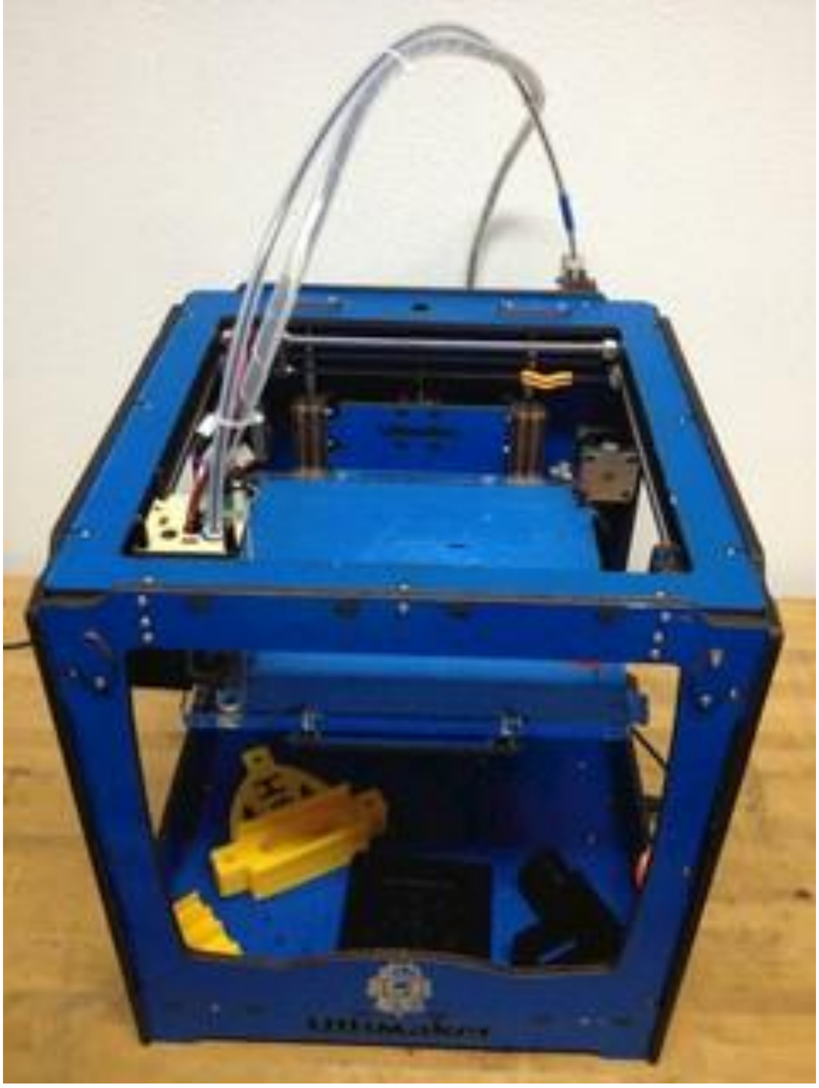
## Introduction

For quite some time (over a year or so) I've been thinking about building a 3d printer or adding a module to the cnc machine I build in the past. At work we own an Ultimaker and 3d printing has some benefits over turning and milling as the approach is more like the 'other way around' as you are not taking material from your stock but actually building the objects layer by layer.

This surely is a different approach and less messy than working on aluminium with cooling and so on.

