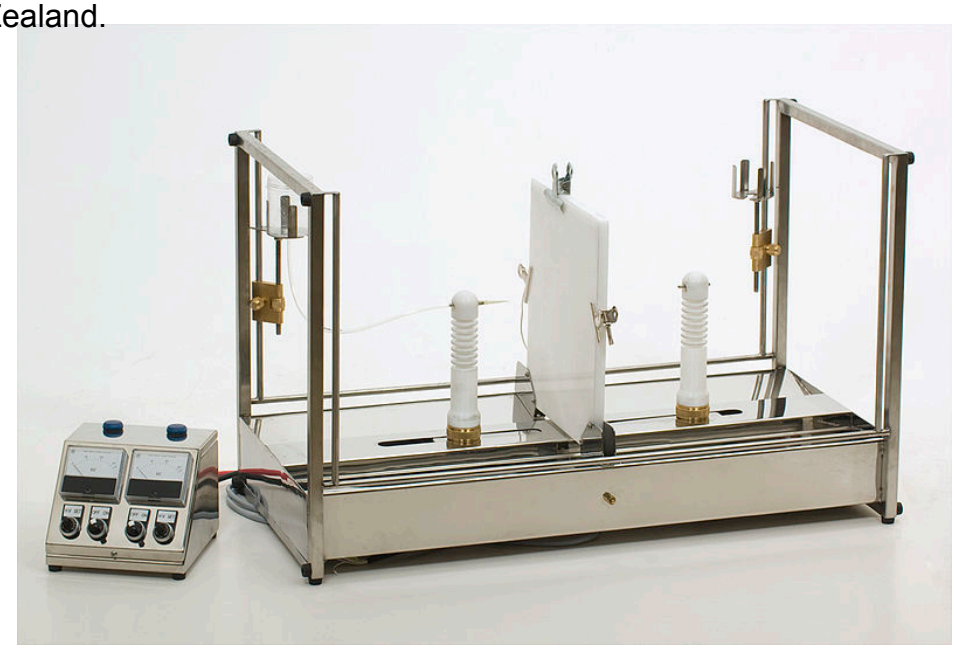
I vaguely remember from my childhood a wool substitute fiber called Vicara which emerged in the 1950s before being replaced by synthetic, petroleum-based fibers a decade or so later. What shool me about the Lawton article was that Vicara was made from zein. The rather high softening temperature of Vicara {~245 C}. If the melting temperature of zein is anywhere near this a whole range of products like coffee makers and the like become possible. Using PEEK instead of PTFE for thermal barriers in extruders, something that we are already beginning to do regularly, ought to let us get by with that.

Oddly, zein's glass transition temperature is very low {~40 C}. It may be that like PLA, zein has no real warping problem during printing compared to petroleum-based polymers.

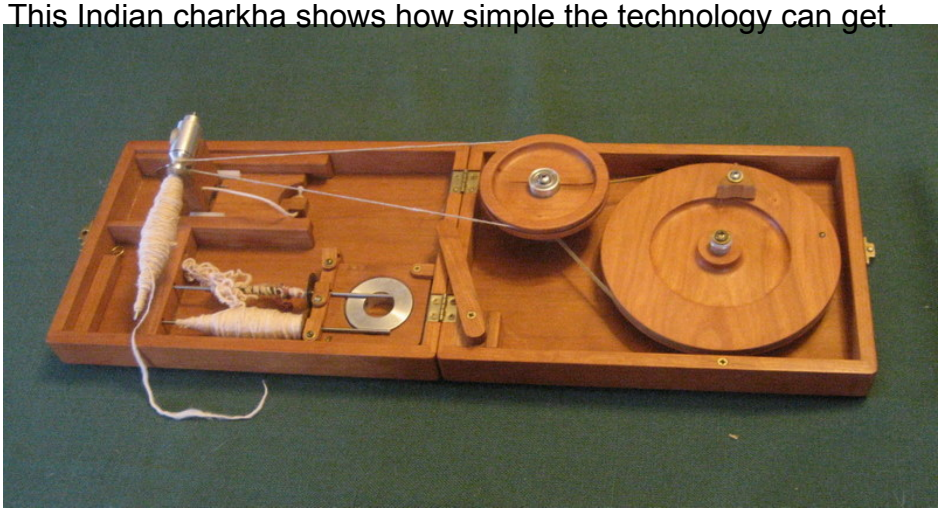
Heretofore, our use of plastic filament as feedstock for printing has caused practical problems when mapped against our desire for a recycling scheme for plastic. With recycling we have to grind the waste plastic, a problem that appears to have been solved recently. Afterwards, we have to extrude the filament. The machinery for doing this extrusion is relatively complicated.

Sadly, you can't just extrude filament. After it is extruded you have to quench, reheat and run it through a series of godet stations before you can spool it. This videoclip gives you a fair notion of the complexity and scale of the process. It seems to me that scaling down that sort of process is going to be an undertaking much more complicated than the Reprap printer

The notion that zein was made into fiber coupled with the application of electrospinning to zein fibres led me to quite a new idea. What would stop us from using spun fiber instead of monofilament in our extruders?
Electrospinning can be achieved with very simple apparatus such at this piece of lab equipment made in New Zealand.



Spinning the filament into thread of the proper diameter is an old, old technology easily scaled to our application. This Indian charkha shows how simple the technology can get.

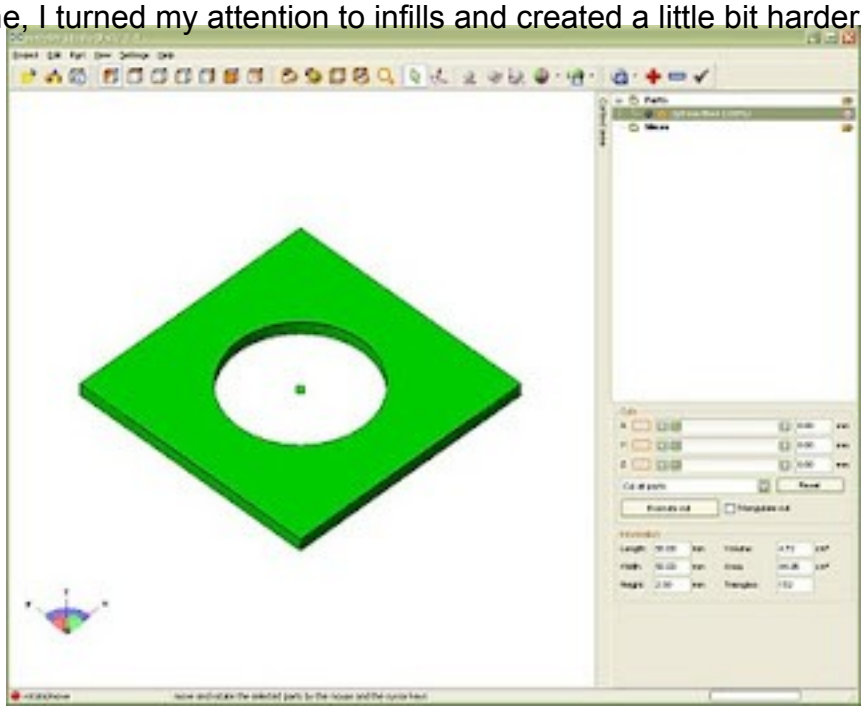


Filament is useful in our extrusion technology because it is stiff enough to be forced into our extruder barrel under pressure. Obviously, fibre thread can't. Keeping in mind, however, that our thread is polymer, we should be able to draw it through a heated die to consolidate the fibres into a single polymer mass.

Improving the infill routine

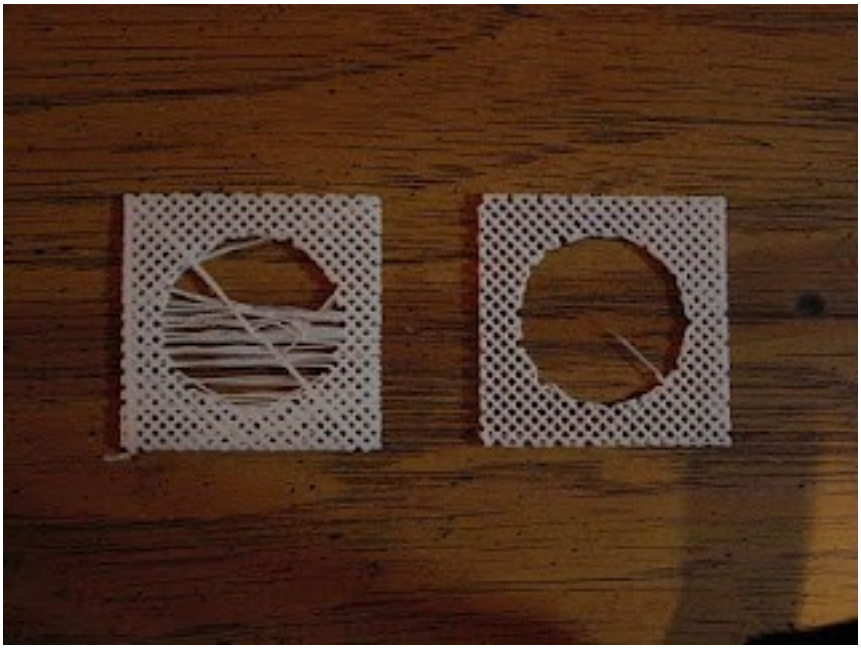
Sunday, 21st February 2010 by Forrest Higgs

I decided to separate the infill and perimeter routines. The main reason for doing that was that I found that with image processing of objects you tended to get a more detailed, almost fractal description of perimeters than you do with a geometric slice and dice routine like Skeinforge uses. What that means is that the apparent speed of the extruder head doing perimeters is much slower than it would be with Skeinforge simply because the perimeters are described with many more, very short (~0.2-0.3 mm) line segments.
I quickly found that that requires closer control of the extrusion rate than would be the case with Skeinforge. I made some headway on that problem on Saturday, but by no means am I finished working on the problem. It was obvious that infills, composed of line segments maybe a magnitude bigger, did not require the close attention that perimeters do. This led me to separating the two kinds of extrusion roads, a task that took most of Saturday.
Once that was done, I turned my attention to infills and created a little bit harder test exercise.



I then started doing test prints of just the infill. It quickly became obvious that my quick and dirty end-to-end approach to reducing the distance between extruded roads on the infill was not going to be enough. That simple method simply took a infill line segment and looked to the next on to see if the line could be reversed to reduce the time between extrusion. While that helped a lot there was still a lot of stringiness happening because the sweep method I used for identifying infill line segments didn't necessarily put them in proximate order.

Today, I decided to rewrite the end-to-end routine to put the infill line segments into as close a proximate order as possible. It was a very frustrating exercise that finally came right a few minutes ago.



The left infill print uses the end-to-end approach. You can see that the sweep method of identifying line segments leads to a lot of jumping back and forth between the sides of the hole in the block. My new proximate approach, seen on the right, reduces the need for long distance jumps to virtually nothing.

It really is an advantage, having a printer to test your code against.

Getting the perimeters right

Wednesday, 24th February 2010 by Forrest Higgs

When using a image processing approach to slicing and dicing when one extracts the perimeter of a curved boundary one is faced with an avalanche of very short (~0.1 - 0.1414 mm) print segments that do not easily transcribe into nice long, straight print roads like infills tend to. Trying to print 0.1 mm line segments presses the limits speed at which the Rapman MCU can take data off of the SD card and process it into stepper instructions.

I spent several days reasoning that if the conversion of vectors into bitmaps is rasterisation, then I ought to be doing the reverse process which is vectorisation. Vectorisation is a well-understood, if unpleasant algorithmic process. After struggling with the likes of the LZ78 algorithm for several days, I finally realised that what I needed to be doing was far simpler than that.

The approach I've finally taken was quite simple. My routines identify a series of pixels that form the boundary of an object slice. I simply begin by taking the first and third pixels in the path and forming a line thereby. Once I have this line I then determine the normal distance of the second pixel to that line. If it is less than a boundary that I can set I go on to test pixels one and four as a line and check the normal distances ob two and three against the the boundary value. I continue that process until one of the internal pixel's distance exceeds the boundary. I then step back one pixel and save that as the end points of a line segment in the compressed boundary. I then repeat the process using the end point of the last line segment in the boundary as the start point for the next until I've finished with the whole boundary description.

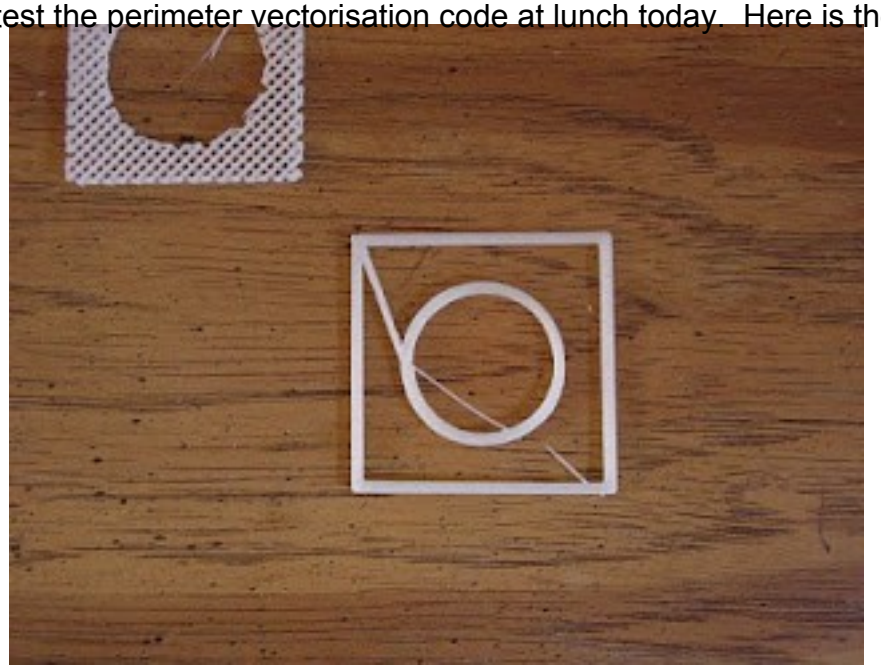
What I've described is a classic smoothing algorithm. The larger one sets the boundary the larger the degree of smoothing the boundary will be subjected to. I typically set it to 0.05 mm at the moment.

The code seems to handle my circular boundary problem adequately for the moment.

Perimeter vectorisation code done for now

Thursday, 25th February 2010 by Forrest Higgs

I was able to fully test the perimeter vectorisation code at lunch today. Here is the test print.



The infill, done earlier, is to the upper right of the perimeter print. I left the extruder on when the print moved from the outer perimeter to the inner so that the two parts would stick together. I've found that photographing white ABS print on a sanded acrylic print surface on the Rapman gives you a very low contrast pic which is a pain to read. Putting white ABS on a wood veneer background photographs very well, on the other hand.

I set the extruder to print a little thick so that it would stick to the print surface without a raft. The outer perimeter is 2 mm wide.

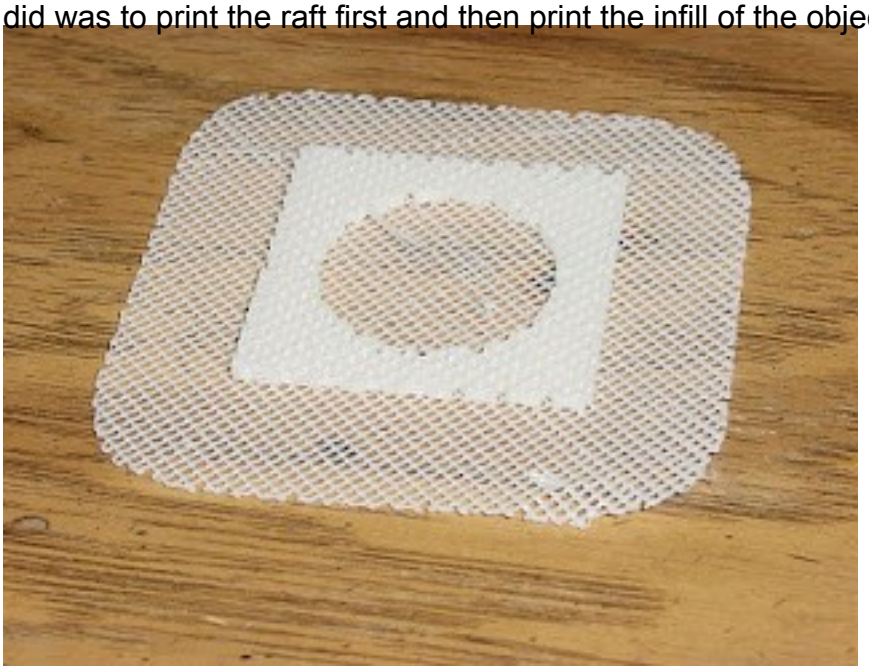
The vectorisation code produces pretty smooth, fast gcode. It is not perfect but it will do for now. The next step is to integrate the code modules which process the perimeter and infill.

Doing prints

Thursday, 25th February 2010 by Forrest Higgs

I had to go back and do some cleaning up in the raft generation coding. Mostly that entailed porting the latest infill code from the object infill back to the raft. Integrating the raft with the infill print was trivial. I keep the perimeter and infill xml in separate files for each layer, so doing a print is largely just a matter of shuffling files.

The first exercise I did was to print the raft first and then print the infill of the object on top of it.



The infill stack was 2.5 mm thick. I've never been able to understand why Skeinforge prints such a heavy raft so slowly when for the Rapman extruder printing at high speeds and high polymer flows is a snap. I printed a two layer, 0.25 mm/layer thick raft at 16 mm/sec. It stuck quite nicely. I then went on to print the infill on top of it a few minutes later with no problems.

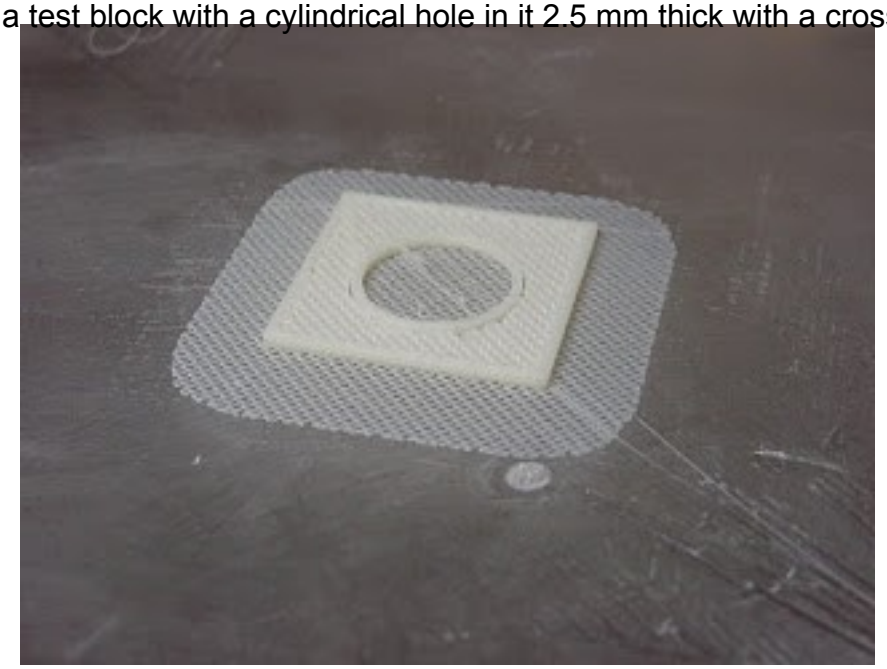
Just for fun I popped the perimeter for the object that I'd printed at lunch over the infill.

It was a bit of a tight fit, not being welded to the infill during the print, but it reassured me that I hadn't got the overall sizes wrong in the analysis.. I will see if I can integrate the perimeter into the object print in the morning before I do my day job.

First collated print

Friday, 26th February 2010 by Forrest Higgs

Slice and Dice processes perimeters, infill and the raft separately as gcode. Because of that, to create a print file of gcode I had to write a collation routine. That was fairly straightforward. Here is the first print of a test block with a cylindrical hole in it 2.5 mm thick with a cross-hatched infill.

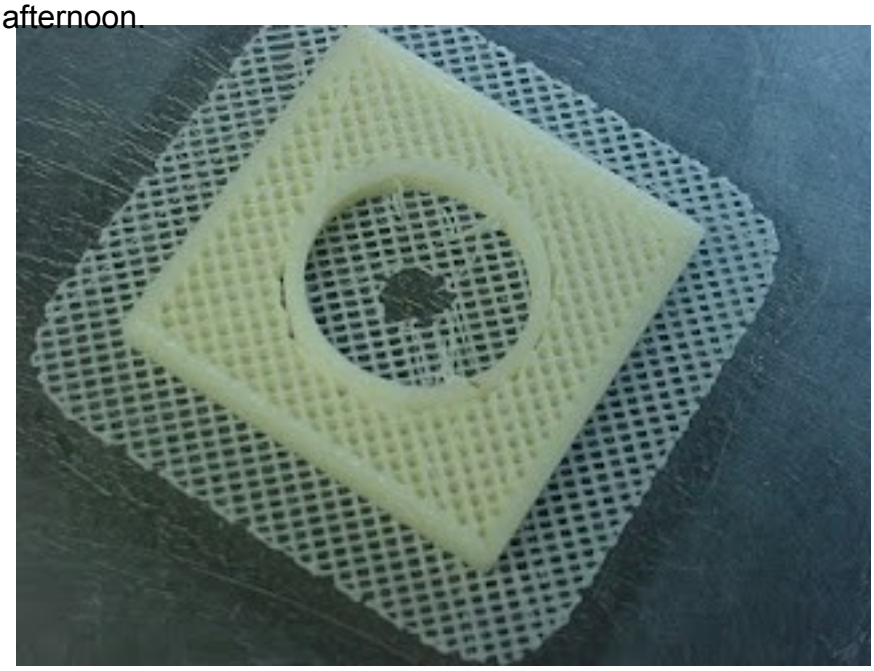


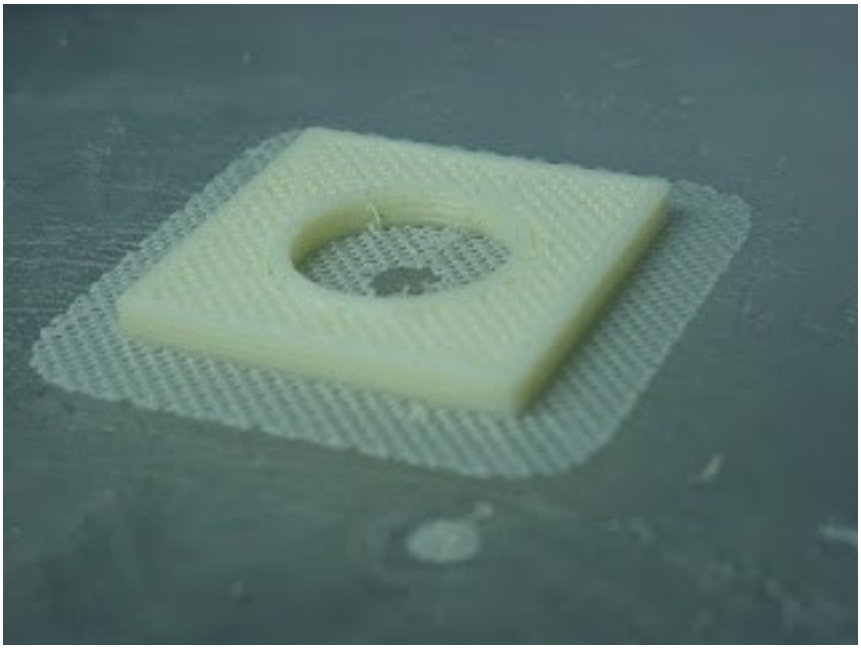
The next task will be to write the code to put solid tops and bottoms on objects. I'm processing a 5 mm thick version of this test block to give me enough layers to be able to do that without a lot of drama.

Sorting a few things out

Saturday, 27th February 2010 by Forrest Higgs

I had a bit of code cleaning to do to prepare for putting tops and bottoms on a infilled solid. Of course, it took longer than I expected. All the same, it's done for now. I was able to run a 5 mm high exercise this afternoon.





You can see with this skirt routine that when I specify a 10 mm skirt, you get a 10 mm skirt. Note the hole in the middle of the raft.

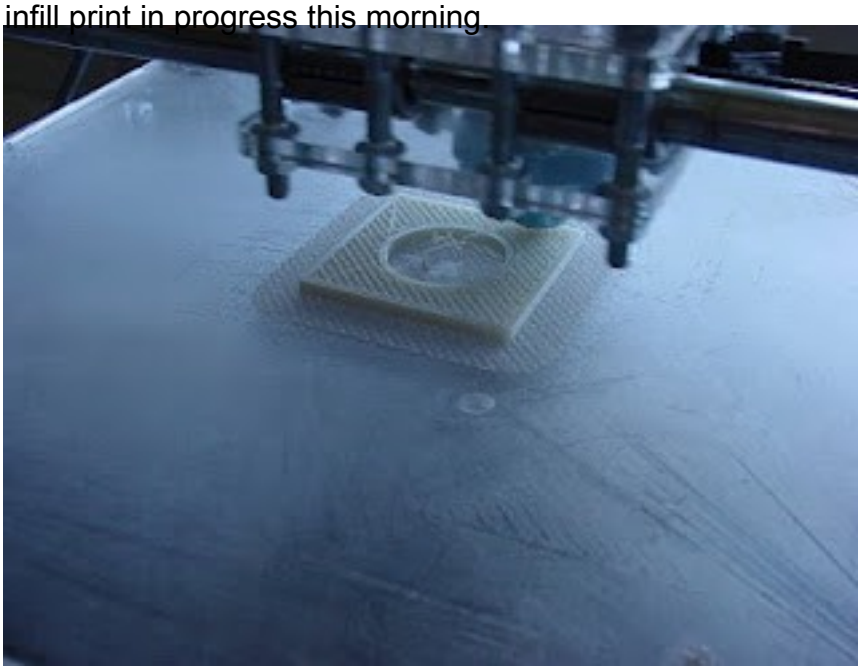
I've got some additional code to put in to convert extrusion/head speed rates into widths and depths of print roads automatically. That should cure the gaps between the infill and the perimeters. Right now, I am doing thumb sucks. I'm getting pretty close but not spot on.

Putting a lid on it

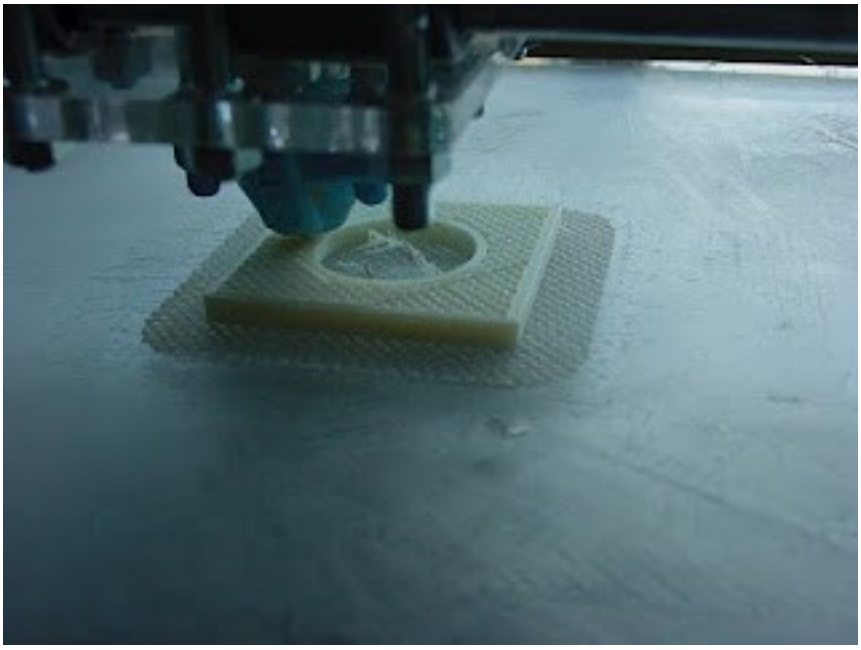
Sunday, 28th February 2010 by Forrest Higgs

I was able to shuffle around image files and, starting from the top of the slices {a "sky" slice that is taken above the print object} find which parts of the print object need lids {closed infills}. Here you can see an earlier test print showing the cross-hatched infill.

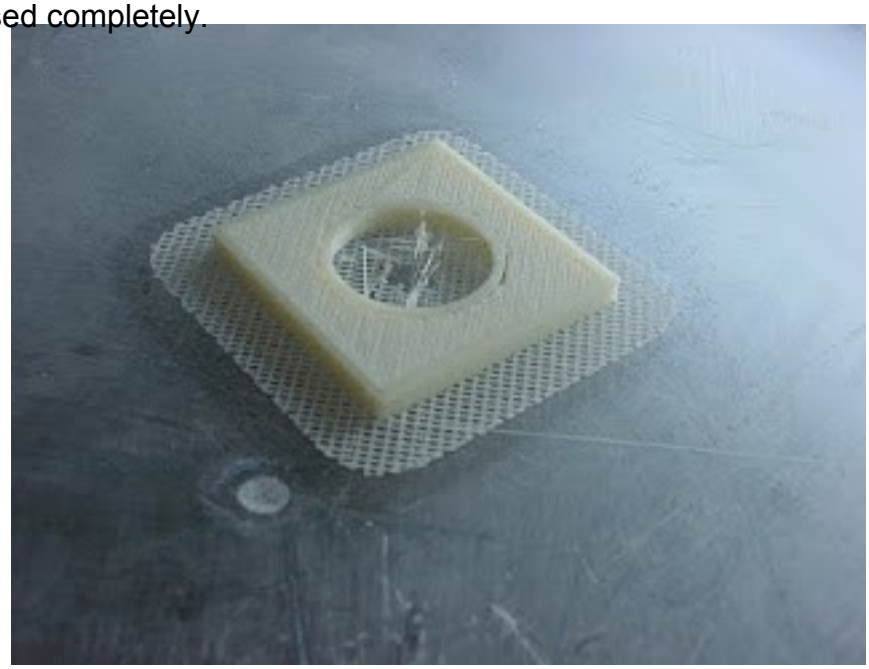
This pic shows the infill print in progress this morning.



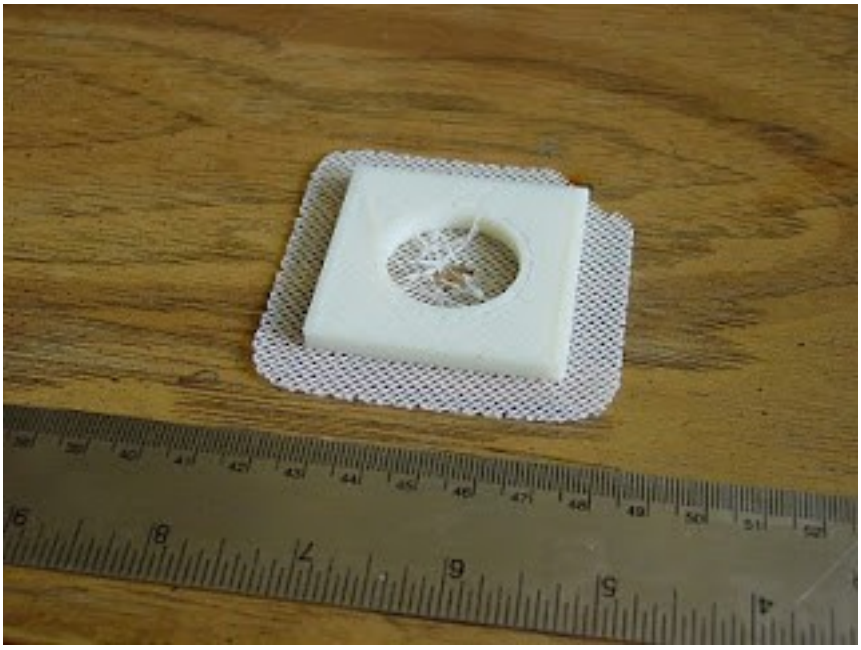
The lid(s) on a print {depending on the complexity of the object there may be many} are taken in several layers by smoothly closing the infill down in several layers till you get a complete seal for the lid itself. Here you see the first layer of the closing of the infill being printed.



Here the lid is closed completely.



Note that I still have some work to do to get the infills and lid to mate properly with the perimeter
Finally, you can get a better look at the print quality of the object removed from the printer.

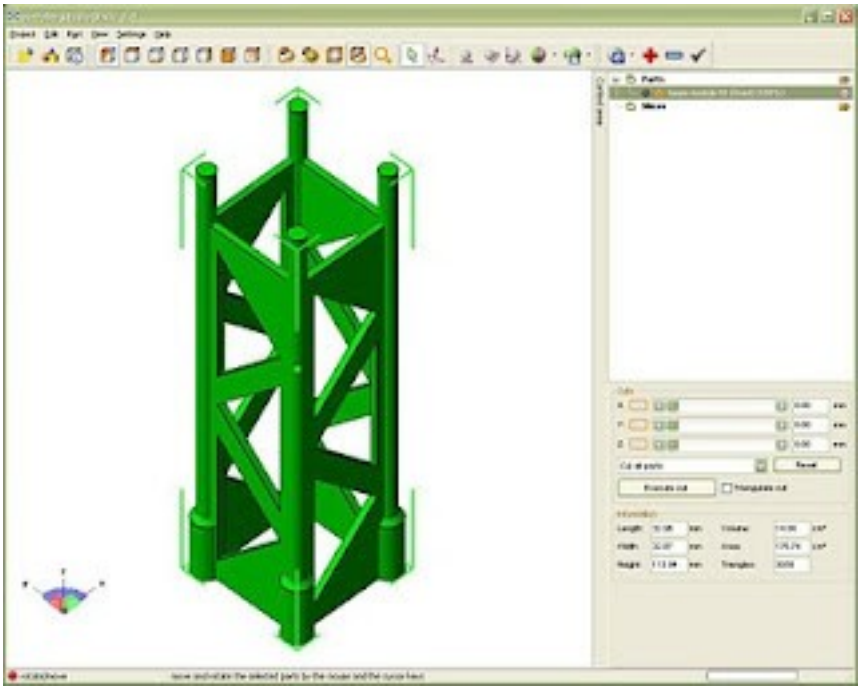


Slice and Dice, as a personal research tool, is basically operational now.

Slice and Dice 0.1

Sunday, 28th February 2010 by Forrest Higgs

Last month, Adrian incorporated some ideas for using pixel maps that I'd used years ago with Tommelise 1.0 into what I understand is his most current version of the Reprap host software. At the time I was happy that my approach had proved to be of use to him. I was, however, happily using Skeinforge with my Rapman printer at that time and didn't think too much about it. Enrique's Skeinforge is a brilliant piece of software for non-mainstream Reprap printers like my Rapman. As is my wont, however, I tend to test the envelope with just about any piece of hardware or software that I undertake to use. The problem with testing the envelope is that too often you push right through it. That happened to me when I started trying to print light, open structures like this.



Enrique invariably handles error notifications within a day or two. I frankly don't know how he manages. It must require superhuman capabilities for one person to manage that for so many users. What I was trying to do, however, wasn't eliciting what could be termed an error as such. The individual slices for what I wanted to print were quite small. As a practical matter, what that means is that is that the layers tend to overheat from the extruder tip being over them too long on average. Let that go on long enough and you wind up with a print that looks like a melted candle. Skeinforge has a "cooling option". What it does is time how long you are spending printing a layer and if it falls below your set point it orbits the extruder tip around the print till it has a chance to cool off. Thanks to the tendency of extruder tips to dribble ever so slightly, if you use this option your print becomes wrapped in what looks like plastic cotton candy. It is a bear to clean parts wrapped up like this. The other way that I developed when printing small pinion gears was simply to print 6-8 of the same thing at once, something that Skeinforge lets you do very easily. While that is very handy if you need a lot of a thing, if you're just trying to develop a part you take 6-8 times longer to print your part and find the errors in the design. With ABS costing $10/lb using Skeinforge's multiply option is not a very practical as a cooling method.

I finally came to the conclusion that Skeinforge, as it presently is written, wasn't going to take me where I wanted to go. I had ordered a copy of the new Netfabb programme to do the same thing,

then learned that their product rollout was going to be at least a month late {1 March}.
With several weeks of time on my hands, I decided to dust off my old Slice and Dice software and see if I could knock it into shape to do what I wanted. Three weeks later, it's running.

Let me tell you from the beginning that Slice and Dice 0.1 is most definitely NOT an alternative to Skeinforge, Netfabb or the Reprap Host software. If you are thinking that, forget it. Now I'm not trying to keep anything proprietary. If you want the source, I will give it to you under a BSD-type license which lets you do pretty much anything you want with it. Indeed, I've handed out several copies already to a few reprappers who wanted a look at this or that bit of the code.
Before you decide you want it anyway, however, here are some facts about it that you should be aware of.

- It is written in Visual Basic .NET {the free Express 2008 edition}
- It is Windows specific
- It is NOT beta code. I'd have a hard time calling it alpha code at this point.
- It is set up to handle only one print object at a time.
- It keeps multiple bitmap images {BMP} of each slice. These are 250x250 mm with a 0.1 mm resolution {figure 22.5 megabytes per image. All the processing is done on images that you can look at. What that means is that a decent design is going to generate many, many gigabytes of images. I don't care since I have a 1 Terabyte external disk committed to the programme.
- I will NOT provide support for Slice and Dice 0.1. If you take it you're on your own. I admire what Enrique is doing but do not envy his situation and most definitely do NOT mean to in any way compete with Skeinforge and Enrique or Reprap Host and Adrian or the good Germans who are fielding Netfabb. I'm not young and life's too, too short.

I've put Slice and Dice 0.1 together for one specific reason. That is to let me try out off-the-wall 3D printing ideas that often times will go beyond the capabilities of ordinary software. Most importantly, its a piece of software that I've written and can easily dive into myself and fiddle with to make it do new things without having to report problems to and make requests from others. I figure it will save me a lot of development time by shortening the development cycle. It for sure will keep me from making a pest of myself making requests for changes in their code to Adrian

and Enrique. That should be good for everybody's nerves.

In spite of all my warnings...

Monday, 1st March 2010 by Forrest Higgs

I'm getting more than a few requests for the code. Please reread the previous blog entry. If you still want the code for whatever reason, you can download it at...
http://3DReplicators.com/Slice and Dice/Slice and Dice 010310.zip
The code is subject to this modified BSD license.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by Forrest S. Higgs in conjunction with the open source Reprap Project

4. Neither the name of the Reprap Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

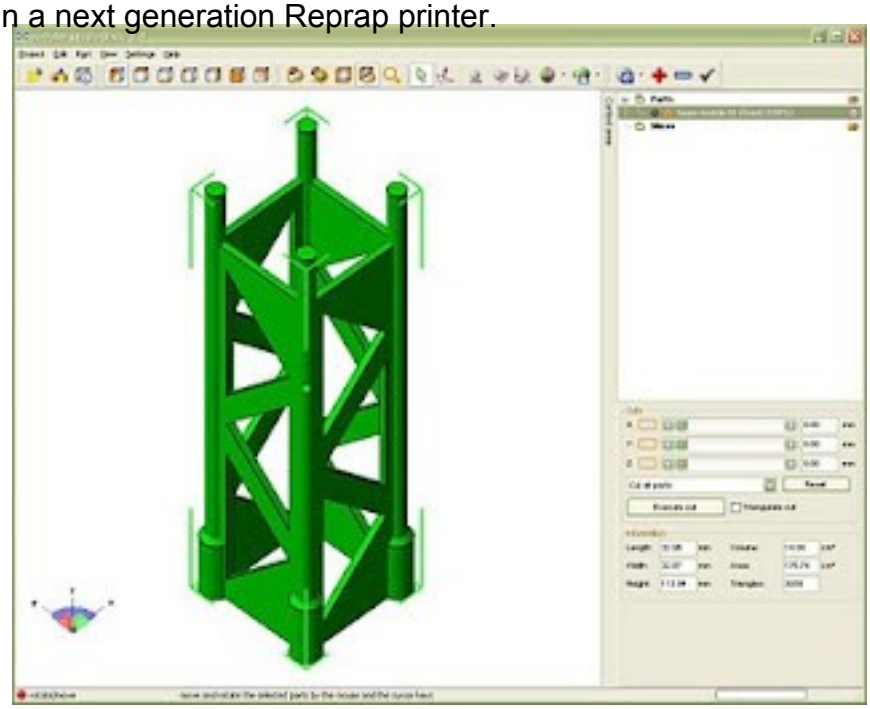
THIS SOFTWARE IS PROVIDED BY FORREST HIGGS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REPRAP PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

If you want to modify or upgrade the Slice and Dice app, feel free. You can either let me distribute your upgrades if they fit in with what I am trying to achieve or distribute them yourself. Just give me credit for where you got the original code, okay?

Printing light structures

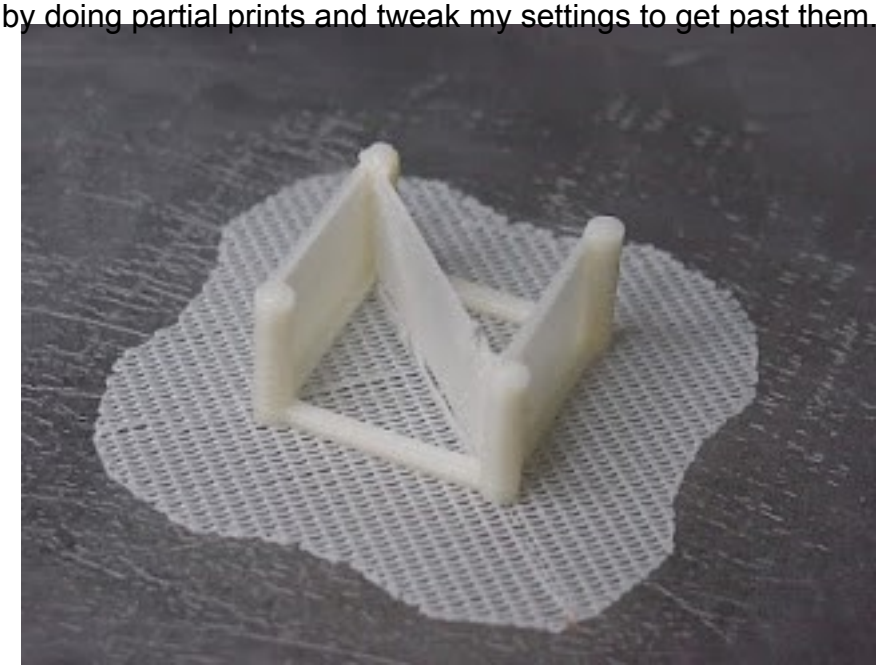
Tuesday, 2nd March 2010 by Forrest Higgs

A month ago, I designed a light frame structure which I could post tension and use to replace much of the steel in a next generation Reprap printer.



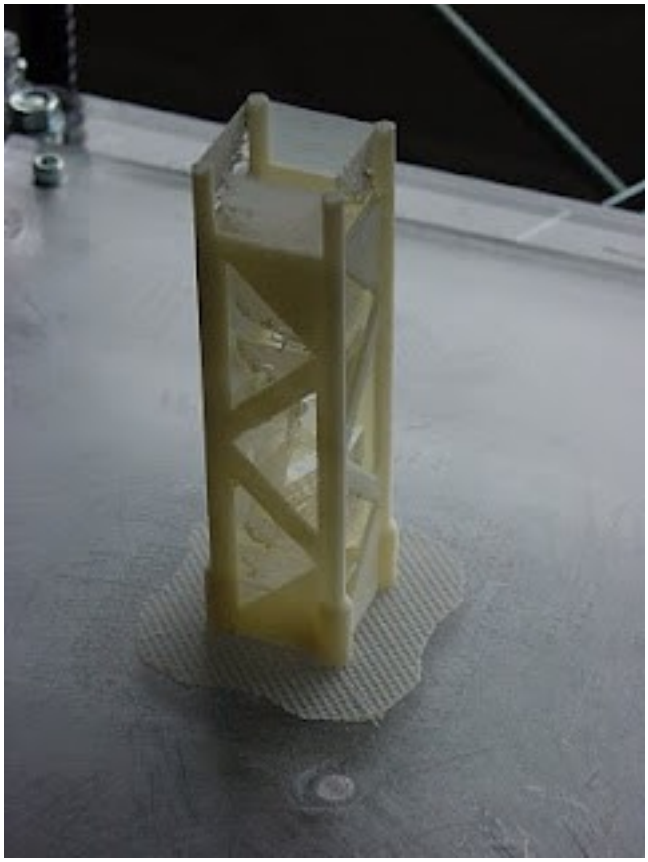
Processing it into gcode with Skeinforge proved to be impossible. That led me into a meandering sidetrack which had me pushing my old Slice and Dice code into working order after several years of neglect after Netfabb ran late with the release date for their STL to gcode processing module. Last weekend, I got Slice and Dice working acceptably.

After work yesterday afternoon, I decided to see if I could make my Slice and Dice alpha code successfully generate gcode for a print of that same light structure. There were a few hiccoughs. Fortunately, Slice and Dice is purposely designed to let me deal with difficult prints. I was able to identify problems by doing partial prints and tweak my settings to get past them.

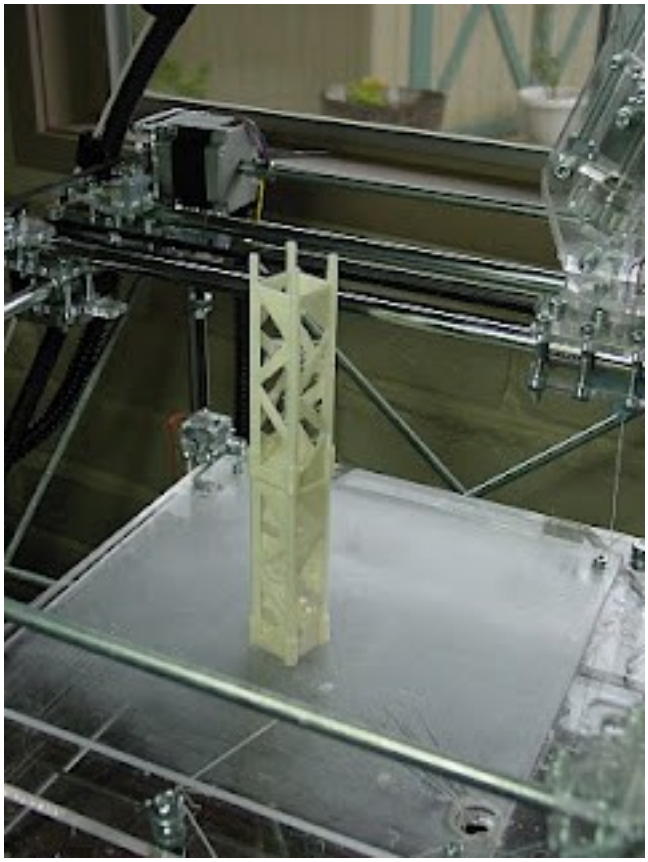


When I got up this morning I finished processing the light structure STL and set my Rapman

printer to work on it. Being a stand-alone 3D printer, it happily worked while I got on with my day job. Roughly 90-100 minutes later, I had a printed module.



I was quite chuffed with my design. Clearing the female connectors on the bottom with a 5/32" bit took a matter of a few seconds and let me mate one of my test prints of the top of the module with the whole printed module. I immediately started printing up a second module. It finished up a few moments ago.



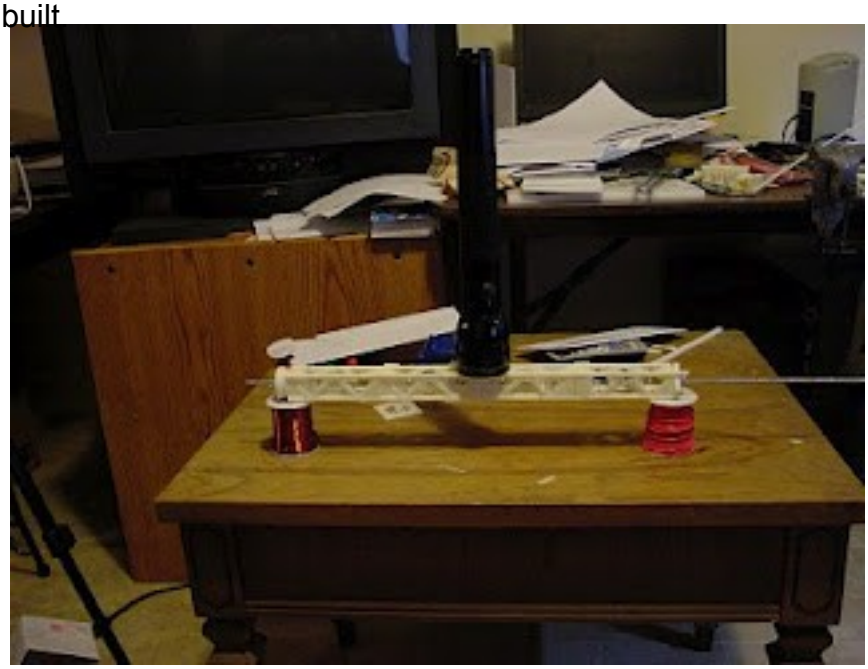
I think it's pretty obvious where this is going. I did a short clip of the print underway so you can get an idea of how fast it proceeds.

I am printing a third module now and plan on printing five over the course of the day. This evening after work, I plan on designing end caps for the assembly so that I can post tension it and then get an idea of how much deflection it exhibits under load for a 500 mm beam.
The lessons I've learned from all this is that if you design light and print vertically, viz, keep the cross-section on the xy print table small, you are going to get what are basically no-warp parts without going through the drama of heated print beds or turkey bags. it's worth thinking about.

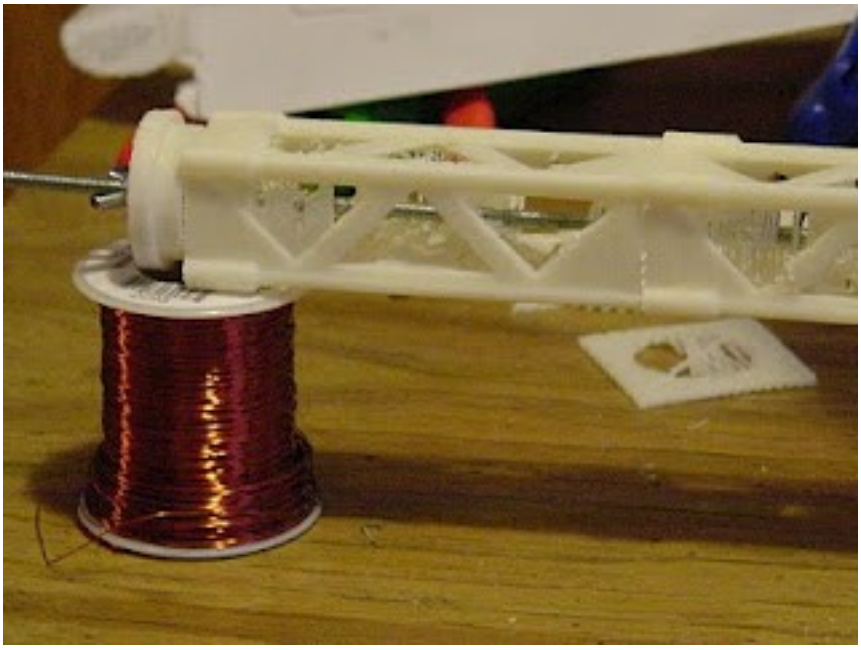
Some crude load tests

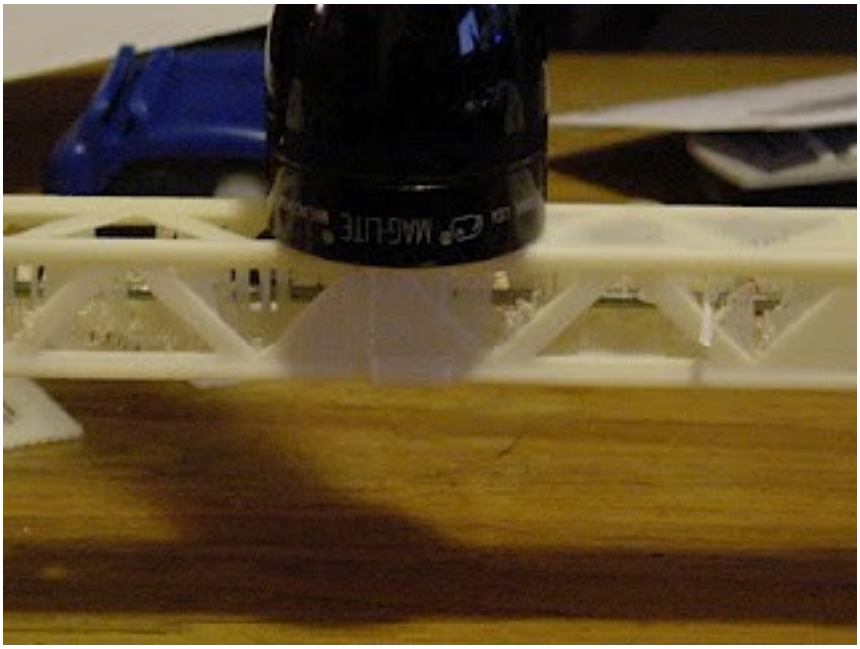
Tuesday, 2nd March 2010 by Forrest Higgs

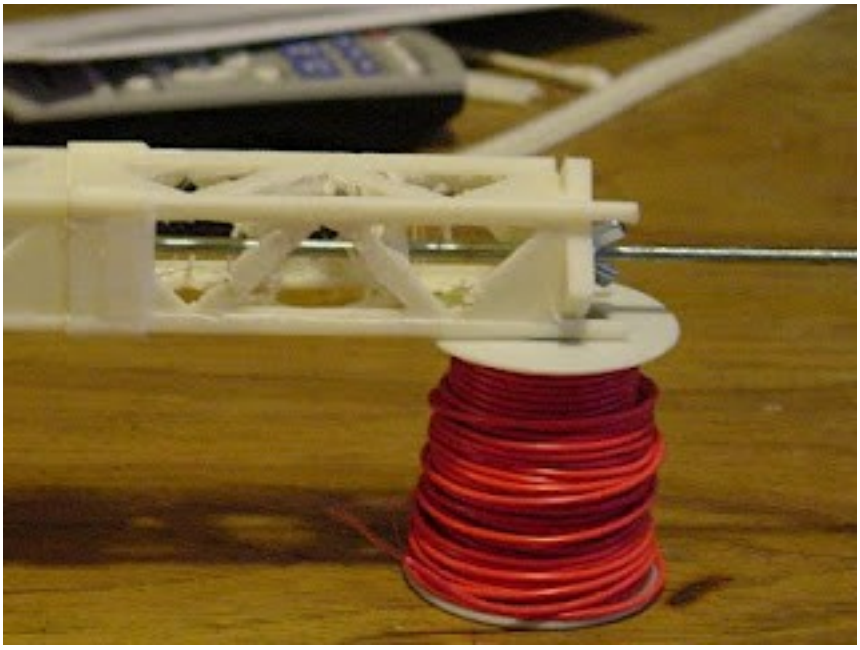
Bogdan began to question me about just how much stress these plastic beam modules could deal with. When he caught on that I was post-tensioning the beams he had no trouble accepting that they'd likely do the job. Just for fun, though, I went ahead and printed up four modules and assembled them into a 400 mm beam, post tensioned them and ran them through their paces.
I was going to print special end plates, but it's getting late and my day job will be rather intense tomorrow, so I used one of the end plates from the last post-tensioning exercise with the herringbone rack column. For the other end plate I used a strongly designed washer that I printed for another purpose some time ago.
First, I did the same loading as last time, but this time the beam is 30 mm deep instead of 20 but much less heavily built.



No deflection to speak of. No problem. That Maglite weighs two pounds and is point loaded at the centre of the beam.

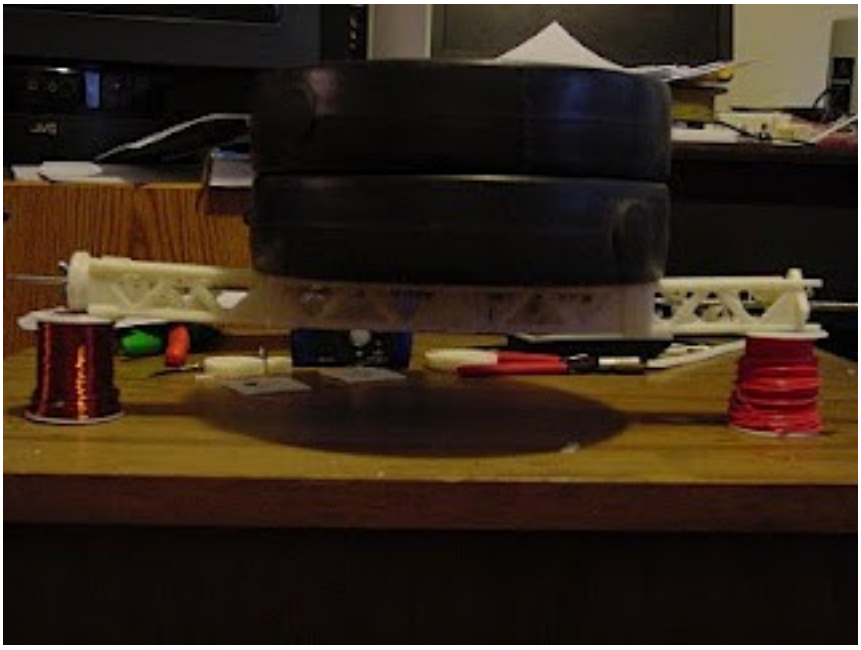



MAGLITE



That was nice, but then I decided to get serious about loading, so I broke out one of my ten pound bar bell weights and partially distributed loaded the beam.

A couple of millimetres deflection, but again not much. I then threw another ten pound bar bell weight on for good measure.



Now you start to see some serious deflection. Interestingly, the deflection is not because the beam is failing but rather because the lightly designed end plate on the right is beginning to fail. I replaced that plate with a heavily designed herringbone gear and repeated the exercise.

The deflection is better, but I began to worry that the shear stress at the ends of the beam, which I had planned to handle with purpose designed end plates is ill-applied to the light frame beam itself.
For a final exercise, I point loaded one of the 10 lb weights at the centre of the beam by blocking it with a couple of pieces of scrap ABS.

Guys, post-tensioning is a wonderful technology and very well suited to the scale that we are working at. Several people have suggested using ABS filament for the tension member. I think the thing to remember is that conventional wisdom in designing post-tensioned beams is that the tension cable or rod needs to be a LOT stronger in tension that the material it is paired with. In real life, you see post tensioning done with steel rod or cable and concrete. Concrete has effectively NO tensile strength. I personally don't think using ABS filament is going to give satisfactory results as a tension element in such beams. I use thin studding, lock washers and wing nuts because that is easy to work with and beams can easily be put in compression by hand. Anyhow, I think that if we want to we can get a lot of the structural steel out of our next generation Reprap machines and greatly improve the printed fraction using this approach.

Making Slice and Dice more practical

Tuesday, 9th March 2010 by Forrest Higgs

Continued shipping delays with Netfabb's shipping have given me more time to work on my Slice and Dice alternative app than I had expected. I was able to use Slice and Dice very effectively to turn my light frame STL into printable gcode as you will have seen in my last few blog entries. I then made two decisions which have put me in present limbo.

The first decision had to do with my desire to start working on the problems implicit in printing with polypropylene. Bogdan had developed a lot of know how with polypropylene and characterised it as being a lot like HDPE. Owning some 80 lbs of surplus polypropylene, I decided to have a go. My first use of it was to try to print my light frame structure. After a few days of that I decided that polypropylene, at least the kind I have, is too rubbery for that kind of application. I then decided to see if I could print something immediately useful, like a bowl, and see if polypropylene was going to be as dishwasher safe as advertised.

I went to Thingiverse and downloaded Vik's beautiful tea bowl. I could have easily designed a bowl, but really liked the look of Vik's. Mind, it wasn't going to look as good in polypropylene as it does in PLA, but never mind. I could throw mine in the dishwasher ... in theory at least.

Vik's tea bowl was a much larger object than my light frame. Hammering the STL through Slice and Dice revealed that the flood routine, which delineates printed and non-printed zones for a slice, did not scale well at all. Replacement of my flood code with a Windows API call for flooding sorted that problem out. Slices that were flooding properly in 8 minutes are now flooding in about 7 seconds.

I am currently working on the fill routine which takes a flooded image and lays out the perimeter and infill roads on the printed areas. That routine, too, isn't scaling well and is taking about 8-9 minutes to process per slice. That means that Vik's bowl will take about 40 hours to lay out the 300-odd slices required for printing. The development of the perimeter roads is taking 99% of the processing time. I am going to have to come up with a new way of doing that process. I have some ideas.

One other little thing that I've found is that the massive amounts of disk space that I am currently using for bitmap image storage can be reduced by more than 98% by the simple expedient of zipping the BMP files using the Windows API and unzipping them only as needed.

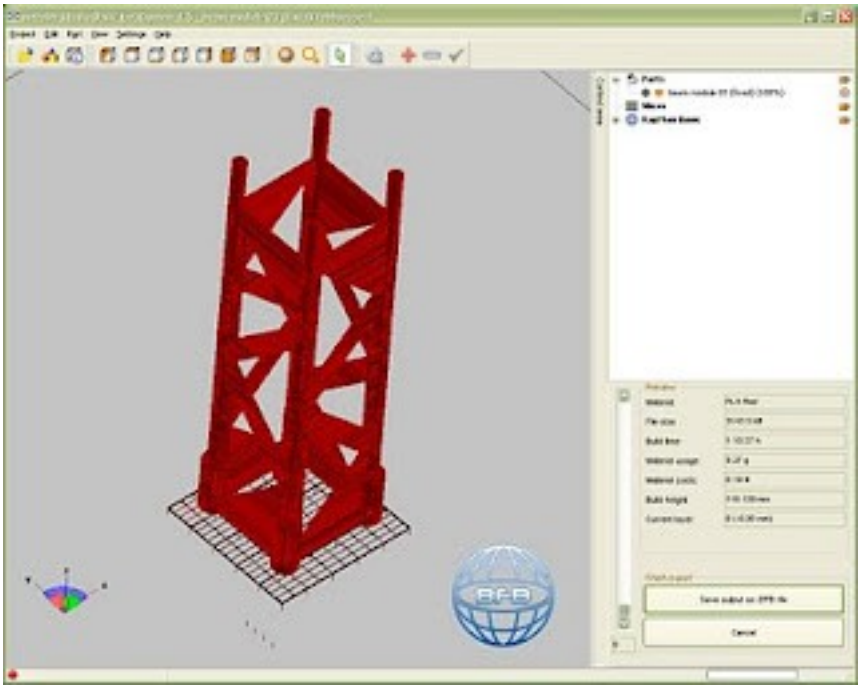
Slowly, I make progress while Netfabb hangs fire.

Speak of the Devil...

Tuesday, 9th March 2010 by Forrest Higgs

...and he doth appear. I'd no sooner blogged about my work about Slice and Dice when I got an email from Alexander at Netfabb that said that I could download a temporary copy of Netfabb Basic for Rapman until my professional add-on was ready.
I downloaded it and cranked in the same tea bowl that Vik designed that I am using with Slice and Dice, calibrated it for PLA Red just for fun, selected "5" and tried "Calculate Toolpath".

It promptly crashed calculating layer 34. I tried the light frame with the same calibrations and it generated a gcode file properly, apparently.

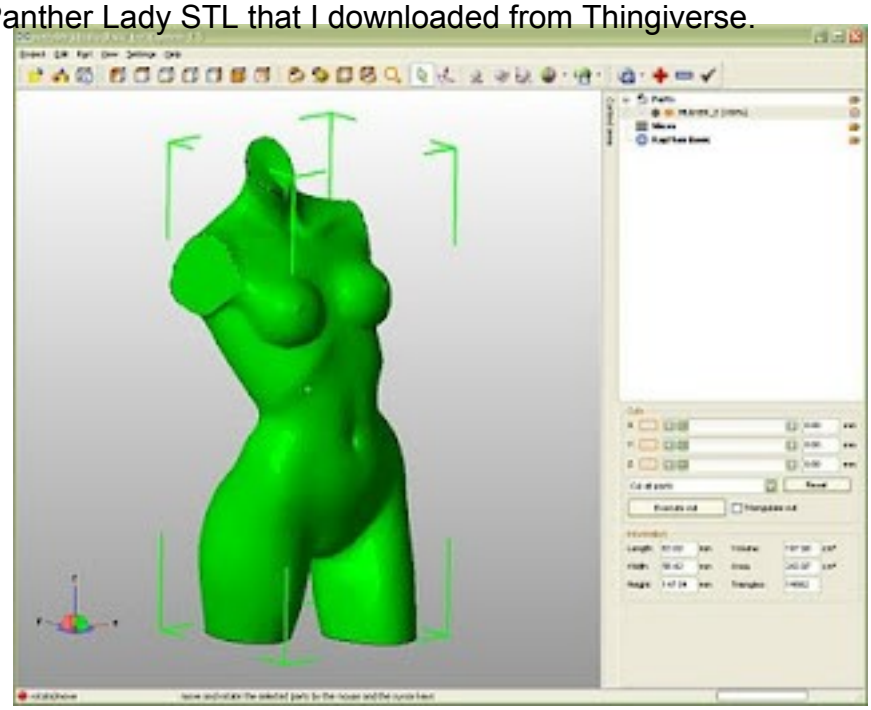


I'm set up to print polypropylene on HDPE so I was going to put things off till my new 3/8 inch acrylic print surfaces arrived. Again, speak of the Devil, I found that the courier service had left them on the front porch.
I need to log some billable hours and there is a big pile of consulting work that wants doing for a good client, so I probably will get back to this Saturday at the earliest. :-(
You guys at Netfabb can get the tea bowl that crashed your system at the link to Thingiverse at the beginning of this blog entry. It looks as likely to give you as much trouble with Netfabb Basic for Rapman as it is giving me with Slice and Dice.

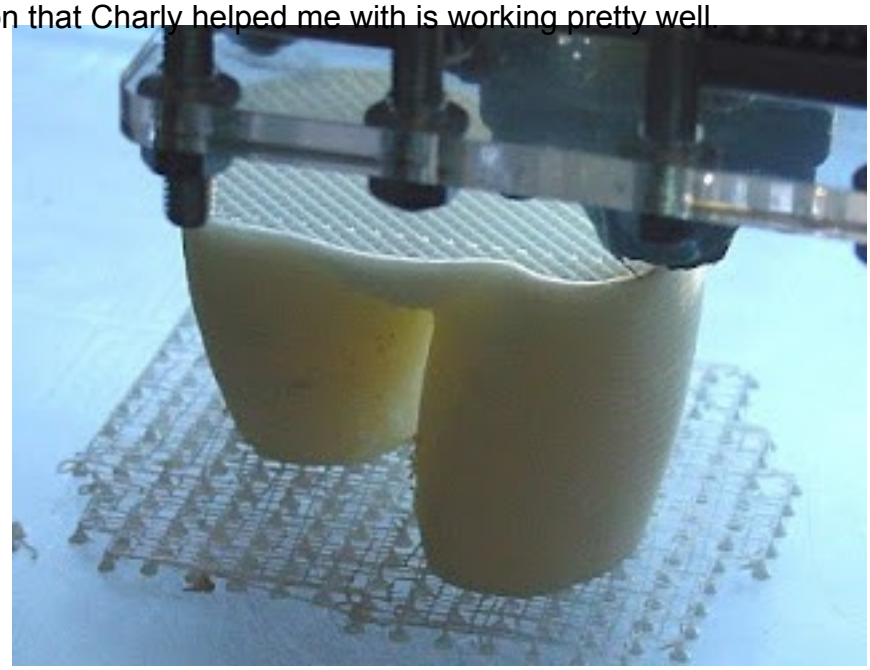
Printing the naked lady

Sunday, 14th March 2010 by Forrest Higgs

A few days ago, I got notice from Alexander at Netfabb that I could download a copy of Netfabb
This weekend I've been working with the new Netfabb for Rapman Basic beta. I'd ordered the Pro edition with upgrade to generate gcode for my Rapman 3.0, but it appears that another month or two is needed for that to be ready, so Alexander kindly gave me a copy of Basic to play with.
This weekend I refurbished my Rapman for ABS with a new 9.5 mm acrylic print bed so that I could print ABS and PLA again.
The big problem I had with Basic is that it is set up for a 0.5 mm extruder orifice while I am using a 0.3 mm. Charly at Netfabb kindly showed me how to sort that out this morning, so now I am printing the Pink Panther Lady STL that I downloaded from Thingiverse.

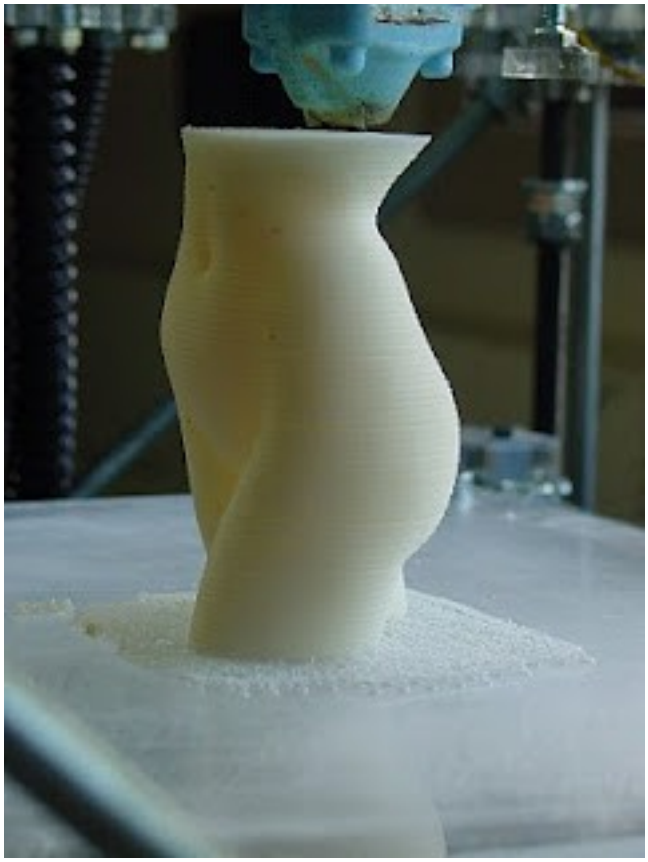


The new calibration that Charly helped me with is working pretty well.

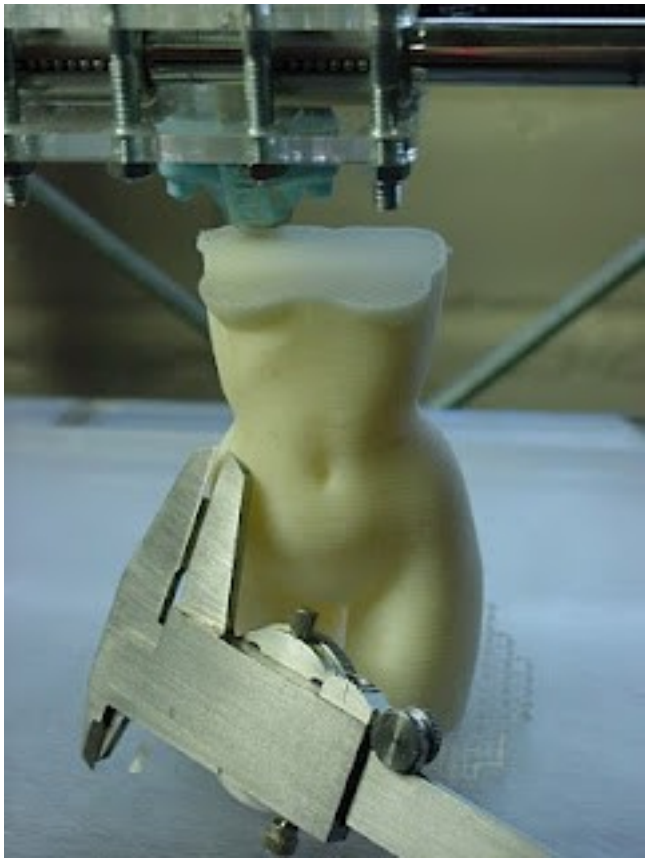


There is a little ripple offset in the print layers that I estimnate at 0.1-0.2 mm that I'm guessing is caused by some sort of truncation error either in the Basic app or in the Rapman firmware. I hope to get to the bottom of that problem in coming days. Other than that, the print quality of the Basic beta app is brilliant.

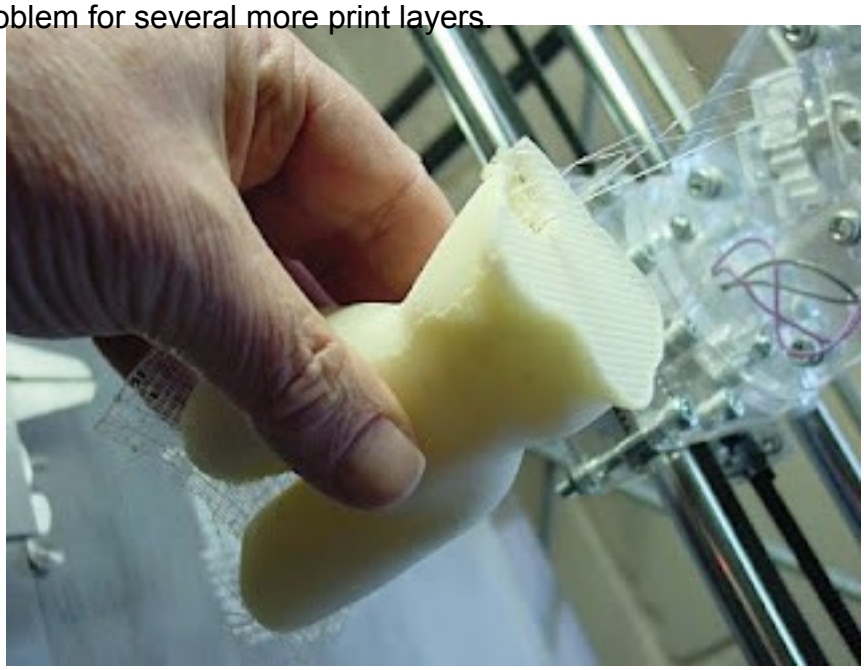
The new raft technique that Netfabb has confected is also brilliant. They basically print a matrix of candy kiss shaped pillars and then print a flexible net over them that is connected only at the edges of the matrix. That enables the print, the naked lady in this case, to shrink safely without warping. It is a REALLY elegant solution to the perennial warping problem that we've all faced for years.



I noticed that the torso started breaking free from the raft webbing when the print was approaching the 100 mm z axis milestone. The print was working on the connection between the right arm and the shoulder and this started catching when the perimeter was being printed and slowly broke the thighs away from the webbing on top of the pillars. I put a caliper against the torso to counteract the movement.



That solved the problem for several more print layers.



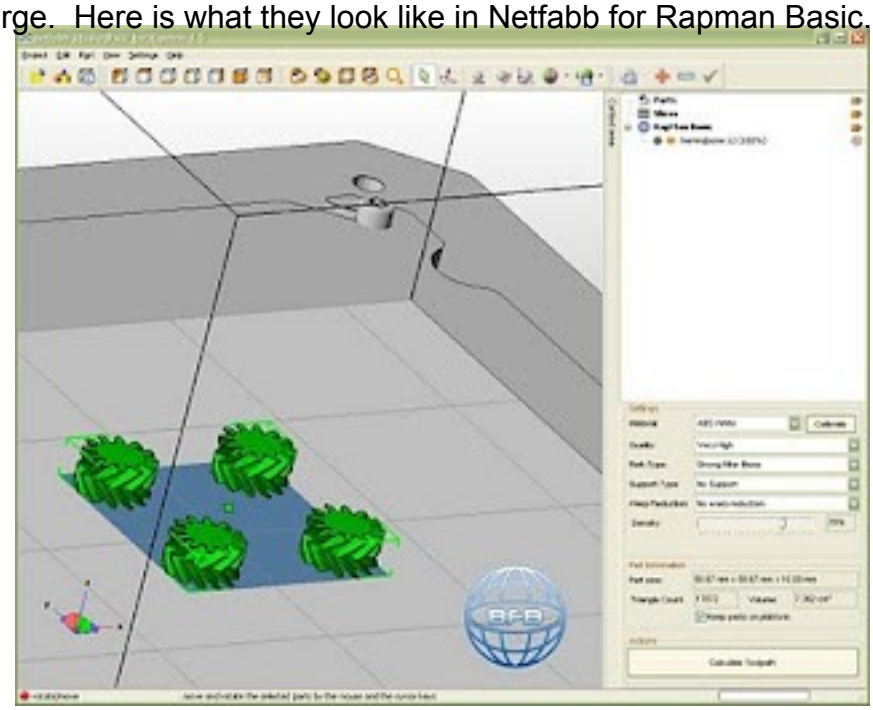
The torso finally snagged on the print head and moved at a diagonal burying the hot extruder head into the infill and ruining the print. The failure occurred precisely at 100 mm, raft not included.

Aside from the slight layer juddering {expansion and contraction}mentioned before, the print quality was quite high.

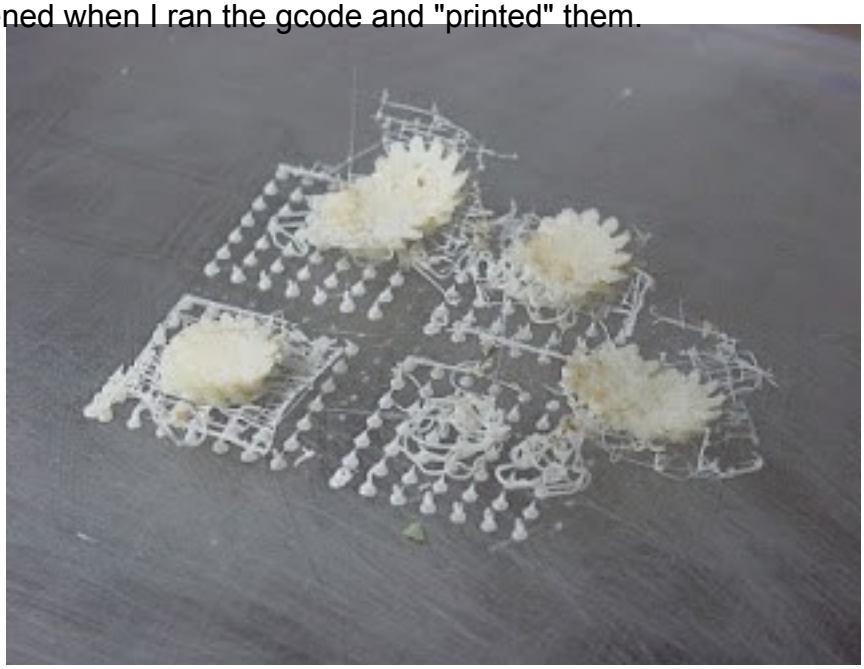
Netfabb for Rapman Basic ... FAIL

Sunday, 14th March 2010 by Forrest Higgs

Okay, now I'm not happy any more. I designed and used Netfabb to clean up a set of 4, 12-toothed herringbone pinion gears that I have successfully printed many times on the August 2009 release of Skeinforge. Here is what they look like in Netfabb for Rapman Basic.



Here's what happened when I ran the gcode and "printed" them.



I think it's pretty clear that the raft technology that Netfabb has designed does not work for small, detailed objects.

Back to Plan B

Tuesday, 16th March 2010 by Forrest Higgs

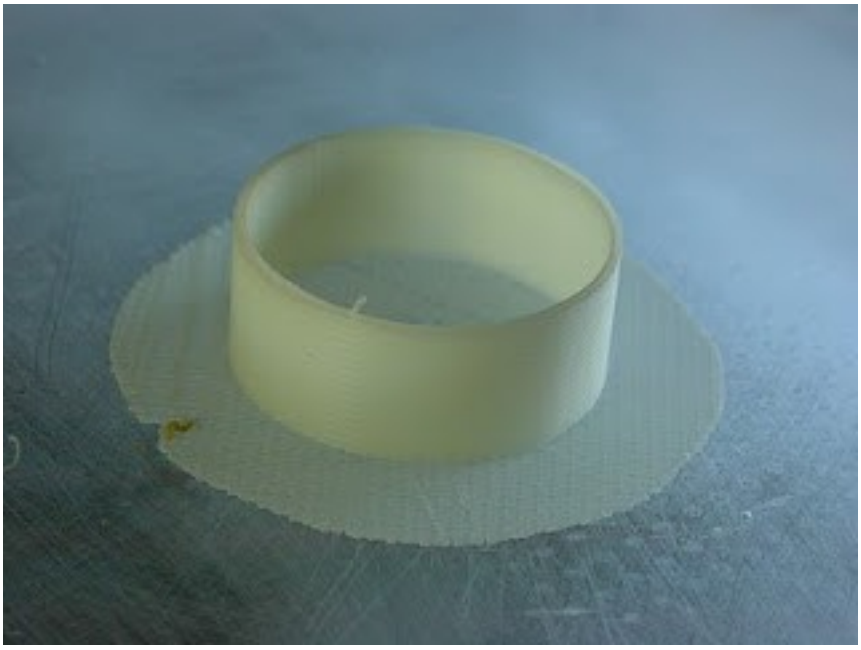
After spending some time with Netfabb for Rapman Beta, I've come to the conclusion that it isn't ready for prime time. That's not to say that it won't be and that, when debugged, it won't be a brilliant piece of software, attractively priced. The problem for me is that it isn't helping me solve the sorts of printing problems that I have ... NOW.

I reached a similar conclusion about Skeinforge over a month ago. At that time I began upgrading and rewriting the Visual Basic .NET Express 2008 code in my old Slice and Dice app that I originally developed for Tommelise 1.0.

The deal breaker for Netfabb was the grim performance with respect to z-axis juddering. I did a comparison not, I'll readily admit, a completely fair one. I created a 50 mm diameter hollow cylinder and ran it first using the "very fine" setting on Netfabb. This is what I got.

This test object gives you a clear idea of the effect of the z-axis juddering. You can also see delamination taking place on a rather large scale and you can see the attempt by Netfabb for Rapman to style some sort of cover over the top of the hollow cylinder.

Now Bogdan has rightly pointed out that a lot of my problems here stem from the fact that Netfabb prints at 260 C and wants a fan that I haven't bothered rigging. That is not, however, the only problem that the beta has at this point, however. For purposes of comparison, I ran the same test object through my own Slice and dice with a 0.25 mm layer setting {Netfabb at "very fine" uses a 0.125 mm layer}. Here is what I got with Slice and Dice.



No delaminations and a much, much less pronounced z-axis juddering. I generated the test object with Art of Illusion and used a 0.05 mm conversion of the cylinder to a triangular mesh. That left me with a 64-sided approximation of a cylinder, if I remember correctly. The facets are clearly visible on the Slice and Dice print and nowhere to be seen on the Netfabb print.

There was simply no comparison between the two insofar as print quality was concerned.
After the disaster that I had printing herringbone pinion gears Bogdan asked for the STL files and did quite a nice job printing them with his 0.5 mm extruder {I use a 0.3 mm extruder}. I decided to run those through Slice and Dice, too. While I was at it I got in and cured, not nicely mind but cured all the same, the flaw in the AoI herringbone pinion gear script that I wrote some time ago. I will be publishing that with a download link in a few days for anybody who needs it. Actually, you can cure the flaw with the Netfabb repair feature, which is a brilliant little facility that has saved me no end of pain and suffering dealing with STLs coming out of AoI.
That done, I did a solid ABS print of the pinion with Slice and Dice.

.

The quality is VERY good.

I printed it in about fifteen minutes including a 5 minute cooling period that I did via the simple expedient of pausing the print midway. I've found that if you let your print get much hotter than 90 degrees you get detail blurring.

Right now I am processing the torso that I earlier attempted to print with Netfabb with Slice and Dice. It will be interesting to see what kind of print quality Slice and Dice gives me with that art object.

Printing the naked lady with Slice and Dice

Wednesday, 17th March 2010 by Forrest Higgs

After my experience trying to print the naked lady with Netfabb I decided to have a go at it with Slice and Dice. I had considerable trouble getting the polymer flow right to bridge the base of the pundendum. Once I got that right, I set up a shell print with zero infill.

The print went well with the exception of one delamination flaw in the left leg as you can see in this pic. The lack of support that a light infill would have provided became apparent as I printed up to the top of the breasts and the area around the collar bones where considerable bridging was required.

Aside from that, however, Slice and Dice produced a superior print quality as you can see.

Very little of the z-axis juddering was apparent and the surface quality was superior. Indeed you could even see the modeling of the body with triangular segments in places.

That was fun. Now back to work.

No peel, no warp, no backlash

Friday, 19th March 2010 by Forrest Higgs

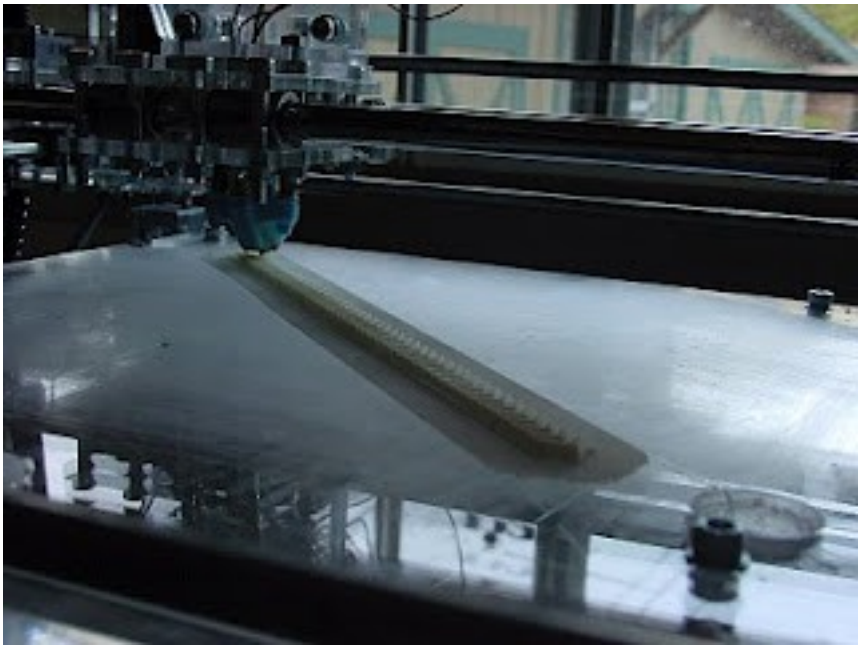
I got Slice and Dice working well enough to let me do some serious R&D. After a side excursion with partial STLs of hands and naked ladies, I got back to work on the herringbone rack and pinion technology.

My first exercise was to see if I could print single herringbone pinions without having a lot of meltdown problems. Skeinforge had a lot of problems in that regard. In fact, Slice and Dice let me do that.

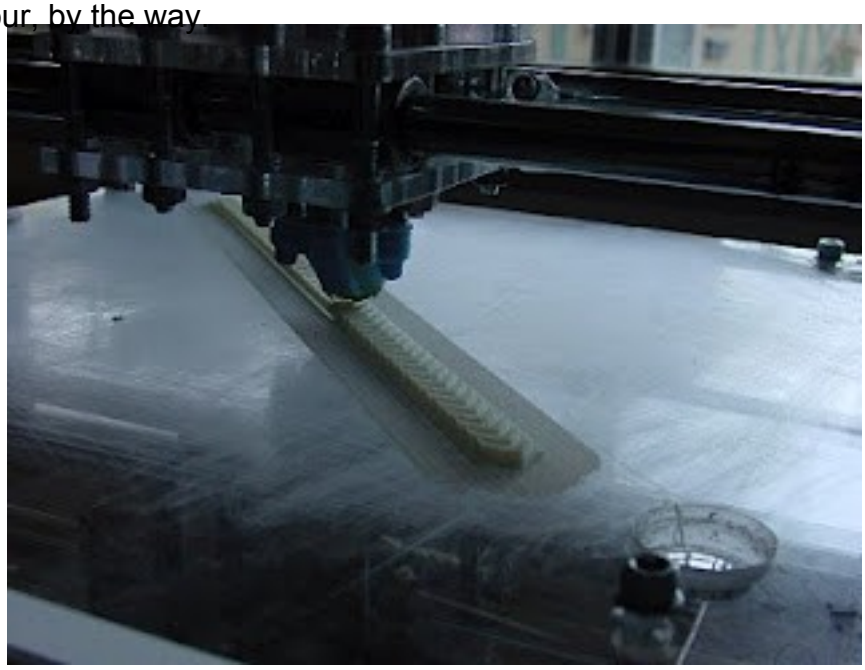
I printed it pretty much solid for strength and paused it for a couple of minutes midway through the print and let it cool down for a few minutes. No problems. This one was printed with 0.25 layers.

Next came the rack. I decided to use the exercise last night to press the limits of Slice and Dice. I figured that I ought to be able to print a 250 mm diagonal layout rack. That let me find a half dozen limits bugs in the S&D code and get them cleared away. I also decided to see if I could print a useful 0.1 mm layer.

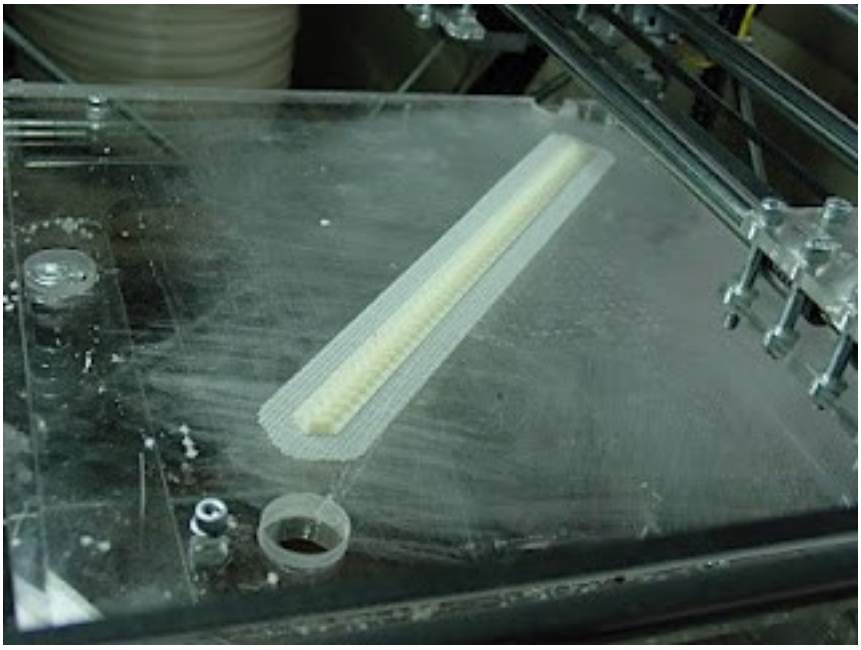
In fact, I could. Here we are about halfway through the print.



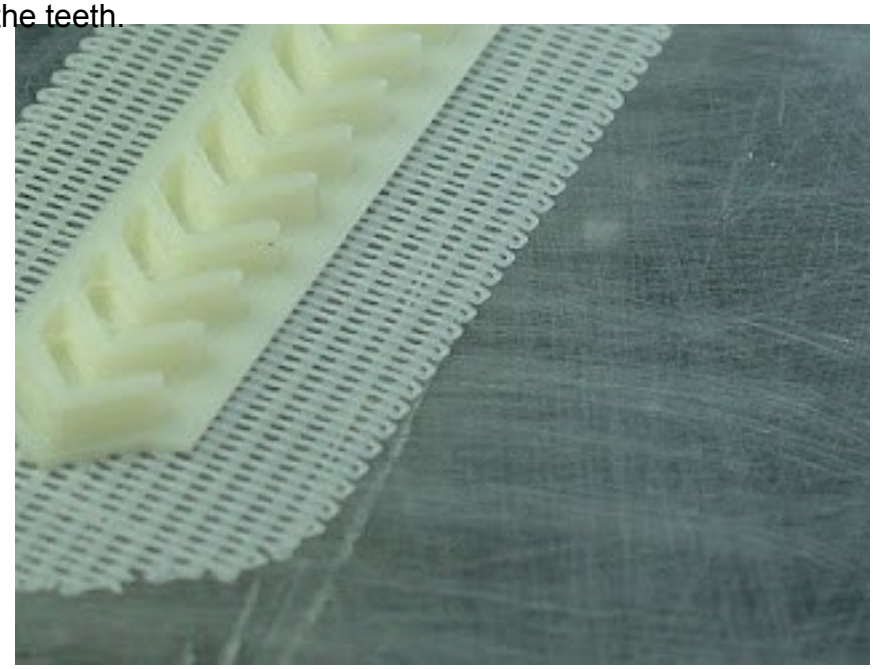
It took about an hour, by the way.



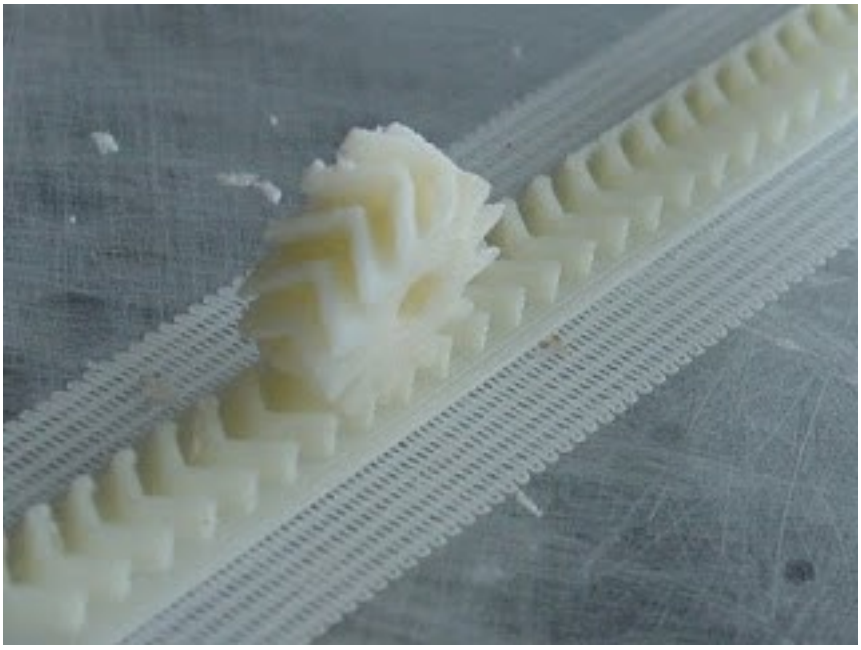
Here it is, complete.



And a closeup of the teeth.



For some reason the extruder is not shutting off between layers. While that is easy enough to clean up with side cutters, I'll be diving into my code to see if I can sort that out in the next few days.
The pinion gear mates with the rack with no backlash.



I didn't make much of an effort to clean up the pinion which you can see in this outdoor pic.

I've found that the colour rendering is always better outdoors.
Finally, for those of you who haven't a sense of scale looking at the Rapman print table.

My next task will be to figure out how to put the pinion on a printed, extended shaft so that it can be secured from both sides and mate with the short axled NEMA 17 that will drive it. Then it will be a matter of integrating my Pololu Allegro 4983 microstepping driver board into my I2C bus and driving the thing from my Microchip 18F4550 uC board.

Time isn't free

Saturday, 20th March 2010 by Forrest Higgs

Slice and Dice takes up a lot of disk space because it creates and operates on bitmap images, each one of which takes up 22 megabytes, and keeps half a dozen for each slice of a print. That eats up the gigabytes in a hurry.

The problem is that if you have a nasty STL that you've pulled out of Thingiverse, for example, you have to repair the faulted slices. On the recent torso that I printed about 6-7% of the slices were faulted. This usually meant dotting in one or at most two bits on an image to put things right. The problem is that after you've done that on 30-40 slices you're rather hesitant to throw all that work away because you need the space.

Some weeks ago, I discovered that if you zipped bitmaps you could reduce their size by about 95%. This morning I looked into converting over the storage modules to do this using a compression/decompression api. I quickly realised that this kind of job was going to require 3-4 man-days to do. Instead, I went down and bought a two terabyte external disk.

Stacking slices to make simple boolean unions

Sunday, 21st March 2010 by Forrest Higgs

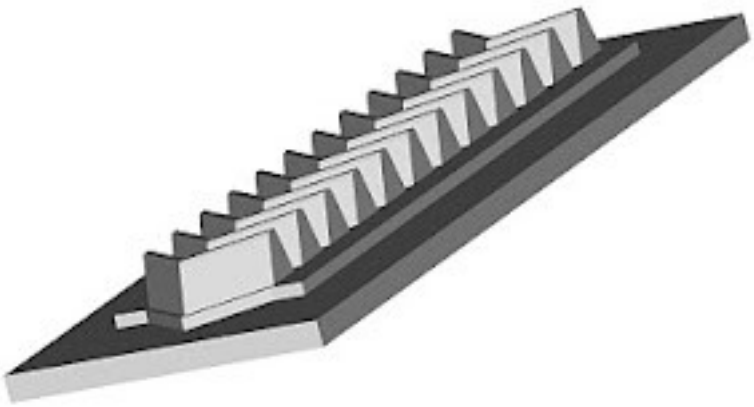
Art of Illusion is a pain in the neck to work with. On the other hand it has the shortest learning curve of any 3D modeling package I've ever encountered and is simple enough for an eight year old to operate.

When I say pain, the primary problem is the boolean ops module. AoI does boolean ops with triangular mesh defined solids. Doing boolean ops on mesh defined solids is a brass bound bastard of a problem. There is a tendency to condemn AoI because its boolean ops module isn't very good. When you go out and start looking at other 3D modeling packages, commercial and open source, you begin to notice that virtually none of them, or all of them for all I know, can import mesh defined solids formats like STL. When you look at how these apps do boolean ops you soon find that the objects you can define have to be very carefully defined out of shape primitives and that these primitives, suitably adjusted by size, orientation and position parameters are what go into the boolean ops module.

That's cheating.

Before I get off into a pointless rant about that, let's get back to the problem I was addressing. AoI is rather simple to write special purpose scripts for. Recently, I wrote one for generating herringbone racks. It was a pain in the neck to design that script, but once it was done it worked perfectly. The rack has a 5 mm flange on either side.

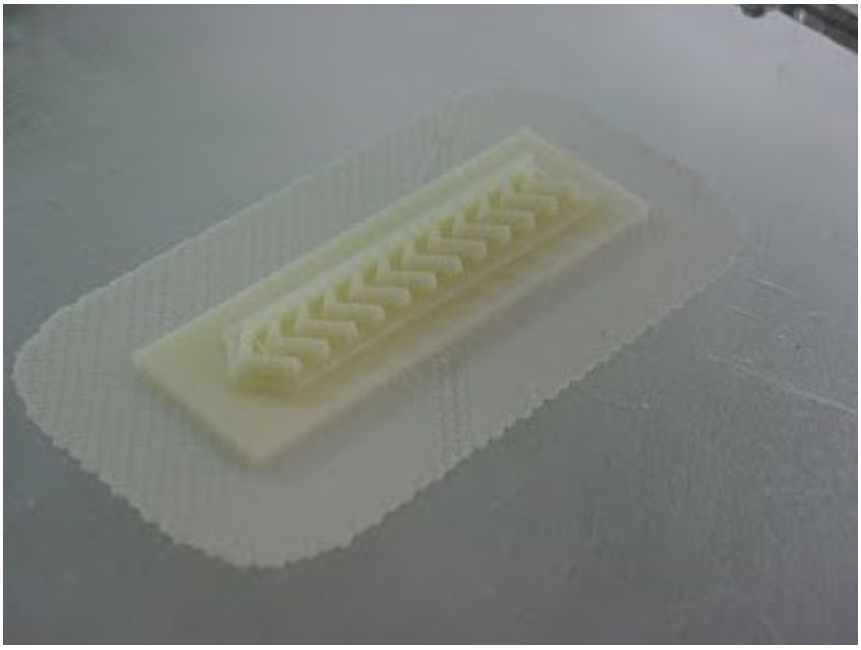
When I set about to design a beam into which that rack could slot, however, I discovered that it would be a very good idea to have some space between where the rack stopped and the beam started. To do that I needed to add a wider, somewhat thicker slab under the rack. I had two choices. I could either go back and rewrite the script {shudder} or I could glue that rack onto the heavier slab using a boolean union op, sort of like this.



While AoI would let me merge the slab and the rack, when I went to do a boolean union op, it simply wouldn't let me. Not nice. It was infuriating to be able to look at what I wanted on the AoI display and not be able to turn it into an STL.

That got me to thinking. Basically, all I wanted to do was put the rack on top of the slab. If I put the slab through Slice and Dice to create a set of slices and then repeated the game with the rack, keeping the same alignment, it would be a simple matter to just stack the rack slices on top of the slab slices. Of course, when I set out to do that I unearthed several previously unsuspected bugs in the Slice and Dice code which took me most of Sunday to chase down and fix.

It works now, though. Slice and Dice can do a simple boolean union between two objects by stacking slices. I've tested the code and ran it through the whole exercise in the morning. And there it is.

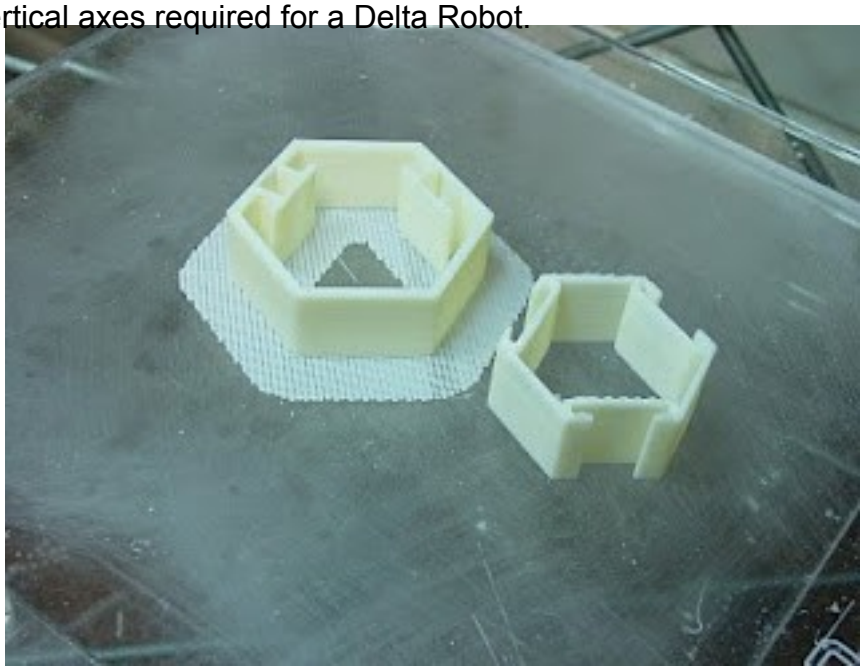


The slab is 80 x 25 x 2 mm. The slab and flange are 100% fill. I printed with 0.1 mm layers using a 0.3 mm extruder at 16 mm/sec. There was no corner curling or warping at all. This print is extremely strong.

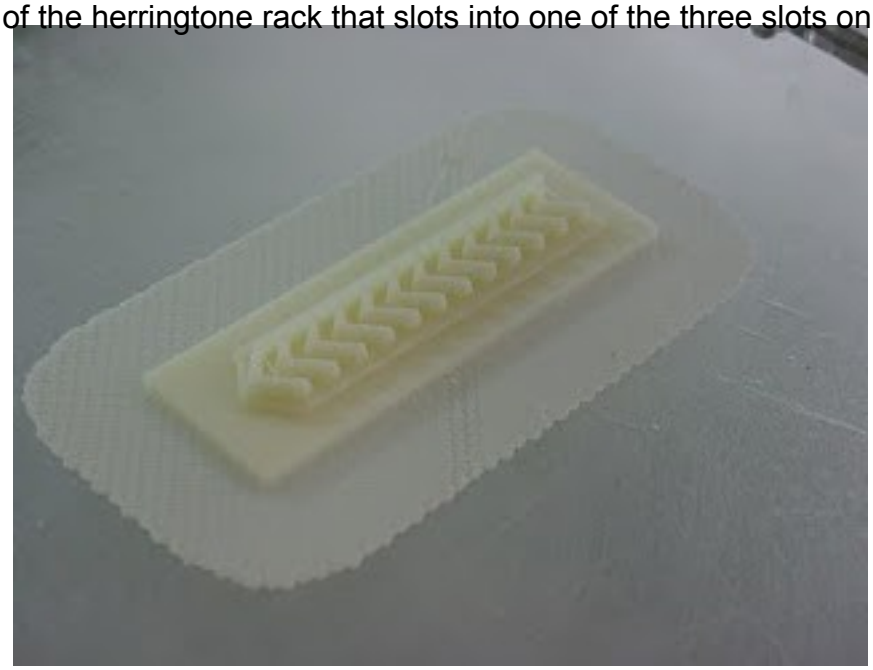
Design study for axis columns and stepper motor sleeve

Wednesday, 24th March 2010 by Forrest Higgs

This is a very preliminary design study for a paired column section and stepper motor sleeve of one of the three vertical axes required for a Delta Robot.



This is a segment of the herringtone rack that slots into one of the three slots on the column.



This is all very tentative. I am trying to learn what fits together with a minimum of fuss. I've found that if I have shapes in my hands I can see their possibilities and what wants doing with them a lot easier than if I just look at them on a screen. Hell of a situation for someone who trained as an architect, don'tchaknow. :-D

Delta 'bot Axis

Saturday, 3rd April 2010 by Forrest Higgs

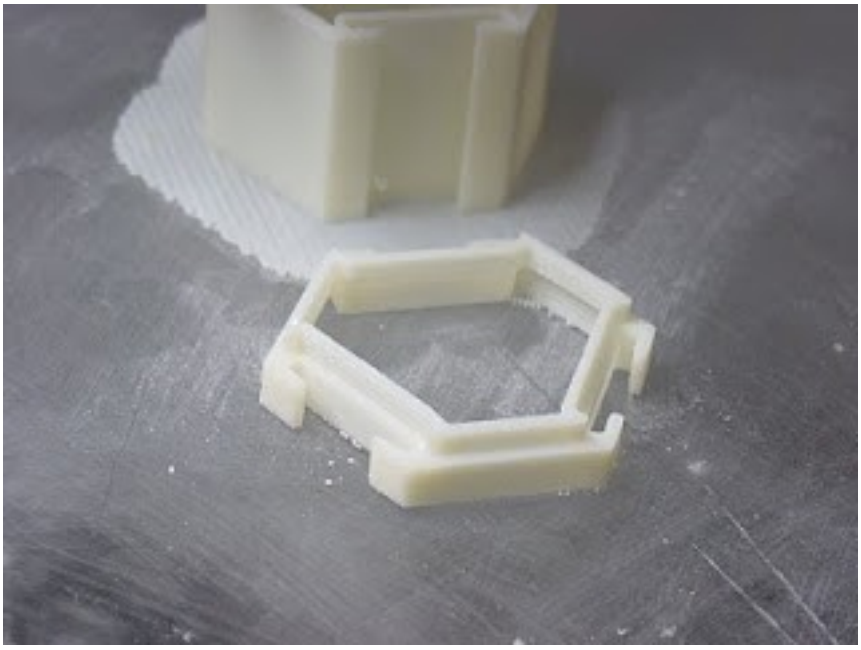
It's one thing to design and print plastic parts that have to mate with steel rods. As I am finding it is quite another to design plastic parts that have to mate with other printed plastic parts. During the last week in my spare time I proved out a design for a hexagonal column which I think might accommodate a herringbone rack and a sleeve seating a pinion gear, stepper motor and two guide plates.

Heretofore, I have been butt seating the column sections and then post-tensioning them. While that was okay for demonstrating the design approach, worries about creep said that I needed to have a male/female attachment technology so that I could fit sections of the column together and avoid having them do a lateral creep out of alignment. That proved to be considerably more difficult than I imagined.

This last weekend, I decided to take a different approach and instead of doing a conventional slice and dice of an STL, I decided instead to work with the print roads of my thin-walled column instead. What I did is a radial 45 degree overhang inside the column onto which I printed the male insert. I think it is worth mentioning how I achieved that.

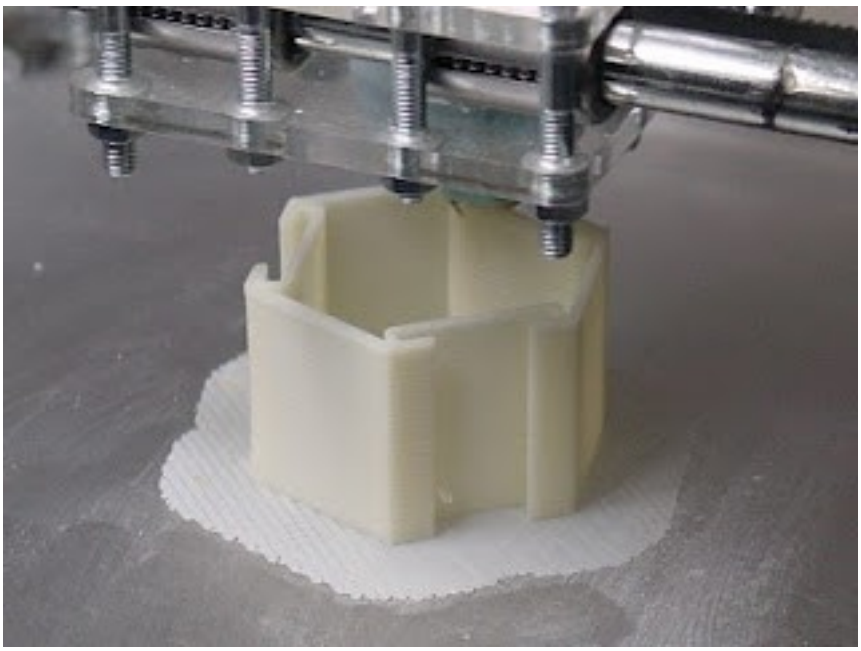
The column was achieved for most of it's length with three concentric print roads of 0.6 mm width. What I needed to do was to build a ledge inside the column. When I began the ledge I widened the innermost print road to 0.8 mm. On the next layer I replaced the 0.8 print road with two 0.6 print roads. This process left me a 0.2 mm miniledge onto which the new innermost print road could adhere. I repeated this process until I had a ledge wide enough to print the male insert onto.

You can see the insert successfully executed here.



I would like to be able to say that that was all an easy exercise. It wasn't. I did exactly thirty text prints like the one you see before I was satisfied with the seating of the guide strips, herringbone rack and the male/female column connector system. I also had to chase several heretofore unsuspected bugs in Slice and Dice AND extend the coding somewhat. All the same, it's done and works so far.

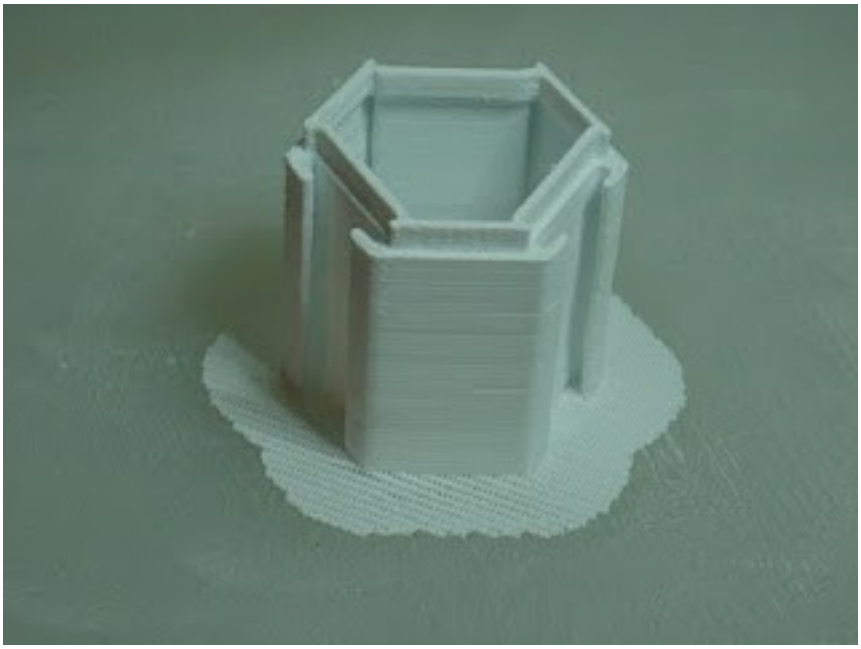
At that point, I decided to see how tall a monolithic column section I could print. It appears that Rapman three can handle about 225 mm in the z-axis. I designed a 200 mm column with a 4 mm male connector and am printing it at present.



It is 2120 and I am up to about 50 mm. I'm doing a 0.25 mm layer every 45 seconds, so I expect to be finished at about 0500 tomorrow morning.

With such a large print I began to encounter xy and z faults. The xy faults were annoying, but the z faults tended to plunge the hot extruder head into the print. Upgrading the Rapman firmware to 2.0.8 got rid of the z faults but the xy faults remained.

I reduced the column test print to 55 mm, relocated the Rapman further away from my PC on a table by itself away from the cold window and switched ABS to HDPE because the acrid fumes were getting to be too much when I wasn't able to open the windows.
A thorough cleaning of the extruder with compressed air was also undertaken. After a few test prints I got the flow rate sorted out for HDPE and was able to successfully print a 55 mm column section in HDPE.



Collapsing the hard disk requirements of Slice and Dice

Thursday, 8th April 2010 by Forrest Higgs

I managed this by the simple expedient, suggested by my son Adriaan, of replacing the BMP format slice image files with the Portable Network Graphics (PNG) format which uses a lossless data compression scheme. This reduced my slice image files in size from 16-22 Mbytes per slice to 80-100 Kbytes.