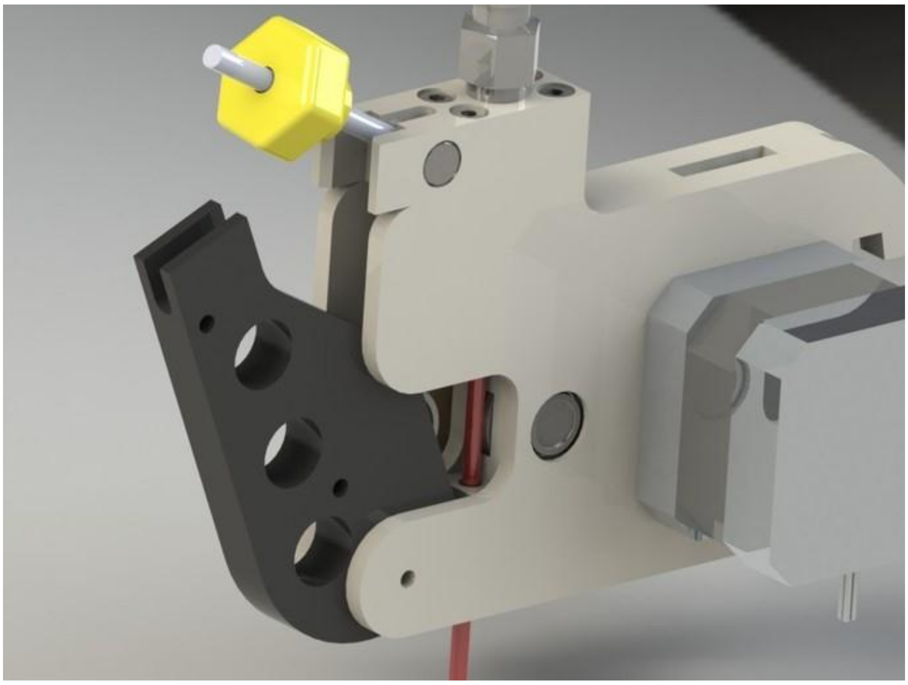
Working with the Ultimaker we learned that the filament feeder (or extruder drive) and hot end are critical parts. We had some issues (slipping of the filament) with the original versions of these and did an upgrade to the latest versions as provided by Ultimaker. You can find them here: https://shop.ultimaker.com/en/parts-and-upgrades.html?p=1

After these upgrades this Ultimaker is now running like a train. Because of the upgrades there were old Ultimaker parts laying around on the desks. This finally made me decide to start working on a 3d printer module for my current cnc machine.



**Extruder Drive**

As we found the spring design (to compensate for diameter differences in PLA) worked quite well I searched on thingiverse for some designs with a spring system. I found one I liked from Geo Hagen. It has a bearing based spring pressure system and some improved metal parts. Below is an image of Geo Hagens design and it can be found here: http://www.thingiverse.com/thing:15897

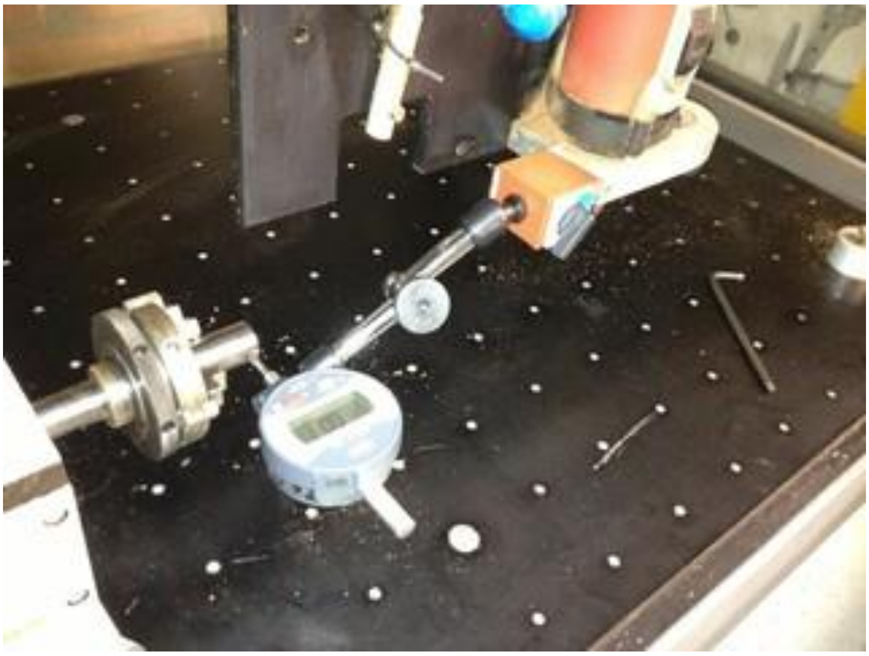


I was lucky to be able to use a 3d printer (Ultimaker) to create the non-metal parts:

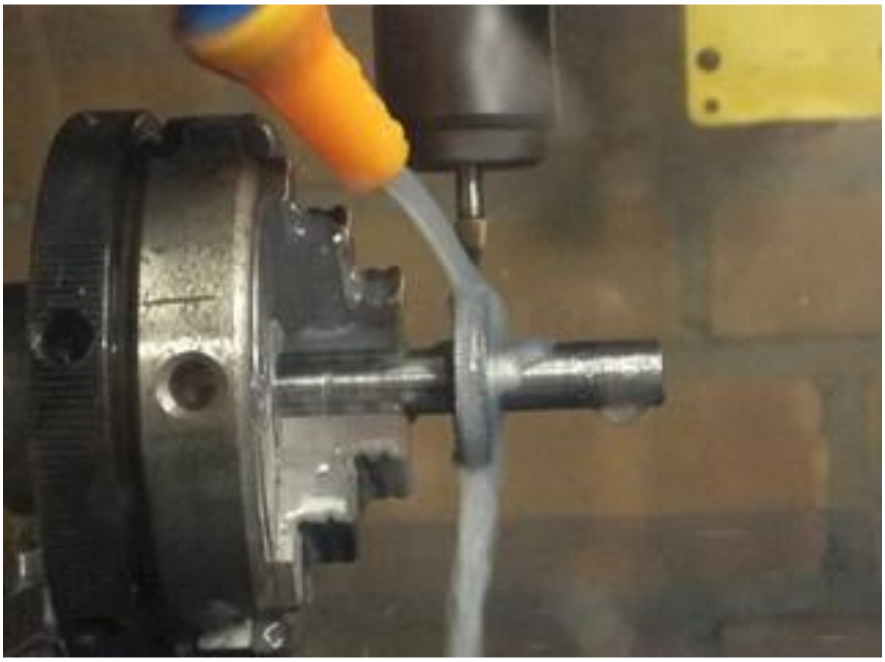
I also used some other parts found on thingiverse for the gear wheels and tube retention. The gear wheels can be found here: http://www.thingiverse.com/thing:25914 and here: http://www.thingiverse.com/thing:43055 . The tube retention part is posted here: http://www.thingiverse.com/thing:26656.

The metal parts in this design were a different cup of tea. I used an Emco 8 turning mill and the cnc router to build these.

This old guy I was used for turning the extruder driver wheel out of silversteel. Notice I just turned the outline. The small groves were made on the cnc router by using the 4th axis. I found the groves just needed to be one tenth of a mm. See the cnc setup with the 4th axis below.



And here you can see the spindle cutting the splines on the part. After this the driver wheel part was put back in the mill to create the large grove in the middle of about 1 mm depth.



It ended up that I had to make two of these drive wheels as the first time I tried to make the small groves to deep (4 tenth of a mm) and had the grove in the middle done first.

The deeper cuts caused some trouble on the router and give a not very nice result. So I have a spare driver wheel. The second time things worked out perfect as can be seen by the result:

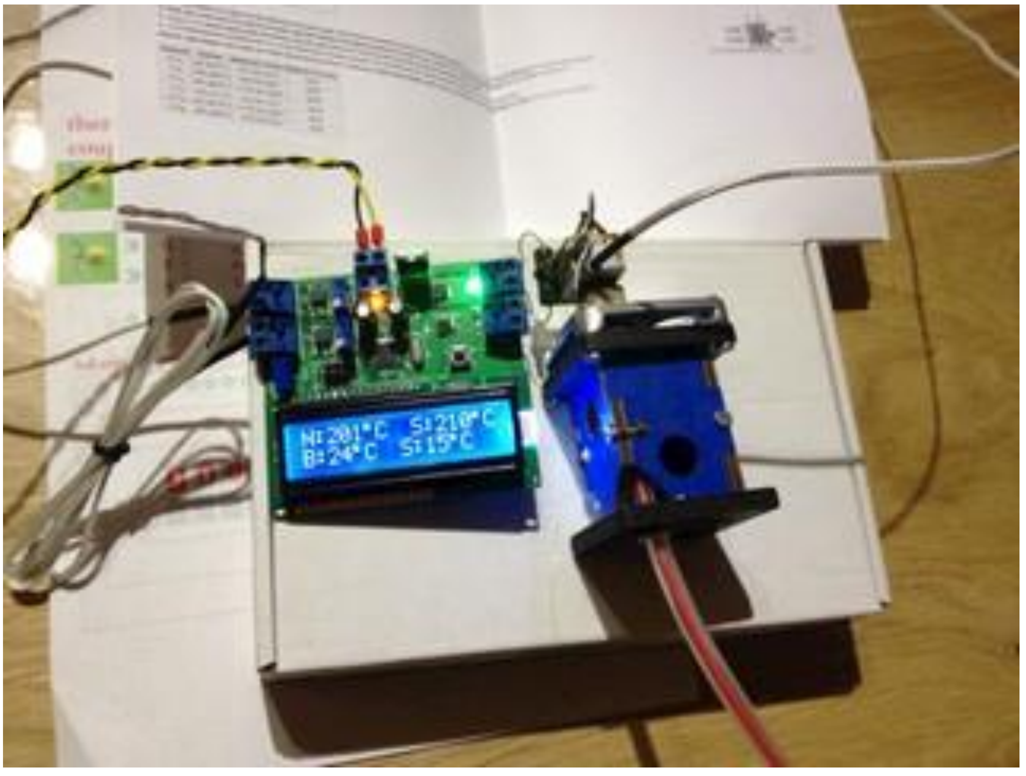
The parts were put together and finally the motor (yes the same one I used for the 4th axis on the router) was mounted. This part looks promising.

**Gnexlab Temperature controller**

First the plan was to create a temperature controller myself from scratch by using Microchip tools and experience from the past. But then on a day I surfed the internet and found the Gnexlab temperature contollers created by Nuri Erginer. This was exactly what I had in mind!

His approach, fair pricing and finding that my Microchip programmer and skills needed an update/refresh convinced me quickly to order one of the controllers and parts.  I went for the Gecs2tc1 as I find the thermocoupler control more for me. Info about the controllers can be found here: [http://cnc2printer3d.wordpress.com](http://cnc2printer3d.wordpress.com) and you can order them on the webshop: [http://store.gnexlab.com](http://store.gnexlab.com)

Some time after ordering I received the controller board and sensors. I did some short tests to check the board and heating of the nozzle. This looked good. The resistor was just hold to the aluminium block with a wire and by doing so I got a maximum of 200 degrees. So the resistor needed to go into the aluminium block.




Next started to work on the mounting all parts on a 3D printing module that can be put my machine.