Bookkeeping and janitorial duties

Monday, 12th April 2010 by Forrest Higgs

There isn't much that is flashy to report at this time. A lot of my time in the past few days has been focussed on a telepresence 'bot project I have intended to design and build for years that the Rapman finally lets me design and print parts for. Oddly, though, getting the basic software modules that will control the 'bot all working together looks to be far harder than printing and building the 'bot itself. I guess that having worked on Reprap for five years now makes me a lot less intimidated by hardware tasks than previously.

Last week, I discovered that Slice and Dice's vectorization routine, which converts pixel defined boundaries to print areas into vectors usable in gcode, wasn't particularly nice. It was not reducing the boundary definitions nearly as much as it ought to have been. What that means as a practical matter is that it generates a lot more gcode than it ought describing very sort little print roads of a few tenths of a mm. Rapman is not designed to deal very efficiently with that kind of gcode and what happens is that the print speed drops well below the nominal rate set with the "F" specification. That not only wastes time but also puts more plastic on the print than it needs to have in areas where the perimeter curves. THAT makes for poor print quality.

I got to the bottom of that problem over the weekend.

As I set up for testing some new approaches to vectorization, I remembered that shifting from BMP to PNG reduced the disk requirements for a print project dramatically. I have plenty of disk space, but the reduction in disk requirements led me to redesign the files handling for Slice and Dice so that one can readily process and work with more than one print project at a time. That job holds no challenges as such, but is merely a bit of tedious restructuring files handling routines and testing to make sure that they all work. I've got that about two-thirds done.

Thus, slowly but surely Slice and Dice begins to look like very preliminary beta code.

What does F960 really mean?

Wednesday, 5th May 2010 by Forrest Higgs

The vectorizer in Slice and Dice is pretty inefficient. With certain slopes and curved surfaces I'd noticed that the print head velocity could drop by as much as 25% down from 16 mm/sec to 12 mm/sec. I'd spent the last weeks working on improving my vectorizer when it occurred to me that I really didn't know how much improvement I needed to make on it.

To sort that out, I did a series of test runs where I moved the inactive print head from x = -75 mm to x = 75 mm using different steps with the nominal head speed set to 16 mm/sec (F960). Here are the very strange results that I got.

Step (mm)	Seconds (s)	Velocity (mm/sec)
0.1	25	6
0.5	5	30
1.0	3	50
10.0	2	75
150.0	2	75

As an example of the gcode I used...

```
G90
G21
M103
M105
M104 S0
G1 X-75.00 Y0.0 Z20.0 F960
G1 X-74.90 Y0.0 Z20.0 F960
G1 X-74.80 Y0.0 Z20.0 F960
....
G1 X74.70 Y0.0 Z20.0 F960
G1 X74.80 Y0.0 Z20.0 F960
G1 X74.90 Y0.0 Z20.0 F960
G1 X75.00 Y0.0 Z20.0 F960
M103
M104 S0
M103
```

I did further tests and discovered that unless the extruder is running the extruder head speed defaults to 75 mm/sec. When it is running F960 seems to be the speed of traverse of the longest dimension {xy} of the print road.

Specifically, I ran a 150 mm diagonal and discovered that F960 {16 mm/sec} clocks in at about 23 mm/sec. when you divide 23/2^.5 you get right at 16 mm/sec which means that the traverse speed for the x and y axes, which with a diagonal are equal, is 16 mm/sec.

BTW, I've started getting an SD Error2 report at the end of one of these test runs. I'm not at all sure what that is all about.

Further tests

Thursday, 6th May 2010 by Forrest Higgs

I did a very careful, new series of speed tests using F960 (16 mm/sec) over a much longer diagonal baseline, viz, 1.7 meters, with the extruder running.

Here's what I found...

Step (mm)	Seconds (s)	Velocity (mm/sec)
0.1	214	5.77
0.2	110	15.40
0.3	121	14.14
0.4	80	21.21
0.5	80	21.21

Note that...

21.21 mm/sec ~ 16 mm/sec x 2^0.5

...which is about what you would expect for firmware that didn't correct for the angle of the print road.

As you can see, the break comes between gcode distances 0.3 and 0.4 mm and is quite abrupt. This means that the vectorization routine for Slice and Dice can be quite crude and work well with Rapman.

Processing gcode instructions with the Rapman firmware

Sunday, 9th May 2010 by Forrest Higgs

After running a variety of speed tests with the Rapman, I gained enough data to estimate the processing time for two kinds of gcode statement. The G1 statement processes in the Rapman 3.1 firmware at a rate of 17 msec/instruction. M108 instructions process at a rate of 19 msec/instruction.

Interestingly, the rate for the M108 instruction is actually significantly longer than for the G1. Given the floating point instructions necessary to interpret a G1 instruction this seems very odd. It would seem that much more time is spent reading off of the SD card than processing the information on a line of code. Either that or carrying out a M108 instruction in the Rapman firmware is more complicated that it would appear at first glance.

As a practical matter what this means is that if one wants to print a layer with 0.1 mm print road lengths you are looking at a maximum print speed of around 5.8 mm/sec. A print speed along a long road component axis of 16 mm/sec would imply that your average road segment would be no less than 0.3 mm.

New blog...

Sunday, 13th June 2010 by Forrest Higgs

I've finally done enough work on my active telepresence project that I need a blog to document what I am doing. For anybody interested, you can access it here.

Vectorization of pixel defined print roads

Monday, 21st June 2010 by Forrest Higgs

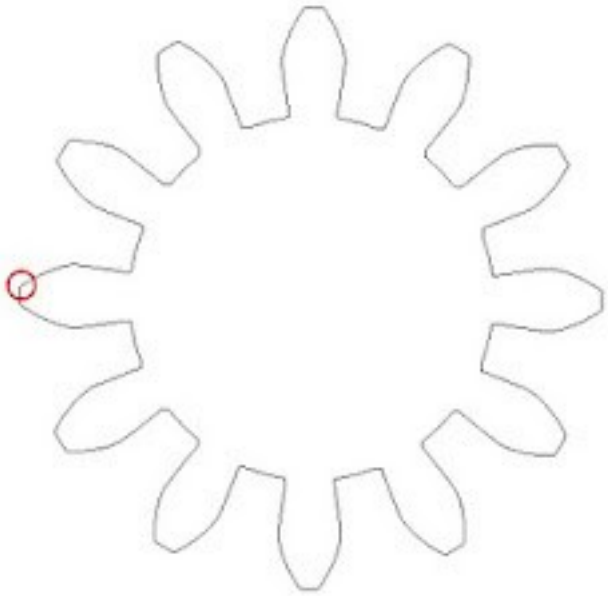
In which your narrator describes the end, hopefully, of a search for a reasonable way of converting pixel-defined print roads calculated by Slice and Dice.

For the past two months I've struggled with what has been, for me, a very nasty problem, viz, converting pixel boundaries defining print roads for my Rapman printer into equivalent vector descriptions. As I mentioned previously, in order to get a major axis velocity of 16 mm/sec on the SD card Rapman 3.x you have to have GCode vectors of no less than 0.3 mm. If you are printing detail finer than that the lag caused by reading off the card and processing the GCode slows down the print head dramatically.

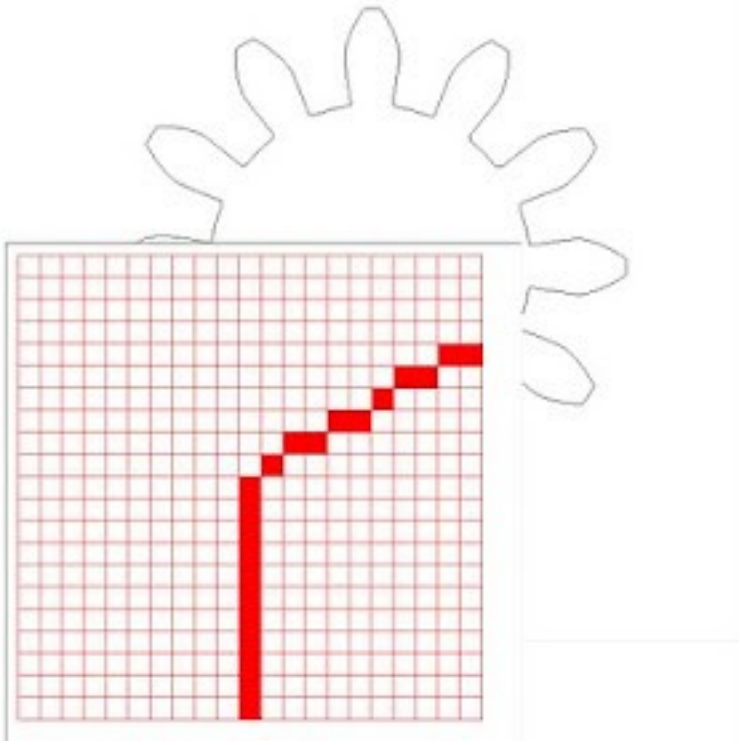
From my literature search, vectorization is a fraught process, especially if you are not willing to accept considerable degradation of your pixel-defined road. With 3D printing, perforce, we really can't accept degradation.

Over the last few weeks, I've finally confected a method which seems to do the job. As with most things I do, it is relatively simple and straightforward.

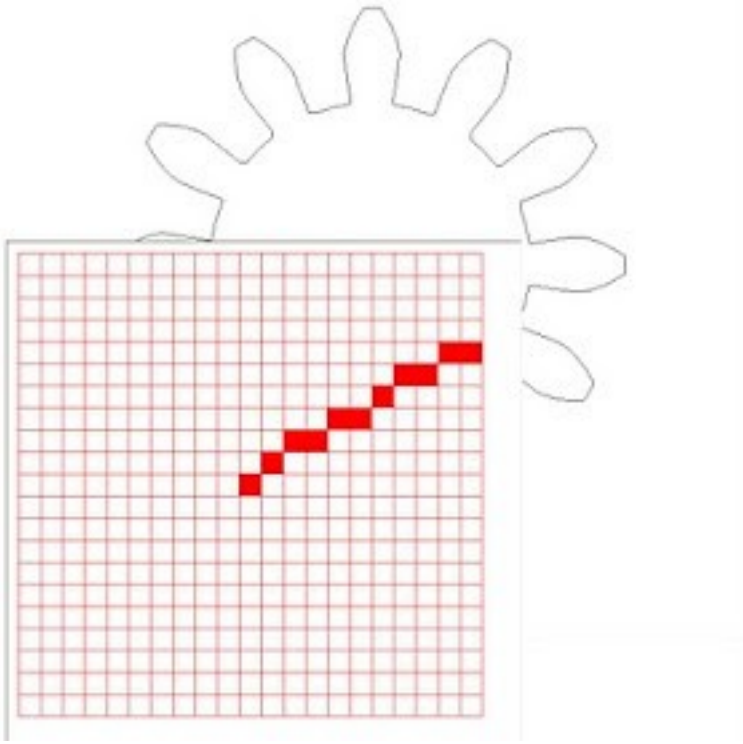
Suppose we have a perimeter path for an involute profile gear...



The method grabs a 2.1 mm patch {note the red circle} at the extreme left of the path which you can see here...

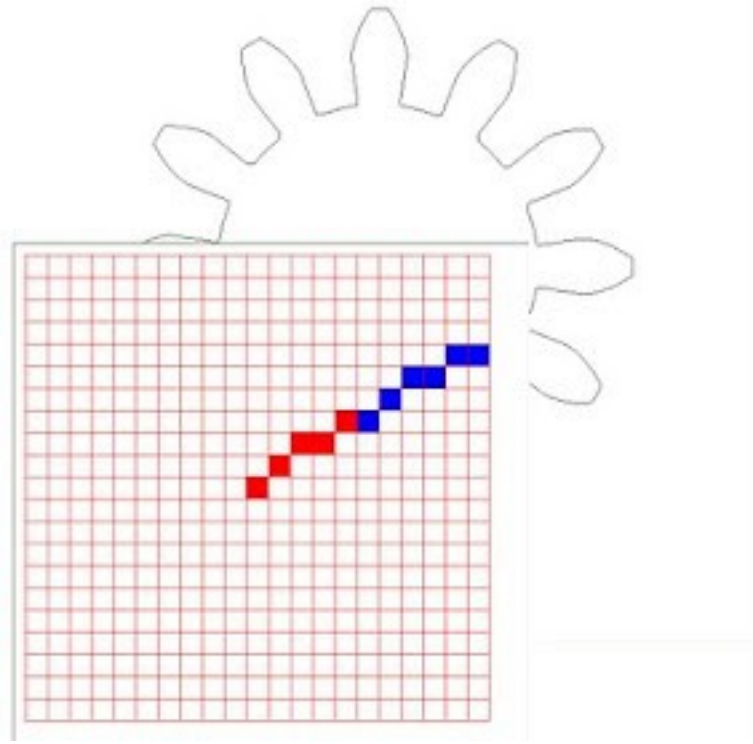


Here you can see the individual pixels making up the print road. The first thing we do is define the starting point and direction of the road from the centre of the patch.

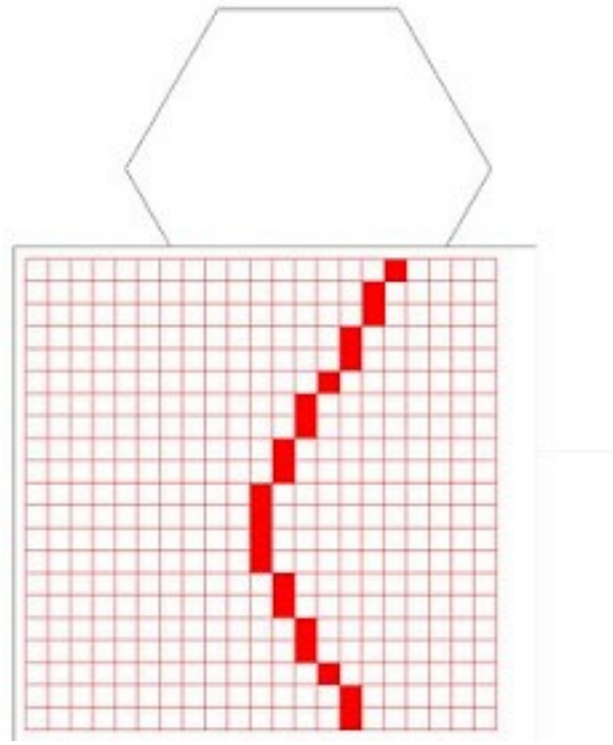


At that point I pivot a scan from the center of the patch and identify the closest fits at ranges varying from 1 pixel to 11 pixels.

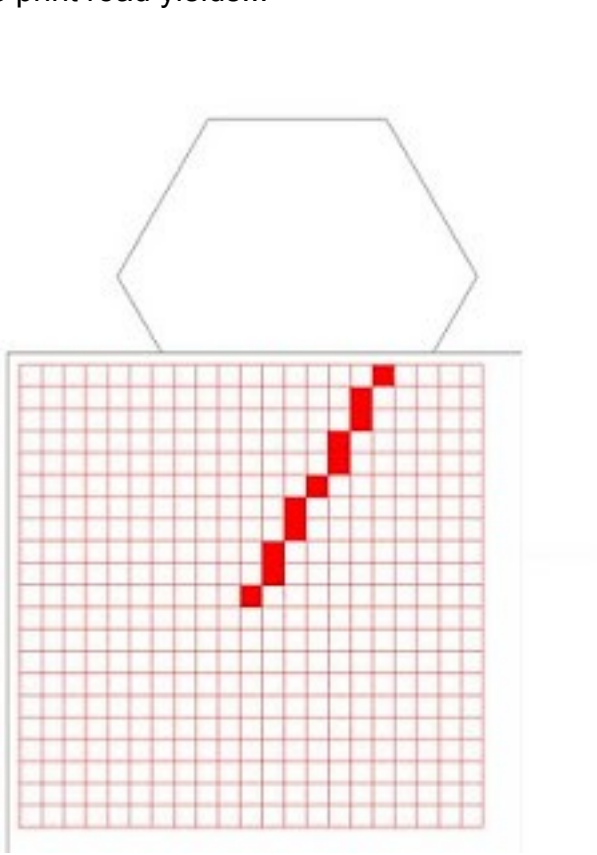
In this case, the longest perfect fit to the involute profile was 4 pixels.



Red pixels represent the fitted vector while blue indicate the remaining pixels. Notice that this vector is four pixels long, just enough for the Rapman to operate at 16 mm/sec.
The patch is then re-centred over the most distant red pixel and the process repeated until the full loop is vectorized.
Looking at a simpler problem, consider a hexagonal print road...



Isolating the beginning of the print road yields...



Scanning this straight line gets a perfect match at 11 pixels. Thus the way I have the code set up now creates vectors of up to 11 pixels in length, a maximum length of about 1.4 mm. It's reasonable to ask why the vectors are kept so short. In fact, it takes much longer to process the print roads with longer vectors and there is no reason, SD cards having huge storage capacity, not to have large print files.
As I get time in the coming weeks I will be fully embedding this method into Slice and Dice.

Vectorization of pixel defined print roads actually working properly

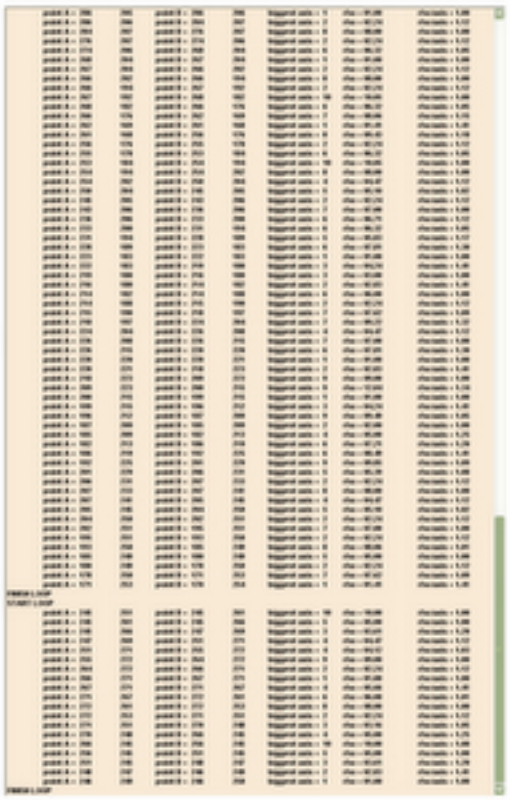
Tuesday, 6th July 2010 by Forrest Higgs

In which a very kind Adrian Bowyer takes pity on your narrator.

In my last posting on 21 June, I laid out a method for vectorizing a pixel-defined perimeter. Having read the posting, Adrian Bowyer, a published expert on this sort of thing, took me aside and showed me how to do the vectorization efficiently instead of just at all, the solution I came up with.

Using Adrian's approach, I was able to achieve my goal of getting exact vectorization of the print road and push out the average segment length to well over the 0.3+ mm that was required to assure a print head speed over the major axis of at least 16 mm/sec.

You can see an example of such a vectorization here. All values, save the last, are in tenths of a millimeter.



The last value in the listbox, "rho/axis" should allow me to adjust the print speed on the fly in G1 statements to assure a constant head velocity for print road segments greater than 0.3 mm in length.

I should be printing ABS parts again by this weekend, which is good since I want to redesign and print one of these.

Operational again

Saturday, 10th July 2010 by Forrest Higgs

I was able to put the final touches on the Slice and Dice upgrade last night. The app is operational again. Now I can get to work on that telepresence hand project. :-)

In production

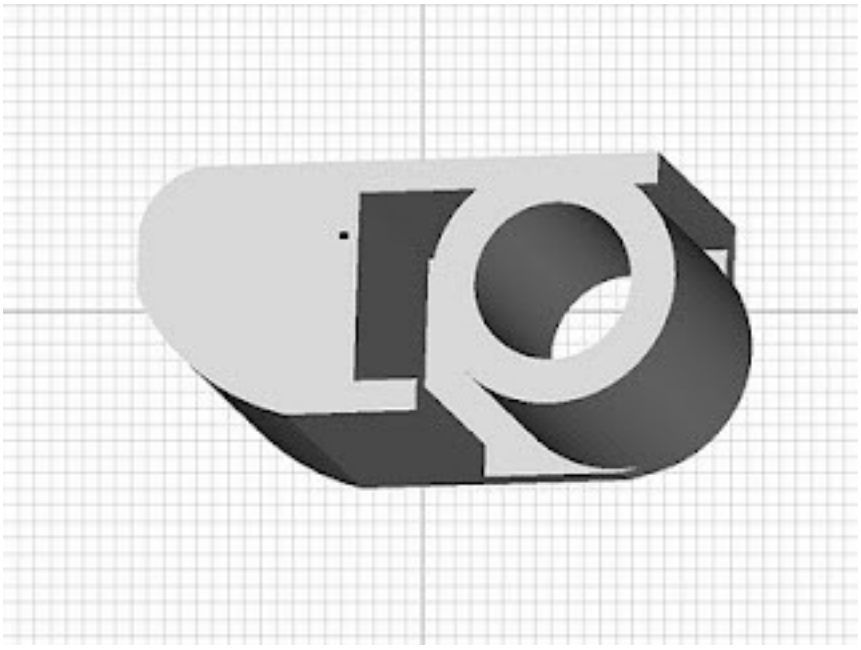
Saturday, 17th July 2010 by Forrest Higgs

Got the whole Slice and Dice ensemble running together last night and did some test prints. Vectorization cured up the extruder head slowdown completely for curved print roads. Sweet! :-D

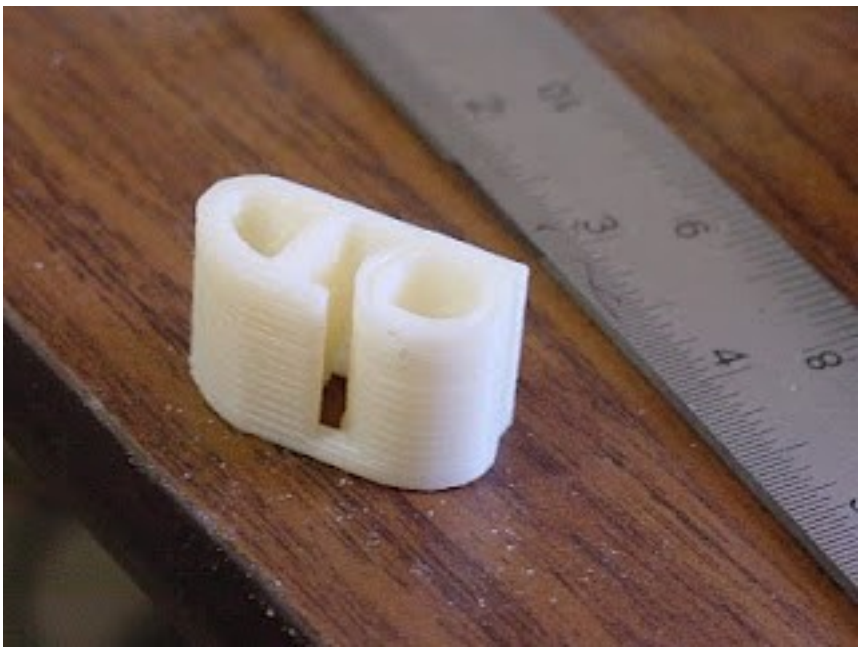
First prints

Sunday, 18th July 2010 by Forrest Higgs

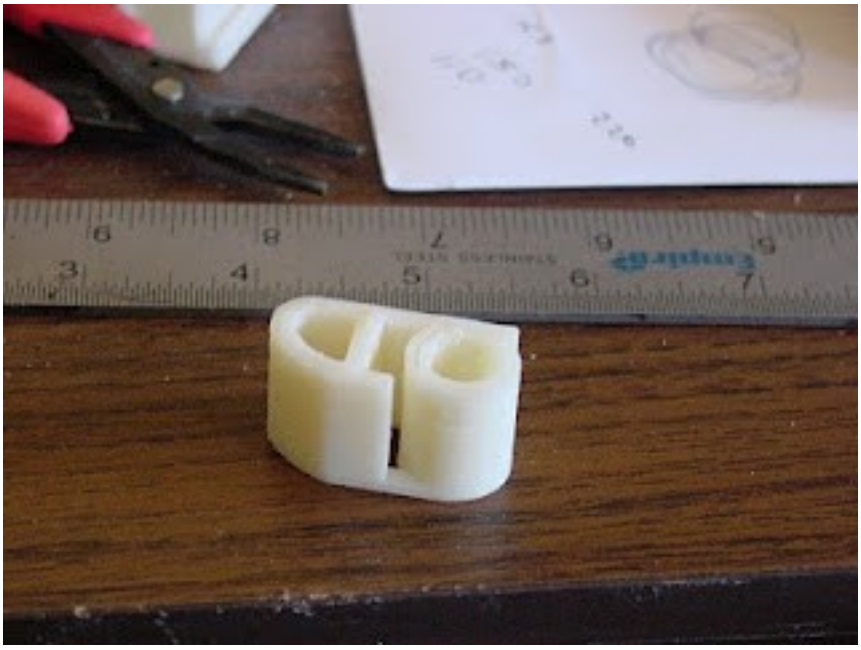
Slice and Dice is pretty much working now. I'm not completely happy with how the infill and the perimeter prints hook up, but that's mostly parameter tuning. I'm debugging Slice and Dice while I am designing a robotic hand. Here is the distal phalange {finger tip} that I've done for a first try.



I've left out the infills in that they are not necessary. Here you can see the first print that I am happy with.



I've really got to break down and find a camera that can do better closeups for small objects. My periodontist uses one for photographing teeth, an Olympus, which might be the ticket. I may well buy one after my quarterly tax payment is done towards the end of this month.



As you can see in this second pic, there is a separation between the outer and inner print road for the flexural axis. I've got to reduce some the outer radius there or lower the extruder flowrate and go for three print roads instead of two on that part of the print.

Memories of plastic model airplanes

Monday, 26th July 2010 by Forrest Higgs

In which your narrator harks back to his youth and fingerprints ruined by trimming plastic flash off of model airplane parts with double edged razor blades.

I have an aversion to infills. Recently, I realised where the aversion came from. During the 1950s you could buy plastic model airplane kits for $1-2. These were made with injection moulded parts that came on little flat plastic Christmas trees. Exacto knives in that era were both expensive, for a child at least, and the blades dulled quickly and were largely beyond the ability of a child to resharpen. As a result, this child nicked paper-thin double edged razors from the medicine cabinet to trim the parts off of the tree and trim the flash off of the parts.

That worked fine, except for the cuts which the blades would make on ones fingertips which eventually grew into permanent scars. That was no big deal in those days.

Having trimmed your parts you glued them together with Testors or Duco cement {ethyl or butyl acetate} and, with a bit of paint, you had a lovely airplane to hang off of your ceiling by a thread and dream about flying.

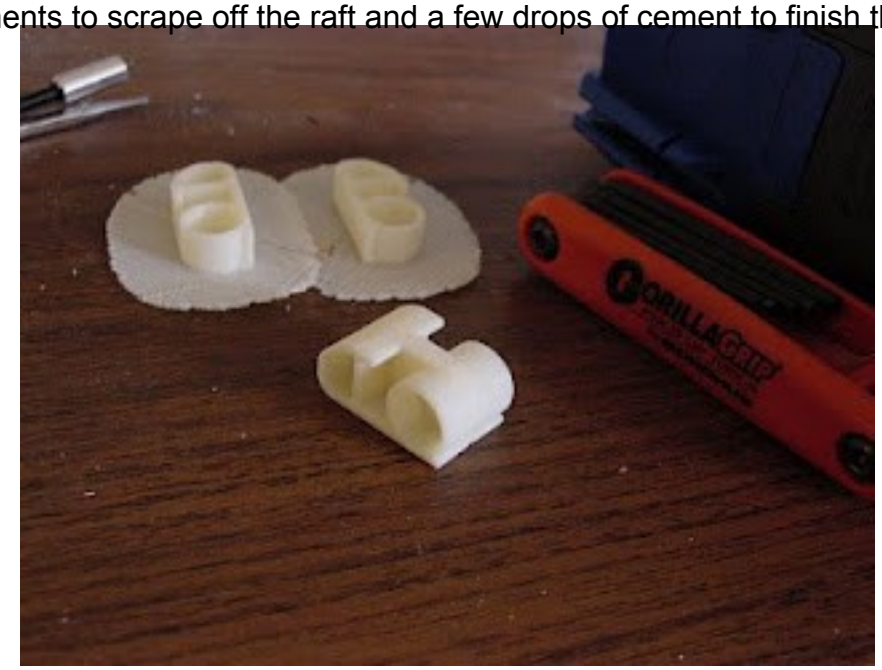
Early on in the Reprap project, we used Solarbotics gearmotors. When you opened one of the gear boxes of, say, a GM-3 you found that the housing had been made using the same injection moulding process that model airplace manufacturers used. With a working Reprap printer I quickly began to loathe the clumsy, infilled parts that we were designing. They wasted both time {to print} and filament as well. Worse still, one tended to stick the parts together with expensive nuts and bolts.

Why not make parts that fit together like the old model airplanes?

The trick there is that the old model airplanes inevitably had a left side and a right side or a top a bottom that were often simply mirror images of each other. This weekend I was design a finger tip with left and right sides. I had designed the left side and got it working acceptably and was going back to do the right side in Art of Illusion {AoI} when it occurred to me that it would be much simpler to just write a script to swap the sign of the coordinates of the axis that I wanted to mirror around. Writing that script took about five minutes and saved me a lot of time in redesign and reprocessing of the resultant STL since my script operated directly on the Gcode.

Printing the two halves was trivial.

It took a few moments to scrape off the raft and a few drops of cement to finish the part.



The result was an elegant looking part that would have been a right bastard to design and print conventionally.
While I've incorporated this simple mirroring script into my own Slice and Dice STL processing software it would be no big deal to incorporate it into the more widely used STL processing routines available to Reprappers either open source or commercial.

Some thoughts and observations about having a Reprap machine in the design cycle

Wednesday, 28th July 2010 by Forrest Higgs

In which your narrator reflects on the rather radical difference between what he perceived the design process would be pre and post the advent of practical Reprap printers.

Do you want to read more?

From 2005 till November of last year, I spent most of my free time trying to build a Reprap printer. It was good fun and I learned a lot of things. Mostly, I learned how not to be intimidated by technology that I was not familiar with. For me, this was an extremely valuable lesson.

When my day job began to require more and more of my attention towards the middle of last year, my development of Tommelise 2.0, my own repstrap project began to suffer. Finally, at the end of October I sat down and did an unpleasant assessment of my situation. I could continue to work on Tommelise, or I could simply buy one of the Reprap-derived kits that were beginning to emerge from small enterprises started up by other core team members. I estimated that it would take something like another 6-8 months to get Tommelise 2.0 printing properly. I subsequently ordered one of Ian Adkins' Rapman 3 printers.

Rapman 3 is a re-engineered clone of the first generation Darwin design. I bought it rather than the somewhat cheaper Makerbot primarily because it had a substantial print volume but also because an associate, Batist Leman, had been blogging his experience with the model. Batist was very impressed with the Rapman and communicated that very well through his video clips of it in operation.

Prior to November, I was designing and making parts with the object of making a Reprap machine. Afterwards, I was designing and making parts WITH a Reprap machine. The distinction can be easily lost on those who have not used a Reprap machine.

What I have discovered in the eight months since is that the availability of a Reprap machine makes a massive change in one's design practices.

Prior to November I lusted after a professional 3D CAD package. Features like dimensioning and reliable boolean operations cluttered my thinking. The problem was that I was thinking like an engineer, viz, I wanted to make careful designs in 3D and then have print them out on an accurate 3D printer. I no longer feel that way.

With the advent of Art of Illusion (AoI) 2.8.1, I stopped lusting after "professional" CAD packages even though by that time I already owned several. My design cycle now looks like this.

- (re)design (a) part(s)
- print the part(s)
- manually and visually see how the parts work together
- repeat the process until satisfied

Precise dimensions became not very important while things fitting together properly did. The linkage between these two factors was not as strong as one might first suppose.

With a Reprap printer it became a simple matter to run through a dozen design cycles to get a parts ensemble to work in a matter that suited me. I currently design with the Reprap printer as

part of the design cycle rather than using it as a final step after design cycles are finished.

At the beginning of this year Skeinforge, which I had been using previously, went through a rough spot. This encouraged me to invest in the upcoming Netfabb software prior to its release. As the Skeinforge rough patch continued and the folks at Netfabb's release dates began to slip, I finally became frustrated and then angry. I wasn't able to do the design work I wanted.

In frustration, I unearthed my old Slice and Dice code from the Tommelise project and began to bring it up to date. Having used both Skeinforge and the nascent Netfabb offering to process STL files into print instructions I evolved a very different approach to processing solids files than what either of these two products offered.

Both Skeinforge and Netfabb basically take your STL and give you print instructions. They do it very quickly and efficiently. Unfortunately, if you have a dodgy design file they can produce some very crazy print files. Ones ability to respond to crazy print files is, perforce, limited. With both Skeinforge and Netfabb, the majority of craziness is generated when ones STL files are flawed. The need for perfect STL files had fueled my previous lusting after "professional" 3D CAD packages.

I responded with Slice and Dice by internalising Mandelbrot's maxim that ***noise is always going to be there and it will be unpredictable***. Instead of striving for perfection, Slice and Dice concentrated on giving me options to deal with imperfect design files. I began by keeping each slice of a design file as an image that I could bring into simple image handling tools like Windows Paint and manipulate. Paint let me patch and alter flawed slices to suit myself. I no longer needed a "professional" 3D CAD system. AoI was just fine for my purposes.

From there, I discovered that if you use 2-3 print roads to define a print's perimeter you wind up with a strong part even without infill. I began working to improve my control of perimeter print roads and soon found that I could make parts not unlike those in old fashioned plastic model airplane kits which used injection-moulded parts that you glued together. Infill became unnecessary in all but the most extreme cases. Hollow parts were tremendously quicker to print and not nearly so prone to warping. as "solid" parts. I don't use a heated bed and doubt that I will anytime soon.

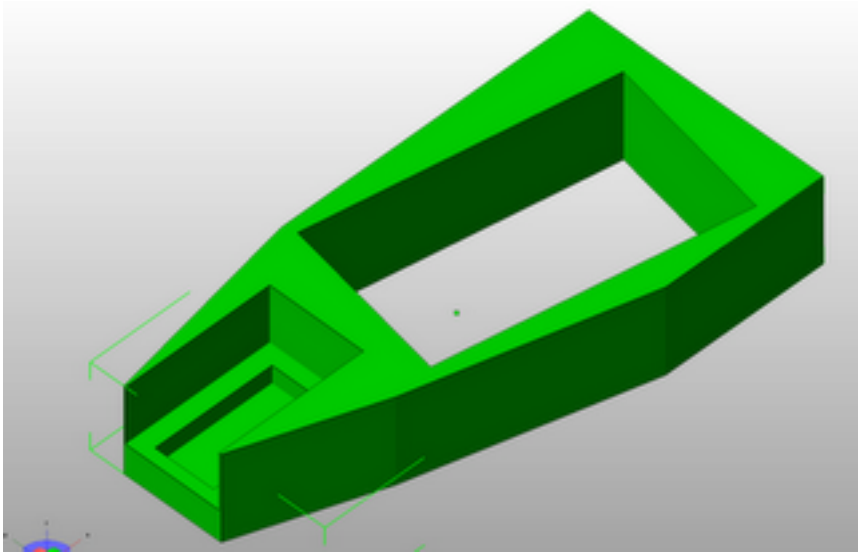
Conventional Reprap thinking views a 3D printer as being able to print, more or less, any 3D object. Development has concentrated on infill and support materials to achieve these ends. Both are wasteful of printer time and materials, in my opinion. I try to design parts which keep in mind the strength and weaknesses of my printer.

As you can see with this telepresence 'bot finger, I think I am getting pretty good results.

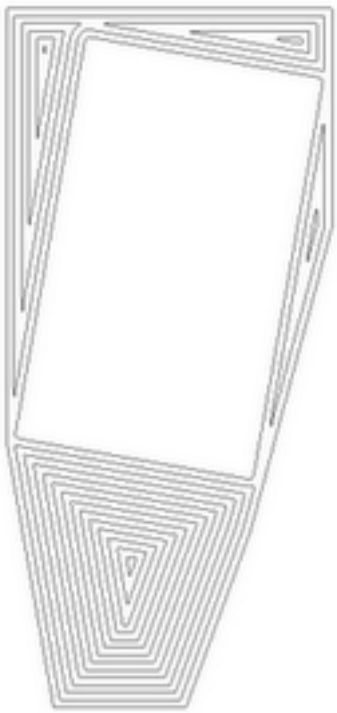
Chasing bugs in Slice and Dice

Thursday, 19th August 2010 by Forrest Higgs

I had completed my narrow profile telepresence finger and then set about designing a mount for the gearmotor and potentiometer to drive it. The mount came out looking a bit weird.



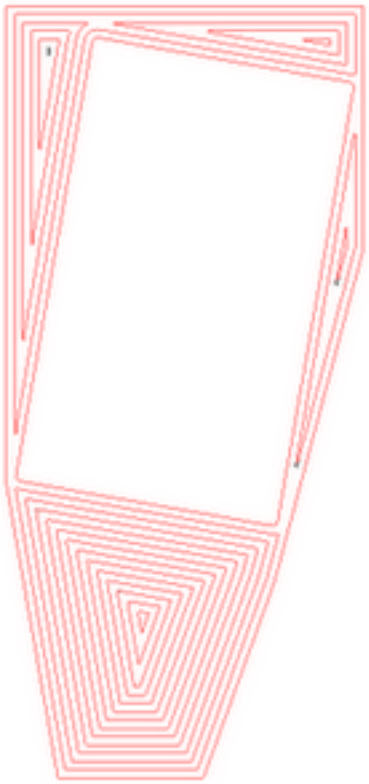
Trust me, though, there were good reasons why it took the shape that it did ... in my twisted mind, at least.
The first thing I discovered was that my print roads routine is not extremely happy with sharp ended print roads like you see here.



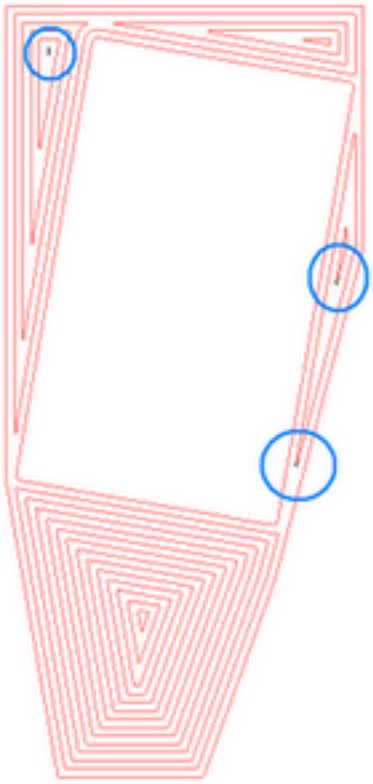
Fortunately, I was able to pull the few unique slice images into Windows Paint and clean them up

rather handily in just a few moment. Slice and Dice is very good about giving you ways around problems you encounter with difficult parts. The messy print roads were only the beginning of my troubles, however. When I tried to turn the image into an XML description of the roads all hell broke out. My pathfinder routine, which worked well enough for most slices, really hated this gearmotor mount.

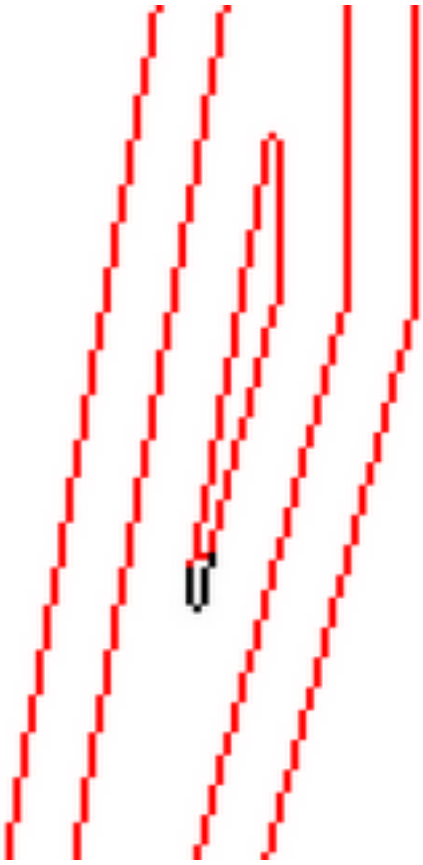
I finally gritted my teeth and spent the time chasing up the many bugs in the pathfinder routine. Once that process, which took about two man-days, was complete, I was able to get a pretty good XML conversion.



Red overlays indicates well formed print roads. The black residue shows where the routine failed.

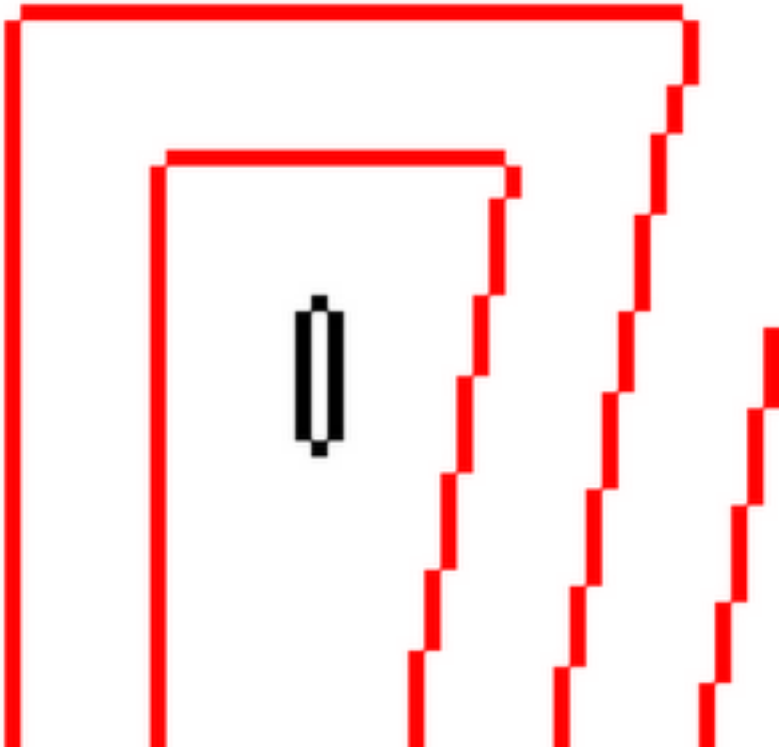


You can see the failures circled in blue. Looking a bit closer at a few of the failures you can see that the routine breaks down when the distance between two parts of a print road drops below 0.2 mm. That's a ridiculous case, but slice and dice routines regularly encounter ridiculous cases. Here is a typical one.



Here you can see that the pathfinder routine jumped the 0.1 mm gap between two sections of the

path. Another thing that became obvious is that the routine can't handle paths which are less than 1 mm in their greatest dimension.

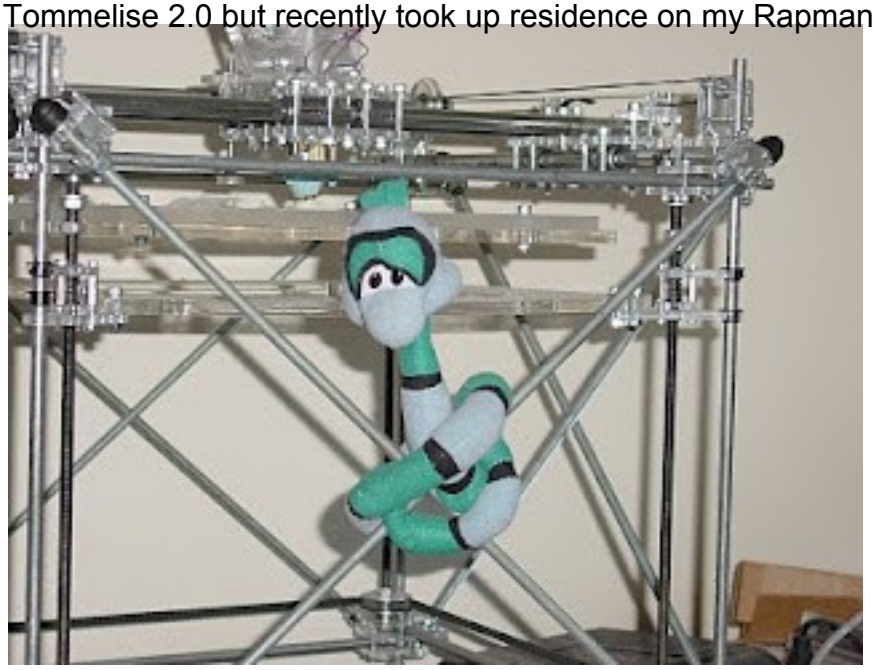


None of these faults are serious enough to cause me to continue working on the code at the moment. I'm back to printing.

Introducing Snakl!

Thursday, 19th August 2010 by Forrest Higgs

He used to live on Tommelise 2.0 but recently took up residence on my Rapman 3.0...



Snakl does all the hissing! :-D

Cloud CAD

Sunday, 22nd August 2010 by Forrest Higgs

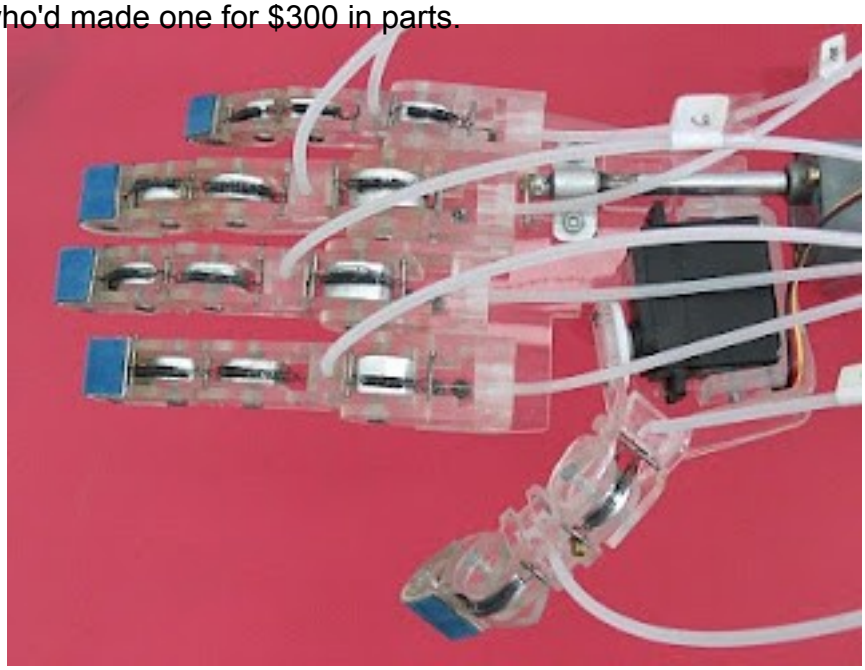
Ordinarily, I publish only my own work on this blog. In this particular case, however, I'd like to pass along a nascent project to make OpenSCAD available in a computing cloud environment. Tony Buser is the man on this undertaking. It's a very worthy one, imo.

Designing with a Reprap machine in the loop

Wednesday, 25th August 2010 by Forrest Higgs

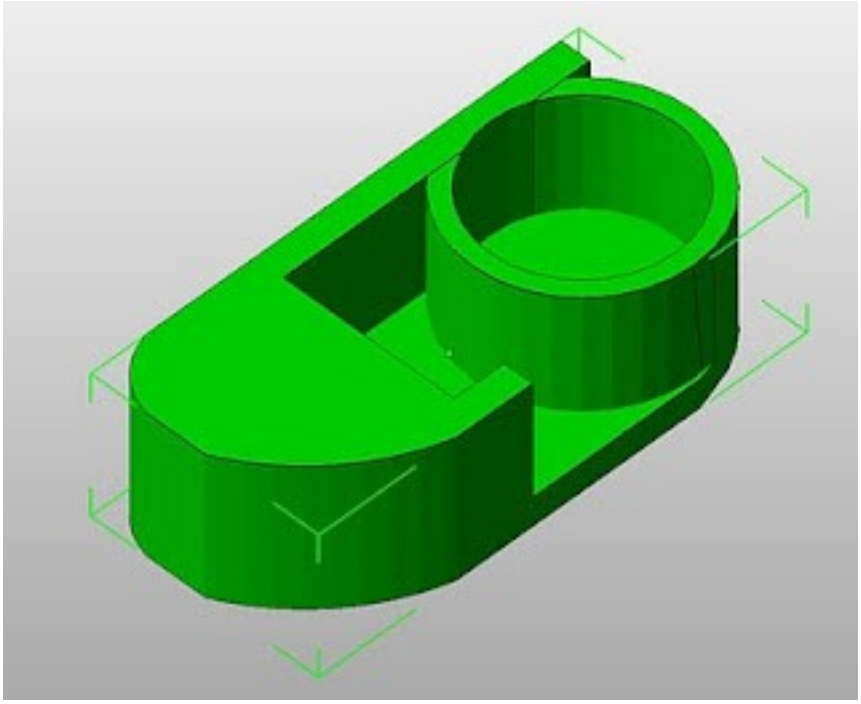
In which your narrator celebrates a return to craftsmanship made possible with the addition of a Reprap printer into the design process.

In the five years I've been on Reprap team I think that possibly the most terrifying and depressing transition of the many I've made during that time was actually *having* a 3D printer on my worktable and having to face up to turning some of my pipe dreams into physical realities.
Let me tell you, it's a *lot* easier to dream up something than to actually cobble it together and get it working.
To begin with I had discovered that thin-walled prints tend to be quite strong, properly designed, and don't have nearly the trouble with warping that we've had with large, filled objects.
For a project I wanted to create an animatronic hand for a telepresence robot project I've wanted to undertake for several years now. If you go out to and buy one you're looking at anywhere from $5-50K for one hand. I haven't got that kind of money so I dug around on the internet till I found a guy in Indonesia who'd made one for $300 in parts.



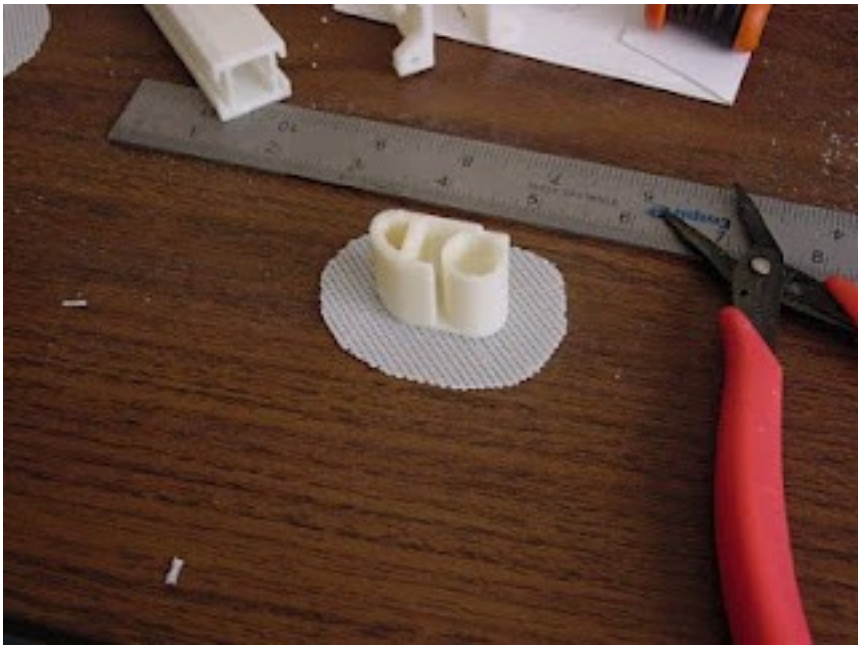
Andreas Maryanto's approach is both brilliant and cheap. I used his project as a starting point and was able to use his blueprints for getting proportions and sizes right.
I was also impressed with the Meka Robotics hand design.

Meka's hand was expensive, but had several interesting points. It was self contained and used elastic bands to return the hand to an opened position. I liked that.
I started small, with a fingertip. Actually, half of a fingertip. I did the design with Art of Illusion 2.8. AoI has always been about the easiest to learn, free 3D CAD app around. With the introduction of version 2.8 the problems with the boolean operators have been largely solved.

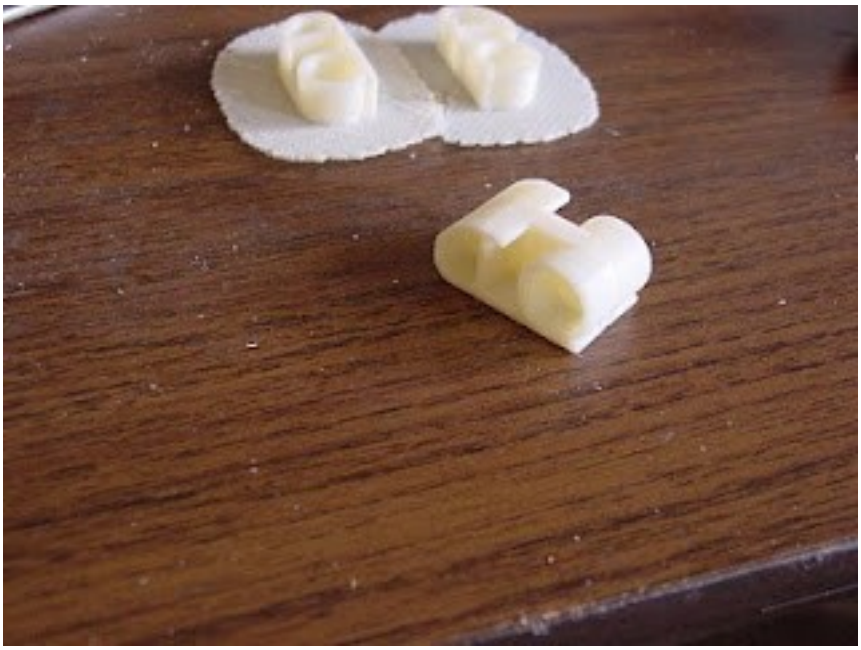


While AoI 2.8 will, occasionally, create a flawed STL file, these can be easily repaired in the free Netfabb Basic app. The combination of Art of Illusion and the free Netfabb app, now available on

the cloud as well, makes for an extremely powerful combination.
I wanted the joint between phalanges in the finger to be integral with the print, that is, not requiring any additional hardware. I printed and worked with the finger tip for quite some time.

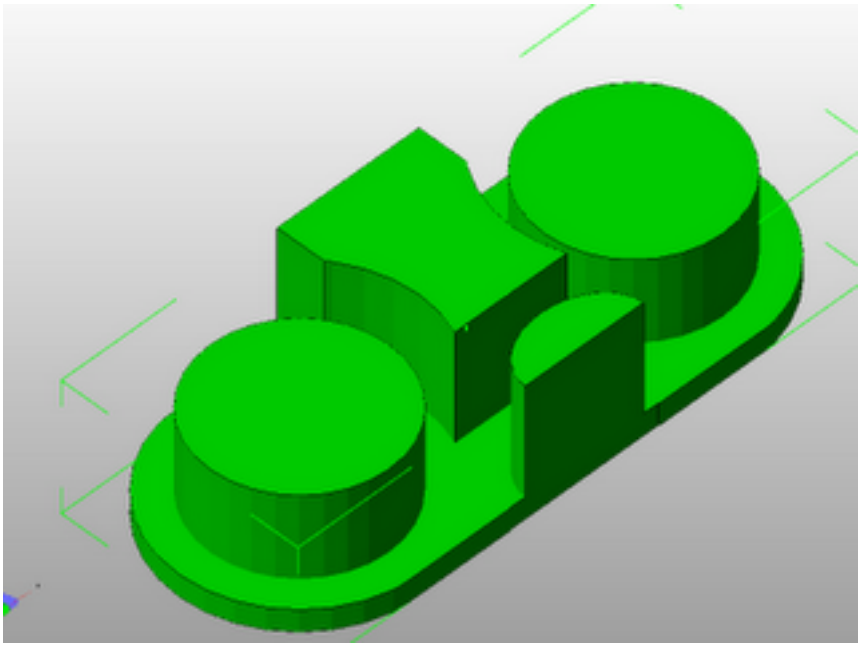


Eventually, after a few dozen prints and redesigns, I got it to do what I wanted.

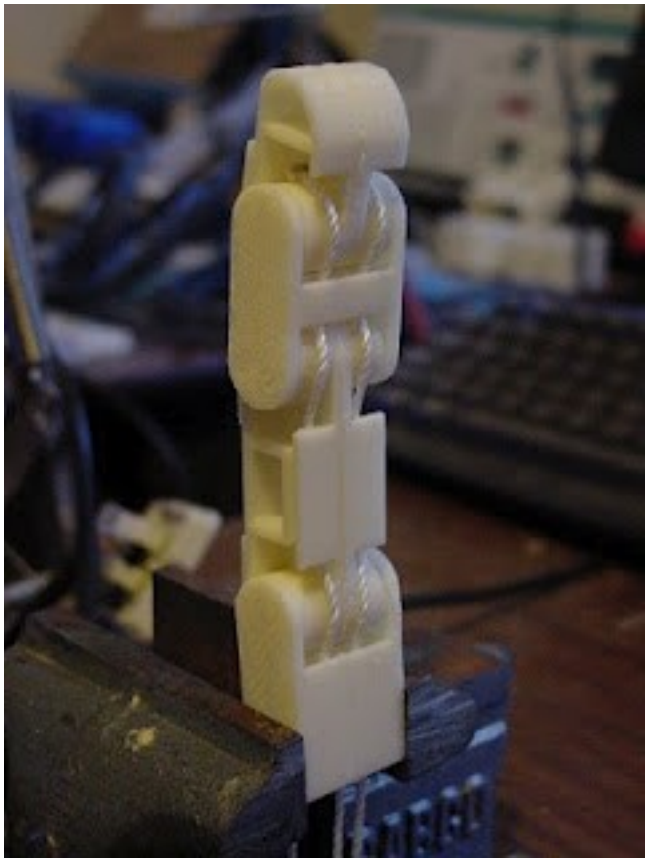


After that, the rest of the finger came together rather easily.

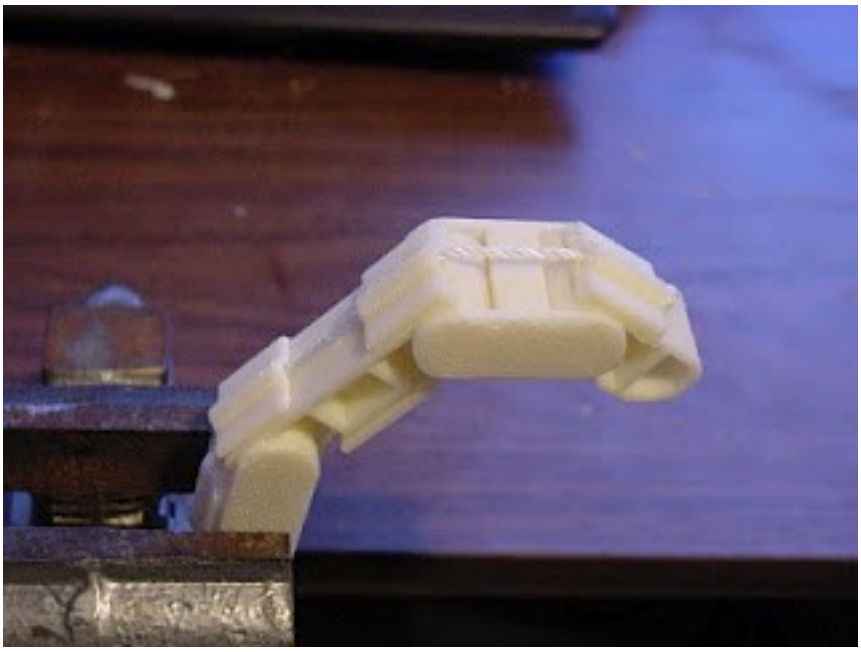
So, I had the joints right, but how to integrate the actuating tendons was a puzzle. Within a week or so I had a tentative solution. I could incorporate slots in the phalanges of the finger to guide and protect the tendons. This meant that I had to import the STL files for the parts back into AoI and carve out the slots to seat the tendons. Here you can see the STL for the second phalange slotted.



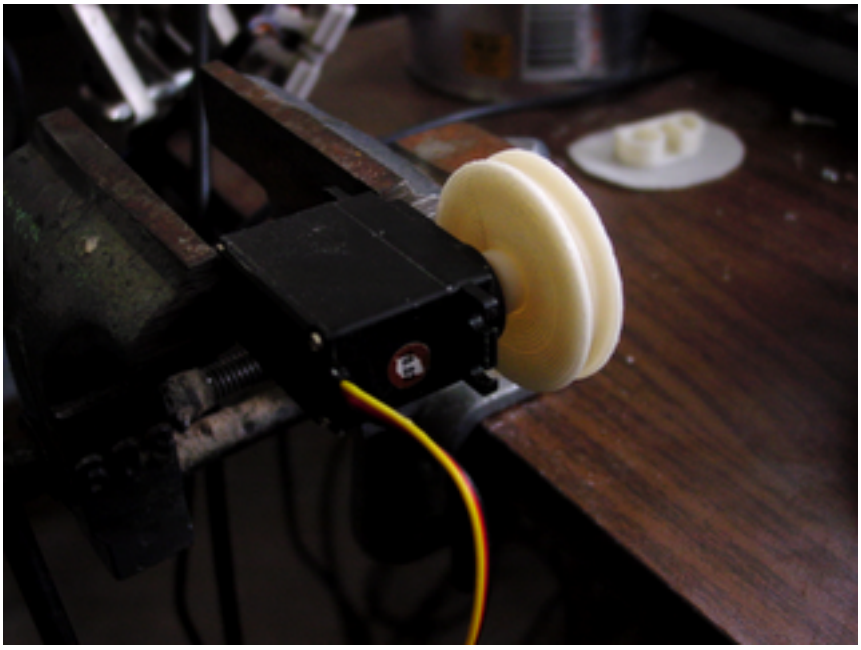
After slotting the second, third and fourth phalange in AoI and reprinting them I had a tentative solution for flexing the finger.



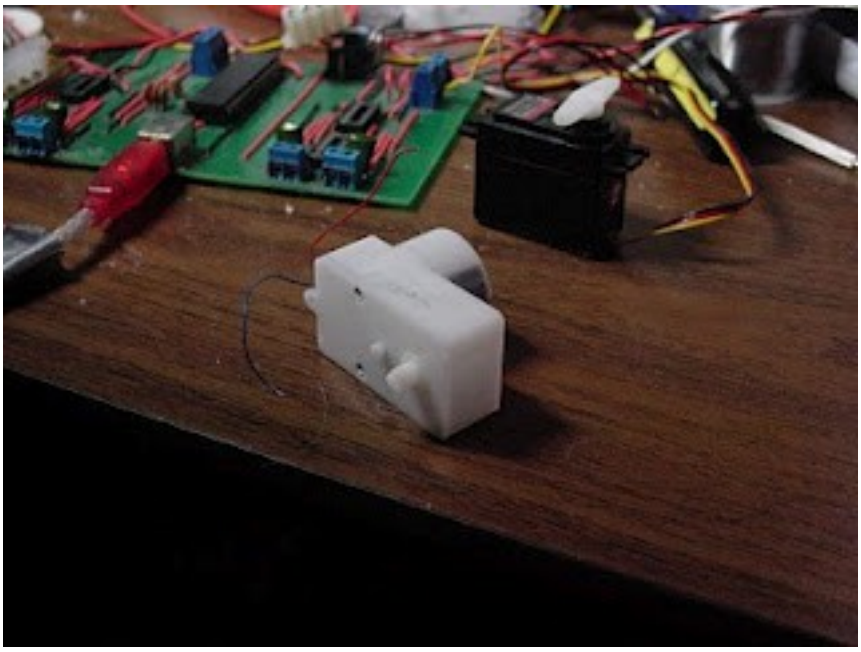
At that time I used a single tendon running through guides along the top of the finger. That was later to change.



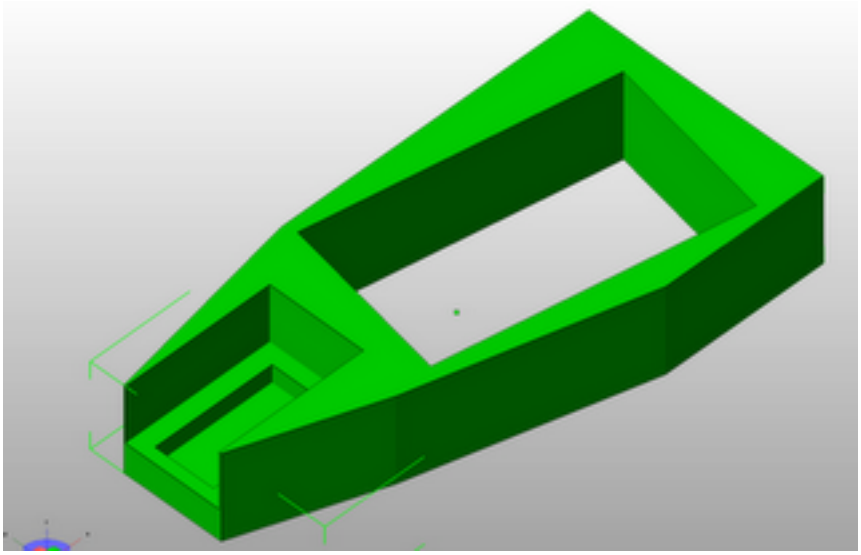
I quickly explored using servos very much like the ones Andreas had used.



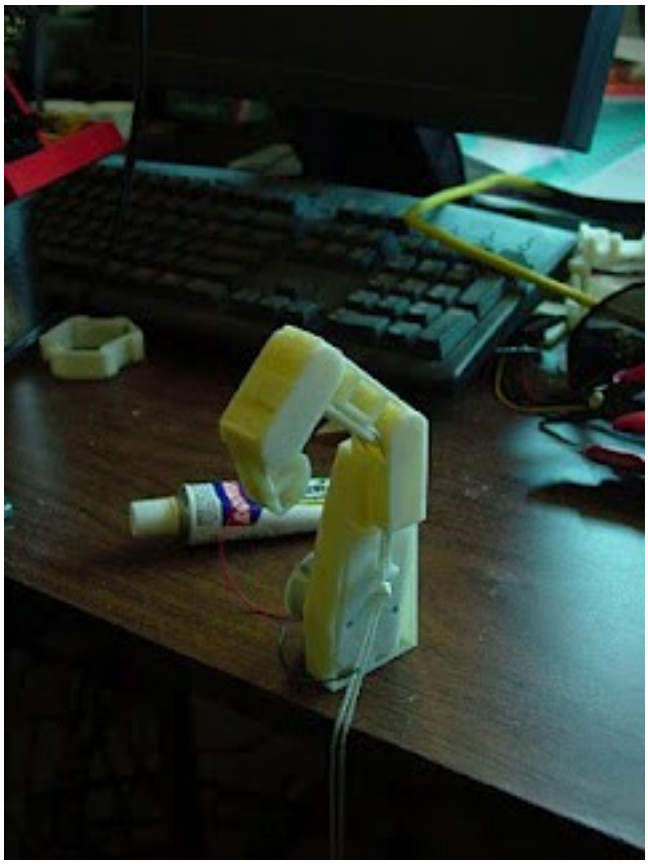
...and just as quickly rejected them as overspecialised and too bulky for use in my hand. From there I went on to creating my own servos from scratch using gearmotors and potentiomenters driven by a micro controller.
Once I'd settled on a gearmotor that I had a supply of...



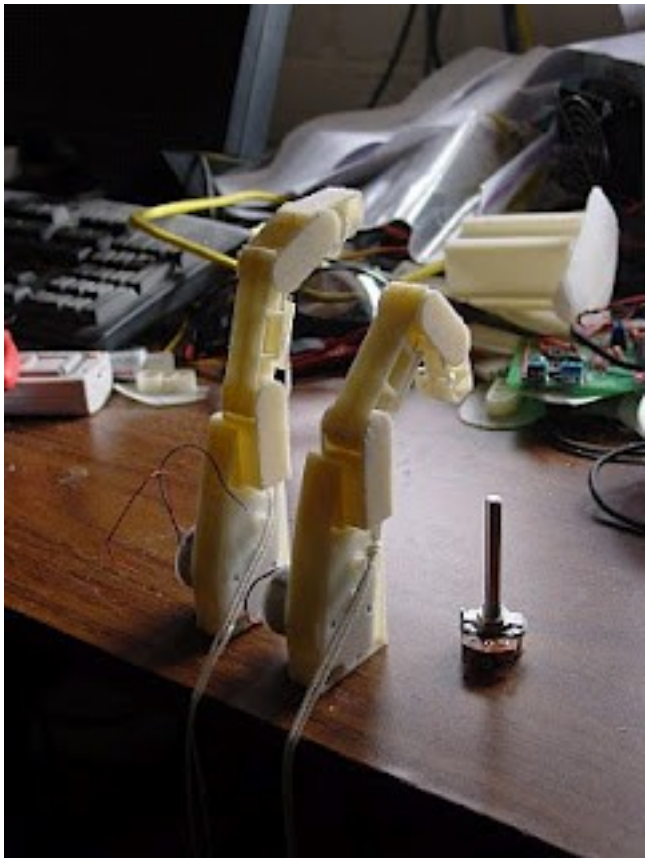
...it became a matter of designing a section of the actual hand and accommodating the gearmotor inside.



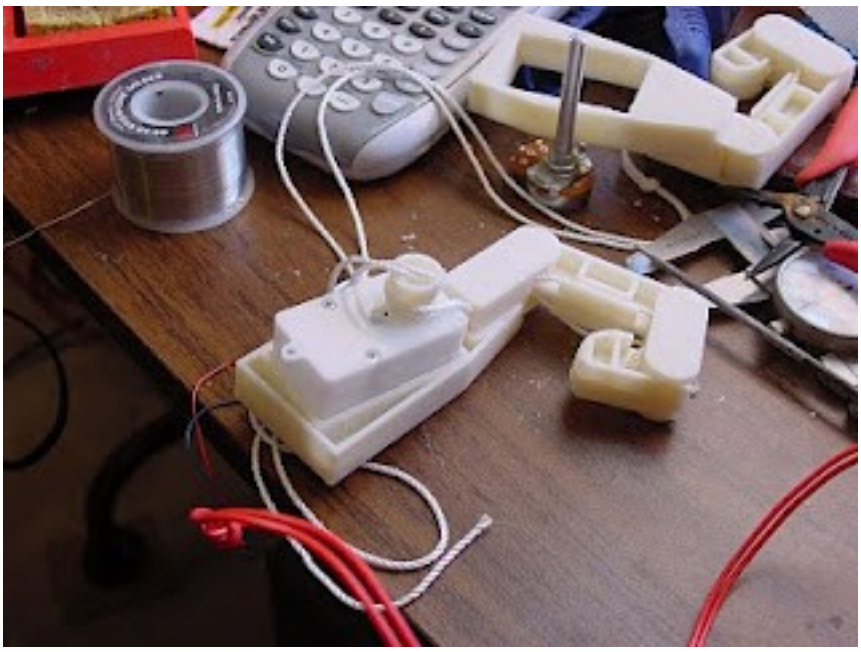
Notice that the design of this part is only aiming at holding the outline of the gearmotor and providing a recess for the finger. The goal is to work out the seating of critical parts and getting the general shape of the part right before trying to accomplish more.



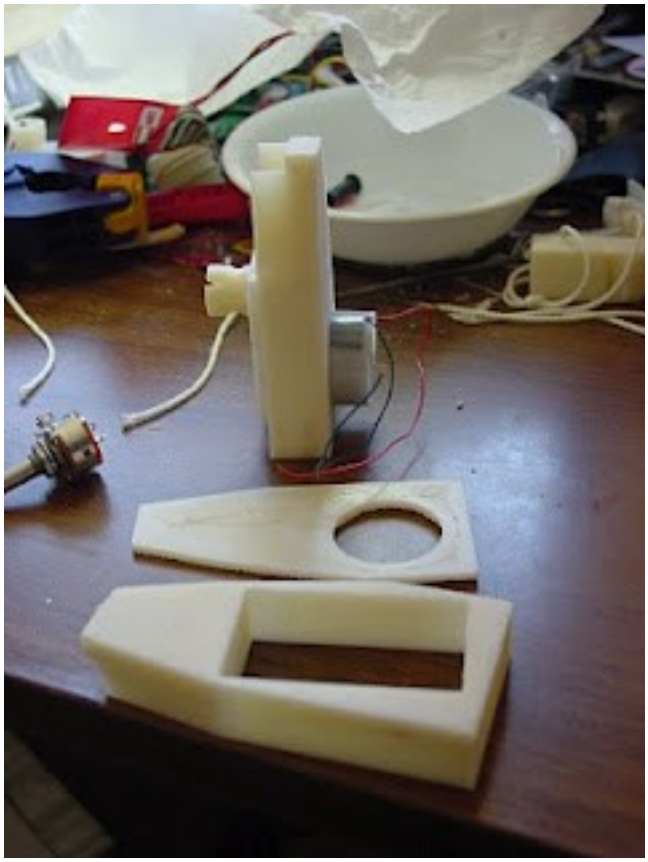
At that point, I needed two finger assemblies to get a feel for the clearances and volumetrics between finger ensembles.



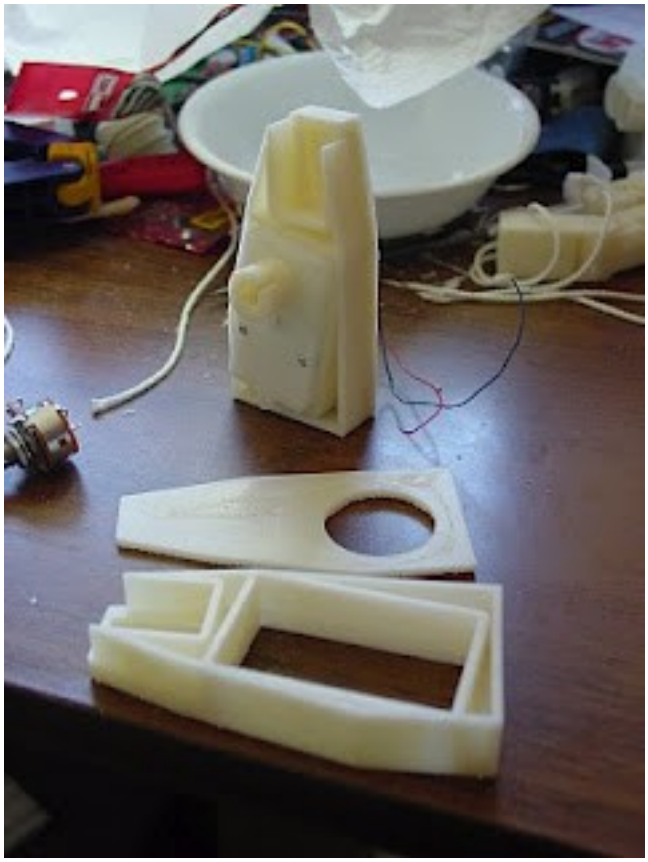
Attention to the reel that would wind up the tendon that contracted the finger was the next step. I'd decided by this time to use a Meka-style elastic band on the top of the finger to return it to the open position.
The reel was a tiny, finicky part, given the clearances in the hand section. I printed two of them at a time to avoid having to do a lot of pausing for the reels to cool between printed layers.



That done, I needed to do a bit more work with the hand section. First, I needed to seat the gear motor. The hand segment was a big, rather complicated part, so rather than trying to redesign that I simply designed a plate that could be glued onto it that would achieve the effect.

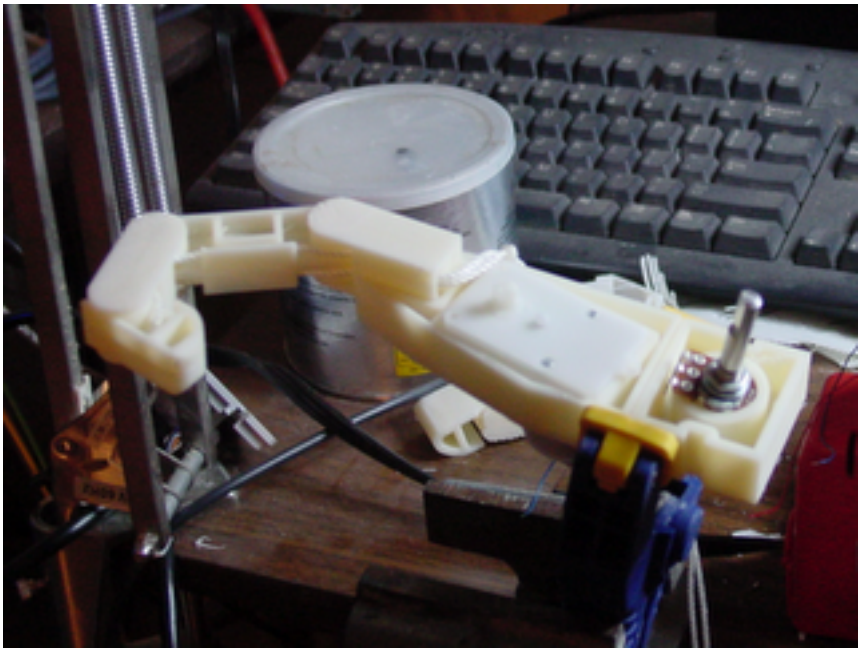


By this time I had slotted the reel to make attaching it to the nylon tendon easier.



While I glued that gear motor support plate to the hand section, I could just as easily do a boolean union in Art of Illusion between the two parts and print them as a single part, or not, as convenience dictates.

From there, I designed a housing for the potentiometer that butted onto the base of the hand section. You can see it on the left-hand side of this picture.



Notice how I haven't bothered gluing this part to the bottom of the hand segment but rather just clamped it. I've not settled on all the dimensions and placement of this part yet, so gluing isn't justified.
Similarly, you can see here that I just took the hand section into the free Netfabb app and flipped it into a mirror image as a start towards completing the hand segment.

The point of all this is that you can take on some really challenging design exercises if you will just

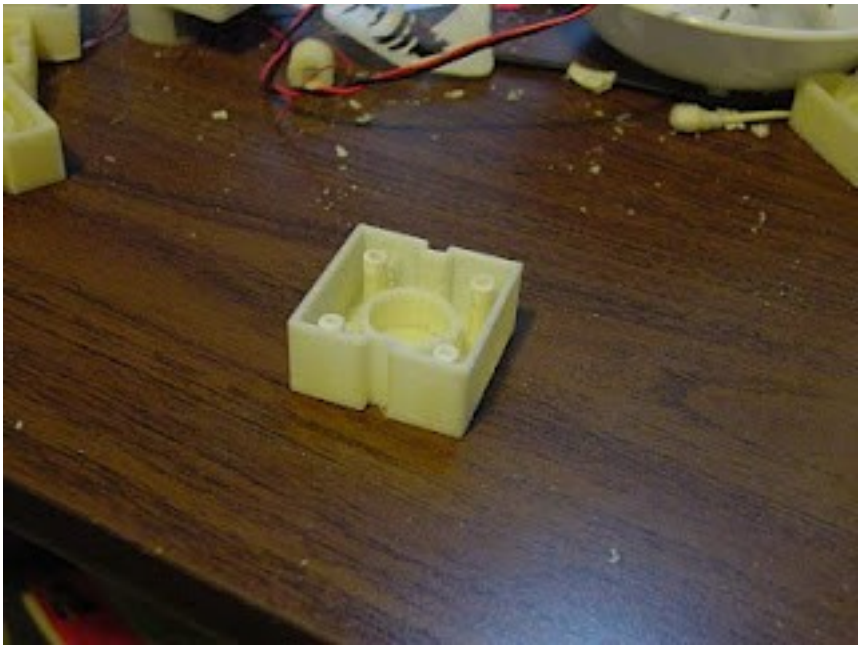
take them a little bit at a time. Having a Reprap 3D printer {in my case a Rapman 3.0} makes it possible to print out a partially designed part and try to match it up with the bits that go in it and the bits with which it must integrate.
While you can do all that in a 3D CAD system you have to have a very good sense of volume and space to do so. With the Reprap printer in the mix, you don't have to be that good.
Don't be afraid to use filament during the design process. You're not wasting it.

Making a lid for a potentiometer mounting box

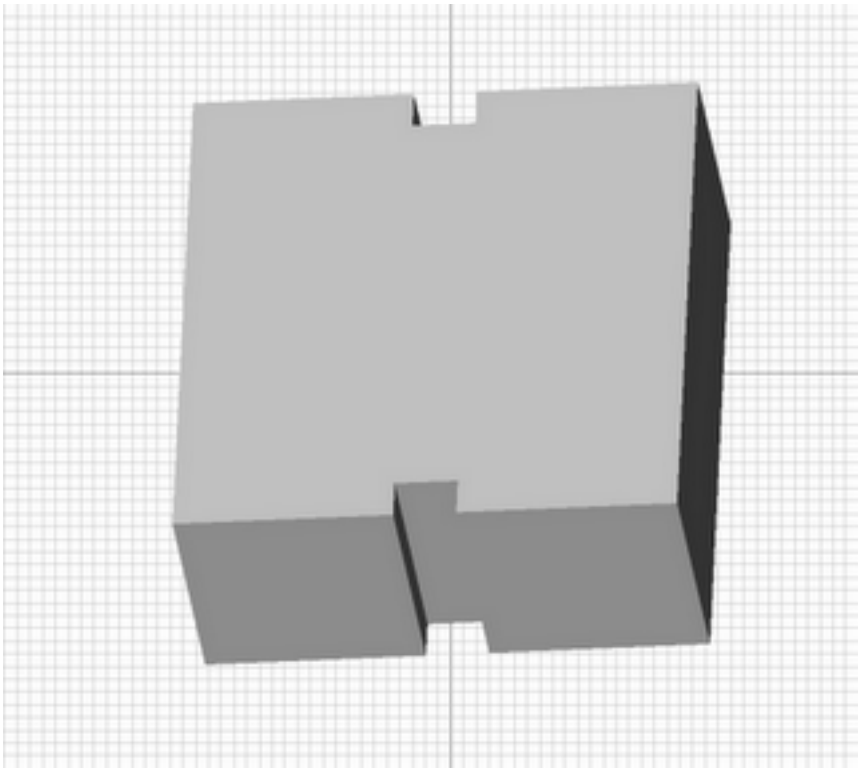
Saturday, 28th August 2010 by Forrest Higgs

In which your narrator shows you a few tricks of the trade with thin-walled prints.

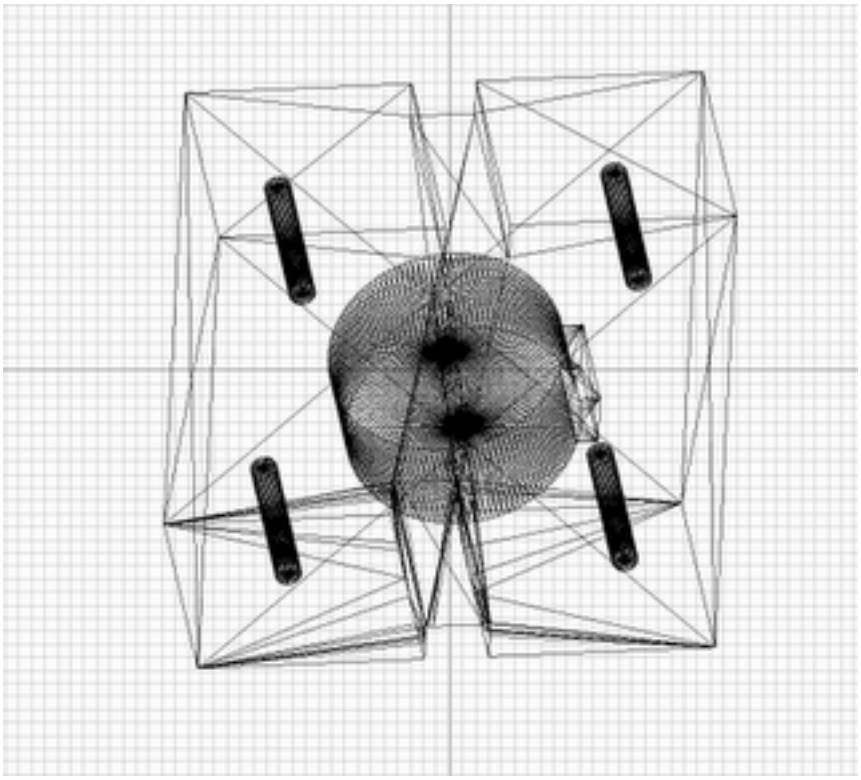
In my telepresence robot hand project I'm not using off-the-shelf servomotors but rather making them from scratch. A servomotor basically consists of a gearmotor, a potentiometer and some electronics, in my case an MCU.
You saw in my previous posting how I'd developed the hand segment and sorted out the gearmotor positioning therein. That done, I had to turn to see to mounting a potentiometer. I decided to simply add an extension to the back of the hand segment to contain the potentiometer. I'll not go into detail about the design of the box itself but rather cover the trick one uses to design the lid for the box. Here is the box.
The mounting box is nothing special. It's a simple square box with mounting rack groves in the sides, seating for the potentiometer and screw posts to secure the lid.



The box is simple to design. It took me about seven design iterations to make everything fit properly and to get the proportions worked out. Using the Reprap 3D printer at every step meant that it was a very low risk exercise.
The box itself in Art of Illusion 2.8 was a simple box in which I'd removed the mounting slots on the sides with two boolean ops.

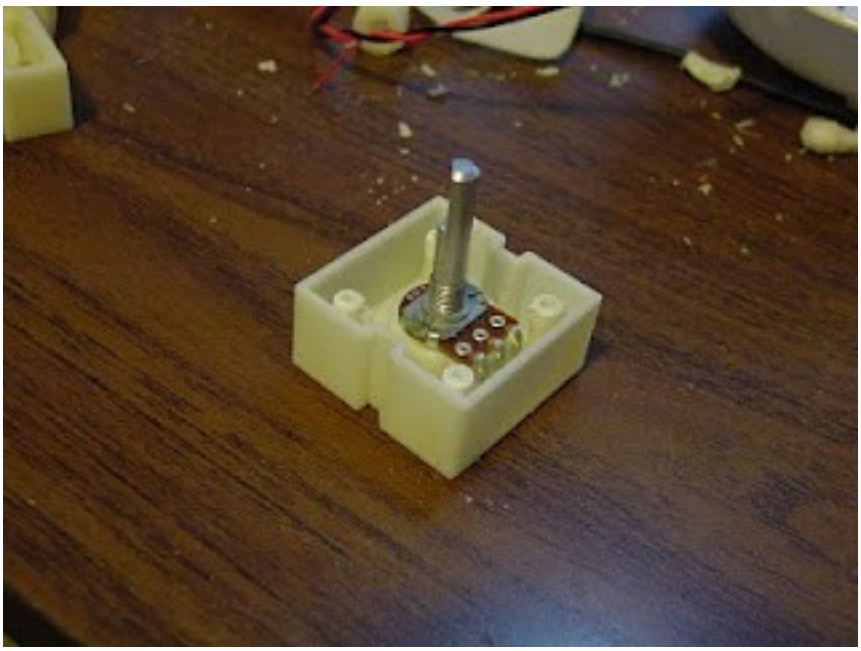


Once that was done I simply removed the voids where the potentiometer would be seated and where the mounting screws would be fitted. I've switched Art of Illusion over to wireframe mode so that you can see the voids since they don't penetrate the surface of the mounting box.

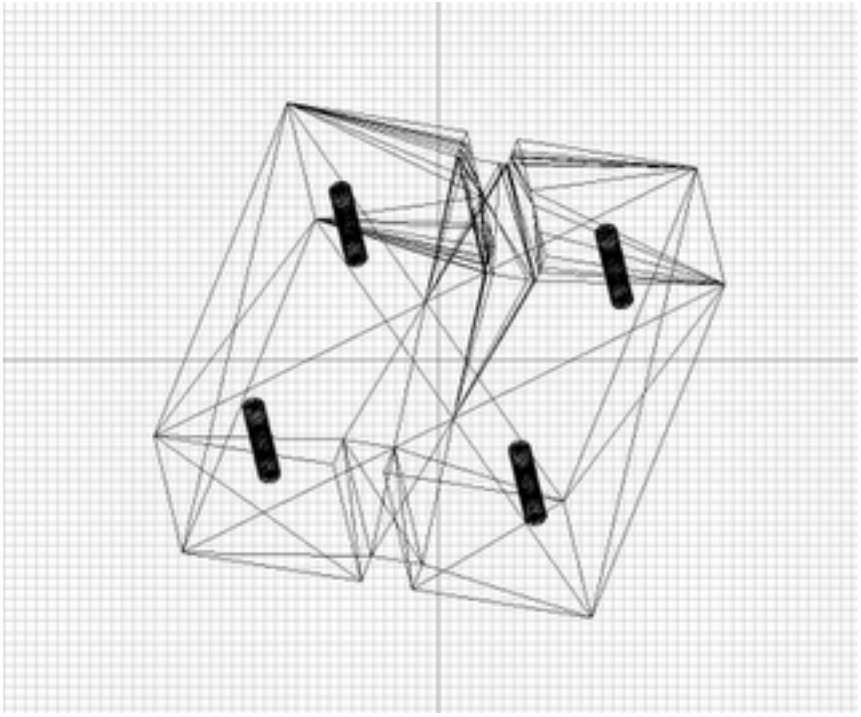


Designing thin walled parts is a bit like designing an old photographic negative. You design the skin of the object and the holes inside of it rather than trying to design the object itself. When you print the object you simply don't use any infill. The top of the box is just another kind of infill, so it is omitted, too.

Here you can see how the printed box seats the potentiomenter.



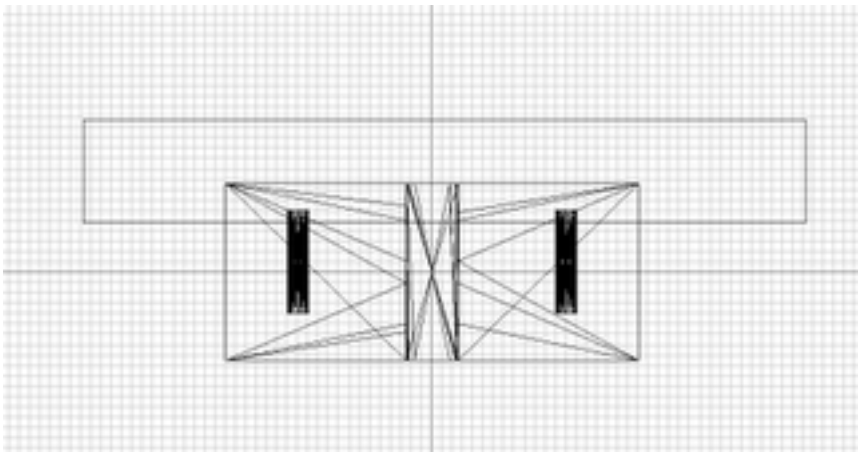
Now, designing the lid you simply take the Art of Illusion file for the box and leave out the void that seats the potentiometer.

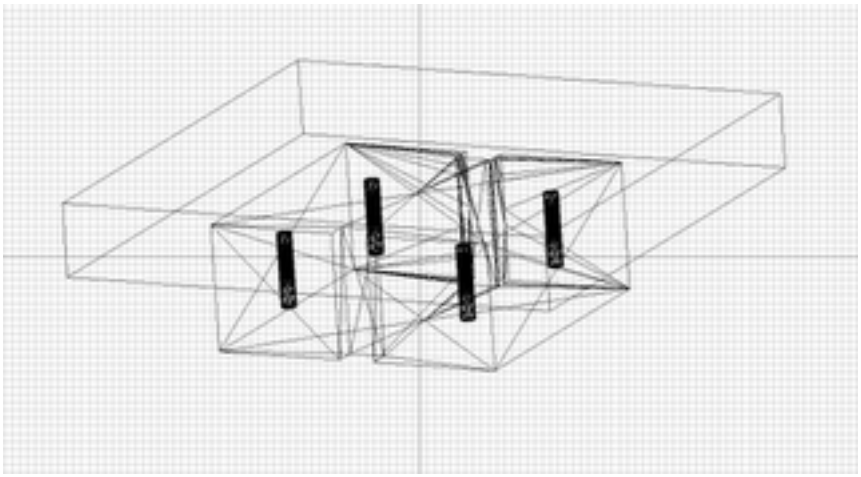


You want the lid to have holes that match the screw posts in the box so you leave those voids in.

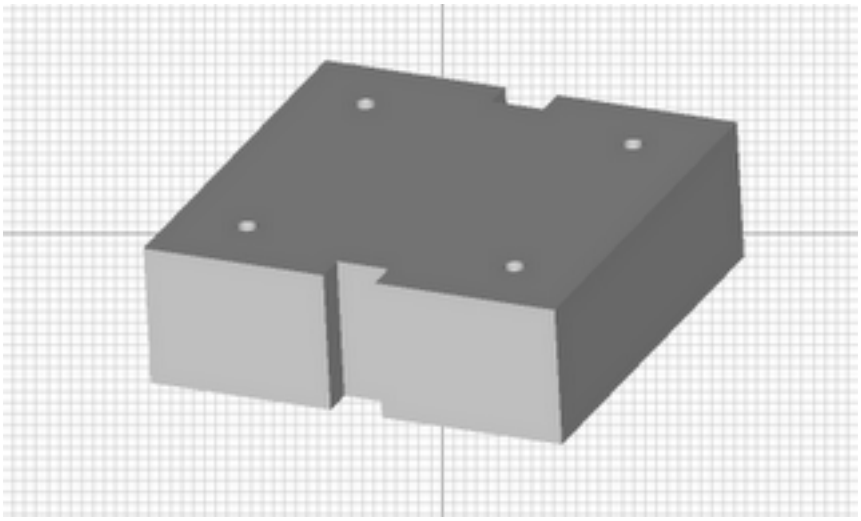
Next you slice off the top of the box leaving the screw hole voids exposed. You do this in Art of

Illusion by simply creating a block and merging it with the top of the box down to where the screw voids begin.

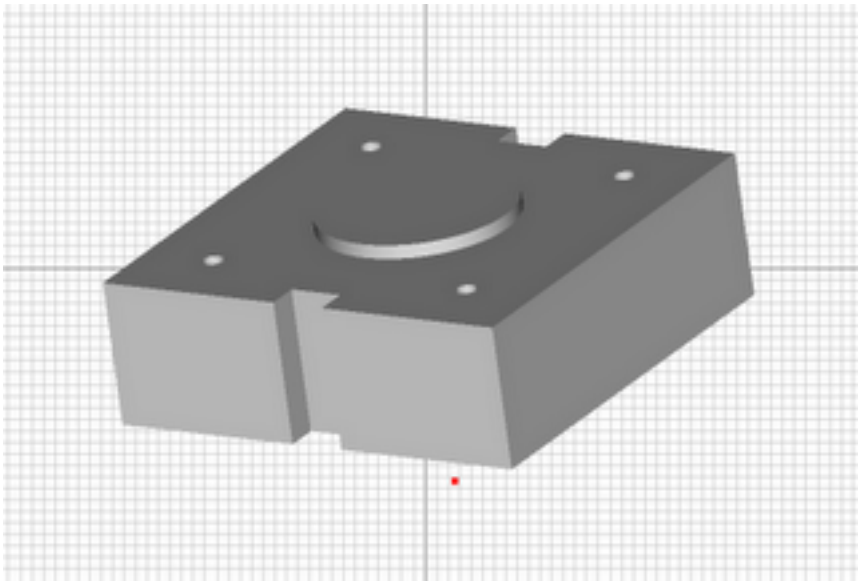




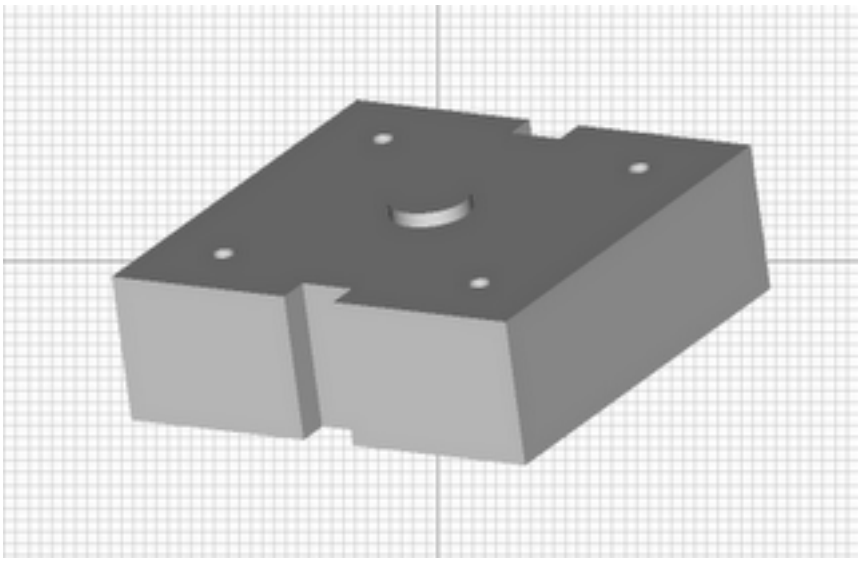
Then you remove the top of the box using Art of Illusion's boolean ops function.



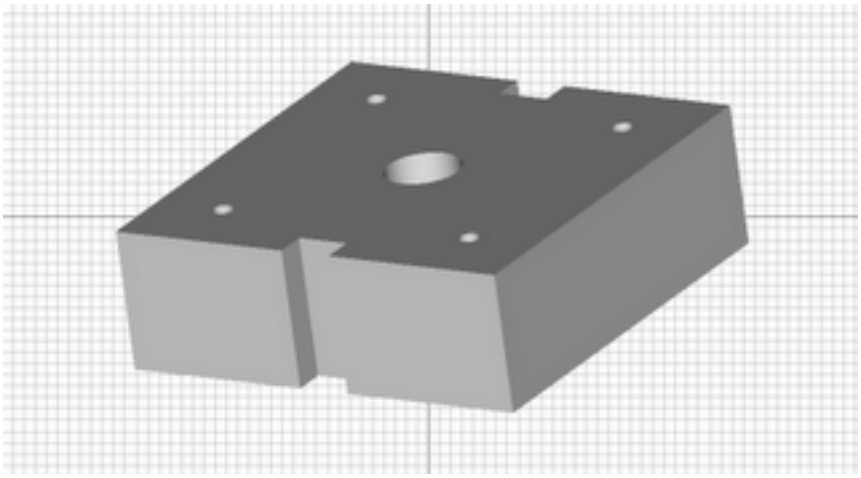
While you don't want to have a hole big enough to slip the whole potentiometer through, you do need to accommodate its shaft. A simple way to do that is to simply take a copy of the seating void cylinder and put it back into the object as a solid rather than a void.



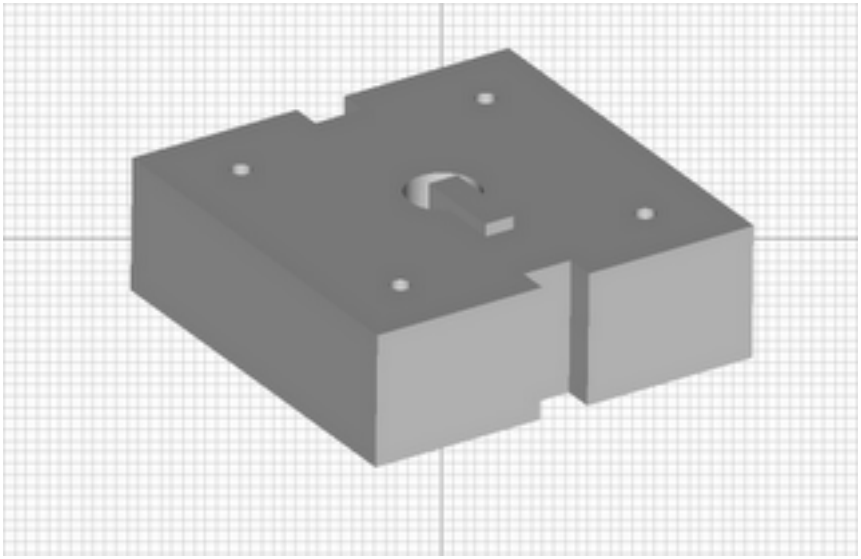
Since we've sliced off the top of the box you can see the top of the cylinder now. Now it's just a simple matter of reducing the radius of the cylinder to a bit more than the radius of the protruding shaft.



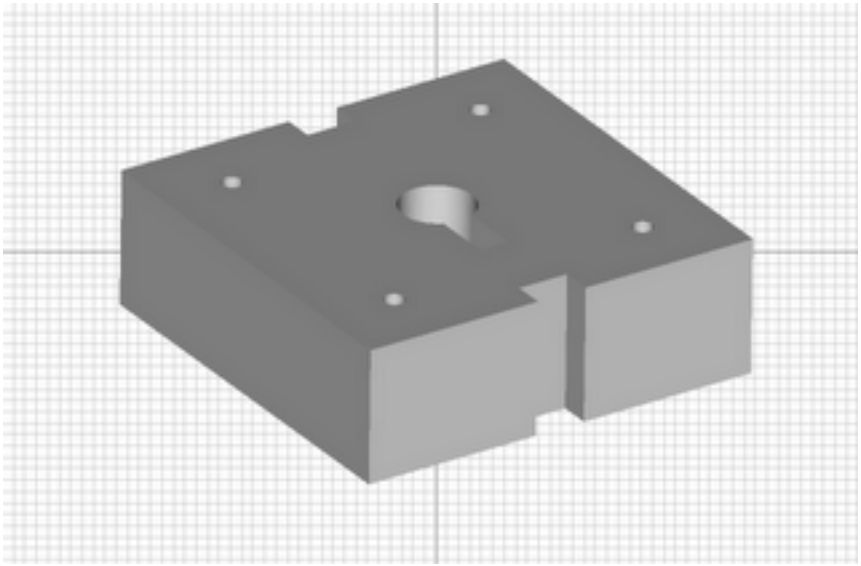
Now do a boolean op to remove that cylinder so that you there will be a hole in your lid.



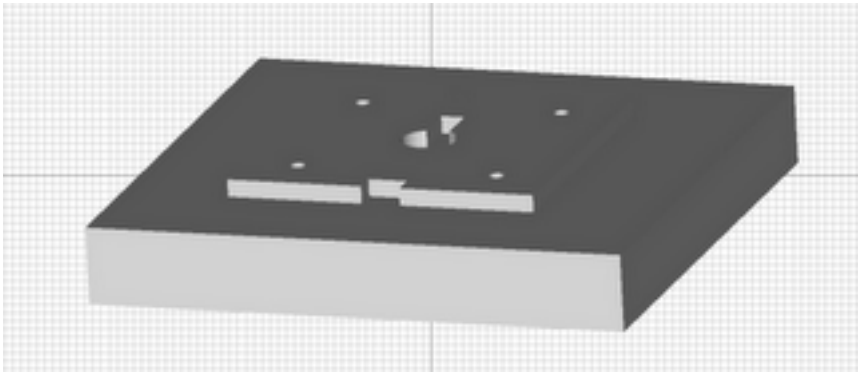
When I looked more closely at the potentiometer I noticed that there was a little metal tab on one side to act as a stop to keep it from rotating around its shaft. I had to design a little slot into the lid to accommodate that tab. I did that by simply locating a little block where I wanted the slot to be.



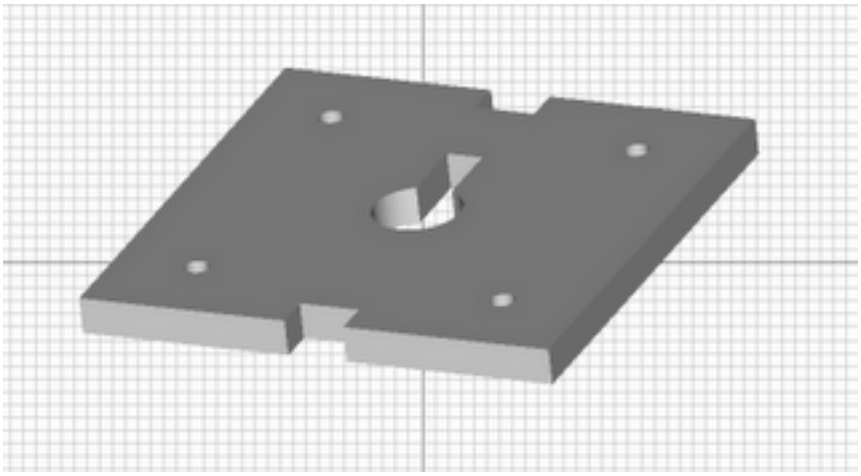
Then I did a boolean op to remove the space occupied by the block.



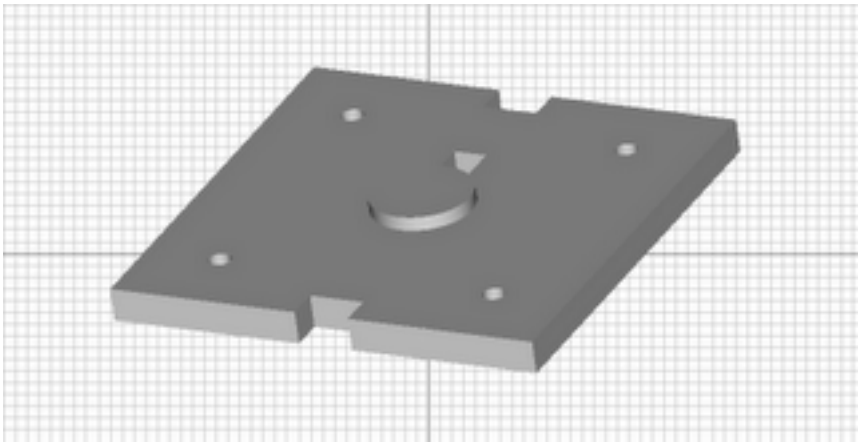
At that point I brought back the cube I used to slice off the top of the box and moved it down so that exactly the thickness of the lid was exposed.



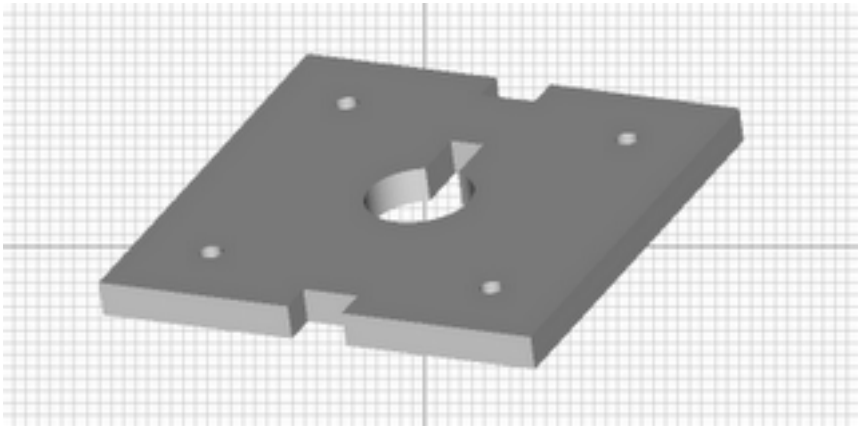
I then did a boolean op that chopped off the bottom of the box so that only the lid remained.



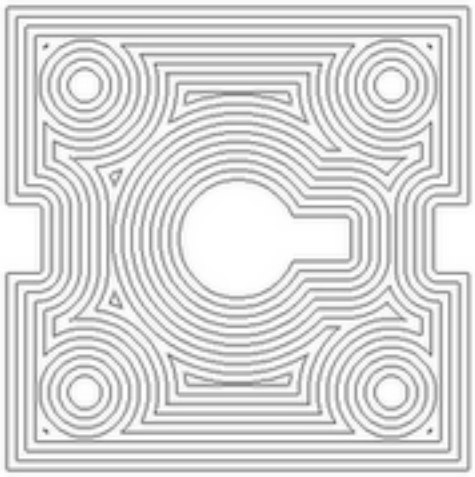
At this point I discovered that I'd made a mistake. I'd designed the hole in the lid so that it fit the shaft. Actually, the shaft fitted into a seating hub in the potentiometer that stuck out about a millimeter. I brought back in the shaft cylinder that I'd used a moment ago and widened the radius to a bit over the radius of the hub.



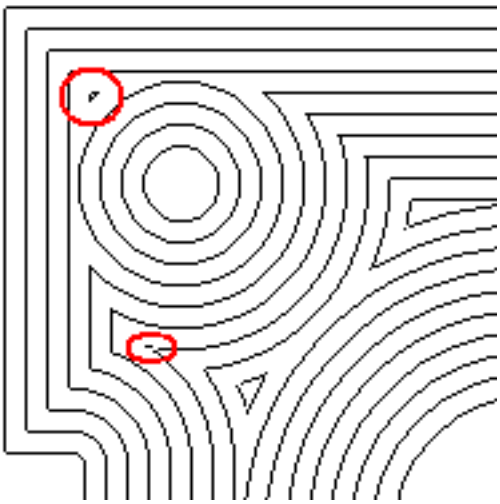
I then removed that cylinder via a boolean op which widened the hole to accommodate the seating hub.



Now comes the tricky part. The lid as designed will sit on *top* of the box. I want it to recess *into* the box. I could try to do this using Art of Illusion, but there is a much simpler way much less likely to fail.
Remember that the box will be printed without infill. In this particular case we will print it with two 0.8 mm print roads describing the perimeter with a thickness of 1.6 mm. Print roads tend to be rounded on the inside of the box, a fact that makes working them with Art of Illusion a bit messy. It is easier to just process the lid in Slice and Dice.
Once you've done that you simply go into the Filled folder and pull out the image of the print roads for the lid.



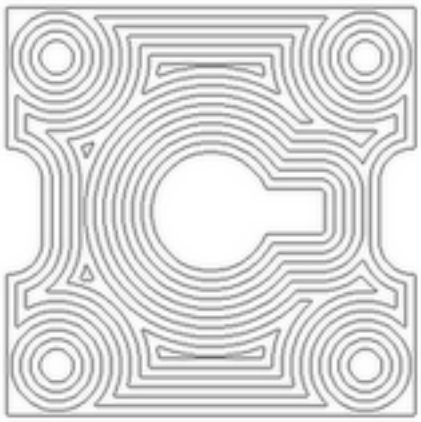
In the present state of development of Slice and Dice with complex prints roads like this it is necessary to go in with Windows Paint and clean up the print road image a bit. Here you can see a few flaws in the roads image.



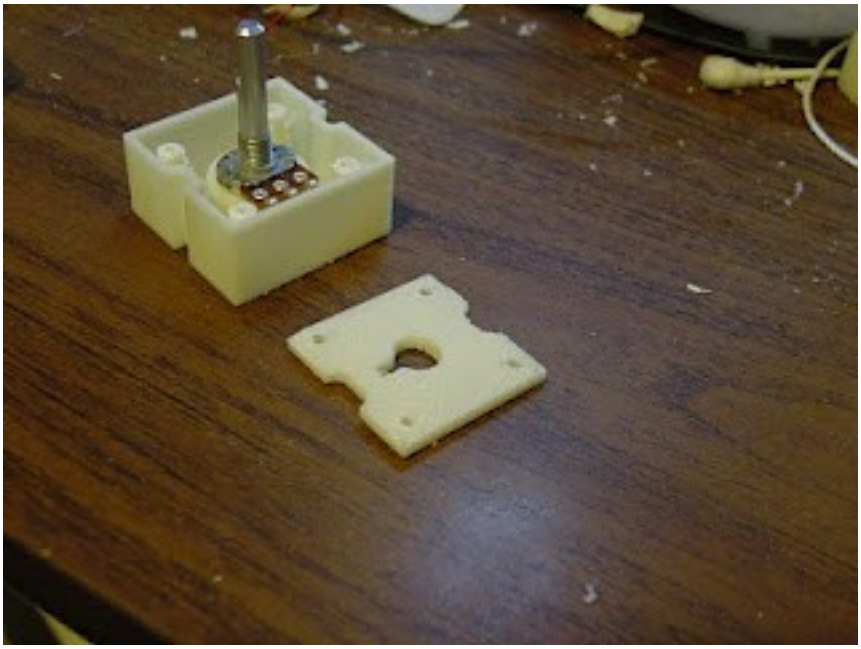
I've circled two of the most obvious. In the upper right hand circle we have a feature that is less than 1 mm in diameter. The lower circle encloses a loose print segment that is too small to print. It's best just to remove those in Paint. You will also, when inspecting the print, note that several of the print road loops are broken by a missing pixel or have tags of pixels hanging on. Those should also be fixed or removed.

You usually only encounter this kind of problem on slices with very complicated print roads. That's not a big deal and it is a quick thing to fix in Paint. Since Slice and Dice processes images rather than arrays of numbers at each step you can go in and make changes with Paint if there is some flaw that you want to fix. You can even alter the print roads in Paint if that suits you.

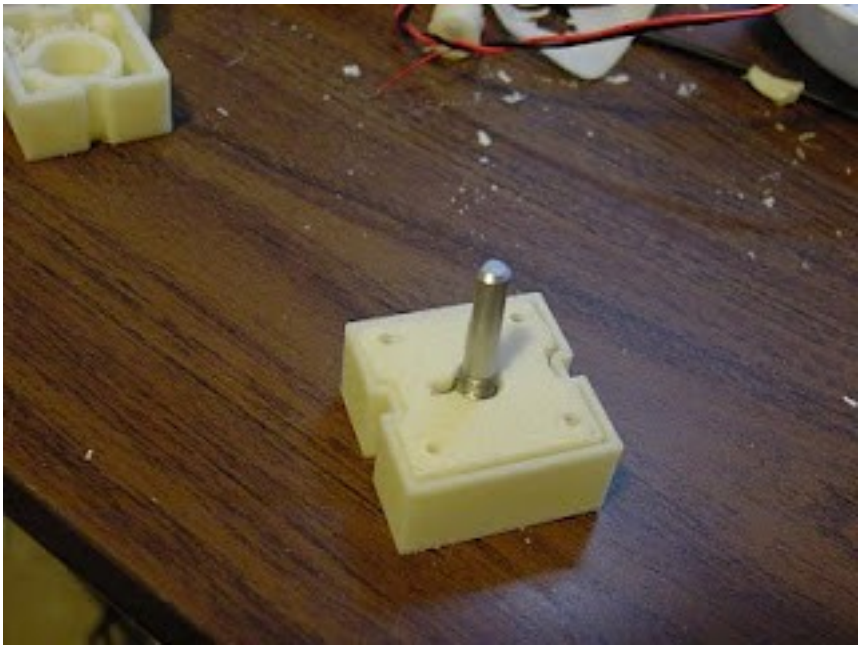
Altering print roads is exactly the trick we will be using to make the lid recessed. The box has a perimeter of two print roads. We simply go into Paint with our lid and erase the outer three print roads.



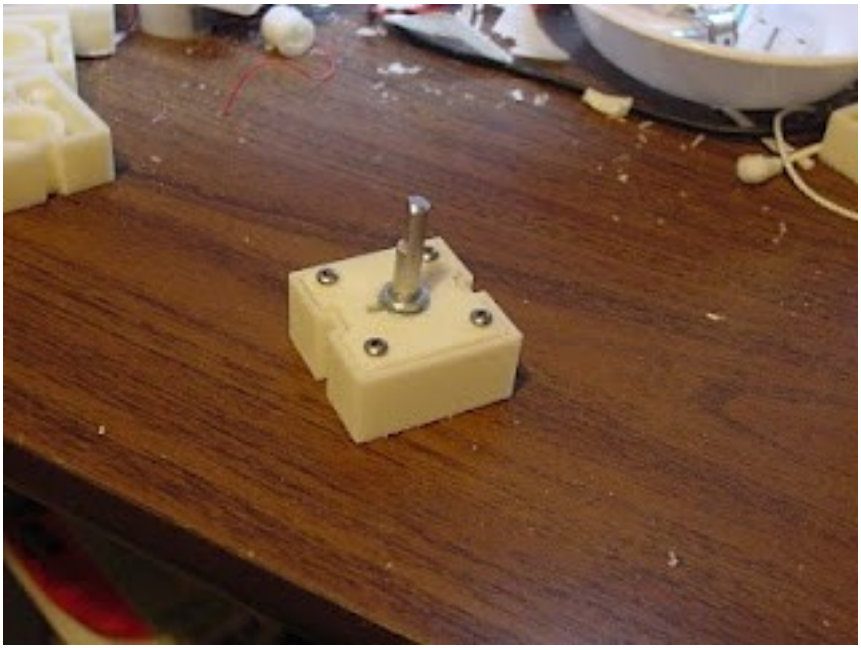
That lets the print box lid fit into the box and sit atop the screw posts inside. We could have simply removed two. In practice, however, that requires a bit of touching up with fine grit sandpaper on the edges to get the lid to fit. Removing three roads leaves a 0.8 mm gap between the box and the edges of the lid. That's good enough for this job.
Here you can see the potentiometer box with the newly printed lid lying beside it.



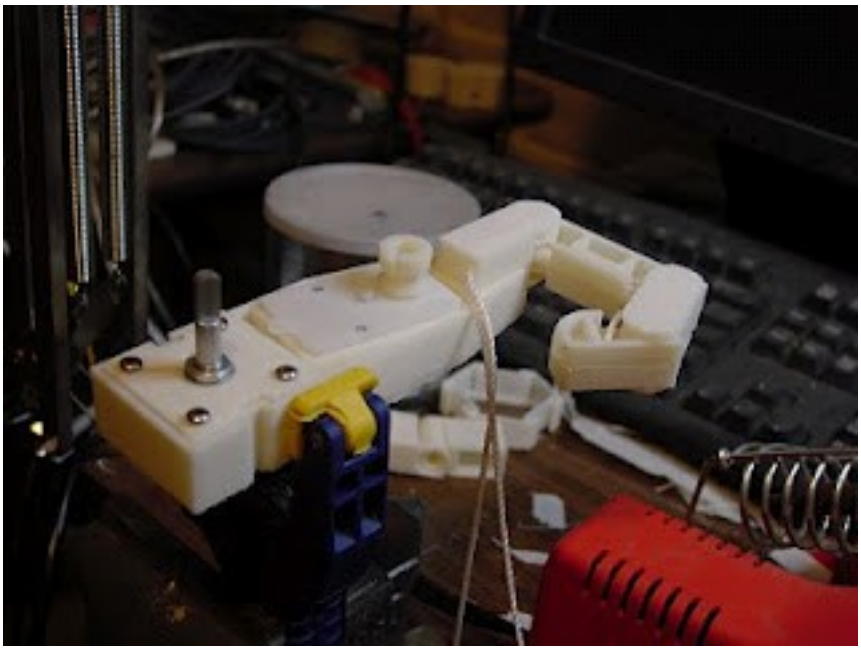
Now you can see that the lid fits nicely over the potentiometer.



After that it's a simple matter of securing the lid with metal screws and putting on the potentiometer's washer and nut to complete the job.



Now you check the fit between the potentiometer box and the hand segment.



You are now ready to connect your potentiometer box onto the back of the hand segment with boolean union ops in Art of Illusion and then print out the resulting large part.
This is how you create a large complicated part by designing it as a set of smaller, simpler parts.

Wondering what the fuss is about

Monday, 6th September 2010 by Forrest Higgs

In which your narrator wonders what all the fuss is all about?

I guess I just don't get it. When you work with a plastic, after a while you get a feel for what it can do and what it can't. I started getting along along quite happily with no heated bed and ABS.

I print with a 0.3 orifice at an axis speed of 16 mm/sec. I suspect that I could kick it up to 22 mm/sec without a lot of drama, but I don't want to take the time out to play "who can print fastest" games at this point.

Once I started doing thin walled pieces and no infill my warping problems virtually disappeared. Most of the pieces I design have a largest dimension <= 90 mm though I've printed a herringbone rack that was 250 mm long. My acrylic print table temperature stays at about 25-30 degrees. You also don't find my prints warping after a few days from the internal stresses that Bogdan has talked about. I've seen that with HDPE. I was also printing with cross-hatched infill in those days, too.

I got into thin walled, no infill after I realised that if I went that way I got pretty fast prints that way at lower print head velocities.

Watching you guys reminds me of the first and second year architectural studio students that I used to lecture to back in my university professor days. They'd make models of buildings out of either "shipboard" {2 mm solid cardboard} or carve them out of expanded polystyrene foam treating it like it was so much cool butter or cheese. We used to say "form follows chipboard". If you followed the careers of those students they usually spent the first five years of their careers designing actual building that got built that looked as if they'd been carved out of cool butter. That's an extremely expensive way to design and build buildings. Those guys either got out of the habit fast or wound up doing architectural detail drawings for other designers who'd developed a feel and respect of the potential and limitations of the materials that went into their buildings.

Around here I see parts designed like you were more used to using an expensive CNC milling machine to carve parts out of a block of steel or aluminum.

That looks really cool but begs the fact that you're not taking advantage of the strengths and avoiding the weaknesses of the material you're working with, viz, plastic and extruded plastic at that. Somewhere along the line with Reprap we got the idea that we ought to be able to design parts in any shape we wished and whatever the material wanted to be be damned. You can see the trouble it's caused us in the chase after things like heated beds and support materials. We're spending a lot of time on that chase when we could be designing killer apps that make having a much simpler reprap machine very desirable.

That's just an old architectural technologist talking, I suppose.

I've included a link of the telepresence hand that I'm currently designing. That is human sized, btw. The largest part dimension is a touch over 80 mm. No warping whatsoever on any part.

Thin walled, no infill, snapped or glued together. There'll be a few screws in it to secure some elastic bands that return fingers to their rest positions. I haven't figured out how to secure elastics bands with snap on parts or glue yet. I'm thinking about how that might be possible all the time, though. :-)

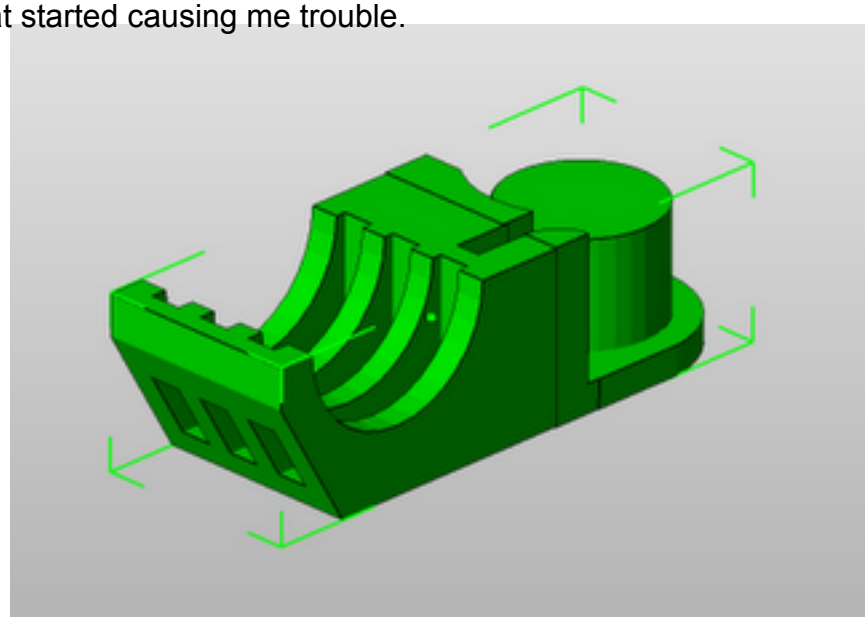
Changes to Slice and Dice

Wednesday, 8th September 2010 by Forrest Higgs

Slice and Dice was fine as long as your parts were more or less continuous along the z-axis and not so complicated that you got a lot of clashing with print loops. When I got to working on the thumb joint on my telepresence hand, however, I started having a lot of trouble with both of those limitations.

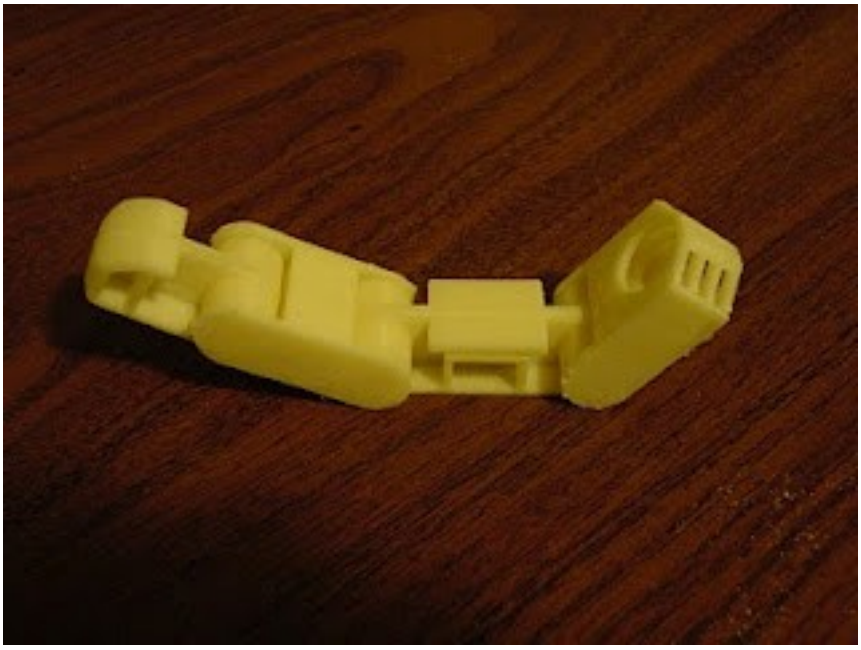
A major strength of Slice and Dice is that if you have a dodgy STL file you can clean up little imperfections on the slice images. That's wonderful until you have a part that you want to print that has little in the way of commonality between slices. I found myself fixing faults on 30-40 slices in Paint. That was seriously not fun. The main problem, as it developed, lay in the implicit dependence of Slice and Dice on looped road descriptions. Once you get into complicated parts it becomes very difficult to meet the app's expectation of clean looped print roads.

This is the part that started causing me trouble.



Virtually every slice is unique.

I rewrote the road making routine to deal with non-looped roads this evening. This is the resulting test print of the part shown above.



A problem that the old Rapman firmware was that it expected minimum print road line segments to be about 0.6 mm long. When I finished the rewrite, I was not looking forward to fitting line segments to the non-loop print roads. I had the output, but it was all 0.1 mm print roads.
Andrew at BitsfromBytes has said that the updated firmware would handle 0.1 mm roads with no problem. I frankly didn't believe him because the older firmware would slow the print down to 6 mm/sec with 0.1 mm roads. I decided to give it a try with the 4.0.2 code, however, and it turned out that Andrew is absolutely correct. The Rapman firmware has no trouble print sequential 0.1 mm line segments at my chosen print speed of 16 mm/sec.
That made the print file for the two halves of the thumb joint some 8 meg long. Since I have a 1 gig SD card, however, that's no trouble at all.

Acid test

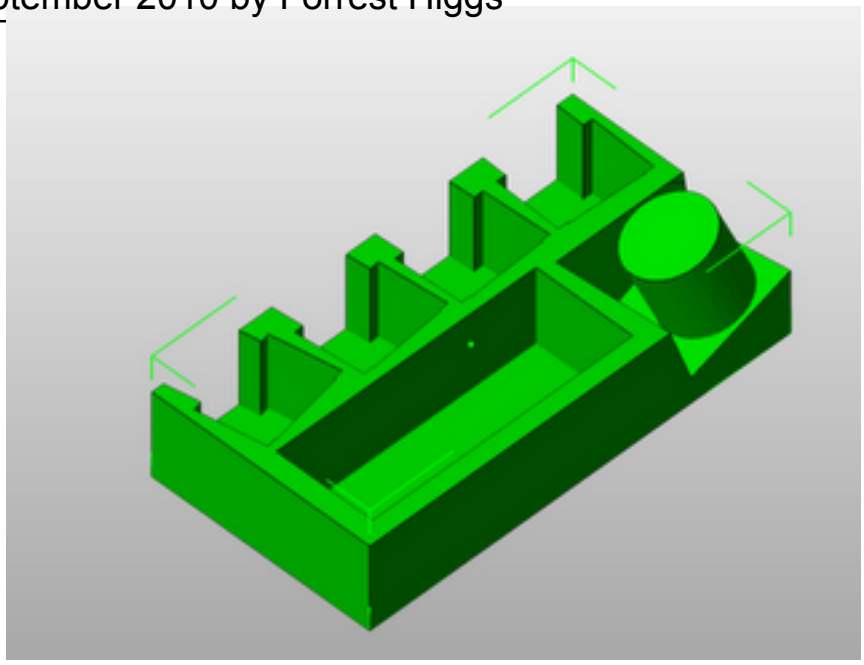
Thursday, 9th September 2010 by Forrest Higgs

I decided to give the new non-loop road finder routine a tough workout to see how robust it is. For that I designed a 20 mm diameter herringbone pinion gear.

I did a solid print for strength. As you can see, there were no problems. The stl's processed without drama and the gcode is good.

Second go at the carpal ensemble

Friday, 10th September 2010 by Forrest Higgs



Dealing with detaching rafts

Thursday, 16th September 2010 by Forrest Higgs

In which your narrator seems to have come up with a way to prevent raft peel for ABS on an acrylic print table.

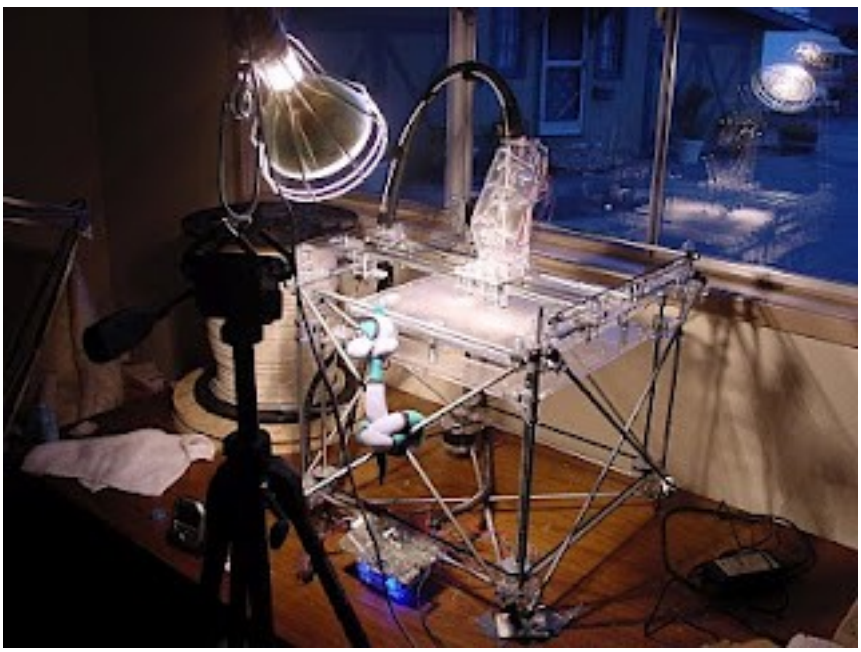
About two weeks ago, the rafts for my ABS prints started detaching from the right hand side of my print table. I was losing one out of two to one out of three prints that way. At first I decided that my acrylic print table had simply got too warped and I had too much variation in level on the acrylic. I removed the acrylic print table and checked its flatness with a milled straight edge. Indeed, it was a little warped so I used a belt sander on it till if was as flat as the milled straightedge. That seemed to work for about an hour but I was soon back to where I began. I then decided that the table was now adjusted properly and went to a great deal of trouble getting it so on my Rapman. Same result.

In desperation I began to print on the left hand side of the print table and the problem went away. It was still troubling, though.

On Tuesday, I finally caught on. In the last weeks the weather had cooled to where the outside temperature was in the teens more often than not and dropped into the single digits {Celsius} in the early mornings when I began to work. I looked at the printer table and noticed that the window I used to ventilate the work area was on the right hand side of the printer. The next time that I had a raft detach I measured the acrylic work surface temperature with my IR thermometer and discovered that whereas it was about 25-26 C on the left hand side of the table the draft from the window dropped that to 22-23 C on the right hand side. I was rather shocked that I got that wide a variation in surface temperature over a few centimeters distance, but I certainly did.

I closed the window and the peeling instantly stopped.

That remedy wasn't workable because of the ABS fumes, so I rigged a portable heat lamp onto my camera tripod to shine on the print table.



The radiant energy keeps the acrylic print table at 34-40 C with the window open.

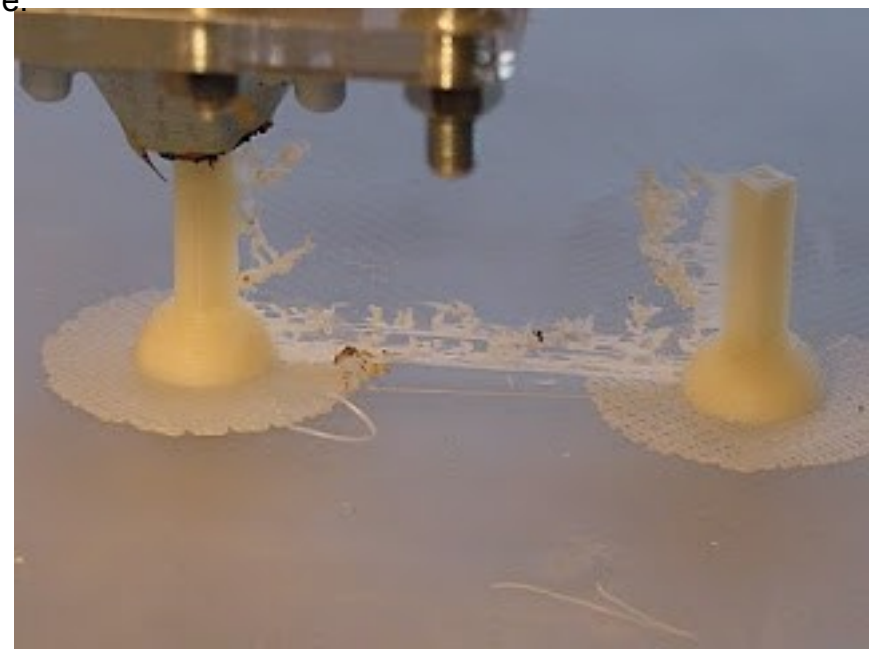
I haven't had a raft detach since then. I've done a few dozen prints, mind.

An interesting stringing behaviour

Sunday, 19th September 2010 by Forrest Higgs

The ability to reverse the extruder on my Rapman printer during transitions between print roads has reduced stringing to an enormous extent. What stringing I do get tends to be very thin and feathery.

I've noticed an interesting stringing behaviour since the release of firmware version 4.0.2, though. You can see it here.



What you will get is a thin string between two objects being printed being propagated and then the string acting as a brush on th extruder orifice at intervals between roads. These brushed accretions build up into quite beautiful forms resembling frost.

When you get just a tiny bit of string hanging off of your print you will see the brushed accretion building up at 45 degree angles into something resembling fractal patterns.

This phenomena isn't a big issue for me. The accretions are thin and fragile and brush off easily without sandpaper.

They are pretty, though, in my opinion at least. :-)

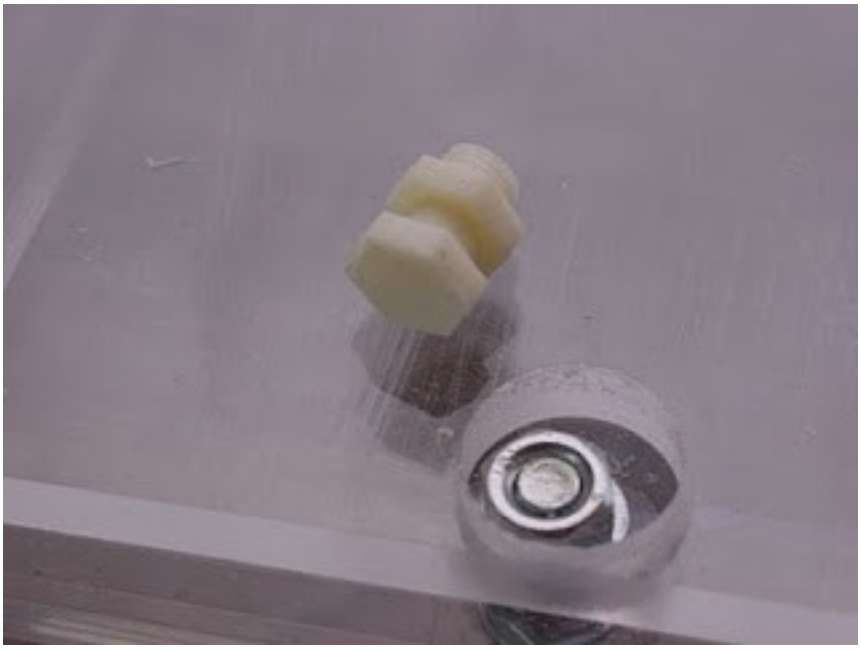
When ego takes charge

Saturday, 25th September 2010 by Forrest Higgs

Nordom recently printed a brilliantly executed M30x70 bolt, nut and washer ensemble.

I was, of course, quite jealous of his accomplishment and still am. It is just a beautiful thing. That got me to wondering, though, just how small a bolt one could print? As usual, I was too impatient to wait till Nordom got permission to distribute the STLs for his bolt and finally found something similar in Thingiverse.
I hate recessed head bolts so I replaced it with a regular hex head and scaled it down to a M10x20. After several tries I had something useful.

I've put it beside a similar metric bolt on the Rapman for scale.



Although the metric scale threads work in ABS, it seems obvious to me that something cut a bit deeper would be more useful.
Now that I've got that out of my system, I'll be going back to working on my carpal assembly.

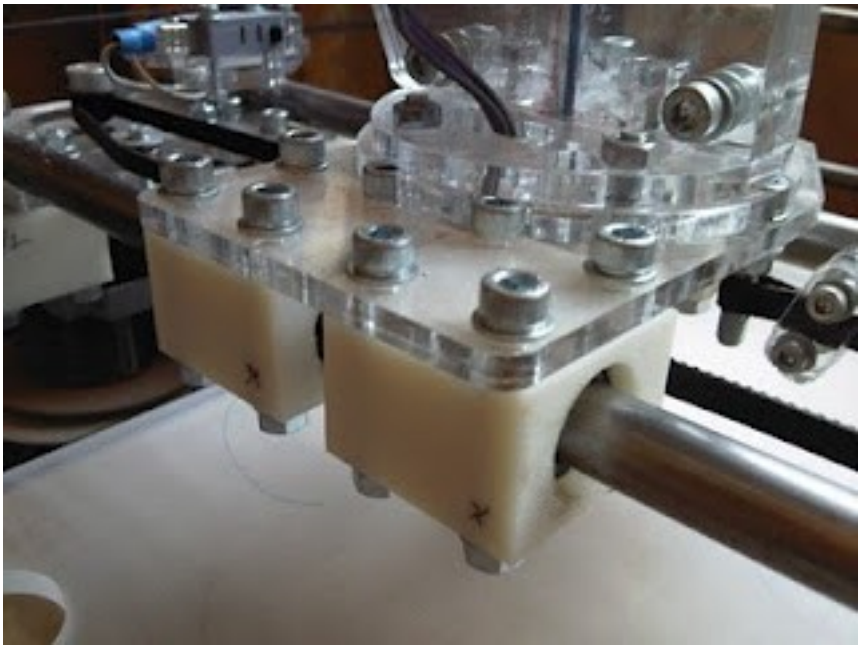
Replicating laser-cut parts

Sunday, 10th October 2010 by Forrest Higgs

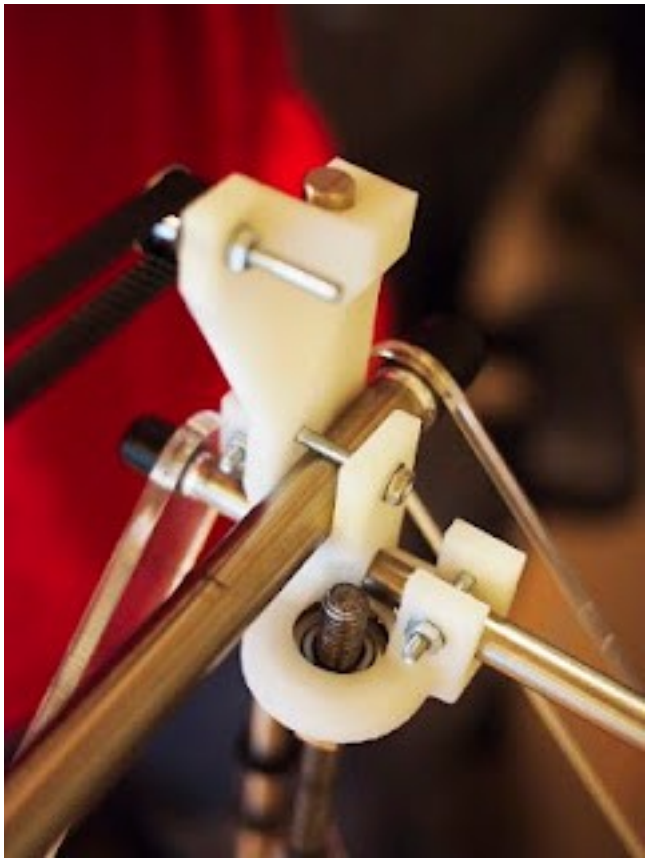
In which your narrator confects his own idiosyncratic method of replicating laser cut parts.

The recent acquisition of the UK manufacturer of the reliable Rapman printer, BitsfromBytes {BfB}, by the American firm 3D Systems has exacerbated a long standing problem that the Rapman's users' community has had, vis, getting drawings of the laser cut acrylic parts that make up the system. While BfB uploaded drawing files of one of their early Rapman designs into the Reprap website, they have neglected to do so with version 3.0 and later models to the best of my knowledge.

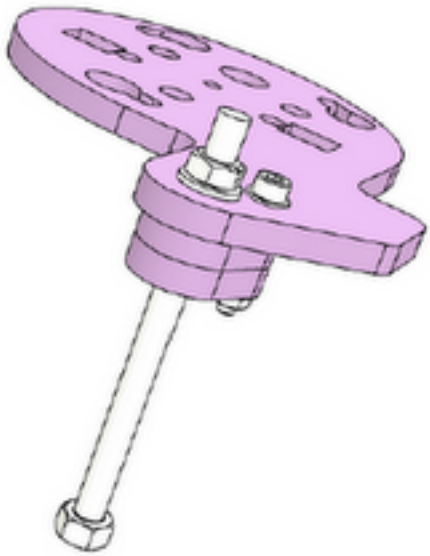
That wouldn't be such a problem save for the fact that the acrylic parts in higher stress parts of the Rapman tend to develop shear cracks and spalls after hundreds of hours of operation. Rapman owners are then left to either request replacement parts from BfB or print their own spares. Interestingly, there is a healthy spares design effort underway for the Rapman, part of which is hosted on the BfB website itself.



Here the printed white ABS grips for the x-axis linear bearings replace the lower plate for the extruder carriage in a much stronger configuration that the original. Indeed, at this moment, there is a very exciting redesign of its printable parts underway by an Australian newcomer that greatly reduces the print time required for what were finicky corner blocks.



All this aside, there are many pieces in a laser-cut printer that simply need to be printed as-is in ABS rather than redesigned. Heretofore, I found myself carefully tracing the parts out on paper and using calipers to get the dimensions. While that approach works well with many parts some, like the extruder z-depth stop plate are much more problematical.



This particular plate is the antithesis of rectilinear and locating the holes and pockets is tedious to put it mildly. I'd thought that it ought to be possible to use an ordinary 2D scanner to capture this sort of information. Today, I gave it a try and discovered that it was not as straightforward an operation as I'd imagined.

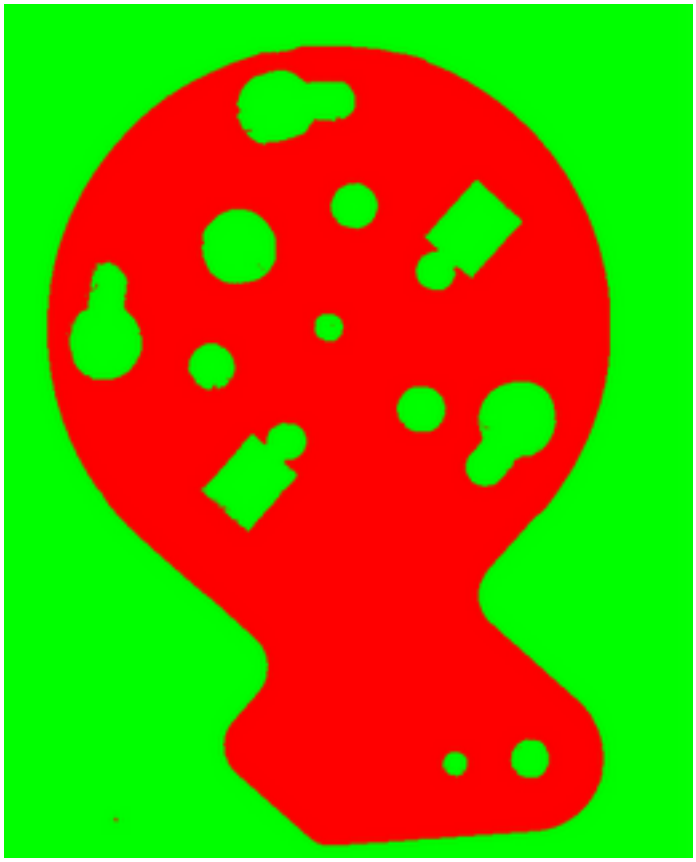
I had an extra copy of this part that I'd bought still in the protective blue film, so I threw it in my Epson Perfection V500 Photo scanner, a very cheap and extremely high resolution machine. Even with the blue film the image captured didn't have a lot of colour information that would let you think that you could separate the part from its background.

I pulled that one into Slice and Dice and tried to separate out the image with RGB manipulations to no avail. I then tried putting a intense red background behind the piece figuring that the blue would override the red and tried playing with the colour mixes in both Slice and Dice and Photoshop, again to no avail.

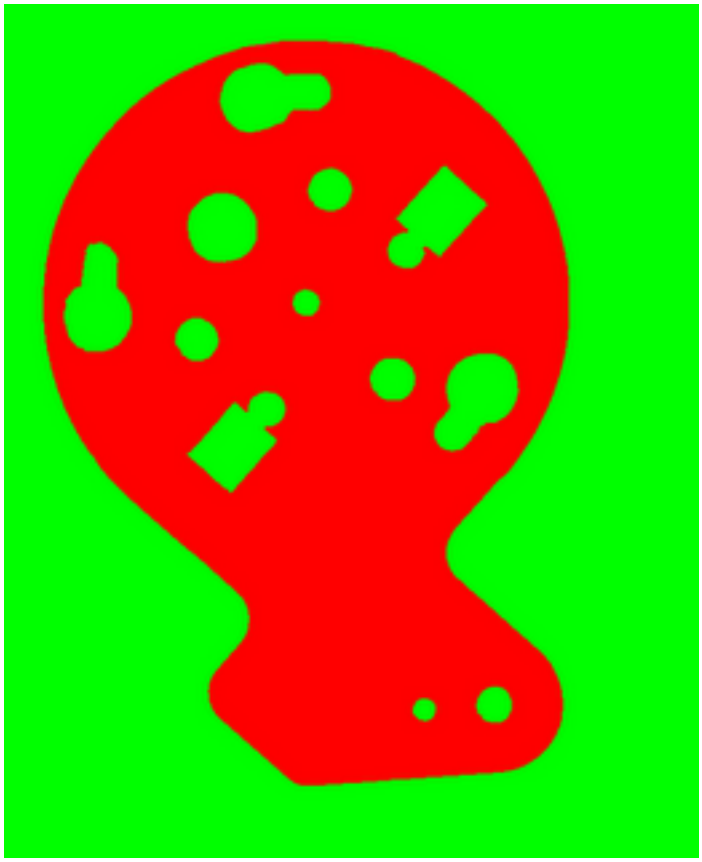
Two things were killing me; the transparency of the part and it's depth, which you can see very clearly in this scan. When I tried to just capture the edges by going high contrast, the depth of the object spoiled everything.

It occurred to me that I could paint the part on one side with tempra paint, which can be wiped off of slick surfaces. Rather than drive into town and given that the part was already protected with a blue film I simply spray painted it with red enamel. I then put it back in the scanner with a piece of dead black HDPE sheet behind it and instantly got the colour separation I needed.

Popping the scan into Slice and Dice and filling the black with green, I had an image I could work with. As you can see, it was a little nasty, largely due to the messy laser cutting on some features and a bit of flare here and there.



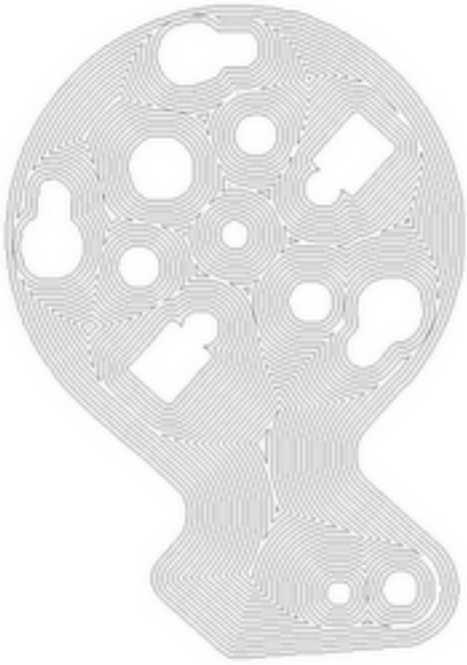
A few moments in Paint cleared that up.



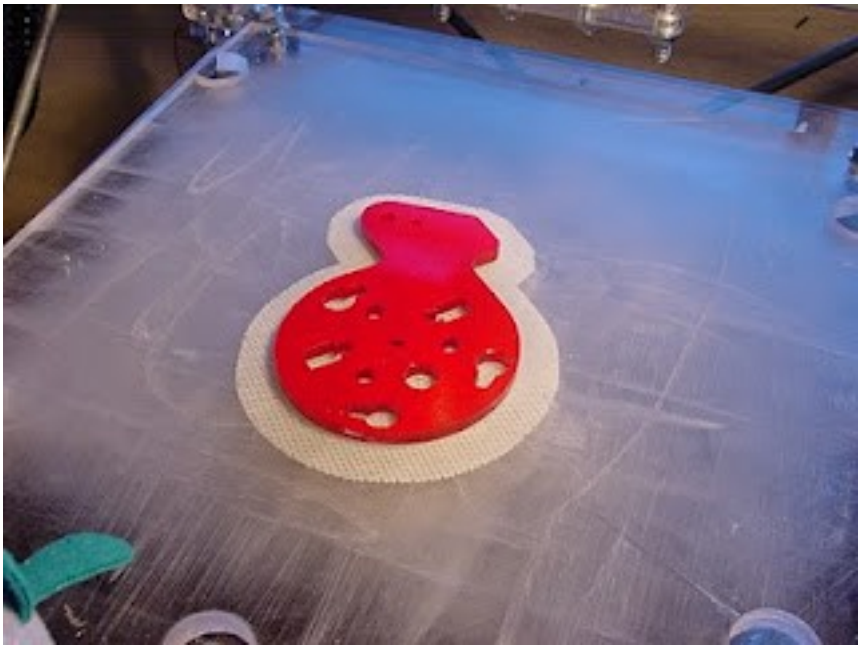
Now that I had crisp colour separation, defining the part boundary in Slice and Dice was trivial.

While I was in paint I took the pixel counts across the diameter of the outer circular feature boundary and measured the same feature on the part with calipers. I then adjusted the size of the image in Paint.

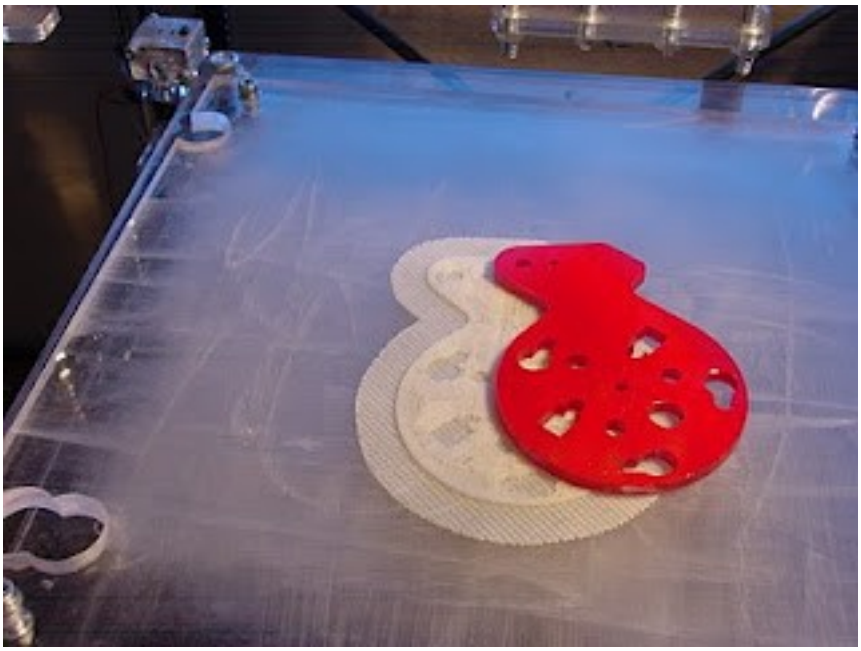
With a bit of pushing and shoving in Slice and Dice, I processed the part and created a print file. Here you can see the print roads for the part.



With a print file, I did a trial print to check the dimensions. The acrylic part lay precisely over the printed part.



I photographed it again with the original part slightly ajar so that you can see the holes in the print.



If you look closely at the several layers of part print that I ran before aborting it you can see that I need to increase the print flow a touch. More importantly, with a part of this size and complexity, however, I am going to have to write a routine that makes a better job of reducing transition distance between print roads. Transitioning was taking far too much of the machine time. That's on my "to do" list now.

What I've demonstrated here is not a smooth operation on Slice and Dice just yet. I was mostly trying to prove the concept. That was a huge success. What this means is that owners of laser cut reprap machines can readily exchange parts information with nothing more complicated than an ordinary 2D scanner and a set of calipers. This should give us considerably more flexibility than we have at the moment.

I am certain as I finish this blog entry that someone is going to show me a simpler, faster way to do this in a few hours. That's certainly happened before. If not, though, we have this approach. :-D

Repairing a catastrophic failure of the Rapman 3.0

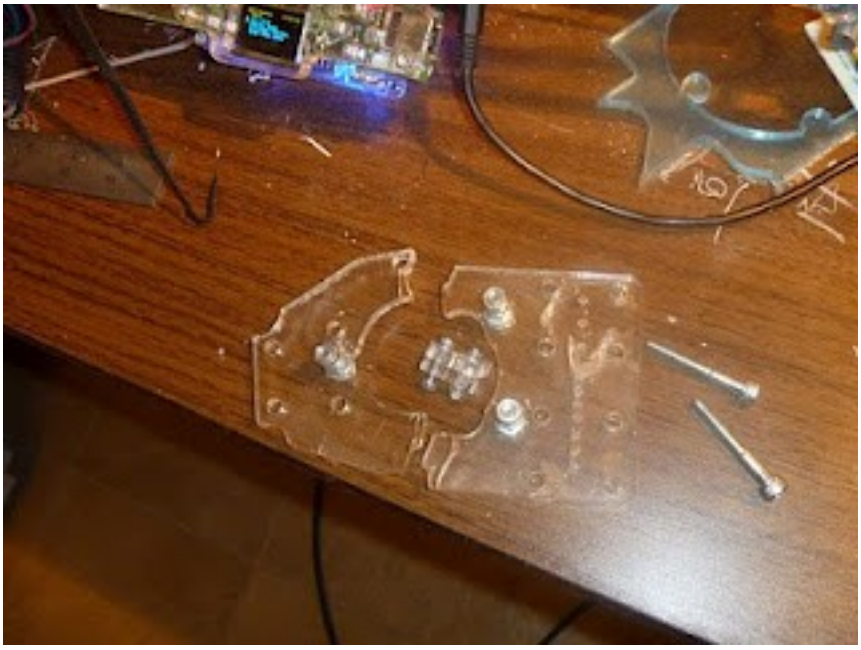
Sunday, 17th October 2010 by Forrest Higgs

Little did I know when I was upgrading Slice and Dice to do prints of laser-cut Rapman parts that I would soon be confronting the problem of replacing such parts without having a 3D printer to print the parts. That is an ENTIRELY different problem.

I had upgraded Slice and Dice to do a better job of printing my sample laser-cut part from last time and was doing trial prints when I noticed that I was getting an enormous number of resets. This was odd because ordinarily I only get one or two resets per month since I uploaded firmware version 4.0.2. I checked the humidity and it was a bit dry, so I fired up the hot mist humidifier and drove the relative humidity up to 52%. No joy.

The next morning I undertook a new print and noticed a mushroom of ABS had formed under the extruder.

When I attempted to remove the extruder, the mushroom of ABS proved to be too big to easily get out of the mounting hole. I then began to disassemble the x-axis carriage that holds the extruder only to discover that the top plate was being held together by the grace of God and nothing else. It crumbled into two major pieces and half a hundred small fragments when I began to remove the bolts.



As well, one of the tiny little bars that hold the x-axis belt had also broken in half as is readily visible in the previous picture.
As you might imagine, this all made me very cranky. A quick calculation revealed that I had historically been spending considerably more on preassembled BfB hot ends that I was on the filament that goes through them. I run a lot of filament, too.
I spent several hours chatting with both Iain and Andy at BfB. They were attempting to be as helpful as they could, but somehow I came away with the same feeling that I get when I take my car into the garage and they say recursively, "It MIGHT be X. We could work on that and see what happens." I began to steam up a bit when they started talking about my not using "BfB gcode". They'd latched on to that fact that I'd written my own STL processing software that produces gcode right out of their manual and tagging this as a possible problem. They then started talking about the fact that I was using very short line segments {0.1-0.1414 mm} and that was possibly touching off heretofore unsuspected bugs in the firmware. I pointed out that the Rapman is supposed to have 0.1 mm resolution and you can't print objects at that level of resolution unless the firmware can handle that short a line segment.
I finally decided that I had to stop talking before I said things I'd regret later {I do that a lot}, and have a good think.
Here's what it came down to...

- I needed a new top plate for the x-carriage and a new hot end.
- I was going to have to pay for these and it was going to take a week to get them
- It was entirely possible that the hot end had been destroyed by a firmware bug
- There was nothing to say that the new top plate would last any longer than the last one, viz, ten months.

No matter how I looked at that equation it just didn't seem to balance. A big bone in my throat was the fact that BfB wanted me to pay to replace a hot end that it was very possible that their firmware had broken. A bigger bone was that there was no guarantee that if I put the new hot end that I bought into the system that the firmware wouldn't ruin it, too, in short order.

It seemed to me that the most reasonable course was to see if the problem was with the firmware.

I typically print with a 0.3 mm hot end. I like what that kind of resolution does for my prints. Before I settled on 0.3 mm, however, I bought two, preassembled 0.5 mm hot ends, so I had those in stock.

I also had Bogdan's experience that replacing the top plate on the x-carriage and grounding the hot end to it would stop the resets. This from the observation that resets were most often caused by a static charge building up on the plastic of the x-axis carriage and extruder and then discharging, causing a reset. BfB which is apparently located in a damp environment had never encountered this issue. I noted that resets tended to get quite common when the relative humidity in the room holding the printer dropped below about 42%. I'd sorted out resets, except for the ones I encountered most recently which have led to the hot end failure, I think, by using a hot vapour humidifier. After seeing how the top plate had crumbled, however, the notion of replacing the acrylic one with an aluminum one began to sound very attractive.

I decided to acquire the means to cut both acrylic and aluminum. Harbour Freight in Salinas had a very nice scroll saw on sale for $69 which would reputedly do the trick.

I bought that, a sheet of 0.22 inch (5 mm) acrylic and a billet of 3.35 mm aluminum plate. I decided to cut an aluminum top plate first. I began the process by simply tracing the bottom plate, which was identical to the top plate onto the aluminum with a fine tip marker.

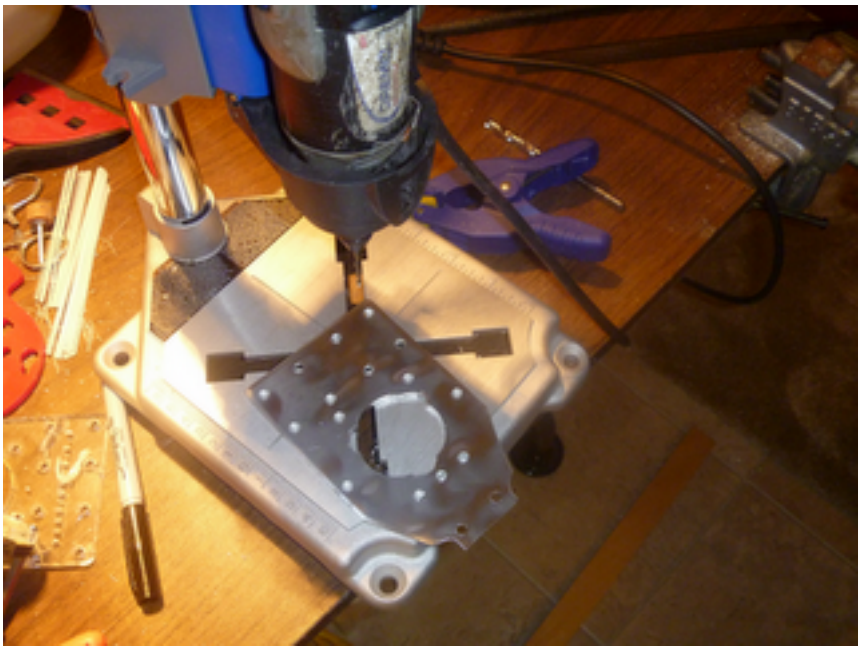
I did the rough cut with the scroll saw and dressed it with a grinding wheel and a half round ring file after having set the plate in a small vise.

I then remarked two holes at diagonal corners of the plate, drilled 1/16th inch guide holes and

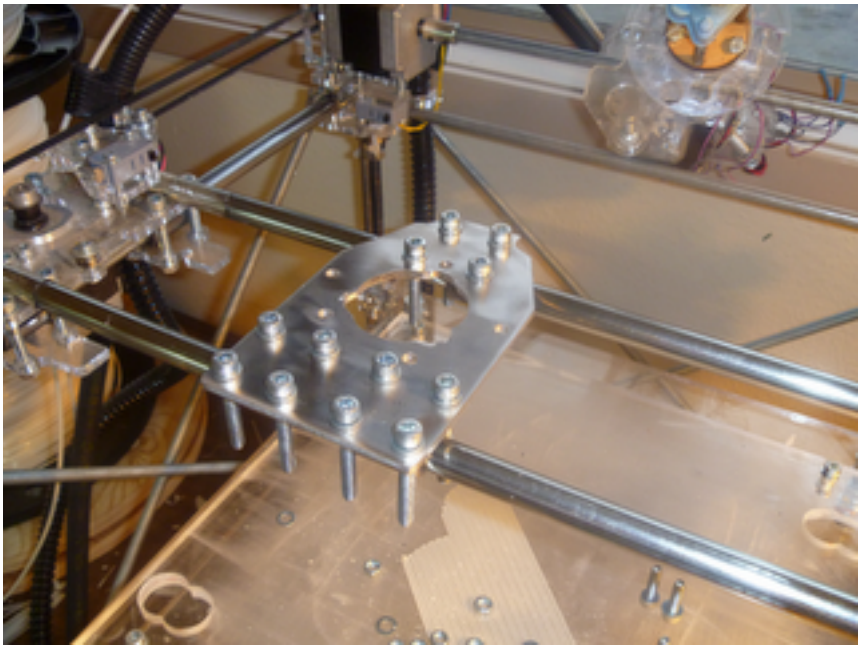
widened them to 13/64th inch {as close as I could get to the 5.2 mm holes as I measured them on the original piece. I did this with a hand drill after securing the plate in a vise.

This done I then bolted the acrylic bottom plate to the developing aluminum one and drilled the guide holes for the rest of the holes using my Dremel drill press.

I then removed the acrylic bottom plate, secured the aluminum plate in the vise again and drilled out the rest of the holes.

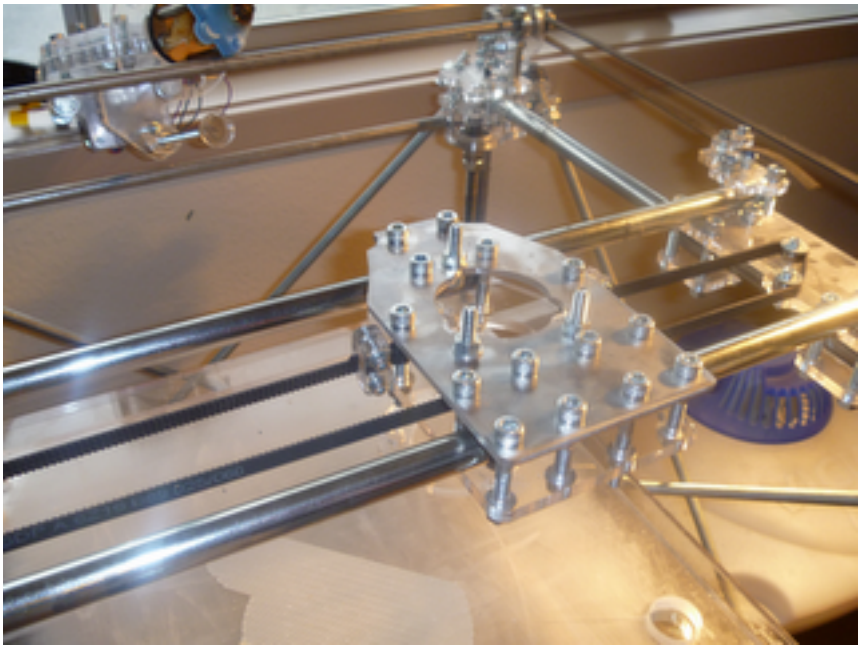


Afterwards I cleaned up the finished plate with a wire wheel and checked it for fit on the x-axis.



At this point my dyslexia set in. The plate is not symmetrical and I'd got it flipped and completely

reassembled and tested the carriage this way. I didn't notice the problem till I tried to fit the extruder into the top plate assembly and discovered the symmetry problem. The next pic is from the original, incorrect assembly.



The new top plate works smoothly. Now I've got to swap out my ruined, 0.3 mm extruder with one of my spare 0.5 mm ones. I will be able to see if I have a serious firmware problem or whether I just have a design fault with the hot end. It could be either, or both.

I'm entering a big of a crisis with respect to 3D printing. I bought into BfB's Rapman because I wanted to do some printing instead of screwing around with printer design and problems all the time. At the time, a year ago, it was a good move. Rapman was a bit pricy, but it was solid and the components of it were affordable. The 32bit MCU board was a delight after all of the Linux/Arduino/Sanguino/Bullshitino nonsense. Some months ago there was talk of extending the Rapman MCU to where you could parameterise the firmware setpoints to deal with different machines and extruders, like the Mendel, for example, or even machines you'd designed yourself. As it stands, it's not clear that BfB can design reliable firmware for their own machine, much less a parametric firmware app that would make it applicable to a wider range of machines.

On top of that, they've recently jumped the price rather dramatically.

As it stands, BfB's Rapman has two Achille's heels; their firmware and their hot end. Neither are reliable and the hot end is very difficult to repair. I'm told that BfB is working on successors to the hot end, but that does me no good at all. I'd like to shift over to something like Nophead's power resistor driven hot end. The problem with that, however, is that I'll have to design a MCU to drive it

and the printer both. By the time I've done that, BfB is out of the picture, since the those two components are what is defensible as corporate worth in the BfB.

I don't know quite what to do.

Operational again at 0.5 mm.

Sunday, 17th October 2010 by Forrest Higgs

Okay, I got one of the spare 0.5 mm extruder heads mounted. I measured the output and discovered that I was getting swell to about 0.66 mm. That works out to 0.34 mm^2 cross sectional area.

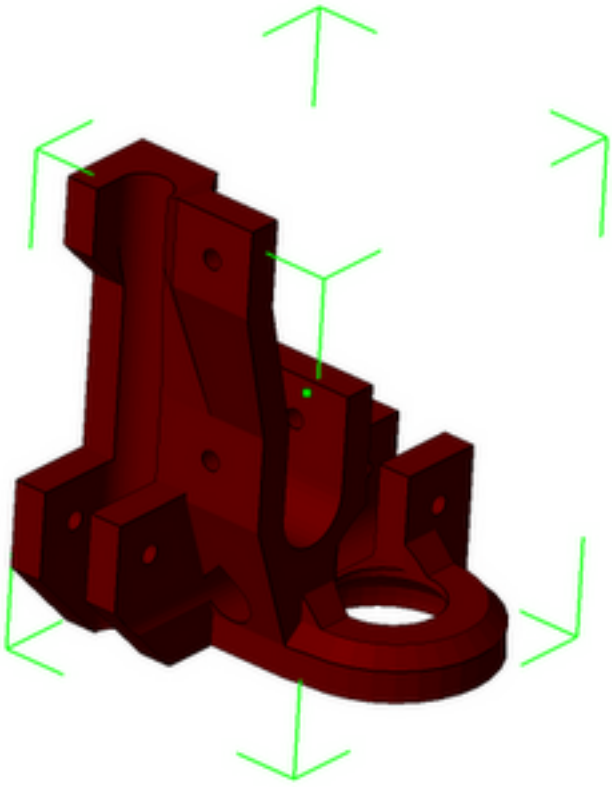
When I was working with the 0.3 mm extruder head I was getting a swell to about 0.42 mm which works out to about 0.14 mm^2 or about 40% as much.

I tried printing a test raft using the old extruder set points and noticed that the base print roads for the raft, which had been well separated before, are overlapping now. In the morning I'll lop 60% off of the extruder's stepper speed and see what happens.

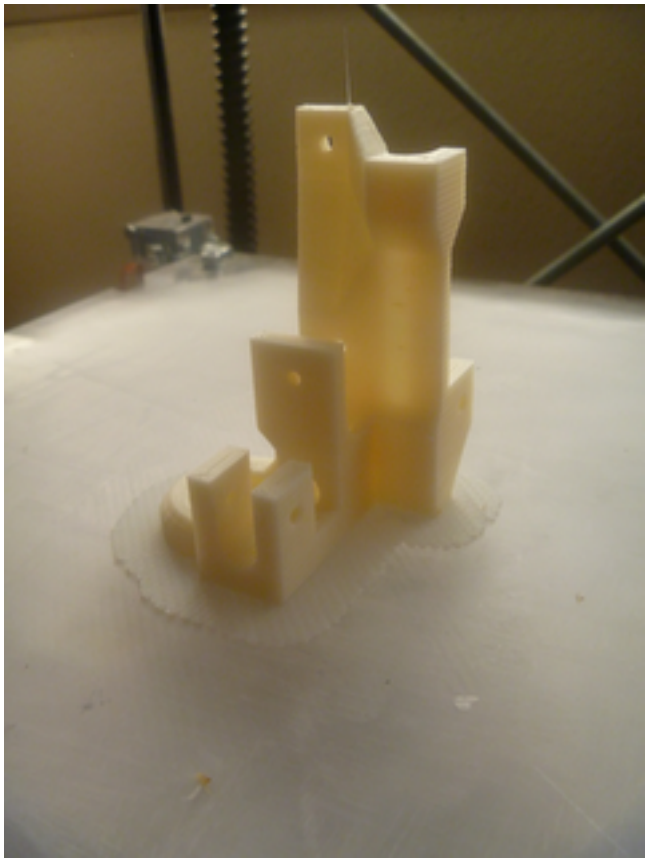
Chylld is a parts design genius

Tuesday, 19th October 2010 by Forrest Higgs

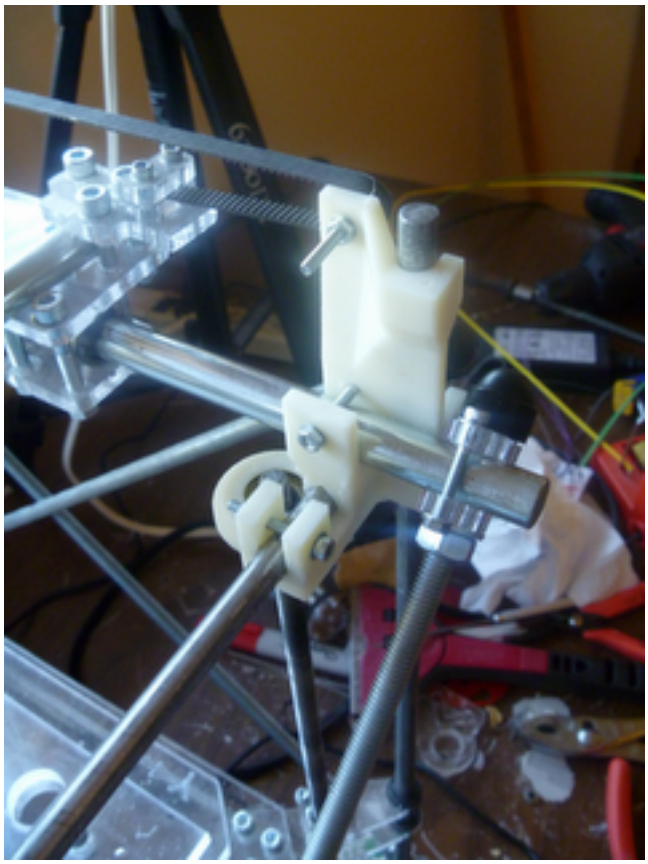
As I mentioned earlier I replaced to destroyed 0.3 mm hot end with one of my spare 0.5 mm hot ends and began to get used to using the larger nozzle size. After some frustrating tries at adjusting flow rates Nophead suggested that I stop fiddling with things and just use the settings that I'd used before. He was right, of course.
A quick print of some of the bones of the hand yielded good results, so I decided to have a try at one of Chylld's corner replacements for Rapman cut acrylic.

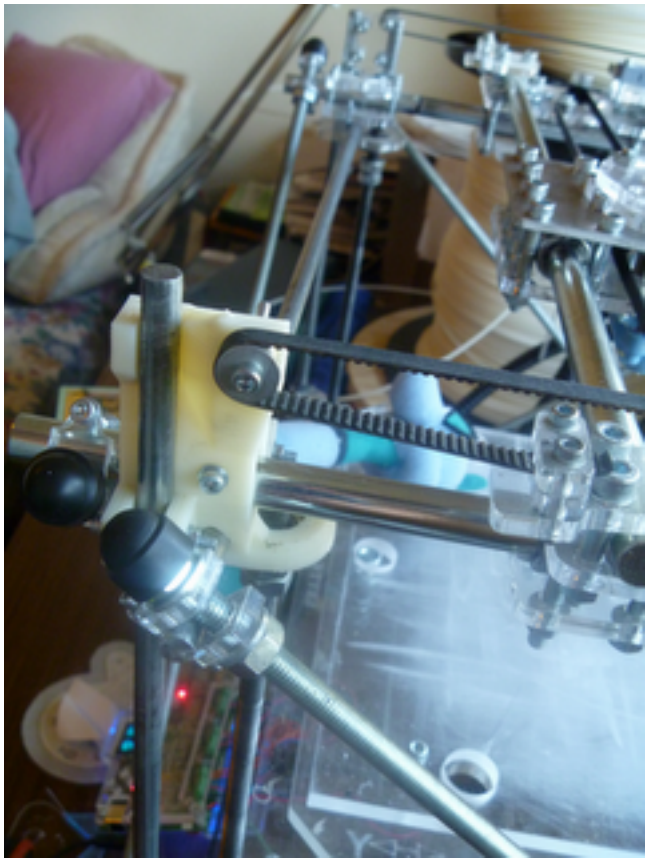


So far there's been no sign of the wall-to-wall resets I was getting with my Rapman in the hours before the 0.3 mm hot end finally failed.



The corner block printed in 2:15 and mounted with a minimum of drama.





There was a minimum of post print part processing required. the x-axis bar was a little tight, but a brushing out of the hole cured that. All bars had a tight, friction fit and the tightening screws were largely cosmetic except when the printer wanted moving.
Chylld is a parts design genius. I've never seen anything work quite this easily.

Getting to the bottom of the Rapman hot end failure

Wednesday, 20th October 2010 by Forrest Higgs

In which your narrator gets to the bottom of why it was that his 0.3 mm Rapman hot end failed and perhaps even why it started causing resets prior to the failure.

The hot end failure that I experienced on 14 October is a relatively rare event. The only other person with a Rapman who reports the sort of ABS leakage that I experienced was Klazlo.

Once I had my Rapman repaired and usefully printing again, I turned my attention to doing forensics on the failed hot end. Once I took a hard look at the hot end and the pictures that I took immediately after the failure hints of the means and mode of the failure began to recommend themselves.

Taking a look at one of the forensic pics that I took on the 14th two items jump out at you. I've circled them in red.

If you look closely at the top red ellipse you will notice that one of the three bolts which secures the spacer on the hot end ensemble was either not tightened adequately at the factory or, more likely, worked its way loose via hundreds of heating and cooling cycles while in operation. This threw the alignment of the hot end out opening a route for hot ABS to escape between the PEEK sleeve and the aluminum extruder barrel sleeve that fits around it. You can clearly see the billowing flow of ABS in the picture. ABS was only leaking on the side of the PEEK barrel where the spacer bolt was loose. That part is straightforward.

The vertical ellipse on the right hand side of the hot end is much more ominous. If you look closely, you can see that one of the thermistor leads has exposed above the sleeve leading down to the extruder tip where the extruder is housed. A bit over 4 mm of exposed thermistor lead is open to the air. You can see that happening a bit more clearly with this picture.

The larger red circle encloses the exposed lead. The smaller one shows the recession of the purple plastic insulation on the lead that occurred once the ABS began to boil out of the gap between the PEEK and the aluminum and contacted the lead.

At first I thought that the hot ABS had actually caused the gap in the insulation. Closer inspection, however, revealed that the lead had most likely been stripped too liberally and originally was only just barely covered by the yellow silicone sleeve. The heat of the billowing ABS almost certainly caused the shrinking of the insulation from just being covered by the silicone sleeve to its present situation. It's easy to see the shrinkage in the second circle.

I then removed the PEEK cylinder from the aluminum extruder head. Originally, I had thought that the PEEK sleeve in contact with the aluminum might have shrunk. It measured 10.09 mm outside of the aluminum sleeve and 9.9 mm inside. That could have been either because of heat shrinkage of the PEEK over time or some sanding of the PEEK tube at the factor to get a clean fit. Unless the parts had been measured before the hot end was put into service, it would be impossible to tell for certain.

I then removed the silicone tube from the inside of the PEEK sleeve as you can see here.

This is the most revealing forensic evidence in the whole exercise. At first, it appears that the

silicone tube has shrunk on the hot end. While that is true to an extent, the reason for the shrinkage is actually because the inside diameter of the PEEK sleeve has shrunk because the PEEK has expanded within the confines of the aluminum sleeve.

It was fortunate that I bought several unassembled hot end kits before BfB began making premade ones because I had some disassembled PEEK and silicone tubes. The unused PEEK sleeves had an outside diameter of 10.0-10.1 mm and a consistent inside diameter of 6.36-6.40 mm.

When I measured the failed PEEK sleeve the cold end inside diameter was 6.40 mm but the hot end had been reduced to 5.88-5.92 mm. The silicone tube outside diameter on the hot end was more or less the same as the inside diameter of the PEEK sleeve.

What was revealing, though, was that the hot end of the silicone tube was virtually entirely closed. It was impossible to slide a 2.9 mm ABS filament through it. I was able to blow tiny bubbles through the silicone tube into a glass of water. What that says is that the ABS filament was being melted in the silicone tube and squirted into the aluminum extruder barrel rather than being melted in the extruder barrel.

I don't think that that was how the hot end design was intended to work.

Getting back to that exposed thermistor lead, it is well known that blowing hot air past plastic surfaces will create a static charge. Not only was that exposed lead within millimeters of billowing hot ABS it was also touching the bottom acrylic plate of the x-axis carriage {part 10034}. Periodic static discharges off of this heated acrylic surface were most likely at least deranging the temperature measurements off of the thermistor and also causing the MCU board to reset. During last winter, I walked in socks across the floor and picked up sufficient charge to where when I touched the MCU board the system reset.

Conclusions:

The failure of the hot end had two aspects.

- The loosening of the spacer bolt, most likely from repeated heating and cooling cycles of the hot end allowed for a hot end misalignment that created a path for leakage of hot ABS out one side of the hot end between the aluminum extruder sleeve and the PEEK sleeve for the silicone tube.
- The swelling of the PEEK sleeve over time reduced it's inside diameter almost entirely closing the silicone tube and causing the ABS to have to melt in the silicone tube rather than in the aluminum hot end. This was certainly not the intent of the hot end designers.

It seems quite obvious that the combination of a PEEK sleeve enclosing a silicone sleeve for the plastic filament need to be rethought.

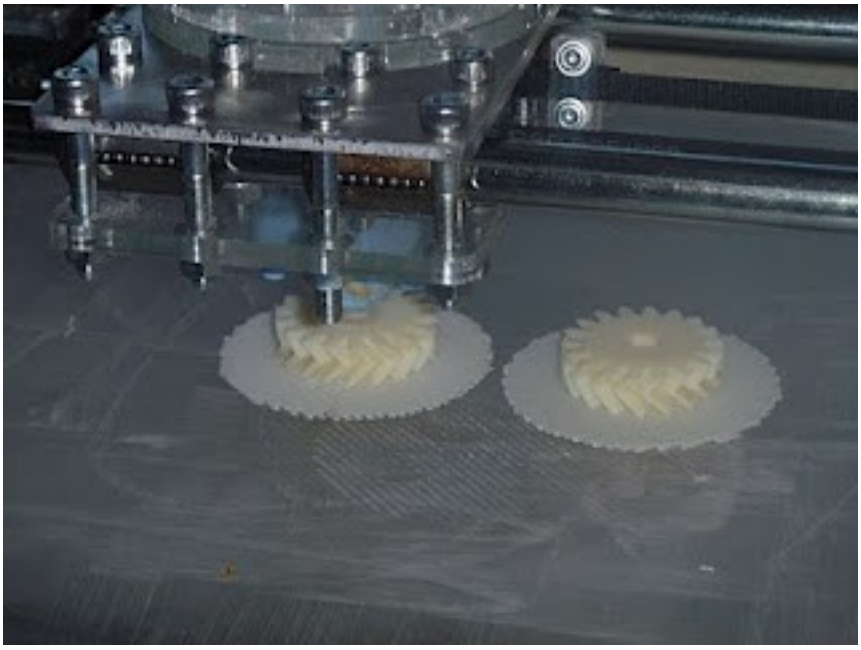
The placing of an MDF plate and an acrylic plate at the top of the hot end spacers through which the securing bolt has to pass also needs some careful. Replacing the MDF with a PEEK plate with the bolts secured by lock washers to the PEEK plate and recessed through the acrylic with the acrylic and PEEK secured by a separate trio of short bolts might be more successful. An aluminum plate as a replacement for the MDF might also be considered.

Annoyances and changes

Thursday, 28th October 2010 by Forrest Higgs

UPDATE: I let my anger run away with me in this blog post. I've just spent time with Ulf Lindhe at Netfabb. Ulf has sorted out the problem I was having with the boolean operations and got me upgraded to version 4.6 at no charge.

It's taken a few days, but I can now print at about the same quality with a 0.5 mm extruder as I was able to print with my now ruined 0.3 mm extruder. I've gone back to work on the active telepresence project and the present task is to create linear servos to as the muscles to operate the hand and fingers. I'm building my own from gearmotors rather than buying after some bad experiences trying to adapt servos designed to operate the flaps and rudders on model airplanes and sailboats to operate hands. Andreas Maryanto was good at this, but I am definitely not. Doing this, I am leveraging the herringbone rack and pinion scripts that I wrote. While Nophead's advice to simply not fiddle with my 0.3 mm hot end settings got me 95% of the way to where I wanted to be, that last 5% took a little tweaking. I'm quite happy with the final results.



The pinions peeled off of the rafts with a minimum of trouble and only took a touch on the belt sander before they were usable.

That was relatively easy. When I started working on the rack, however, I quickly ran up against the limits of Art of Illusion. While it is easy to assemble a set of solids to represent what I want to print in AoI...

Doing the Boolean opeations to make them into one solid is quite beyond AoI's boolean ops capabilities. Bogdan told me repeatedly that I could do this sort of thing in Netfabb Professional, so I read up on the method and set about to use my now copy and discovered that the Netfabb people had somehow disabled this feature in my copy. ~~I had a email from them about version 4.6 {I have 4.4} but no clear idea if I was going to get a free upgrade or have to pay them even more to do something that I am already supposed to be able to do.~~

~~The more I think about that the more pissed off I get. If I don't hear from them in the morning, I am going to build boolean ops into Slice and Dice and give it away for free. The way I feel just now I might just do that anyway. I am getting sick and tired of getting ripped off.~~

Taking a different direction

Tuesday, 16th November 2010 by Forrest Higgs

A few months ago, I had planned on acquiring another BfB Rapman with two print heads this time so that I could explore the use of PLA as a support material for ABS prints.
With the sale of BfB to 3D Systems, continuing troubles with BfB firmware and the catastrophic failure of my expensive Rapman hot end and the discovery that BfB isn't warranting such failures, I've decided to build my own second printer.
To that end, I've just purchased a Sherline 4410 metric lathe.

It will be arriving Thursday.

Arrived

Saturday, 20th November 2010 by Forrest Higgs

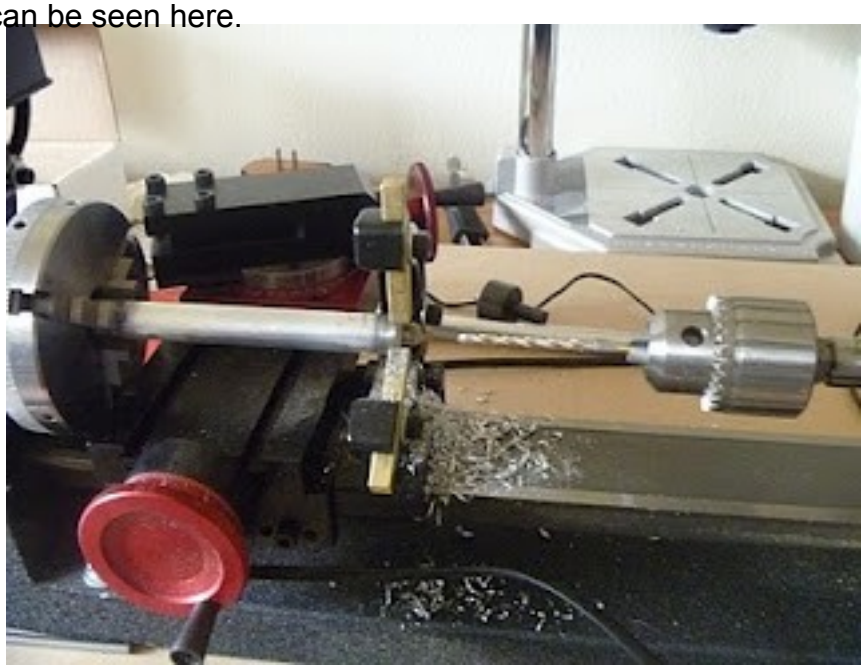
The Sherline lathe arrived on Thursday. I was working that night but stayed up extra late and more or less got it assembled. It went together without a lot of drama. The digital readout box makes whether you bought the lathe with American or SI verniers irrelevant since you can select metric or "Imperial" on the digital readout box.
The readout box also includes an optical tachometer to keep track of the rpm rate of the lathe.
The DC controller knob will give you stable rotational rates down to about 200-250 rpm.
This morning, I mounted the lathe on a piece of chipboard for stability.

I bought a couple of pieces of aluminum round bar, a piece of stainless round bar and 3 feet of hot rolled steel round bar to practice on. You can see one of the aluminum round bars mounted in the lathe.

First lathe experiments

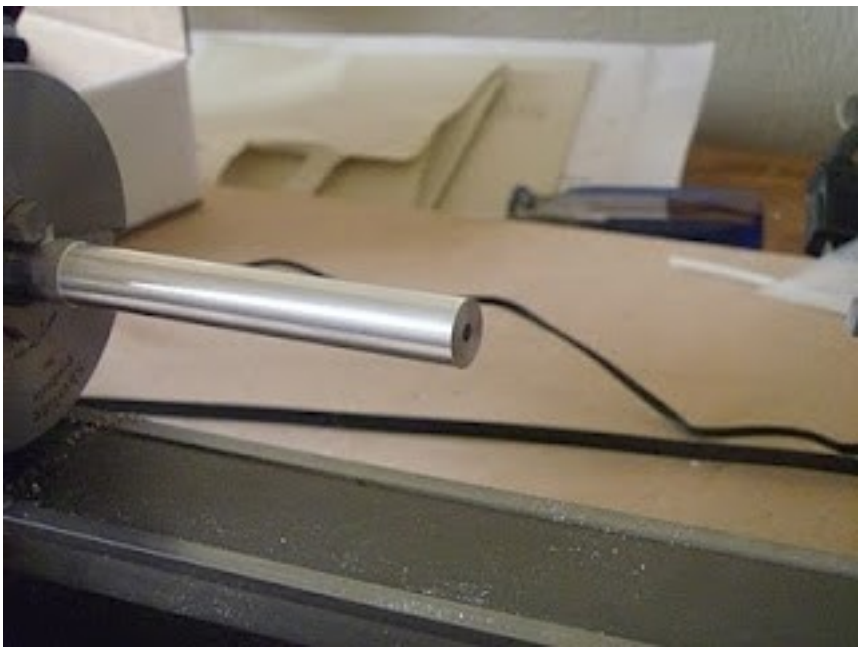
Monday, 13th December 2010 by Forrest Higgs

I finally got a few days off, so this morning I began to do things with my new lathe.
My initial motive in getting the lathe was to have the capacity to make my own extruder hot ends.
Given my total lack of experience with lathes, I decided to see if I could try sinking an 1/8 bore in a piece of aluminum round stock using a fixed drill bit in a chuck inserted into the tailstock. An idea of what I did can be seen here.



After a considerable time, I was able to sink a 50 mm hole into the centre of the round stock. The steady rest that you see supporting the end of the aluminum round stock nearest the drill bit reduces vibration and misalignment.

When I worked with Tommelise 1.0, I used a 5/32 inch braised copper tube (1/8 inch ID) hot end several inches long with its top only heated using nichrome wire. I was able to get well past the whole issue of having filament melt in the PTFE thermal break with that approach. These days, I hope to do something similar using aluminum.



Here you see the drilled round stock after I've trimmed it down to 12 mm outside diameter and trued up the end. It takes a nice finish.

Getting into threading...

Sunday, 19th December 2010 by Forrest Higgs

When I bought my Sherline lathe, I went for the loaded option. As I am finding out, it was a very good decision. I don't think I'd have been able to use at all it without the digital readout box. I understand how people in times pass could have used a micrometer to do measurements between cuts. I doubt that I could have had the patience, though.

After my last attempt at doing a deep drilling and cutting down a piece of aluminum bar stock, the next thing I wanted to do was use the threading attachment. Between Joe Martin's Sherline book, Tabletop machining and the Sherline Accessories Shop Guide I was able to puzzle out how to re-rig the lathe for threading. My motive in getting a threading attachment was to have the possibility to cut my own lead screws and thrust collars. The studding used in the Rapman and Mendel 3D printers, not to put too fine a point on it, suck the middle out of canned Vienna sausages. When you buy studding it is rarely, if ever, straight. When you try using it as a z-axis lead screw, it does not very pleasant things to your print quality.

As well, I am quite fond of the polymer pump design on the Rapman. My primary reason for wanting to do my own threads was to be able to make the threaded drive shafts for those, too.

The biggest shock for me in this adventure happened when the instructions book told me to demount the lathe motor. It turns out that you cut threads on the Sherline lathe with a hand-powered handle. Using it reminds me of my grandmother's old foot pedal powered Singer sewing machine that I learned to use when I was about four. It has a very similar, 19th century feel to it.

By and large the threading system design is quite good. The real pain in the tail, however, is that little black lever that you see below the spindle in the above picture. Getting that to work right had me taking the whole lathe apart about three times. Sherline needs to rethink that part of the design. It really sucks.

Other than that, the threading attachment was quite easy to use. It is set up to do standard and metric 60 degree threads. The carbide cutting tool did that quite easily.

For a first try I did a really coarse metric thread, 2 mm.

I figured that that deep a thread would require the most muscle. It was a bit tiring but not at all impossible.

For lead screws, the Sherline will handle about 15 inches of threading. If the diameter of the stock is less that about 9 mm, a longer unthreaded piece of stock can be snaked through the spindle. Given that manufactured lead screws about cost $1/cm and thrust collars typically cost $30-35, this capability will save me some money over the mid-term.

A proper conductive polymer mix?

Friday, 24th December 2010 by Forrest Higgs

In which your narrator begins to test a new conductive polymer mix.
Do you want to read more?
One of the long term grails of the reprap project is to be able to print circuitry. Currently, Rhys Jones, a PhD candidate working for Adrian Bowyer at the University of Bath, is devoting a lot of his time to addressing this issue.
Heretofore, we've looked at very low temperature eutectic alloys like Rose's and Wood's metal and confected methods of extruding it into preprinted channels. More recently, Rhys has demonstrated that ordinary solder can be successfully printed onto ABS.
We've looked at various concoctions of carbon, silver and other conductive materials with polymers at arm's length. Always, though, the conductivity of these mixes has been too low for them to be useful in printed circuits. Rhys, however, has looked at a two step process which may get us by that.
The main problem with printing PCBs with a reprap machine is that PCBs are not that hard or expensive to have made using conventional, proved materials. This makes coming up with a method of printing them a bit tricky because it not only has to work, but it also has to yield a product that is not all that much difficult to use than conventional boards.
Recently, I ran across a high-tech company, Integral Technologies, located in Bellingham, Washington. They've come up with a different approach to creating a conductive polymer mix which uses conductive fibers mixed with conventional polymers. Probably the most exciting is their mix of very fine stainless steel fibres mixed with a blend of polycarbonate and ABS.
The volume resistivity for this material on the data sheet runs to 0.06 Ohm-meter. This is far too high for a useful conductive plastic. I rather shrugged the material off until I got a look at a few non-technical demonstrations of the material. This one shows two small bars of the material used to conduct 110 v current running an electric drill.
Interestingly, Bill quoted me a price range of between $9-45/lb depending on the composition. More interestingly, the PC/ABS mixes which resemble what we are already used to using in Repraps are at the cheap end of the price spectrum.

As well, they demonstrate measured resistance across an extruded plate of their material using an analog multimeter.

There was obviously a mismatch between what their data sheets said and what the non-tech demos indicated. I explored this question with Bill Robinson at Integral. It boils down to the "skinning effect" mentioned in the first clip. Simply explained, it means that the metal fiber doesn't tend to show up at the surface of an extrusion. This means that the material is partially insulated at the surface of an extruded surface.
That left the issue of how, in the second demo clip, ordinary multimeter probes registered effectively zero resistance when measuring resistance across a extruded plate of their plastic. Bill sent me some demo plates of the material a few days ago. I have a digital multimeter rather than the analog Radio Shack multimeter.

Multimeters are not particularly good at accurately measuring low resistances. Anything reading below about 5 ohms for most multimeters is not something you want to bet your life on.
I tried duplicating the measurements in the clip and kept running into skinning effects. In places I could get readings less than 1 ohm, but others I would be in the mega ohm range. I made a cut across one end of one of the samples. You could barely see the fiber ends. Integral is using VERY fine fibres.
I then drilled 1.5 mm holes through the plaques and put the probes through those. I was able the get more consistent and lower resistance measurements. Mechanical contact between the probes and the fibre ends is how good conductivity is achieved.

The best I was able to get consistently was 0.6-0.8 ohms over about 100 mm of the material.
Bill at Integral suggested that if we used a 0.5 - 1.0 mm extrusion nozzle we out to be able to avoid jamming of our extrusion nozzle with the stainless steel fibres. Making strip extrusion as we can with a Reprap printer should tend to align the fibres much better than a simple injection moulding would do. The problem as I see it will be making a conductive connection between two traces and

making a connection between a trace and a component.
The question now is whether what we know about this material now justifies having a sample turned into fibre for further testing.

Restocked

Tuesday, 11th January 2011 by Forrest Higgs

Many months ago, I ordered new filament for my stocks. I order exclusively from New Image Plastics in that I know that Jim Waring there charges low rates and has consistently high quality. What with the economic troubles Jim has had trouble in recent years finding a broker willing to deal with small orders of less common polymers like ABS. He found such a broker recently, though, and was able to fill my order. :-)

Yesterday, my filament arrived; thirty lbs of 3 mm ABS at $7.95/lb and 10 lbs of polypropylene at $4.25/lb. With the ABS I already have, I think I am now set for the year.

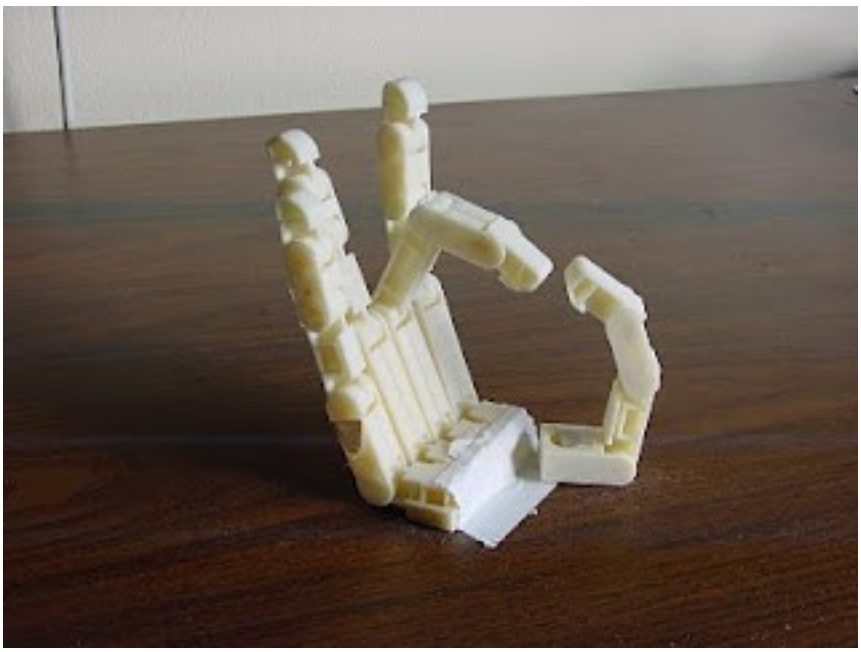
Printing small holes in small features

Monday, 14th February 2011 by Forrest Higgs

In which your narrator discovers an out-of-the-box method of making small holes for pinned hinges in small features.

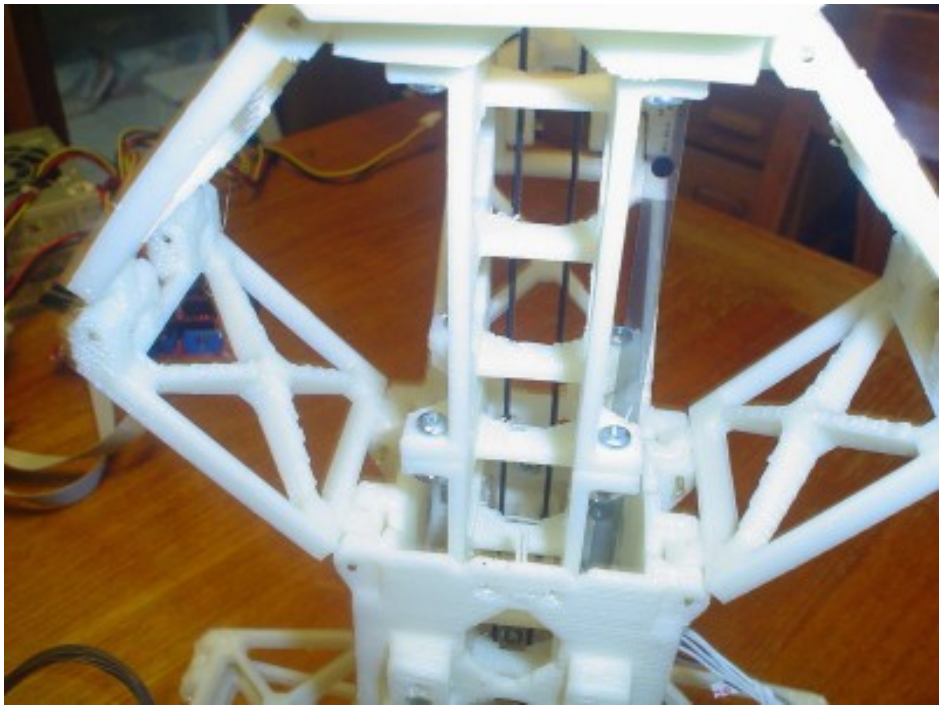
Recently, Chris Palmer blogged an exquisite article on the pitfalls of getting the diameters right in holes made in objects. It related very closely to a problem I was having in developing a printed robot hand.

Previously, I'd had no trouble in that the joints between the phalanges of the robotic fingers were very large and printed.



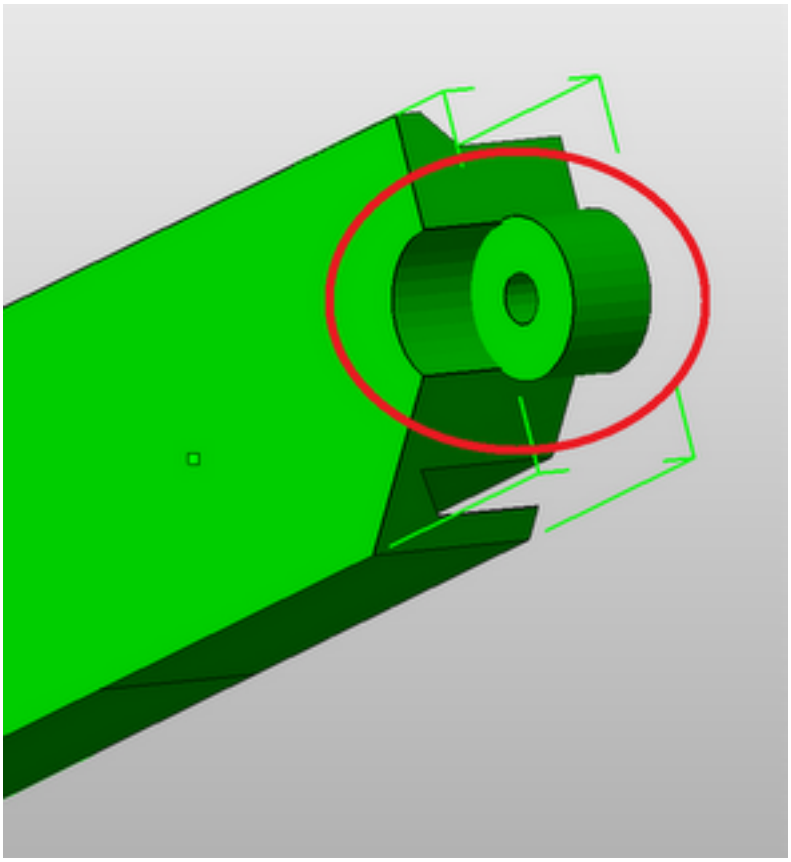
I loved this approach, but sadly had to abandon it simply because the large contact areas between the phalanges created excessive friction and because the number of things that I needed to have happening in a phalang made the large joints unhelpful. The need for smaller, more compact hinged joints brought to mind the work of Frank Davies with his Sarrus Linkage positioning system .

Frank used pinned hinges to great effect.



As I was designing such pinned hinges into my phalanges, I soon discovered that the substantial pins that Frank was able to use were too large for the delicate phalanges that I was working on. In fact, I finally settled on using simple paper clips (0.84 mm diameter) for the pins in my work.

In designing the joints, I followed the usual Reprap gambit and simply included the pin holes in the STLs for the parts.

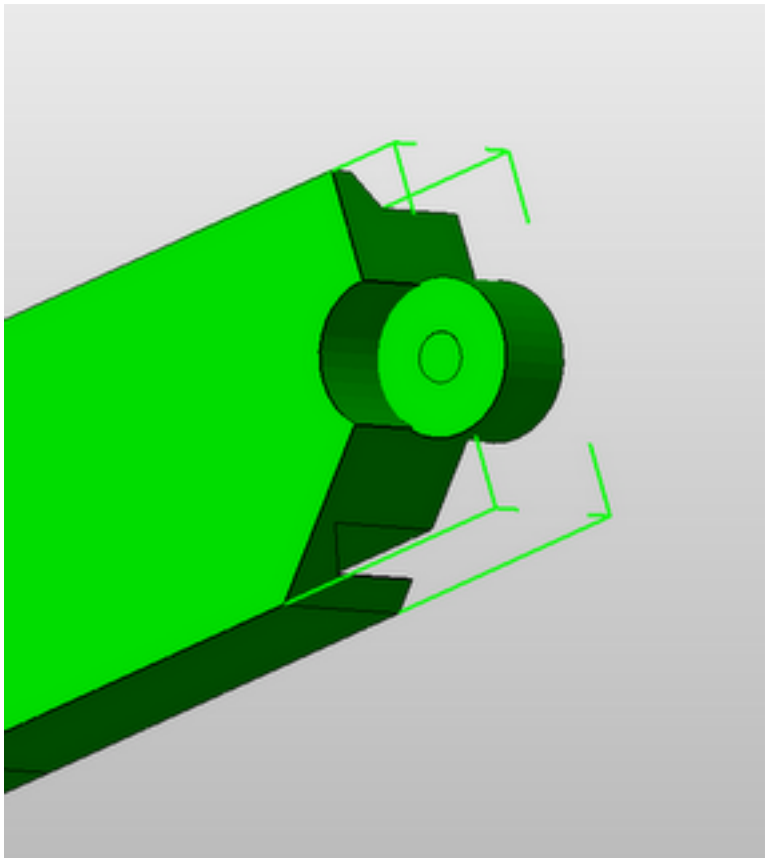


The only problem with this approach was that both the hole and the hinge joint that it was seated in were very small. The joint had a radius of only 3 mm and the hole 0.42 mm.

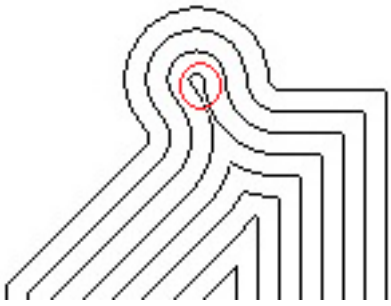
Ordinarily Reprappers use a few print paths around the perimeter and then infill the rest. I design using thin walled parts glued together after printing, an approach that lets me create fine featured parts that are fast to print. With a feature this small, however, print roads radiating out from the pin hole very quickly clash with print roads radiating in from the joint.

It's bad design practice to let a hot print head hover for extended periods of time over or near a small feature. You want the head doing the feature then going far away quickly so that the molten plastic thread has a chance to cool a bit before the next layer is applied. If you don't get this your small feature becomes a featureless blob.

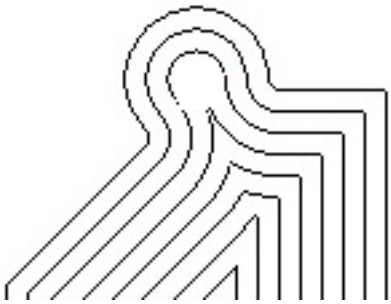
Print road clashing between the pin hole and outer perimeter of the hinge had me fiddling around with print road width for the better part of a week with indifferent results. Yesterday, however, an out-of-the-box solution to the problem finally hit me. I'd do better at printing the pin hole if I simply didn't include it in the STL, something like this.



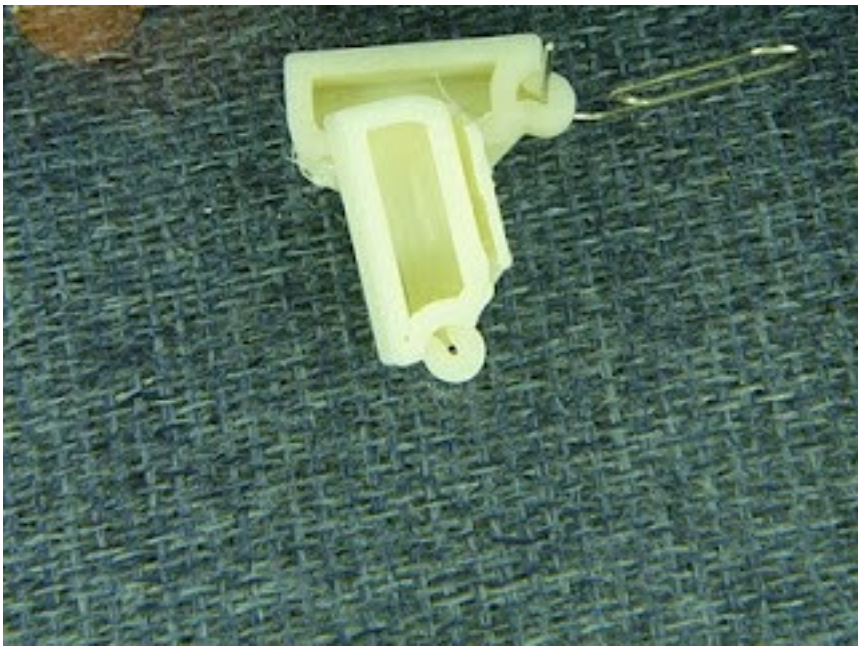
Basically, I just plugged the hole. What that did is to limit the print road propagation to those radiating inwards from the outside perimeter of the hinge. Since I don't use infill because of the problems getting the perimeter roads to match with the infill roads, for the first few millimeters of the print had no hole due to the use of perimeter roads to completely fill the layer.



Once I had the roads calculated I pulled up the .PNG images for the completely filled layers. You can see the bit that will fill the pin hole circled in red.



It was a simple matter to pull the images into Paint and remove the inconvenient loop from the images and then continue processing the images into Gcode.



By calculating the width of the print roads appropriately, I was able to get the proper diameter for the pin correct on the first try.

The annoying part of this whole epiphany was that it took me over a week and dozens of trial prints to see the simple way of solving the problem.

Going for 32 bit embedded processors

Tuesday, 1st March 2011 by Forrest Higgs

In which your narrator begins to cobble a next-generation Reprap controller board together out of much bigger vitamins than has previously been possible.

The controller for the Bits from Bytes Rapman has long been the most sophisticated in the world of Reprap printers. It's use of the PIC32 MCU gives it the power to do things, like support an LCD display, SD cards and 0.1 mm gcode steps without difficulty are very difficult to do with 8 bit MCUs. There are two barriers to it's widespread adoption in Reprap printers, however.

The first is that PIC32s typically come as 100 pin surface mount chips. That means that you have to do some fairly sophisticated PCB design and get the damned chip soldered on properly. That's not impossible, but it's not easy, either.

The big barrier, however, has been the lack of inexpensive compilers for PIC32s. The one from Microchip is very good, but costs over $1K. Recently, that situation has been remedied by a Serbian firm, Mikroelektronika. I had had experience with their PIC 18F compiler and found it both very reliable and possessed of an enormous library of library functions which made it extremely valuable as a development tool.

Thus, when I heard last year that Mikroelektronika had undertaken to create an inexpensive compiler for PIC32 chips I was immediately interested. After a long development and beta cycle they did their formal release this morning. They offer inexpensive C, Basic and Pascal compilers for the PIC32.

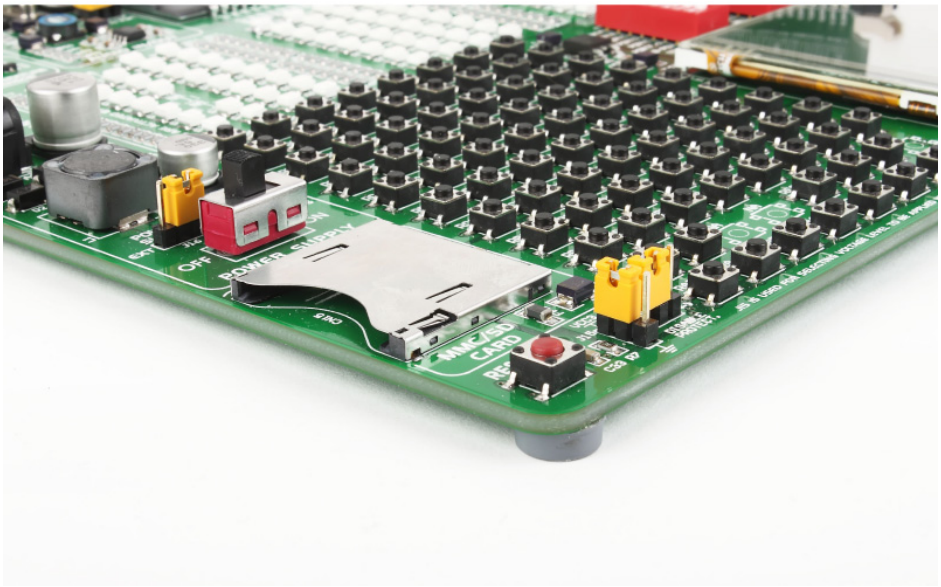
Some time ago, I resolved to build up my next printer rather than buy another BfB product. The ultra-reliable BfB extruder and controller board were two technologies that I intended to bring across. I wanted, however, to have two extruders and a somewhat bigger print table to accommodate them better than Rapman allows.

Laszlo created an inexpensive, easier to repair hot end than BfB offers which I also intend to use in the design.

As well, recent purchase of a hobby lathe ...

... will let me experiment with large diameter precision lead screws which can double as structural elements should cut down on the amount of steel in the printer rather dramatically.

The big advantage to the new Mikroelektronika compilers, however, is that they also offer an inexpensive line of prototyping boards. That will save me the development time of putting together a new board with technology that I am not familiar with. I ordered their most expensive board {$169}...



which includes everything that I need save stepper drivers. I also ordered one of their stepper driver boards. to test with it to test my firmware with ...

before I integrate the heavier amperage Pololu driver board that has been used in a number of other Reprap controller boards.

The current situation is so much better than we faced when we began the Reprap project in 2005. Then you were pretty much doomed to building up boards from scratch. These days you can buy inexpensive boards and add-on miniboards to build up an inexpensive yet very sophisticated Reprap controller board quite easily.
Should my controller design catch on, I suspect that other Reprappers would want to use this much smaller and less expensive graphics board in Mikroelektronika's stable ...

... which has pretty much everything the larger board does.

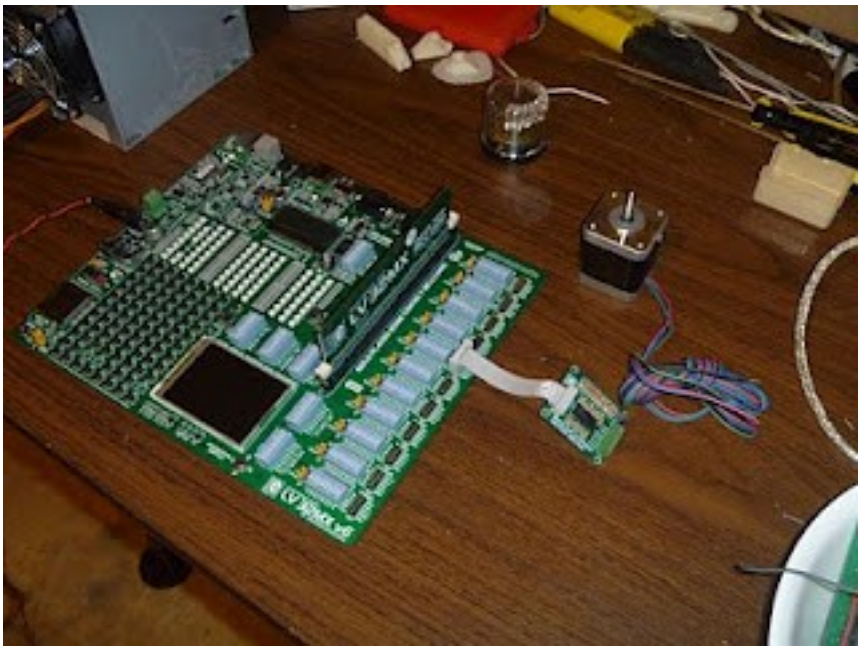
Image not found.

Development board arrived

Thursday, 10th March 2011 by Forrest Higgs

My PIC32 development board arrived today!

I also got a trial stepper motor controller with it.



The development board looks as good in my hands as it did in the sales literature.
My old PC power supply even had a power connector that fit with the board already installed. It is a relict of a battery charger for an old Sony digital camera that I finally retired last year.
I am also wanting to adopt Rapman's use of milled linear guides and linear bearings. If you look hard these are not all that expensive. For bearings, I found this little unit...

12mm Linear Motion Bearing/Bushing LB12UU
Code: **Kit12098**
Price: **$3.95**

add one to basket

(1 review)

To be well within budget. Guide shafts are a little pricier. In the US it appears that 12 mm shafting is the least expensive while still being stiff enough for a printer. Most pricing seems to be running around $0.50/inch. That's not cheap, but not impossible.
Being a Scots-Irish tightwad, however, I kept looking for a better deal and ran across this...

Compare / Select Item

60 Plus Metric Linear Shafting - 12mm Dia. Shaft
Item #: T9FB393287

Our Price: **$0.80**

Qty: Add to Cart

See all 9 items in product family

...which comes in at a bit under $0.07/inch. I spoke with the vendor today and they are checking to make sure that it meets CNC requirements for a linear guide. The general impression was that it did, but I want to be sure.

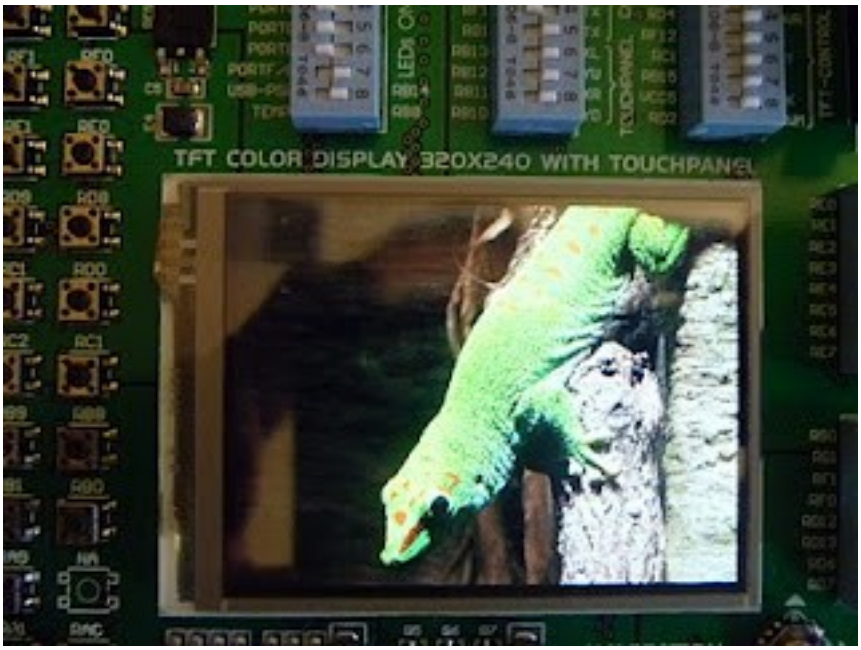
If that price does follow, it should be possible to replace the effective, but overcomplicated z-axis arrangement of Rapman which derives from the old Darwin design...

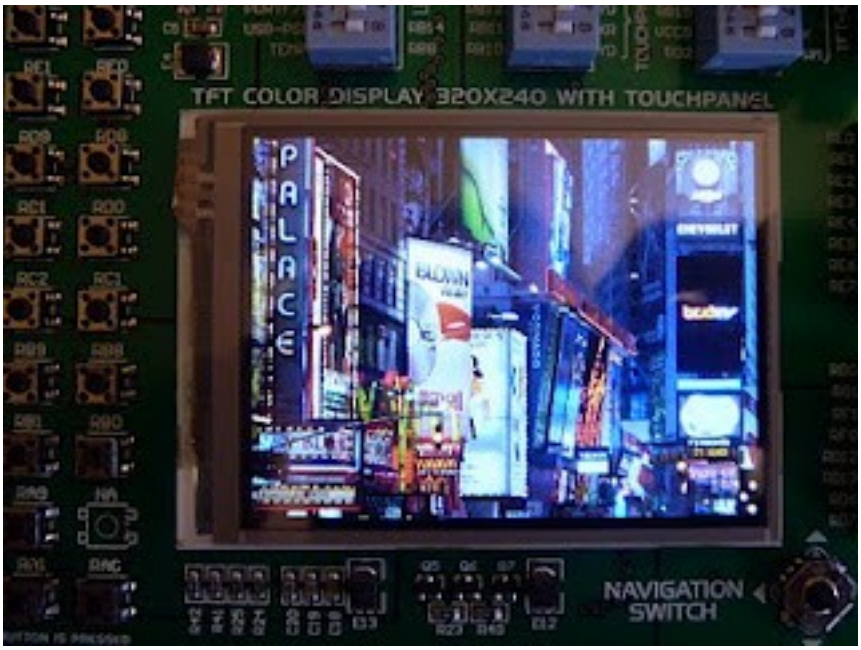
...which has 8 mm guide rod and a plastic bushing with a 12 mm guide rod and a proper linear bearing. That extra rigidity might allow me to control the position of the print table with an single, proper lead screw attached to a cantilever beam which contacts the bottom of the print table in the center instead of three or four lead screws, or cheap studding {threaded rods}, more likely. Certainly the leveraging the guide rods as vertical structural members should simplify the design a bit.

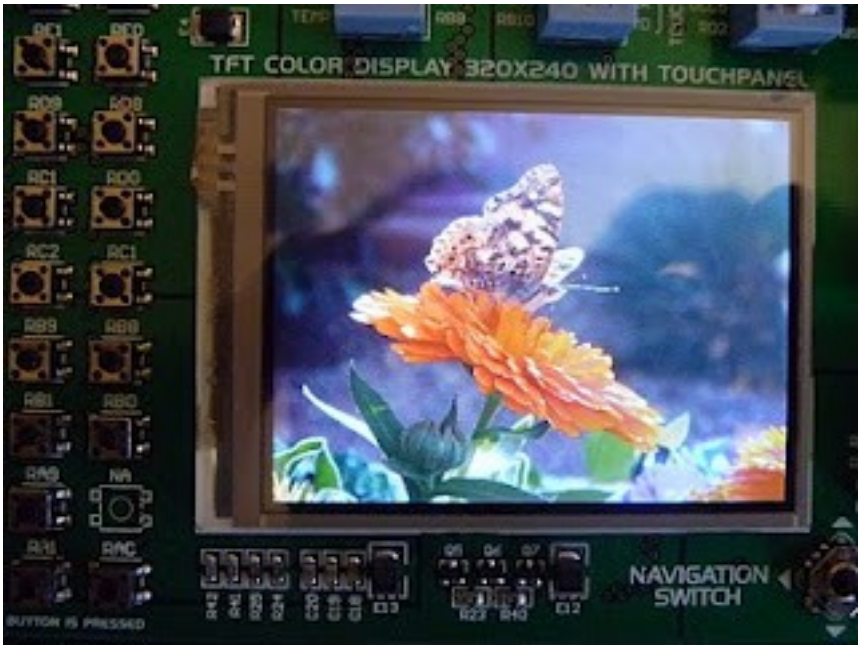
Eat your heart out! :-D

Friday, 11th March 2011 by Forrest Higgs

Fired up the development board and it fell into a roundabout of pics.






TFT COLOR DISPLAY 320X240 WITH TOUCHPANEL
NAVIGATION SWITCH


TFT COLOR DISPLAY 320X240 WITH TOUCHPANEL
NAVIGATION SWITCH


TFT COLOR DISPLAY 320X240 WITH TOUCHPANEL
NAVIGATION SWITCH

Not bad for a little 320x240 display.

Buying parts

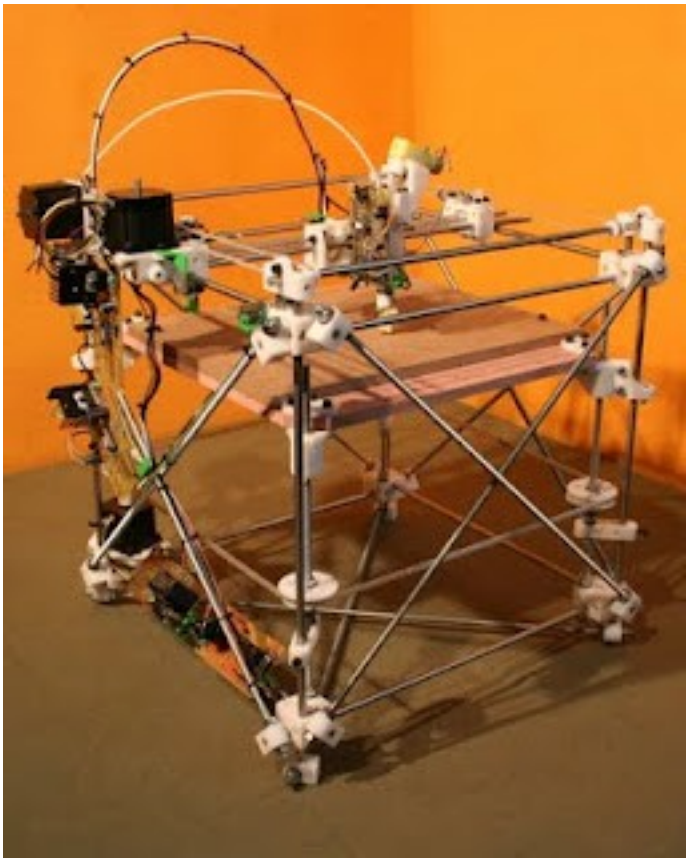
Thursday, 17th March 2011 by Forrest Higgs

This morning, I bought the linear shafting; 16x2ft 8 mm diameter segments and the linear bearings; 24x8 mm linear bearings.
The first test NEMA 23 stepper and the 24 volt power supply should be arriving today. I plan on using Bogdan's mount design for the linear bearings and an aluminum plate for mounting the extruders. Corner connectors will owe a lot to Chylld's corner designs for the Rapman, though I will not be slavishly following his approach.
I hope to be testing the firmware routines for controlling the steppers this weekend.

Sampo: The return of Darwin...

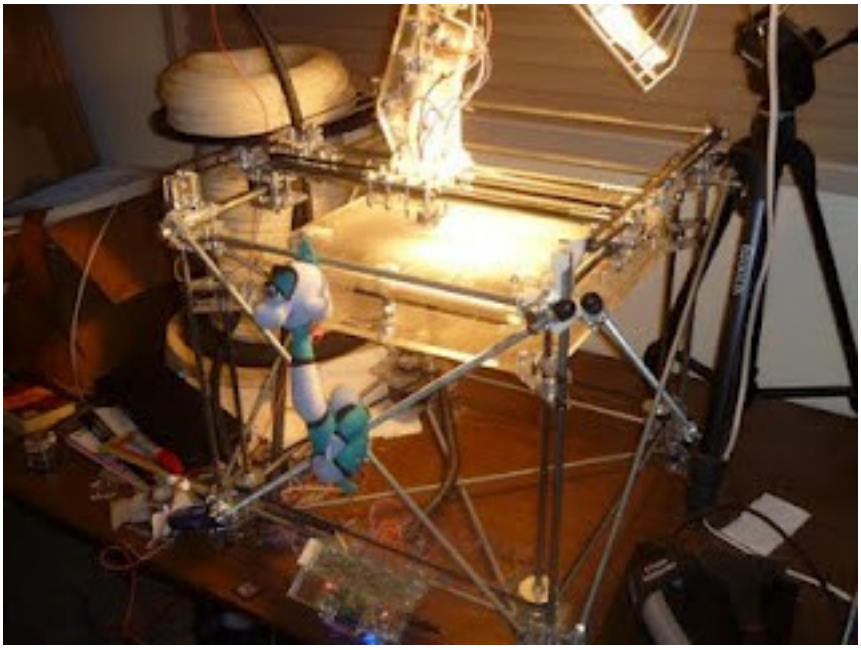
Wednesday, 27th April 2011 by Forrest Higgs

Actually, Darwin never really left.



Despite all of the hoopla about Mendel, a very robust Reprap machine which is virally spreading like a bad flu, Darwin derivatives have been quietly building up numbers largely through Bits From Bytes' Darwin-derived Rapman printer. It would be fairly safe to say that there are upwards of two thousand of these Darwin clones of one flavour or another in the field.

I bought a Rapman 3.0 in the Fall of 2009.

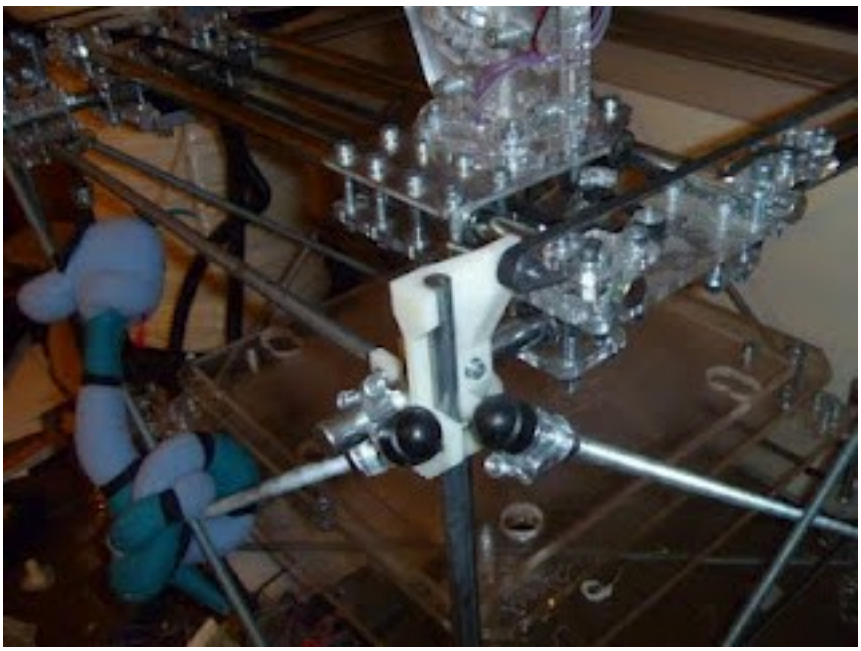


Rapman is basically a Darwin re-engineered for laser cut acrylic plastic instead of printed parts. I've never been able to decide whether Rapman is a crib of the Ponoko laser cut acrylic Darwin or vice versa. The Rapman was a brilliant choice for me. I had been working on a series of Repstrap systems. Early on in 2009 I realised that getting Tommelise 2.0 printing was going to take another half year at the rate I was going and what I really wanted to do was design things with a 3D printer in the design loop. I saw Batist Lehman's video of his Rapman in action and was sold.

Of course, Rapman has its little ways. It arrived disassembled as a flat pack. Inside were several sacks of metric nuts and bolts heavy enough, if put in a sock, to serve as a very effective cosh. Assembly was a daunting task and once finished there was a strong sense that you don't want to fool around with it for fear that it would break or some apart.

I was puzzled at first that lock washers hadn't been included with the nuts and bolts. It quickly became apparent, however, that if you put enough torque on nuts to make a lock washer compress the high impact acrylic that Bits from Bytes used would shatter. The net effect of this was that I keep a bowl beside my Rapman to collect the constant drizzle of nuts and bolts which unscrew themselves and fall on my work table.

Since then I've put well over three thousand hours on my Rapman. It became apparent some months ago that Rapman was not a heavy duty machine. I started seeing stress cracks all over the machine and was faced with a x-axis extruder mounting plate that just crumbled away from the heat of the extruder and the mechanical stress of being moved around. I replaced it with an equivalent aluminum plate.



Early on Rapman users began to design replacement parts for the bits of the Rapman laser cut acrylic that fell apart. You can see a corner block designed in white ABS by Chylld in the picture above.

You notice that I only put ONE of Chylld's excellently designed, fast printing corner blocks onto my Reprap machine. Once I got it on it has performed beautifully. Unfortunately, I had to half way disassemble my Rapman to get the damned thing installed. That wasn't an experience that I thought worth repeating.

Late in 2010 I decided that I wanted to experiment with dual extruders. Bits from Bytes had come out with one some months before and I decided that I would get one. When I called, they were so focussed on their new 3000 printer that they wanted me to switch over to that and did a hard sell. I never react positively to hard sells, so I put off the purchase. As well, the dual head Rapman had the same dimensions as the old one which meant that it had a reduced print table area as a result.

After the beginning of the new year Bits from Bytes was acquired by 3D Systems. I attended the telephone meeting announcing the takeover and was so put off by the 3D Systems CEO, Abe Reichental, that I abandoned any plans to purchase any more Bits from Bytes equipment at all.

That left me with quite a dilemma. I had never been particularly happy with the Mendel design. Tommelise had used very similar kinematics. The problem with it that it requires a larger footprint than the print table and is unstable in the x axis direction. Mendel addresses this last issue with a

electronics mounting board that doubles as a reinforcing plate to cure the x-axis problem. Because of that, I decided to stay with the Darwin kinematics and run the system through another kaizen exercise.

Basically, I want to look into several possible improvements to my Rapman

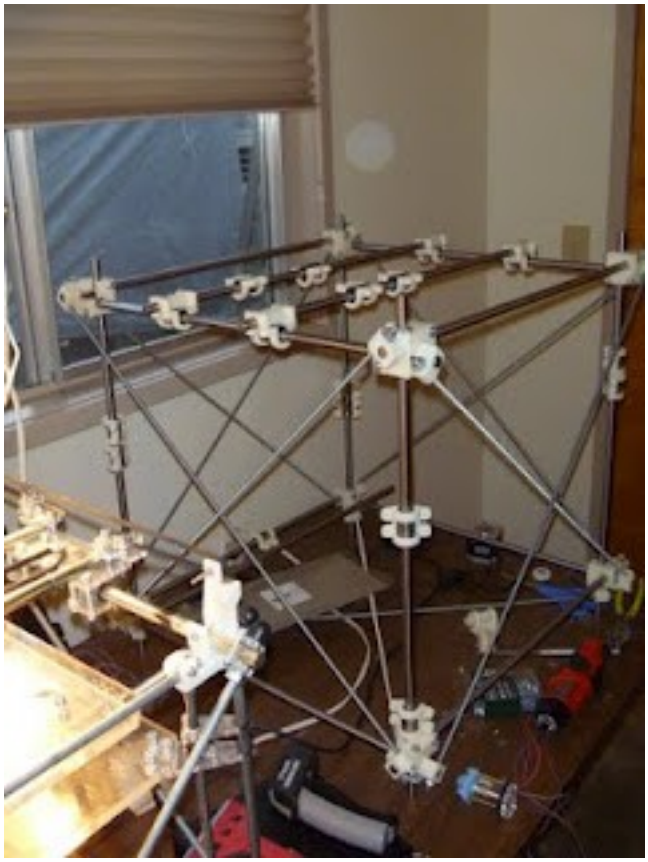
- a return to printed parts. My experience has shown that laser cut acrylic does not make for a robust machine.
- a massive reduction in the variety, as opposed to the number, of parts needed to construct the machine.
- parts design which allows for easy partial demounting of the printer.
- a design robust enough to be easily shipped fully assembled or quickly broken down and reassembled for exhibitions
- room for two extruders without compromising the size of the print table.
- cleaner y and z axis kinematics
- a more capable controller board
- shifting to the successful Wade/Adrian pinch wheel extruder

My experience with trying to install the Chylld corner on my Rapman has caused me to want to rethink the whole issue of parts design for a Reprap printer. I feel that we have, in a way, gone wrong when we designed the parts for Darwin and Mendel. Because you can custom tailor individual parts with a 3D printer doesn't, to me, mean that it is good design practice. Right now, to have a proper set of spares for a Darwin or Mendel you pretty much have to print a full parts set. A lot of that stems from, in my opinion, from custom tailoring parts.

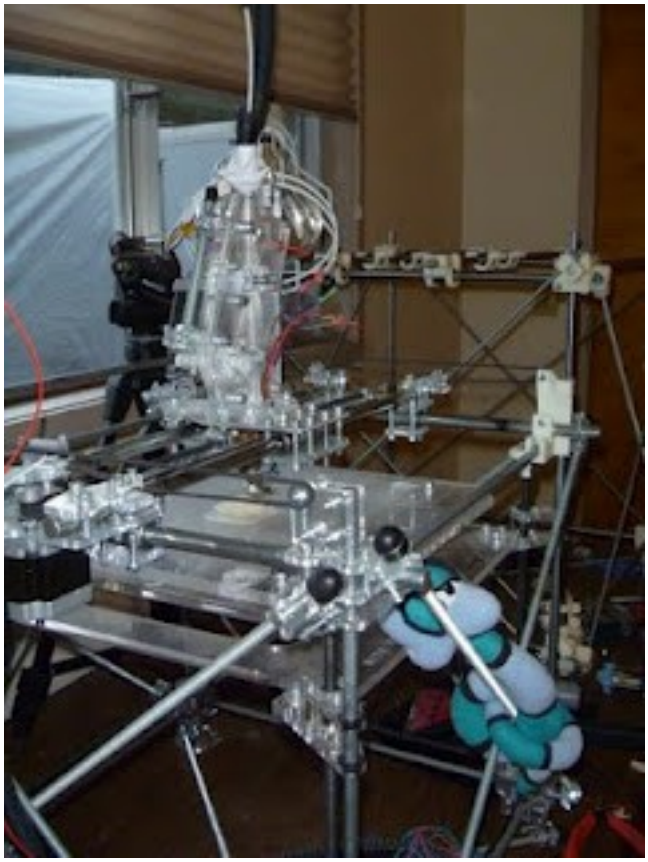
I don't think too many Reprappers are going to bite the bullet like Adrian Bowyer and Nophead {Chris Palmer} have and print whole sets of parts day after day and week after week. Most of us want Reprap machines so that we can design and print other things beside new Reprap machines. To that end, having a limited set of parts types that you can print a few at a time during down time when you are doing design, seems to me to be the way most of us are going to propagate more printers.

With respect to a new y and z axis design I want to revisit the cabling approach that eD did with the Darwin precursor ARNIE back in 2006.

But, enough talk. Here is what I have so far.



Sampo's frame is built on a 600 mm module to give it that extra space for dual extruder print frame that Rapman's 400 mm does not. So far, the entire frame has been assembled with exactly five different parts.



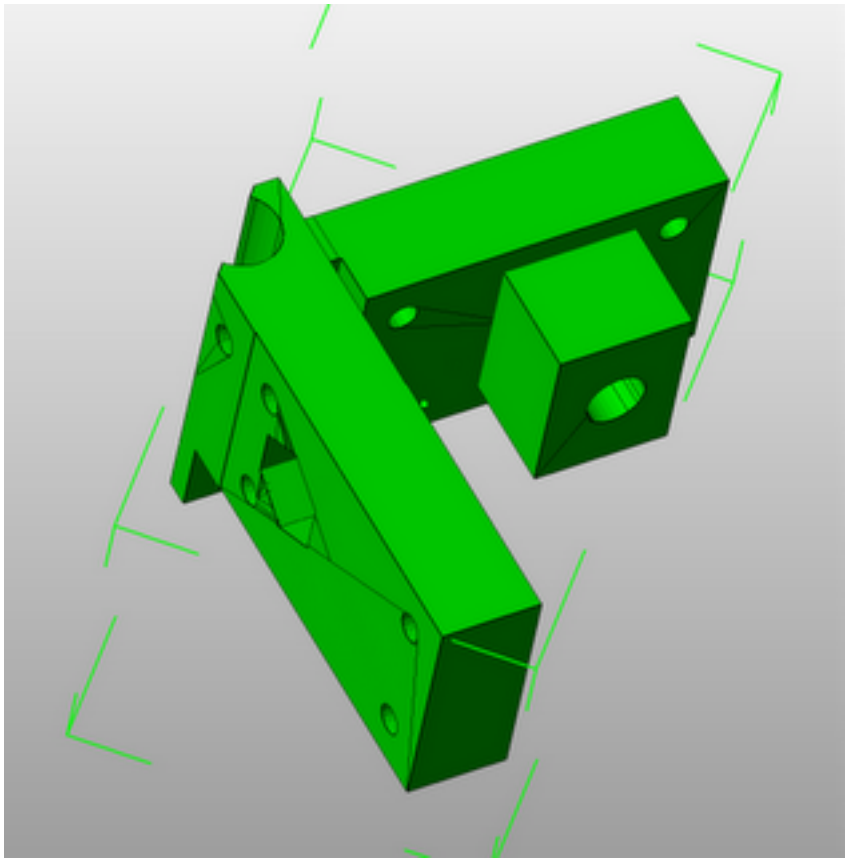
Here you can see the Sampo frame beyond my Rapman 3.0 for scale. I am working on the cable housings for the z-axis as this blog entry goes to press.

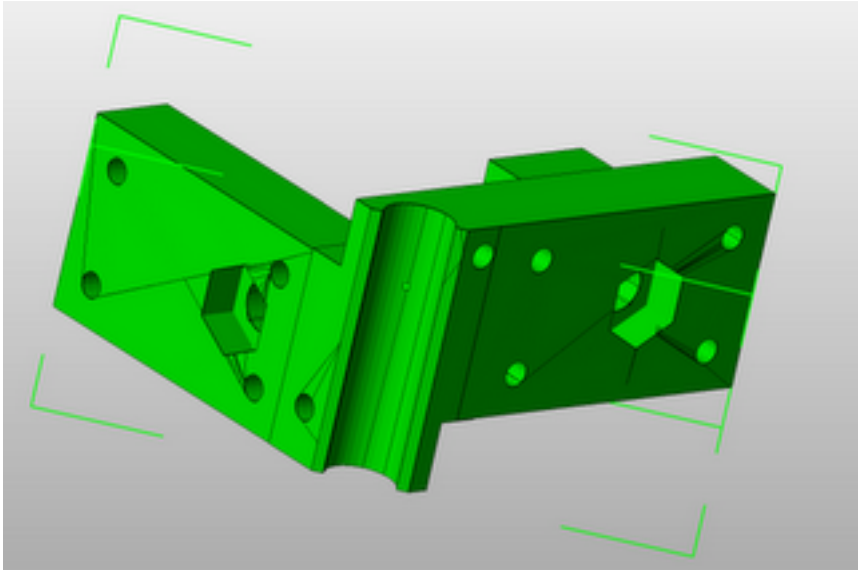
Solid prints

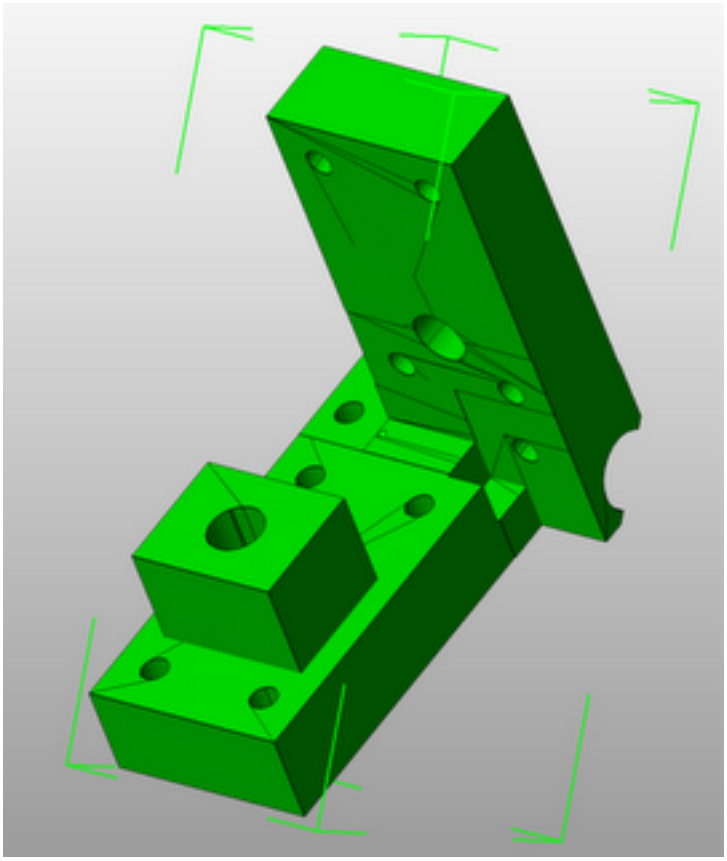
Monday, 9th May 2011 by Forrest Higgs

Printing solid, structural objects is quite an art. As I mentioned earlier, I am harking back to a very early effort that eD made with A.R.N.I.E, the precursor to Darwin. eD wanted to use a cabling system for several of the axes not unlike what you used to see for parallel bars on traditional drafting tables. eD finally gave up the effort when he couldn't get the cable to grip a sprocket wheel properly. I have another idea about how to do that which I will talk about in the near future.

Securing the cable and pulleys for the z-axis is tricky. I've spent quite a few days designing a lower corner mounting block for the pulleys.

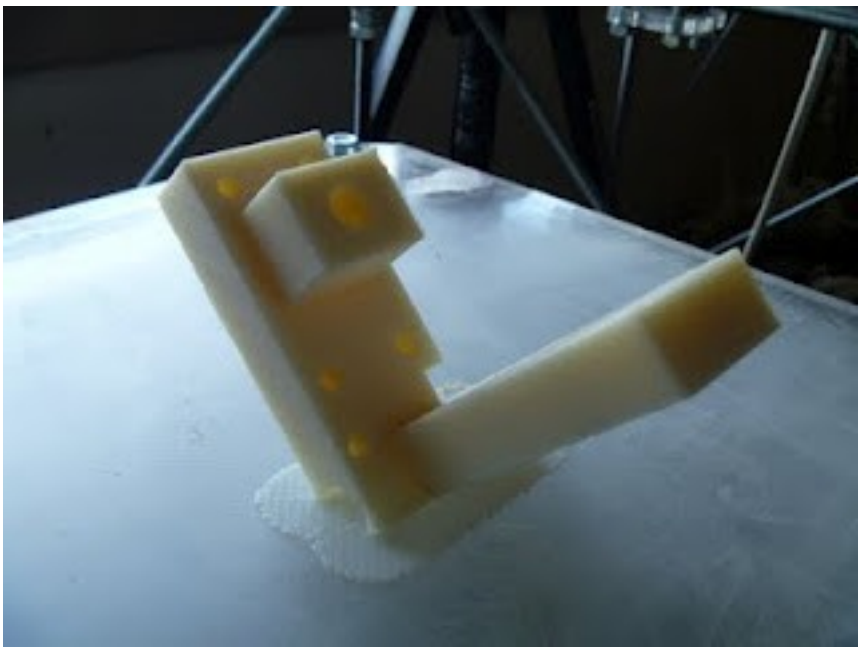




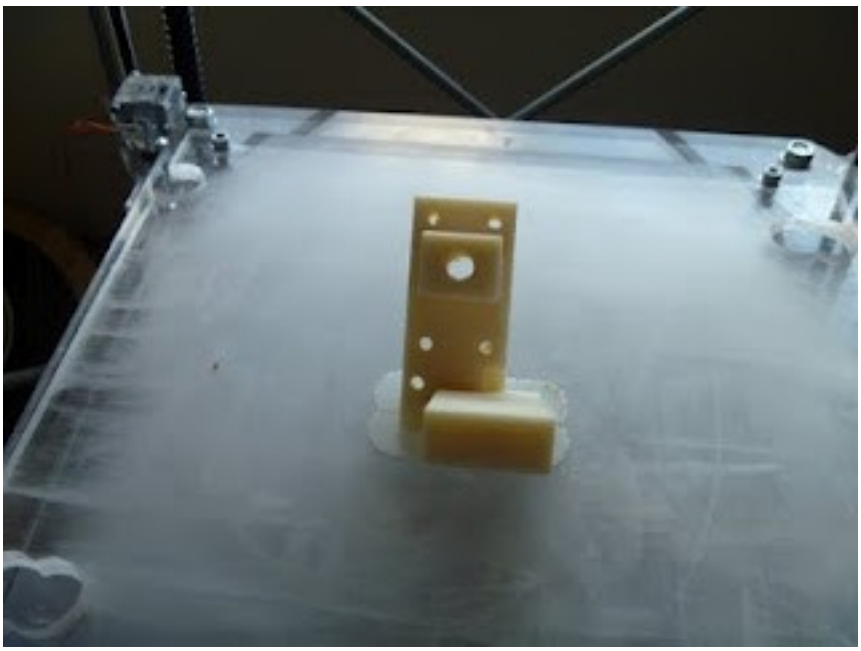


At 69 cubic centimeters it's quite a large piece. One thing you learn very quickly is that when you are designing structural pieces you have to remember that you not printing an isotropic material but rather a grained material not unlike wood. As a result of this, it doesn't do to lay this block on one flat surface. That gets you a part that is strong in one plane and fatally weak on the other. Given it's size that's asking for corner curling in any case.

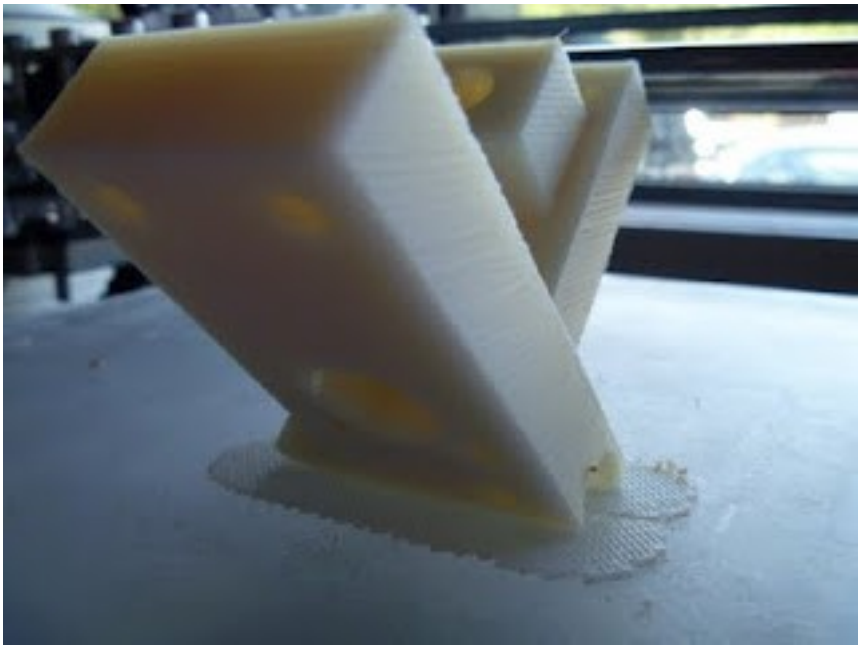
As a result, I decided to put the grain of the print at a 45 degree angle to both major planes.



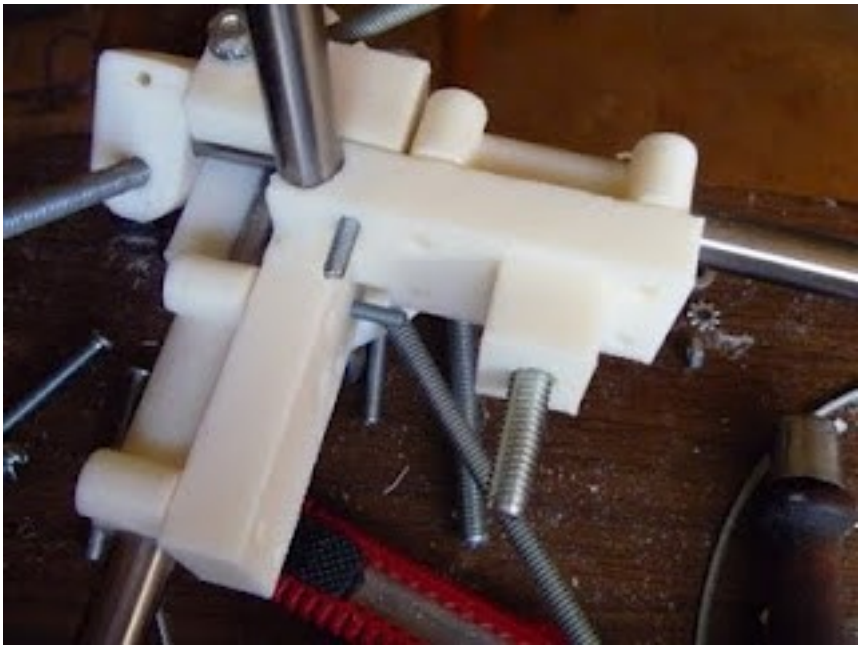
Several tries at printing it finally got me settings that yielded a very clean, handsome print. The preparation time for this piece was about 20 seconds after I removed it from the print table.



You can see how clean the bolt holes are.



As are the recessed pockets for the hex head 5/16ths inch bolts. No warping. No corner curling. Pretty much a perfect print. The longest dimension is 80 mm.

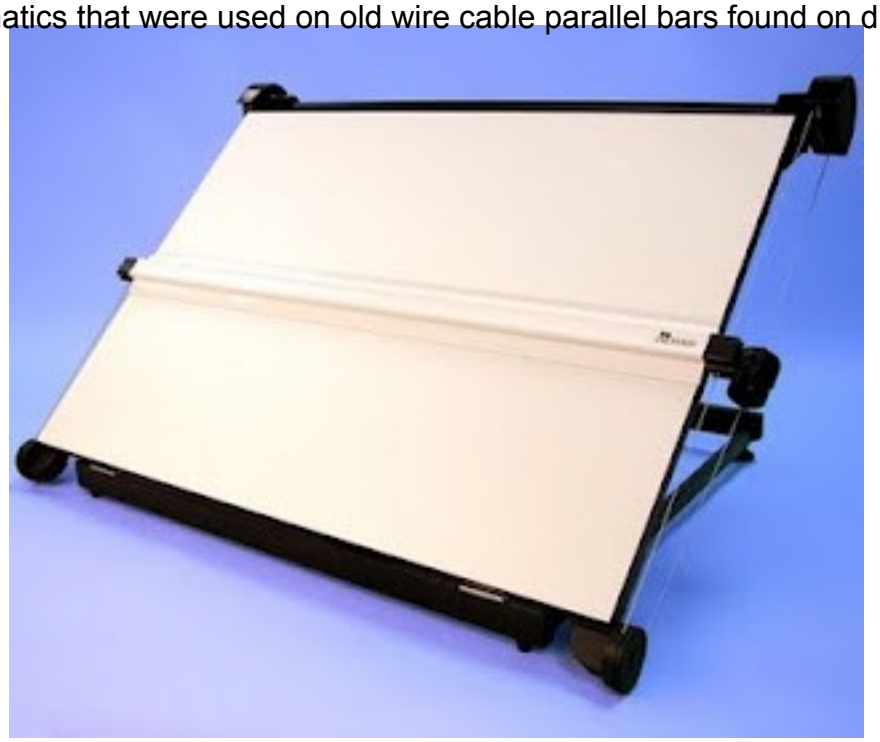


The part mounted properly without problems.

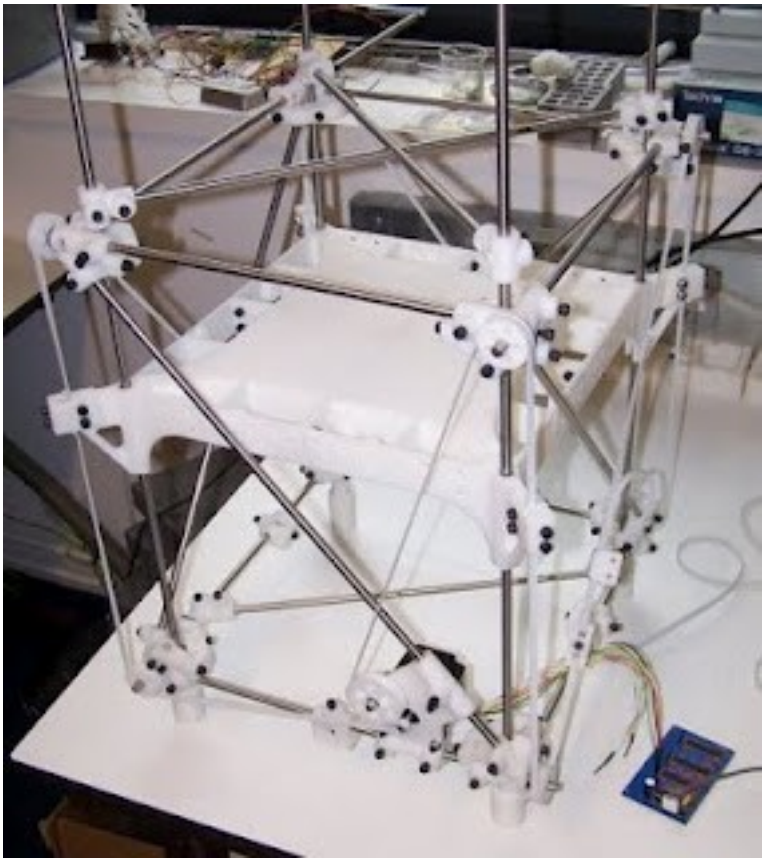
Refighting the String Wars.

Sunday, 15th May 2011 by Forrest Higgs

Back at the beginning of 2006, before there was even a Darwin, eD Sells at the University of Bath was designing Darwin's predecessor, ARNIE. Confronting the problem of designing a z-axis, eD adapted the kinematics that were used on old wire cable parallel bars found on drafting tables.

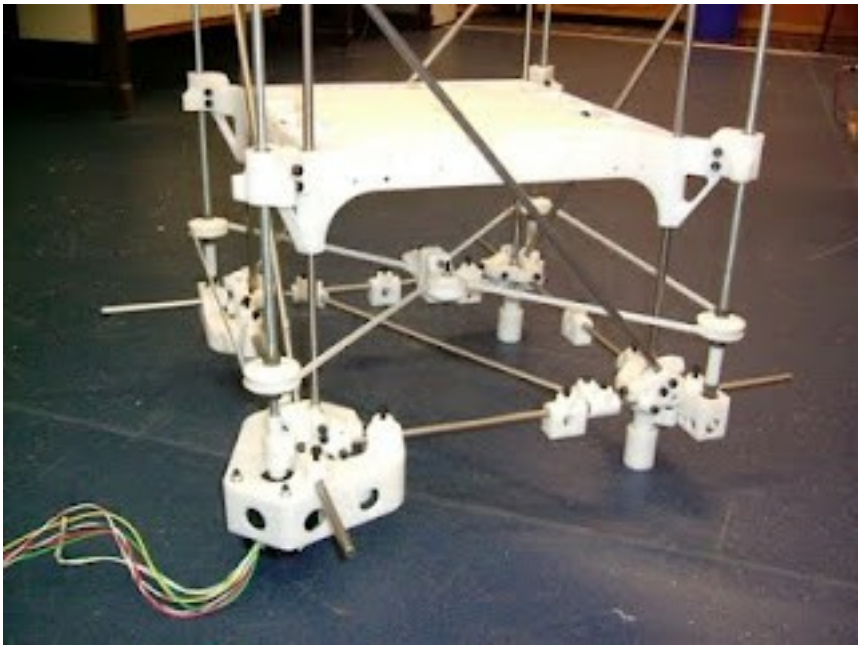


eD adapted this technology in 3 dimensions to allow a single stepper motor to raise and lower ARNIE's print table.

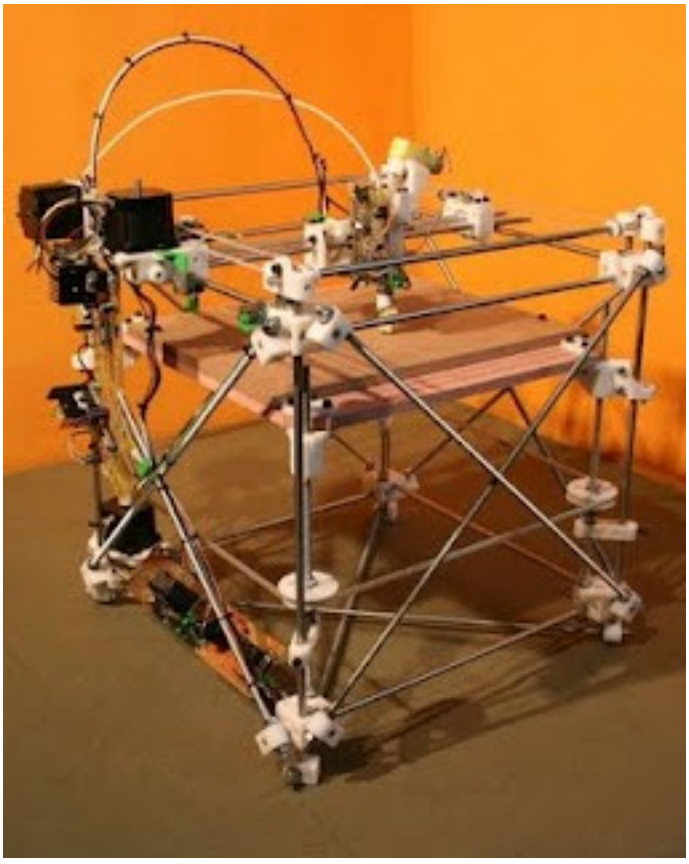


Having trained as an architect before the Great Flood, I immediately fell in love the idea and adapted it to my failed Godzilla Repstrap design.

eD encountered no end of trouble with the cabling idea and eventually abandoned it for an approach which used four pieces of studding {threaded rod}.



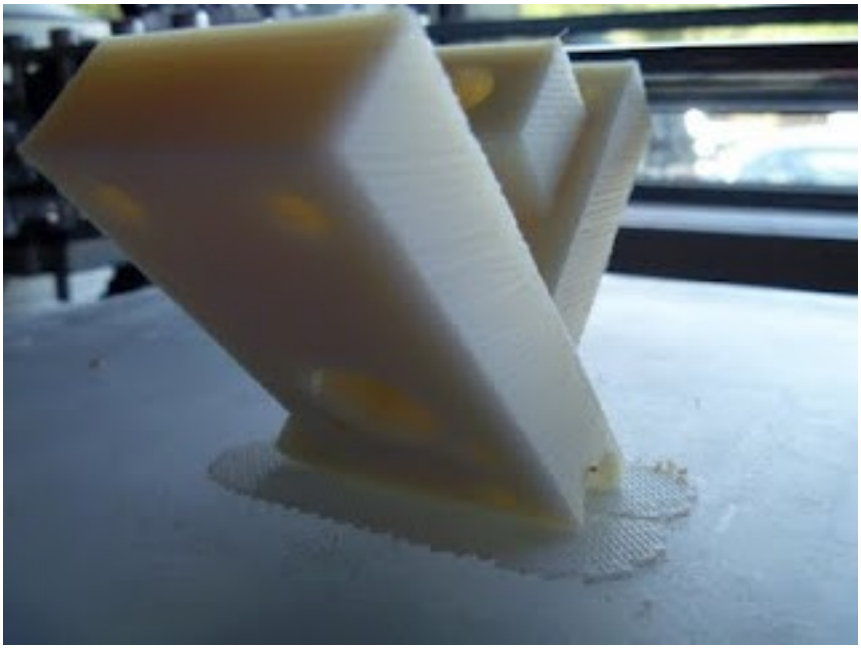
It tends to be forgotten but the first fully operational Reprap machine at Bath was ARNIE, not Darwin. Indeed, Bath's traditional whiskey shot glass, the second one printed after Vik Olliver's in New Zealand, was printed on ARNIE. This approach was refined in Darwin.



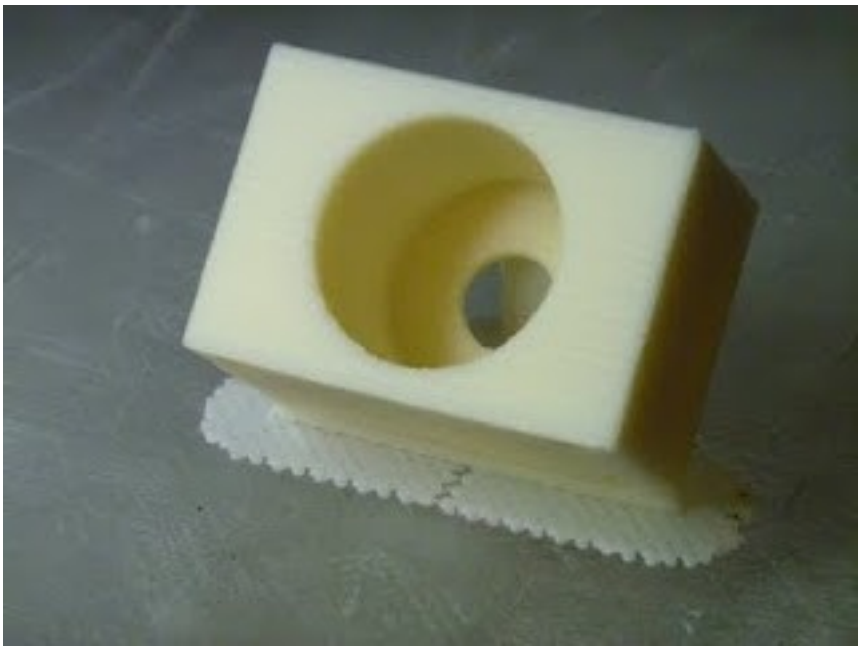
Rapman, a Darwin derivative, was put into serial production by Bits from Bytes and is still selling quite well, today.

This z-axis approach does have its problems, though. Studding is most definitely NOT a proper lead screw. When you undertake to use four pieces of studding to raise a 3D printer's print table, the tendency of studding to be not quite straight plus the fact that you are using four pieces of not quite straight studding can lead to some unpleasant consequences. Here is an extreme example of what can happen.

If you expand the pic, you can see a nasty juddering of layers taking place. Here is a more usual example of the effect.

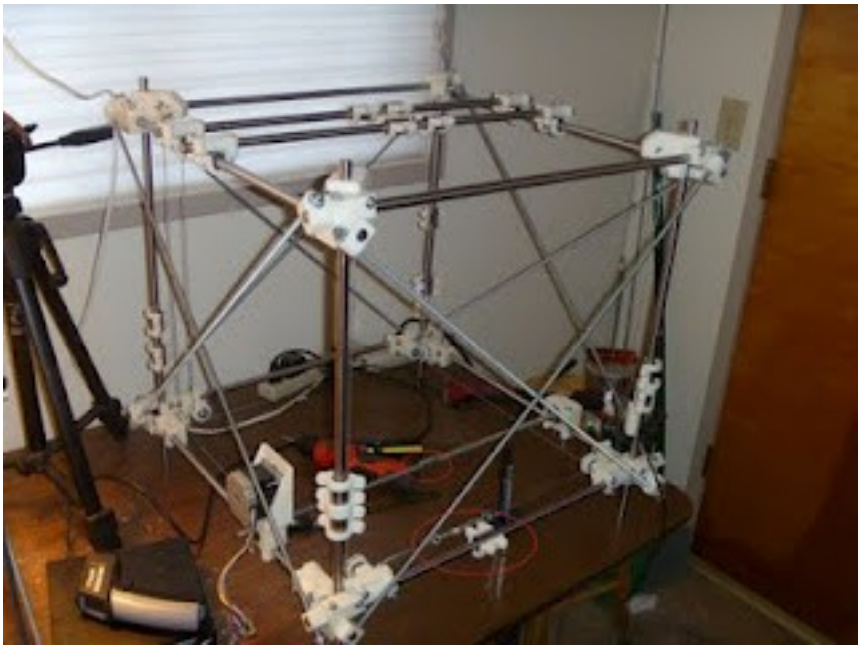


This is an extreme closeup with the light accentuating the effect. The object is quite smooth to the touch. The effect is still there, though. Here is a more usual picture showing the effect.

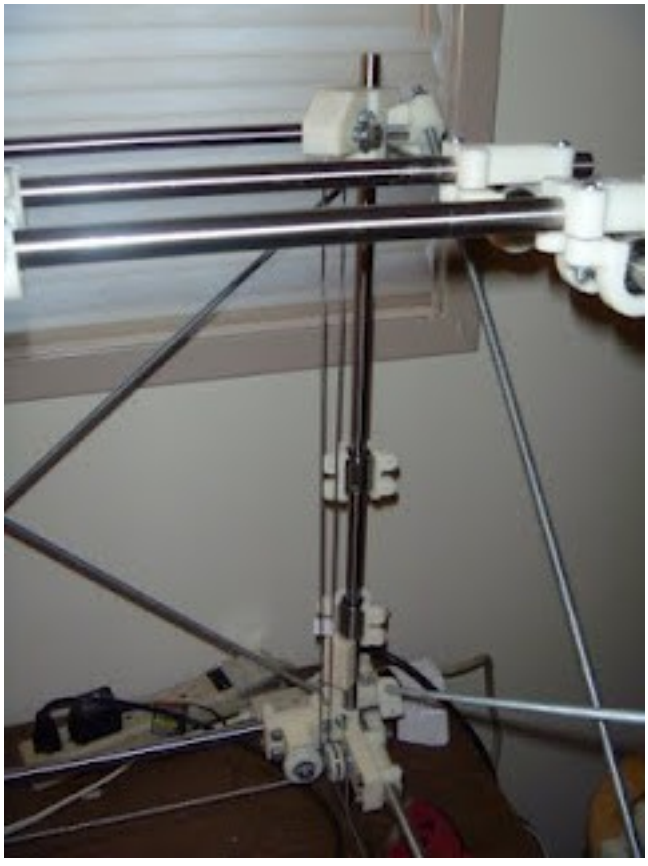


If you expand the pic you can see a regular pulse peaking at every seventh layer. This varies depending on how you adjust your machine and how straight your studding rods are. The closer the alignment, the better your print quality.
Now Bits from Bytes set out to solve this problem in their out-of-the-box BfB 3000 printer. They used a single proper lead screw to drive a cantilevered print table.

Both the Ultimaker and Makerbot's Thingomatic use the same approach.
Recently, when I decided to kaizen the old Darwin design, I decided to see what I could do about the z-axis situation. I didn't like the cantilevered print table approach and I did not want to simply duplicate the 4 studding solution originally used. That got me to thinking about the old cabling approach that eD had used back in 2006. The problem with it seemed to be applying force to the cable to move the print table. eD tried to use a friction wheel and eventually gave it up.
It occurred to me that it might be reasonable to use a single studding lead screw to apply force to the cabling. Lead screws can apply LOTS of force. So why not just attach one to the cable at a convenient point and be off?
I am in the process of doing just that.



I have circled the lead screw's thrust collar, the cabling turnbuckle and a linear bearing. Those three elements will be connected and a NEMA 23 stepper used to drive the cabling to raise and lower the print table.



Here you can see a detail of the cabling scheme associated with a pair of linear bearings on a vertical shaft. I have got to design a connector between the cable, the linear bearings and a corner of the print table. Hopefully, this approach will let me get a smoother z-axis operation without the juddering so characteristic of the Darwin design.

Another battle in the string wars: cabled z-axis working

Saturday, 21st May 2011 by Forrest Higgs

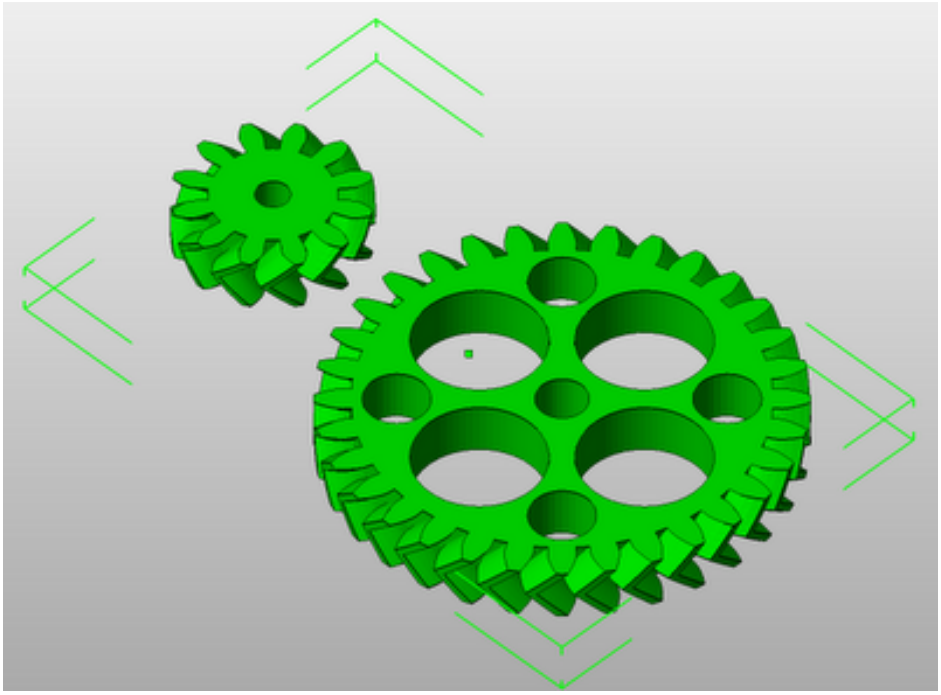
It took me 10 hours and fifty minutes to print the first print table bracket and I got a few measurements wrong, but it appears that the cabled z-axis concept is going to work.

Since the speed of the z-axis is not particularly critical, I intend to use a fairly high gear ratio between the NEMA 23 stepper and the lead screw that drives the cabling to insure that I have adequate force to overcome friction in the system.

A Z-axis gear set for the Sampo 3D printer

Sunday, 22nd May 2011 by Forrest Higgs

Few experiences are more satisfying than a creating a useful device of great intrinsic beauty

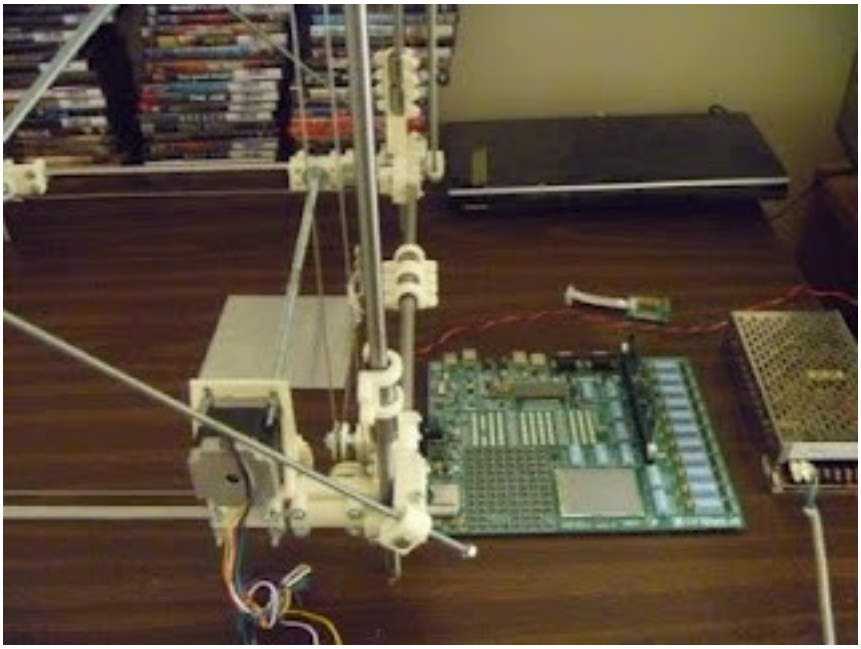


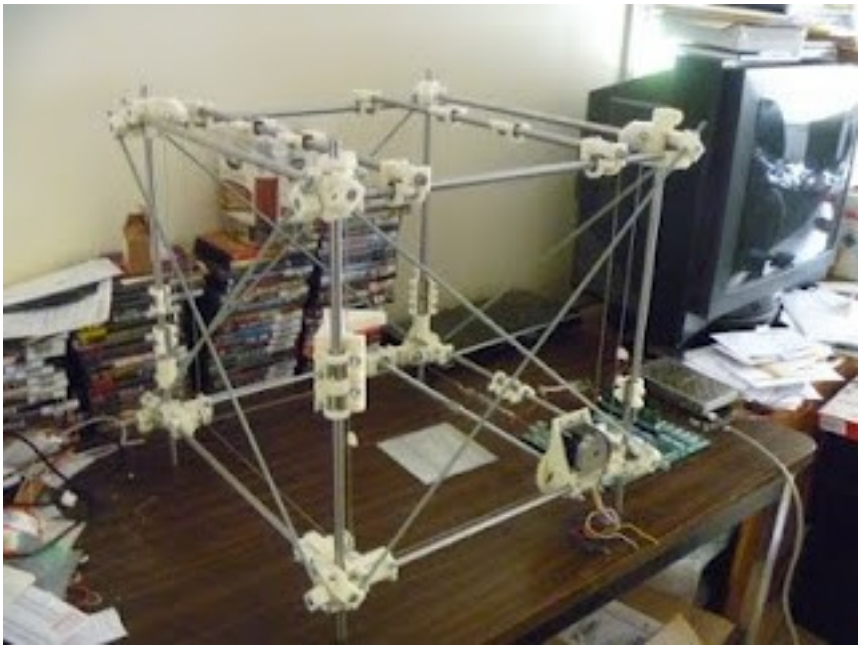
NEMA 23 connected to the Z-axis lead screw

Monday, 23rd May 2011 by Forrest Higgs

With the completion of the gear pair last night, the connection between the NEMA 23 and the Z-axis lead screw is complete.

That milestone meant that I had to move Sampo into the front room so that firmware development could begin.



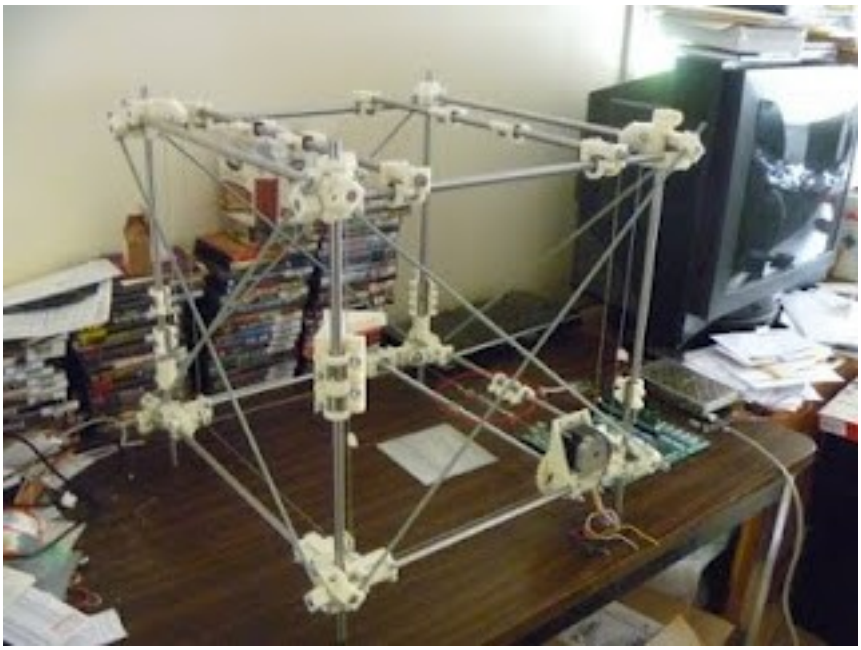


Winding up the String Wars

Tuesday, 24th May 2011 by Forrest Higgs

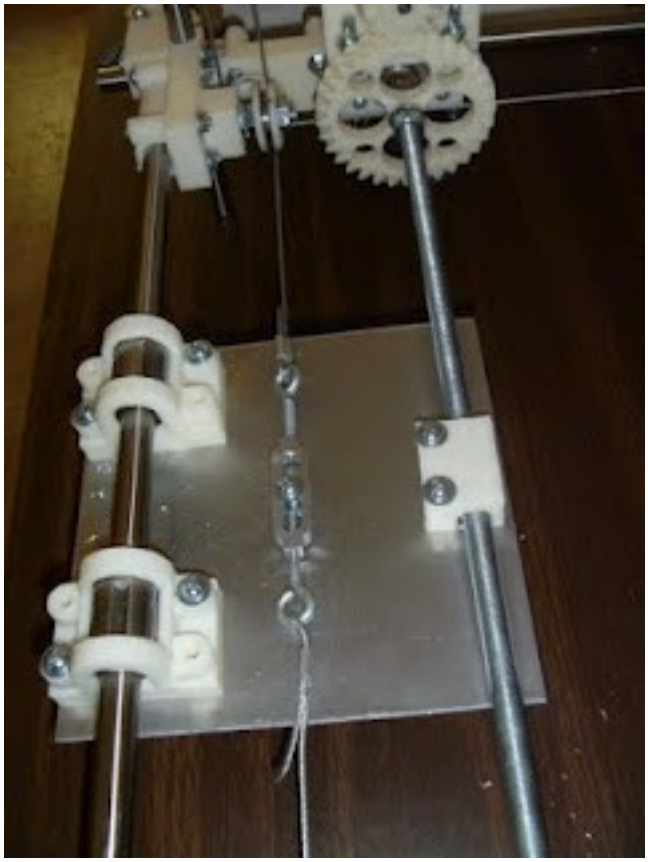
I was finally able to finish making the connection between the z-axis lead screw and the cable turnbuckle for the z-axis positioning system. Given that the 3/8-24 threaded rod moves 0.945 mm per full turn and that the 32:12 gear reduction coupled with the 200 step/full turn for the NEMA 23 we get something like 533.33 steps per turn or 0.00177 mm movement per step.

You can see the general layout of the z-axis lead screw with this pic...



I've circled, from left to right, the thrust collar nut, the cable turnbuckle and the linear bearings that make up the elements to transfer power from the stepper motor to the cable.

Here you can see the three connected...



Here is a detail of one of the four cable-driven lifts for the print table with the bracket in place...

I plan on having one fixed joint between the brackets and the table and two sliding joints constrained in the x and y-axes in the two brackets adjacent to it and a sliding joint in the xy plane opposite. I hope that will be stable enough.

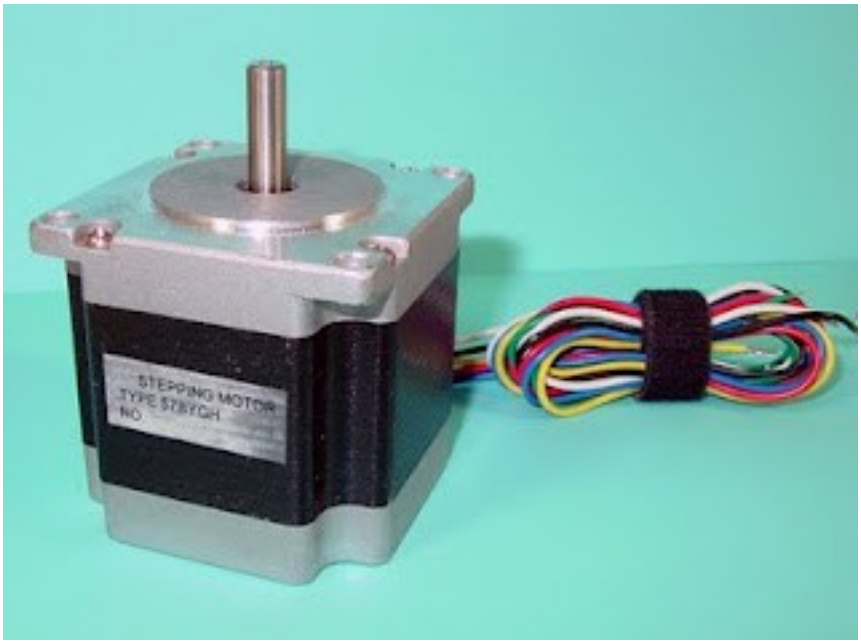
Next, though, I have to see if I can finish the design and printing of the BfB hot end adapted Wade extruder derivative. If that takes too long I will fall back on the two full BfB extruders that I have in stock.

Stepping into firmware

Sunday, 29th May 2011 by Forrest Higgs

Having had good experience with the Rapman's PIC32-based controller, I decided to stick with that MCU for my new printer. As I mentioned earlier, rather than buy a $1k+ C compiler from Microchip, I bought a much less expensive, full-featured BASIC compiler for the PIC32 {they also offer C and Pascal compilers} and a full development board from Mikroelektronika in Belgrade. Friday night, with the last of the z-axis brackets being printed on my Rapman and the 19 June exhibition in San Francisco coming up, I decided that I'd better get cracking on the firmware.
When I bought the PIC32 development board from Mikroelektronika, I also picked up a little stepper controller board from them.

It uses an Allegro A3967 driver chip rated at .75 Amps. Now ordinarily I wouldn't have considered getting such a thing, but in this case it seemed reasonable to have a ready made stepper tester board that I knew worked with my development board and I knew I had a number of small stepper motors that I could use with it. Nothing I'd consider using on the printer, mind, but all the same useful in the learning process.
Like Darwin and Rapman, I'd decided to be conservative and use NEMA 23 steppers. The price on these has dropped dramatically since we were building Darwins several years ago. While shopping for NEMA 23 steppers, I happened across this little jewel.



Oddly enough, this 6 wire NEMA only drew 0.4 amps but produced a lot of torque. I bought it, too, just so I could have have a NEMA the right size that I didn't necessarily want to use on the printer. At that time I was looking at using the much heavier capacity Pololu stepper drivers that have recently proved so popular with Mendel electronics.

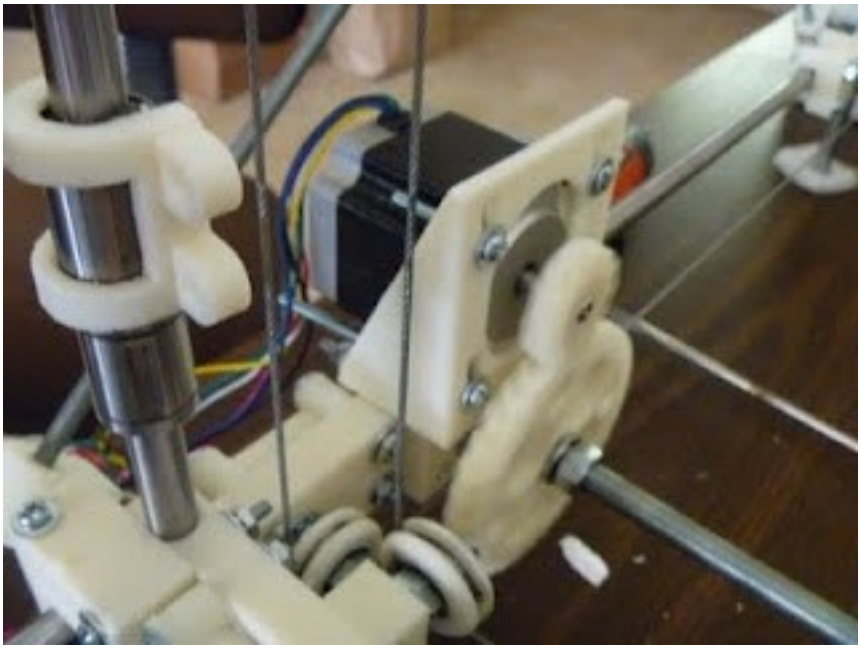
Now here is where serendipitous good fortune intruded. The firm selling this stepper also had a special on 24 volt power supplies.

When we began with Reprap about the only reasonably priced power supply we could lay hands on was a salvaged 5-12v ATX box out of old PCs. 24 volt supplies at the time were quite dear.

We knew very well that we could get a lot better performance out of steppers if we used 24 volts, but nobody wanted to invest in a 24 volt supply. This bad boy put out 6.5 amps at 24 volts for $19. The economics of that were hard to argue with given that my development board power conditioning circuit would eat anything up to 30 v DC.

Friday night and Saturday I spent the necessary hours skating down the learning curve of the Mikroelektronika development board and compiler IDE. This took longer than it should have in that the PIC32 boards and compiler are very new to Mikroelektronika. As a result, while they had code samples in BASIC for their stepper controller board, they were for 8 bit PIC chips and development boards, not their new 32 bit offerings. I don't know why firms don't offer extremely simple sample code patches. Instead, they always clutter it up with nonsense that runs LCD boards and makes LEDs flash on and off prettily. Of course, how that works on an 8 bit board is very different than it is for a 32 bit board.

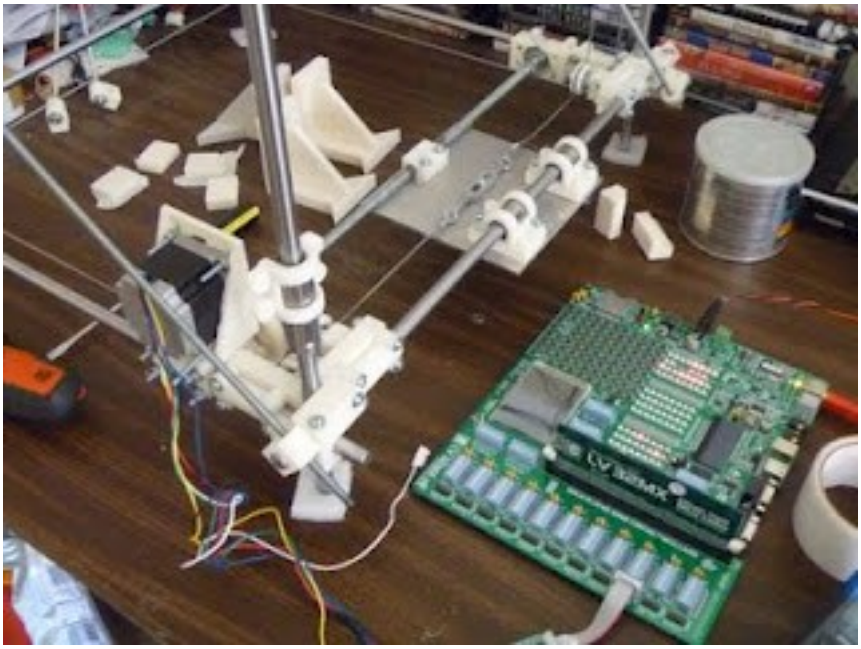
By Saturday afternoon, I'd managed to unclutter and migrate their code to PIC32 and had the stepper controller connected to the NEMA 23 working properly.



I knew there was a lot of friction in my cable z-axis system, so I did an initial gear design of 3.5:1 to insure that I got plenty of torque. I had rigged the 6 wire stepper in series at Bogdan's suggestion so that the amperage pull was considerably below 0.4 amps. Imagine my surprise when I discovered that there was ample torque even at half-step to happily push that stiff z-axis lead screw collar back and forth under serious load at 660 pps, a step rate just short of the resonance speed of the stepper. Even under those loads the controller chip never got above about 50 C even after several hours under load. That means that no heat sink is necessary. When you translate that pulse rate out to an MXL belt driven x or y axis powered by an 18 groove pulley you get a calculated top speed of about 60 mm/sec. That's about three times the head velocity that I print at. It would appear that running a stepper with 24 volt power makes a very big

performance difference.

Here you can see the stepper controller attached to the NEMA 23 and the PIC32 development board.



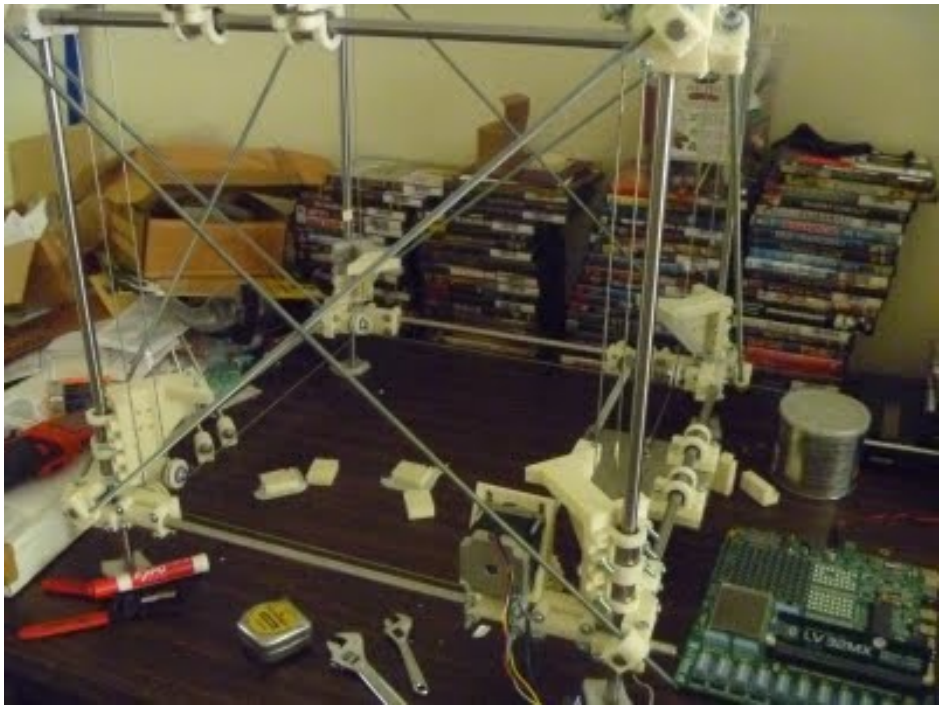
I've been thinking about that cool controller chip and that NEMA 23 and wondering about the possibility of driving a Wade extruder design with a NEMA 23. The technology and economics are certainly attractive.
I'm going to buy some more of those controllers and also a relay card so that I can control the hot ends and heat lamps on the printer.

Mikroelektronika certainly has a very big toolbox of accessory boards that let you prototype just about anything without having to build up circuitry from scratch. They're not as cheap as you could build from scratch, but if you count the time and cost of building up purpose made boards while you are developing a printer and not sure of everything you want in it, they're very cost effective.

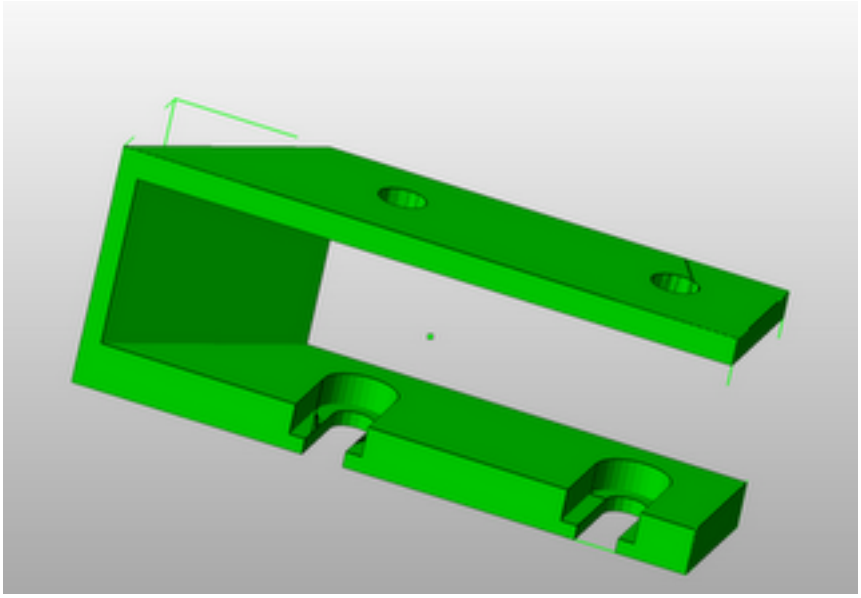
Ready to install the print table

Monday, 30th May 2011 by Forrest Higgs

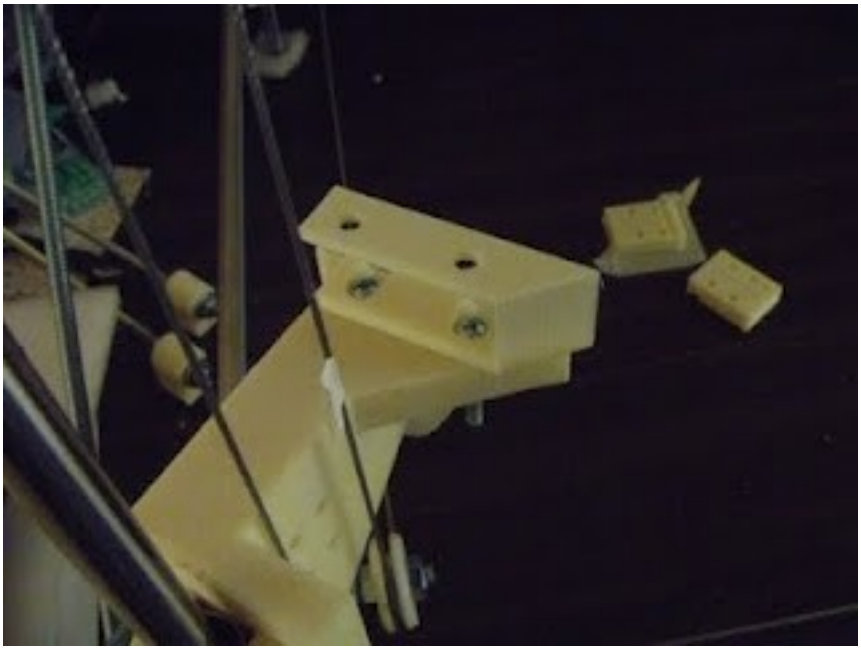
I got the last of the y-axis brackets printed and light mounted.



It looks like the print table with be 420x420 mm. I've designed a bolt-on template that seats the table on the brackets and shows me where to site the guide holes for the mounting bolts.



And here is the first drill template mounted on a z-axis bracket.

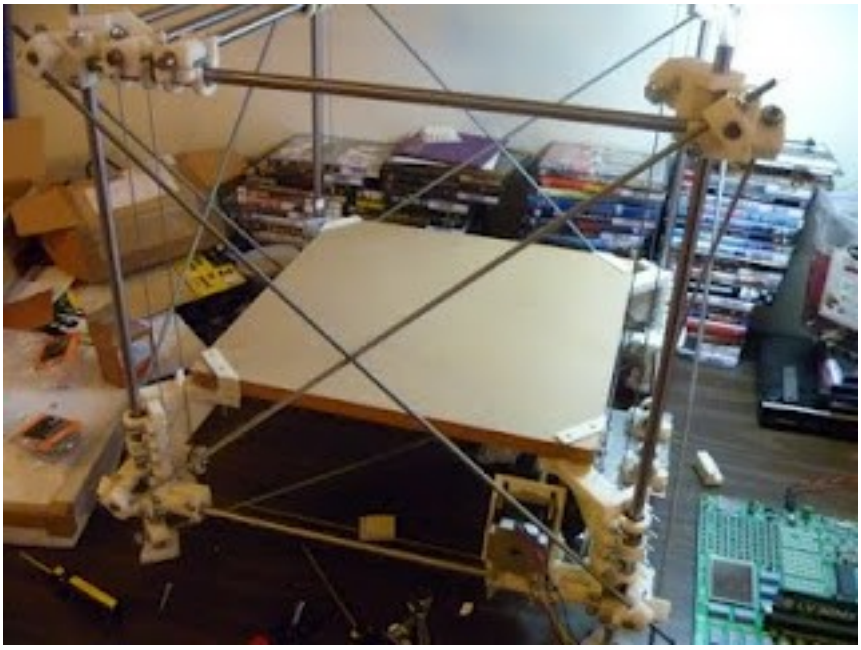


Got it right on the second try! :-)

Print table installed

Tuesday, 31st May 2011 by Forrest Higgs

I had gone to considerable trouble to make sure that I sited the bolt holes in the MDF {medium density fibreboard} print table properly. To that end I designed a template for each of the brackets that would show me where the guide holes should be put.



Once I had the table suitably aligned, I was able to drill the guide holes with my Dremel tool quite easily.

I then removed the table and took it to the workshop to drill out the holes to #8 bolt diameter. When I returned and tried to mount the board I discovered that the holes didn't line up. I had neglected to mark the lower left corner of the print table and had no way of knowing which side our orientation matched my drilled holes. We are talking about a few mm here, mind. There were 16 possible orientations and that was complicated by the fact that the brackets were able to rotate around the z-axis linear shafts in the xy plane. After about the tenth possible orientation, I found one that fit 3 of the four brackets and just redrilled both the MDF and the bracket. Mercifully, solid ABS is very amenable to drilling so other than having the lower right corner showing an extra set of bolt holes, it all worked out quite well.

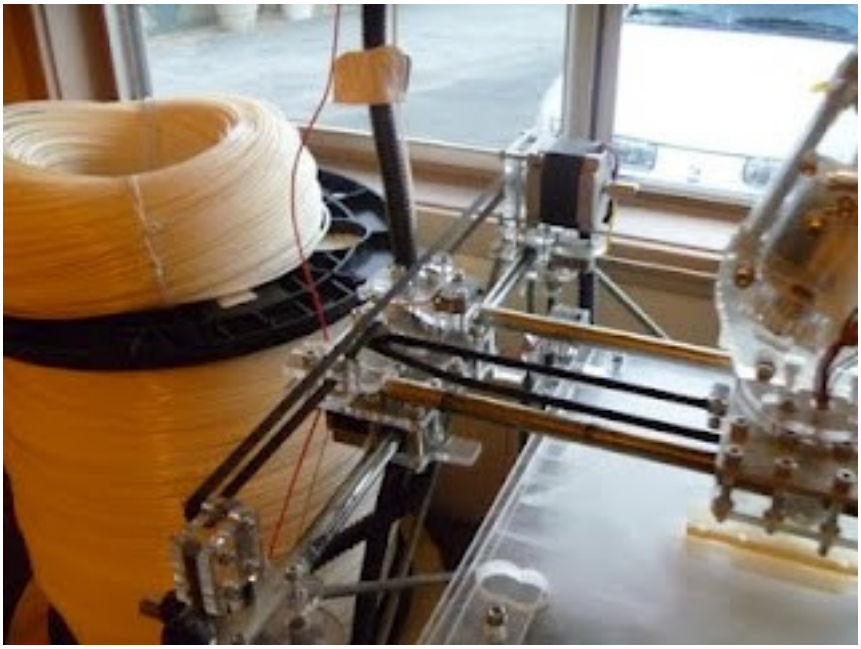
Hopefully, I will remember to mark the lower left corner the next time I build one of these. The print table moves quite freely as the video will demonstrate.

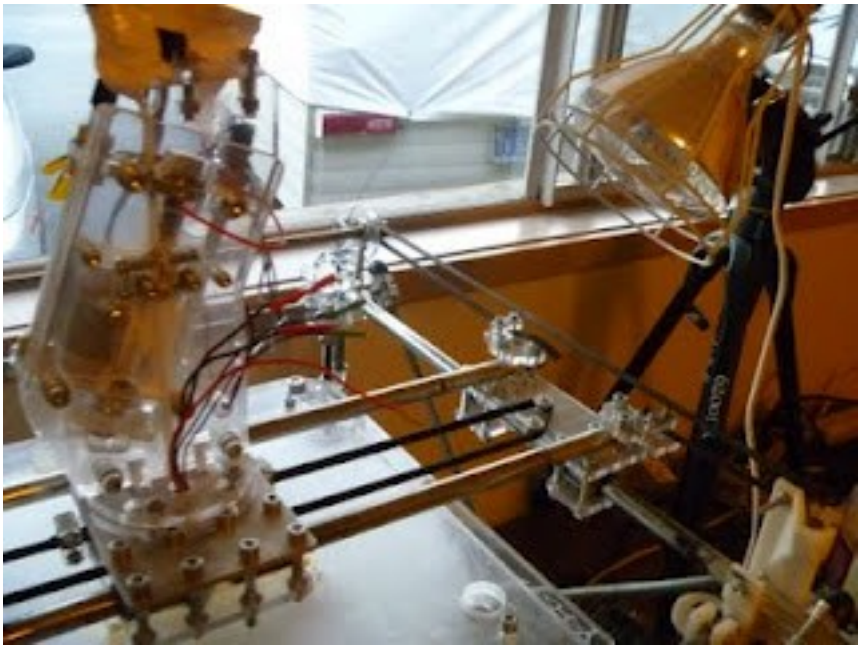
Now all that remains is for me to reprint the z-axis cable grippers and mount them and the z-axis will be complete.

A "string wars" approach to the y-axis

Tuesday, 14th June 2011 by Forrest Higgs

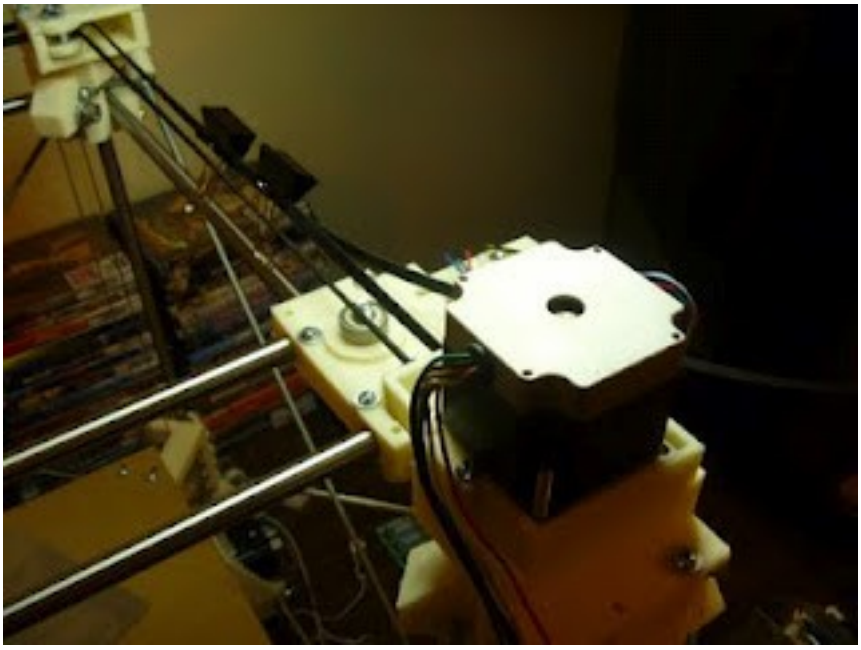
Darwin and it's direct derivatives uses two belt loops driven by a shaft connected to a NEMA 23 stepper.



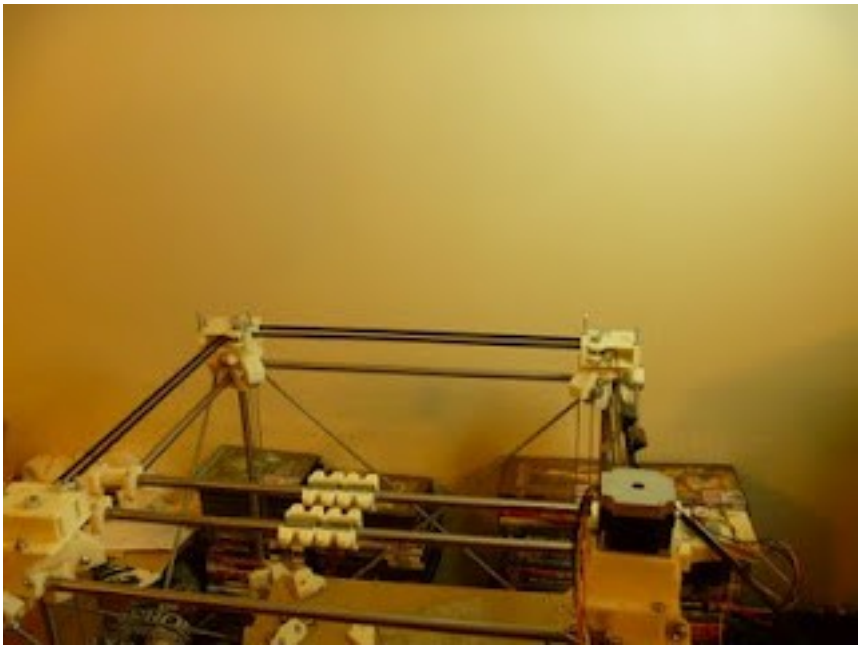


In Sampo, that dual loop arrangement has been replaced with a single large loop

The belt guides are equipped with standard 608 skateboard bearings with printed fenders.



Topologically, the y-axis is simply an attenuated version of the x-axis.



Flex cable carrier

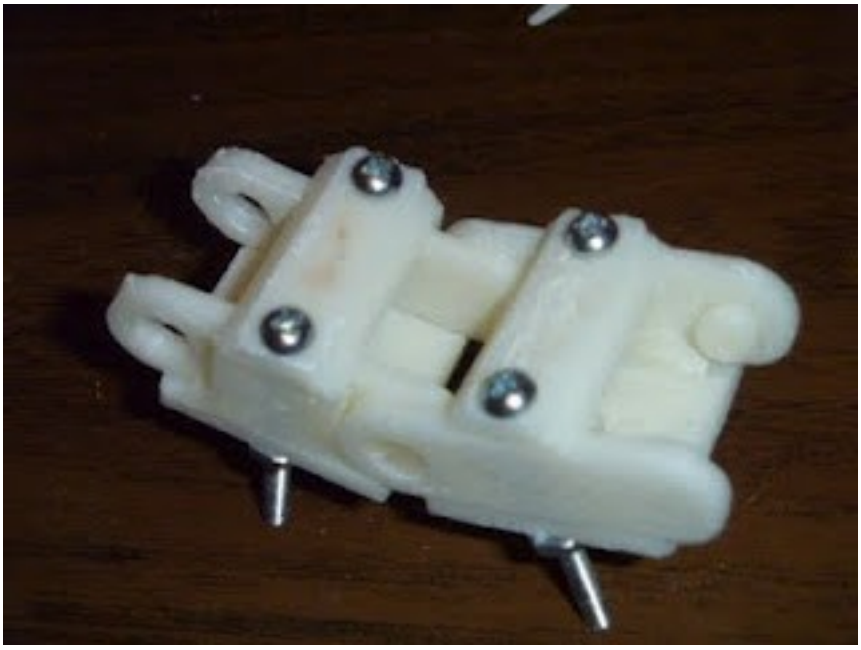
Sunday, 26th June 2011 by Forrest Higgs

I've never been happy with the way that Rapman handles axis and extruder cabling, so I decided to print my own flex cable carrier system. I saw several possibilities in Thingiverse. Most of them were knockoffs of existing injection molded parts, however, and printed very poorly.

A few looked as if they were designed specifically for a 3D printer like this one...

I didn't much like this one largely because of the large turning radius for the flex. I wanted something more like this...

Since my wiring was considerably more modest, however, I wanted something with a bit sharper turning radius still. After several hours of trying out alternatives in Art of Illusion, I came up with this as a first try.



It is shown here with #4 bolts 1.5 inches long. It can work interchangeably with #4 UTS/SAE - 3/4 inch or M3 - 20 mm bolts. With fasteners it costs about $1.50/ft. Commercially available alternatives average about $12.50/ft.

The system can make a 180 degree flex within 50 mm. The next move is to get a couple of packets of #4 bolts and nuts and print a few feet of this to try with the x-axis cabling.

Printing flexible cable guides...

Wednesday, 29th June 2011 by Forrest Higgs

There is not much to say about this. Once I got two of the links printed and assembled so that I knew everything fit together, I bought a few hundred #4-40 3/4 inch machine screws and nuts to hold the parts together. I'd designed the parts to perfectly seat a 3/4 inch machine screw and nut.

When I got home with my trove of fasteners from my stockist I discovered that his Chinese supplier had been making a little extra money by trimming his 3/4 inch screws (0.75 inch) down to 0.714. What that meant was that the screws went all the way through the guide assembly but didn't emerge on the other side to allow the nut to be seated on the end of the machine screw.

My stockist is getting me some 7/8 inch machine screws as replacements and writing a hot note to the warehouse. Quality assurance at the Chinese plant needs a bit of a rework, I think.

Interestingly, I had designed the holes for the #4 machine screws so that the threads engaged the sides of the holes, so actually nuts weren't required. With that in mind I went ahead and assembled the flexible cable guide for the x-axis. It seems to work perfectly.

I get a tight turn like I'd hoped with no clashing. Right now I am up to 16 inches of a 24 inch assembly for the x-axis. When I get the full 24 inches printed and assembled I will design and print the end mounts.

It will be interestingly how many hours of operation this kind of flexible cable guide will handle before something wears out.

Y-axis test firmware operational

Monday, 4th July 2011 by Forrest Higgs

After a delightful chat with Bogdan this morning about the PIC32's MIPS core processor, I was able to get the y-axis test firmware working. Bogdan has done a lot of work with the PIC32 and is very generous with his knowledge. A few hours later, I had the y-axis responding to the limits switches.

Right now with sampling from the limits switches in the same loop that runs the stepper I'm getting 15 mm/sec. On its own the stepper can do about 52 mm/sec. I suspect that the speed of the axis will be getting a lot closer to that upper limit once I get the limits switches into an interrupt loop. :-)

Picking up speed

Tuesday, 5th July 2011 by Forrest Higgs

The MIPS core that PIC32 uses is a very high performance CPU that was used in high end Windows workstations in the early to mid-1990s. It doesn't behave much like the 8 and 16 bit PIC chips, so it's taken me a while to get down the learning curve. The button function in the Mikroelektronika compiler library works, but requires about 10 msec to filter out the bounce when a limits switch is encountered.

Processing a button function for each step when I was running at half step slowed the y-axis down to 12-15 mm/sec. To get around that problem in firmware I wrote a smart limits switch routine that runs slow until it finds the first limits switch and then kicks the stepper motor up to full speed until it nears the other limits switch.

Using this approach lets me increase the maximum transition speed for the y-axis from 15 mm/sec to about 65 mm/sec for my firmware testing as you can see in the video clip.

It's worth noting that the Allegro driver chip has a maximum rating of 0.75 amps and the NEMA 23 is a six wire model drawing 0.5 amps per phase. I've wired it in series which brings that amperage down considerably. In spite of this I'm getting 65 mm/sec and both the stepper and the NEMA 23 are running quite cool. The driver chip requires no heat sink or fan.

Printing a Mendel derivative

Saturday, 9th July 2011 by Forrest Higgs

After talking with my son recently, I concluded that he needs to start printing a lot faster than I'm going to have Sampo debugged and duplicated. I have a lot of 8 mm linear shafting and linear bearings in stock, so I decided to build him a Mendel derivative using Sampo firmware and controllers. That should get him printing a lot faster than would otherwise be the case.
I started printing parts for a Prusa Mendel yesterday. So far, so good. Got the gantries finished and am printing the rest of the parts now.

5/16th inch threaded rod is pretty much a one on one replacement for M8, #8 machine screws replace M4s and #4 machine screws replace M3s. I couldn't see much point in printing the SAE Mendel. I'm going to have to redesign the extruder carriage to seat linear bearings instead of those strange PLA things it ordinarily uses.

Mendel frame takes shape...

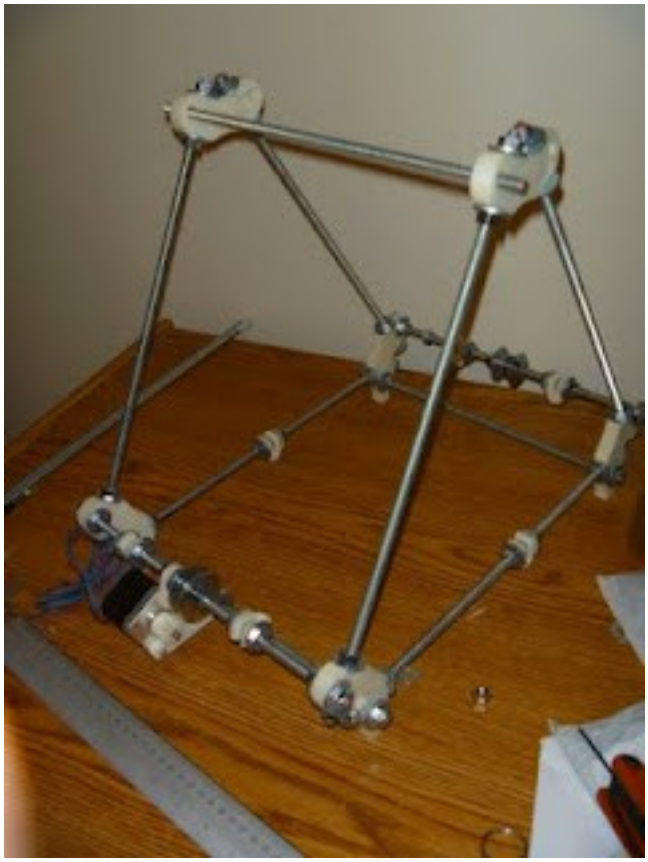
Sunday, 10th July 2011 by Forrest Higgs

I had a major crash of the Rapman 3.0 printer and for several hours I thought I was really knackered. One of the leads to a phase of the extruder stepper parted because of fatigue from thousands of hours of vibration. The system shut down and reset. At first I thought I had a simple static discharge event, the first in many months. I fired the system back up and discovered that the extruder stepper would dance around but wouldn't pump filament.

After serious prayers that the stepper driver chip for the extruder hadn't fried I rewired the connector and got the extruder pumping ABS again only to discover that the intense vibration from the stepper before the reset had actually shaken apart the Arcol extruder hot end that I had bought from Lszl Krekcs in Hungary.

This sounds worse than it was. I simply screwed it all back together and cleaned the extruder end and it worked perfectly again. Unfortunately, I will have to recalibrate Rapman now. That should take a few hours that I didn't have available today.

In any case, I was able to cobble the Prusa Mendel frame together this morning.

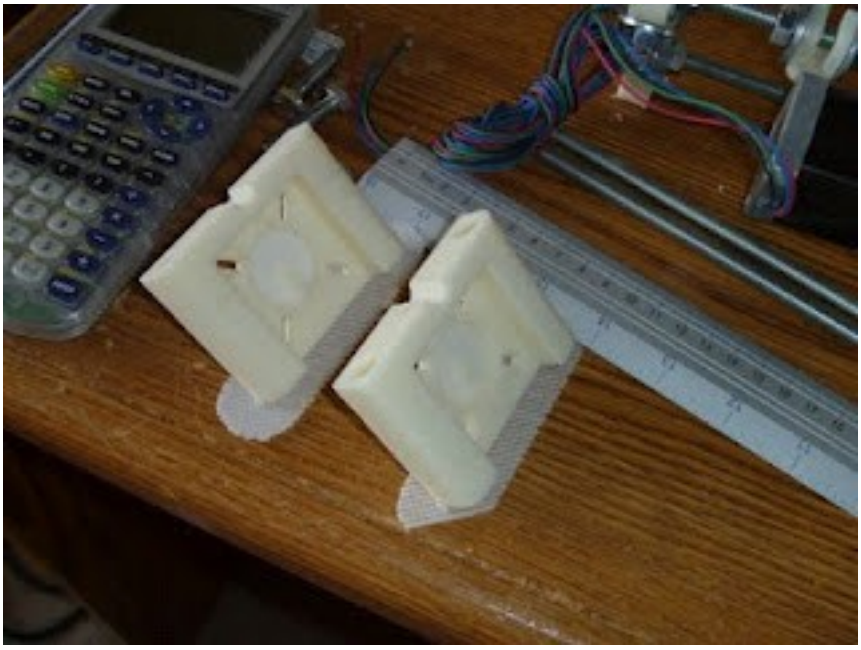


It's a dinky little thing, but interesting all the same.

Mendel z-axis stepper mounts done

Wednesday, 13th July 2011 by Forrest Higgs

I managed to get some hours together to do some more work on my son's Prusa Mendel, the z-axis stepper mounts this time.



I printed the z-axis stepper mounts at a 45 degree angle to minimize parts preparation time and avoid warping.

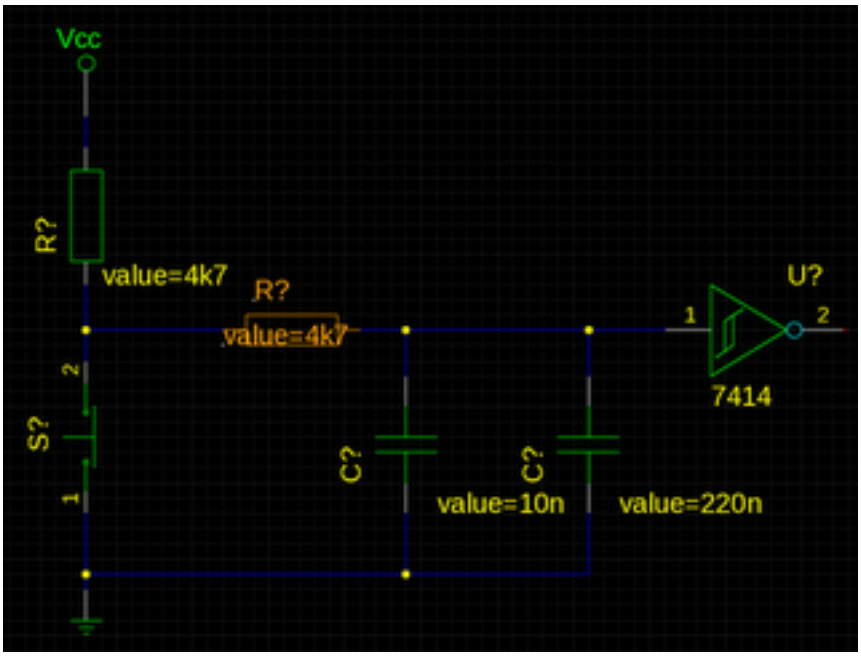
I'm now working on the x-axis stepper and idler mounts. My son processed the stepper mount last night and I did the idler mount this morning. They're printing this evening.

Leveraging Bogdan's anti-bounce circuit for Sampo...

Wednesday, 20th July 2011 by Forrest Higgs

In developing the Darwin-derivative, Rapman-derivative Sampo 3D printer project as a kaizen exercise I utilized the same sort of microswitches for limits checking as are specified in the Rapman design. I soon discovered that the switches have a formidable electronic bounce. I was able to control that using the button function in my firmware compiler for the y-axis. The computations taken for a firmware fix, however, were going to put a terrific drag of my MCU that I didn't want to have to deal with.

Enter Bogdan Kecman with helpful suggestions on how to put together an antibounce circuit for the limits switches.

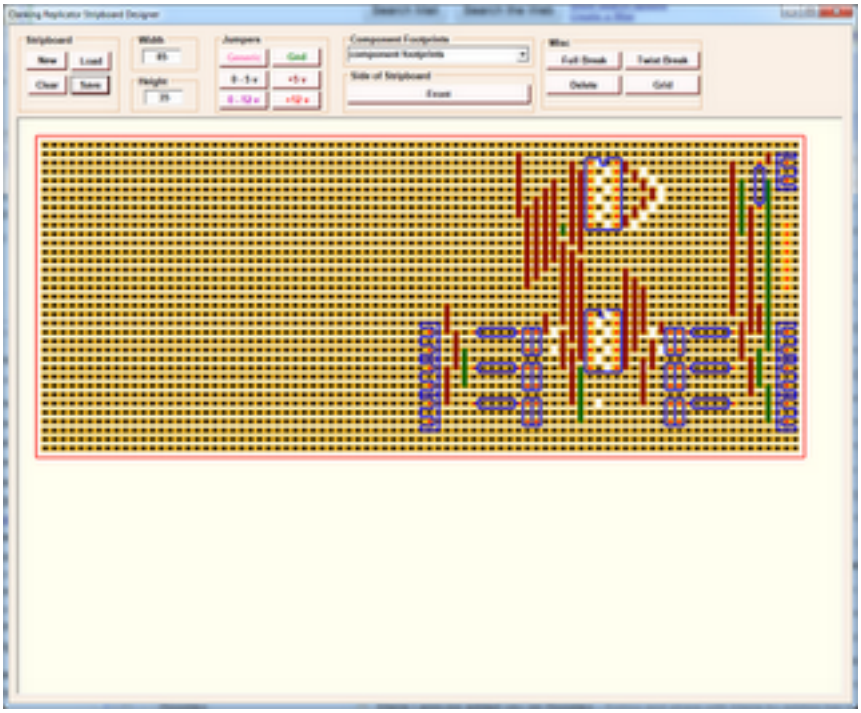


I had last built an antibounce circuit in 1981, so his help was greatly appreciated. I built a lashup of the circuit to check to see that the component values were right and then went on to design a board to handle all six limts switches. I wanted six instead of Rapman's three because a lot of problems that I'd had with Rapman stemmed from the fact that it has limits switches only on one end of its axes. When things went bad one could find steppers trying to skate off of the unchecked far end of axes. As well, Rapman limits checking only seems to be done when one is resetting the axes at the beginning of a print. I want to do better than that.

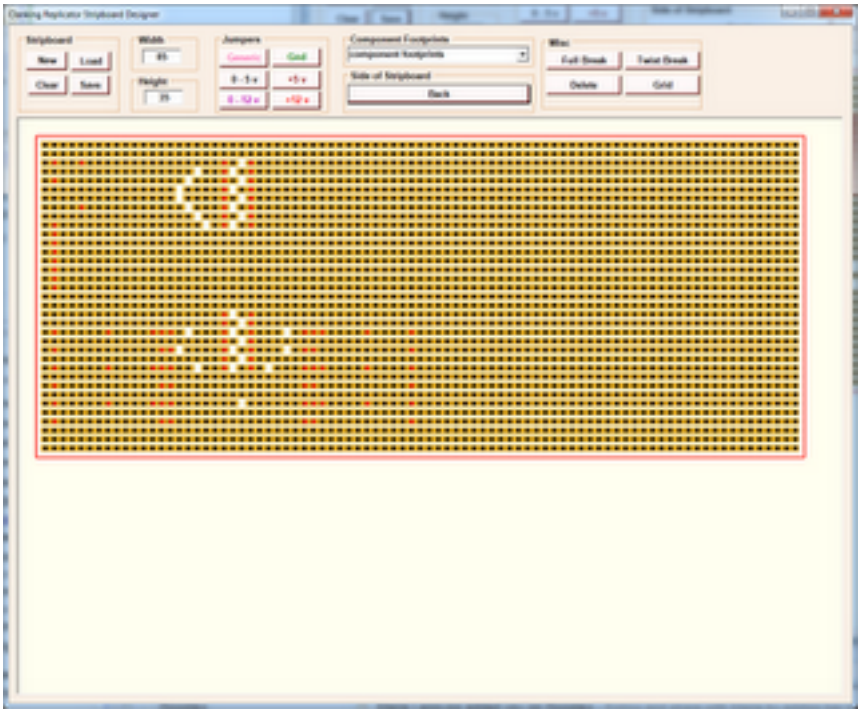
I bought components and dug out my stripboard and had a go at the design. Some time before I

put together a stripboard design program after having had no luck with the ones I was able to access on the web. Eventually, I evolved this board.

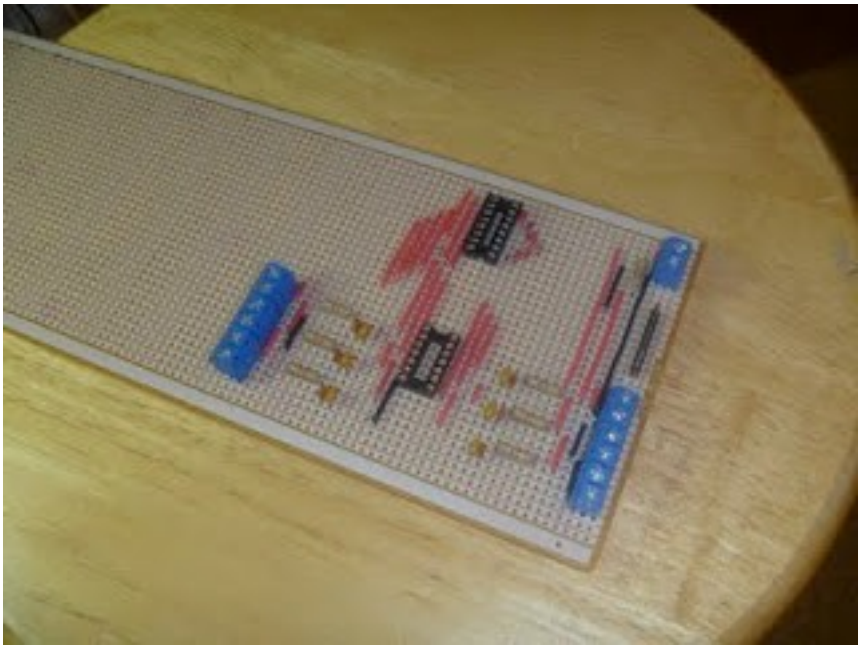
Frontside...



Backside...



It has been some time since I built a board, so I found putting this one together quite frustrating. I was about to give up this evening after making a bunch of mistakes and then got angry to the point of rage. The adrenalin let me get the @#$#@$ thing finished.



Tomorrow I will drill out the breaks in the strips, check the board for continuity and, if I have enough time, try to rig it into Sampo and extend the firmware to utilize it. I suspect that will have to wait till the weekend, however.

X & Y Axes operational from the controller

Wednesday, 27th July 2011 by Forrest Higgs

I was finally able to get time to integrate the full anti-bounce board with the x and y axis limit switches.

The two axes are playing ping pong and running at a speed of 65 mm/sec with no slippage and no heating of either the steppers or the driver chips. I've run them all morning with no mishaps.

Now I am going to have to see to writing a gcode interpreter and taking a shot at the TFT 320x240 graphics touch screen for system control.

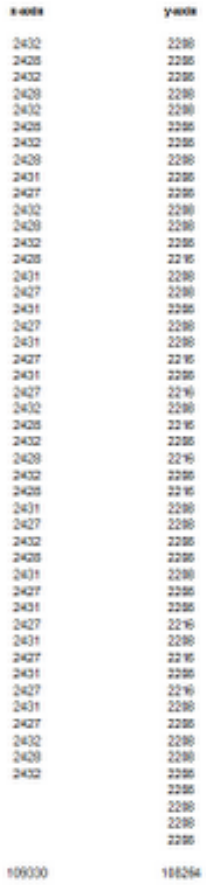
Bogdan makes a measurement suggestion...

Wednesday, 27th July 2011 by Forrest Higgs

After I got the x and y axes operating independently, Bogdan suggested that I measure the steps between the limits switches to see how much difference in measurements might be attributable to the mechanical microswitches that I am using. I had already been recording that with the y axis, so I decided to extend the monitoring code a bit.

With the y-axis, I had been simply writing the number of pulses to an SD card whenever a limit switch was tripped. In that I was running the axes at 65 mm/sec, I was getting a rather substantial thump whenever a switch was encountered. When I thought about it, I began to suspect that the impact was a result of the time it was taking the write to the SD card to happen in that I was doing that immediately after the switch was tripped. I changed the code to record a set number of triggering events for each axis and then exit the stepper loop and print the whole set of measurements at one time. That reduced the noise of switch triggering on the x axis to almost nothing. It also reduced the noise from triggering events on the y axis, but not as much. Considering the y-axis is shifting the whole weight of the x axis assembly, the extra momentum generated thereby is probably causing the larger thump.

I first took a set of 50 triggering events running at 65 mm/sec.



x-axis	y-axis
2432	2286
2428	2286
2432	2286
2429	2286
2432	2286
2428	2286
2432	2286
2429	2286
2431	2286
2427	2286
2432	2286
2429	2286
2432	2286
2428	2216
2431	2286
2427	2286
2431	2286
2427	2286
2431	2286
2427	2216
2431	2286
2427	2216
2432	2286
2428	2216
2432	2286
2429	2216
2432	2286
2428	2216
2431	2286
2427	2286
2432	2286
2428	2286
2431	2286
2427	2286
2431	2286
2427	2216
2431	2286
2427	2216
2431	2286
2427	2216
2431	2286
2427	2286
2432	2286
2429	2286
2432	2286
	2286
	2286
	2286
	2286
109330	188264

You can see that the two limits switches on serving the x axis trigger with slightly different sensitivities, one triggering about 0.3 mm greater than the other {transition is running at 0.89 mm/step for both axes}. The both y axis limits switches trigger at the same place except that occasionally one gets moody and triggers 8 steps {~0.6-0.7 mm} longer than the first.

I then took another set of measurements at 32.5 mm/sec. The NEMA 23s were near resonance frequencies at this speed raising the noise level of the printer considerably. I will have to see about damping this.

x-axis	y-axis
2426	2194
2422	2194
2425	2194
2421	2194
2425	2195
2421	2195
2425	2194
2421	2194
2425	2195
2421	2195
2426	2195
2422	2195
2426	2195
2422	2195
2426	2194
2422	2194
2426	2194
2422	2194
2426	2194
2422	2194
2426	2194
2422	2194
2425	2195
2421	2195
2425	2194
2421	2194
2425	2194
2421	2194
2425	2194
2421	2194
2426	2195
2422	2195
2426	2194
2422	2194
2426	2194
2422	2194
2426	2195
2422	2195
2426	2194
2422	2194
2426	2194
2422	2194
2425	2194
2421	2194
2425	2194
	2194
	2195
	2195
	2194
109063	107522

What you can see is that the variation on the x-axis stayed about the same while the moodiness of the one limit switch on the y axis disappeared. Notice also that the steps between switches are down.

From there, I took a set of measurements at my usual printing speed for Rapman at 22 mm/sec.

xsum	ysum
2423	2435
2419	2436
2423	2435
2419	2436
2423	2436
2419	2436
2423	2436
2419	2436
2421	2436
2419	2436
2423	2436
2419	2436
2421	2435
2419	2435
2421	2436
2419	2436
2421	2437
2419	2437
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
2419	2436
2421	2436
118591	119357

Decreasing the transition velocity got us further away from the resonance frequencies of the NEMA 23s. It must be said, however, that the printer was still louder than when I was running it at 65 mm/sec. You can notice here that the variation in limit switch triggering has dropped to 1-2 steps.

This has been an interesting exercise. One thing that is obvious now is that to control noise levels I should be controlling the stepper speed both by the delays between steps and by adjusting the level of microstepping that I am using.

Sampo's touch screen begins to work...

Tuesday, 16th August 2011 by Forrest Higgs

Adriaan has been working hard learning the TFT programming protocols and has his first touch screen menu working on the Sampo controller board.

A new acquisition...

Friday, 9th December 2011 by Forrest Higgs

Details

Solving a nagging question about print adhension

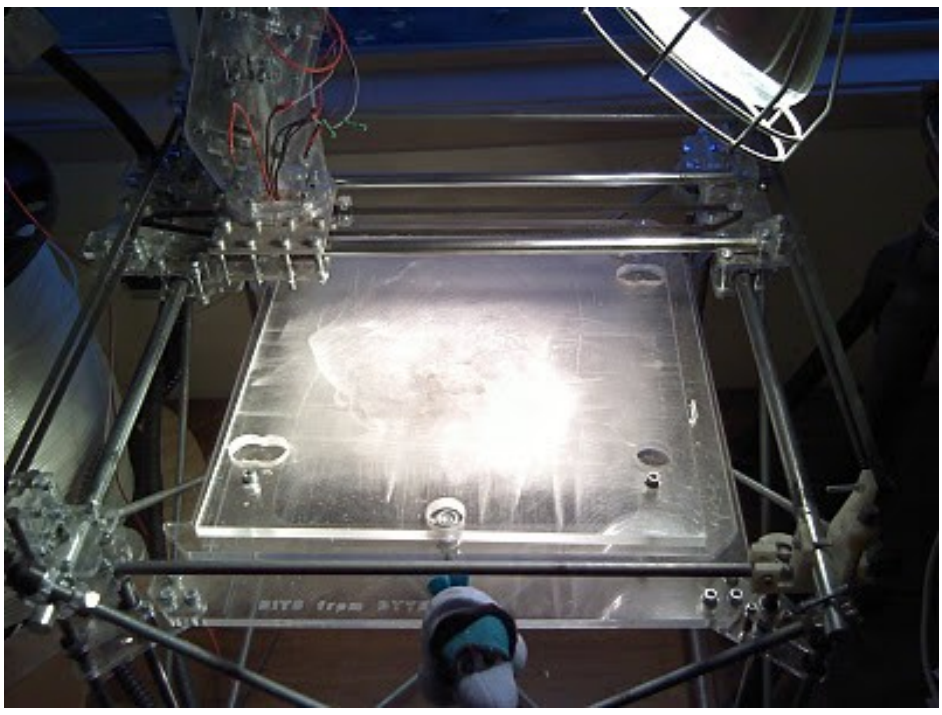
Friday, 16th December 2011 by Forrest Higgs

Unlike most of you, I don't use an electrically heated print surface. Some time ago I bought a Rapman 3.1, which used an acrylic 3 mm print table. I soon discovered that 3 mm was far too thin and quickly warped beyond use. Switching to 10 mm solved that problem.

After a long time of successful prints, I noticed that with winter causing colder temperatures in the print room I was having more and more trouble getting my prints to stick to the acrylic. I tried cleaning it and sanding it with little avail. Electrically heated print tables were just coming available but insofar as printing was concerned, I thought that things were already complicated enough without adding that sort of equipment to my Rapman.

I had an IR heat lamp in the lab, detritus of another experiment, and discovered that using it on a tripod to raise the temperature of the acrylic print table above 40 degrees Celsius measured with a non-contact IR thermometer gave me consistent adhesion. I soon discovered that I could turn off the IR lamp after 4-5 print layers with no ill effects. It was not needed for the rest of the print.

The rig looked a bit like this...



Note that the lamp is placed at a 45 degree angle to the acrylic print table.

I soon noticed that adhesion at the near side of the print table was much less firm than that at the back and less firm at the left side than the right. I attributed this to various things, uneven heating being one of the possibilities. While the left/right difference made sense the front/back difference didn't seeing as the IR lamp was aligned with the left/right axis.

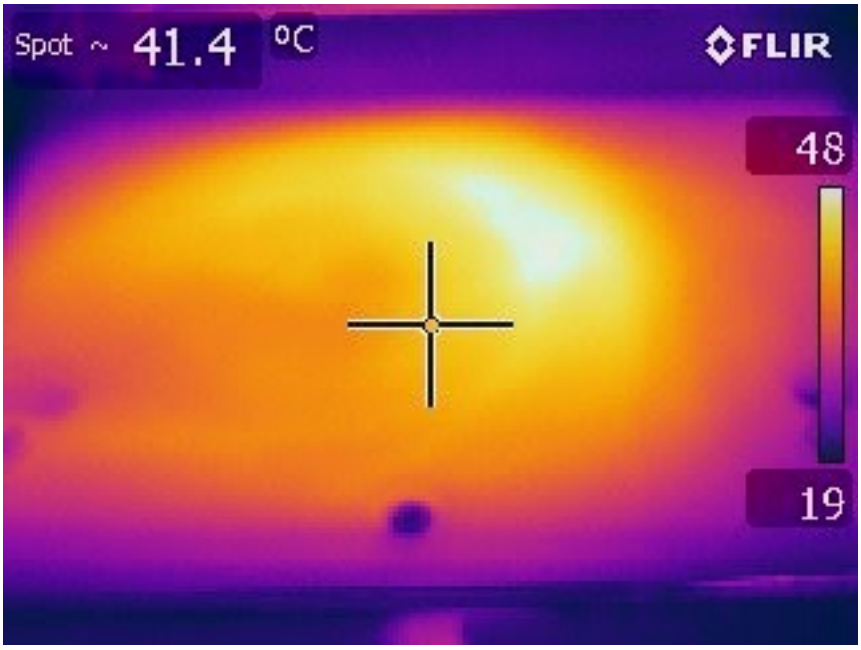
Cranking the terminal heating temperature before starting a print to about 50 degrees solved most of the problem for the center of the table and I was able to print along the front/back axis with reliable success. Unfortunately, the back side of the print area seemed to have the print pad melting into the acrylic while the front side would separate easily.

It made no sense. I thought for a while that it had something to do with the acrylic plate and rotated it with no effect. Swapping ends and sides always left the back side of the print table very firmly attached to the print pad. While that wasn't a horrible situation it was annoying, because it meant that processing the printed objects after separation became more time consuming.

A few weeks ago, I purchased a FLIR E30 thermal imaging camera with the intention of learning more about what was happening with prints as they were being laid down, the ultimate goal being building in advanced heuristics into my Slice and Dice app which converts STL files into Gcode. I also had hopes about eventually doing some research into what actually happens thermally with

extruder hot ends with the notion that I might be able to design a better one.

Yesterday, the E30 arrived and I decided that a good beginning exercise might be to look at the distribution of heat on my acrylic print table when I used the IR lamp in its standard configuration to heat it. The results were quite unexpected.



>

The lamp put down a marked hot spot at the upper right rather than at the right as I expected. The upper right was exactly where I had the most trouble with print pad melting. Obviously, the IR lamp did not give even heating when tilted but overheated in on the upper right.

This was nasty. I had previously thought about using several smaller IR lamps at the corners of the Sampo printer that I have been developing. If the smaller lamps behaved like my single, large one, however, this might not be a good idea at all.

I then got to thinking about how IR lamps are actually used in food heating cabinets. They are almost always placed point straight down. I rearranged my tripod to place the lamp almost vertically over the acrylic print table.

That sorted out the temperature distribution problem...

FLIR
58
15
Spot ~ 31.9 °C