**Building the 3D print module**

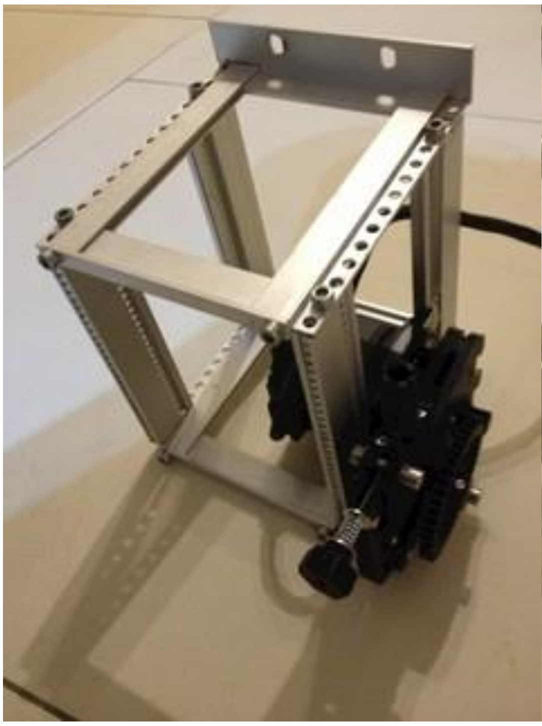
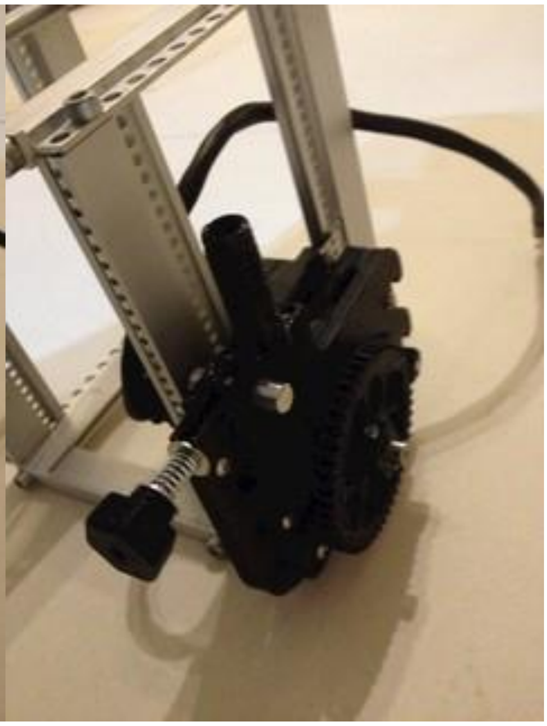
At this time several parts were build and available: the extruder driver, heating controller electronics and the parts (hot end) from the Ultimaker. The goal is to create an 'Add-on' module for the current machine that will 'convert a CNC machine to a 3D printer' as nicely stated on the Gnexlab website. The module needs to be easily 'added on to' or 'taken from' the machine.

The reason for this is simple: The machine is covered with screens and when milling or routing metal parts the complete machine can get dirty and (metal) debris and cooling liquid remains can be everywhere. Clearly this is not an environment where you want to have your delicate 3D printing head, electronics and parts in!

I went to my dad's barn (in our family we call this the general store…) and found a larger euro-card print frame available. I removed the electronic cards and adapted it to the size I needed. This is a good start for a module:

Some small aluminium parts are made or adapted to fixate the extruder driver and controller pcb.

Next I mounted the extruder drive unit (wit motor) in the frame. As my motor is of a larger type and by far the heaviest part I put it inside the rack so it is resting on the aluminium frame. The motor is more to the size of Nema 23 (and not Nema 17) therefore I also turned an aluminium adapter shaft for the small stepper gear. For the mounting of the extruder drive I only fixated the motor as the rest will be hold by the motor housing:

I also fixated the Gnexlab controller pcb to the frame construction. I used plastic spacers to isolate the pcb from the frame:

**Hot end**

As mentioned I had some Ultimaker hot end parts that I planned on using. This included the 0.4mm nozzle, some of the pieces from wood and a heating block (where the thermocouple and heating element are put in). I had to make or buy the aluminium plate for mounting the extruder hot end.

I first drew this mounting plate and then I created it on the cnc router. This didn’t went all that well but after some manual old school metal work (filing and drilling etc.) I had a usable mounting plate. I first drilled the four holes for the M3 thread and then used a 6mm end mill to route it and can provide the drawings/gcodes on request.

So after I made the mounting plate the above (with some wooden parts) was the starting point for the hot end.

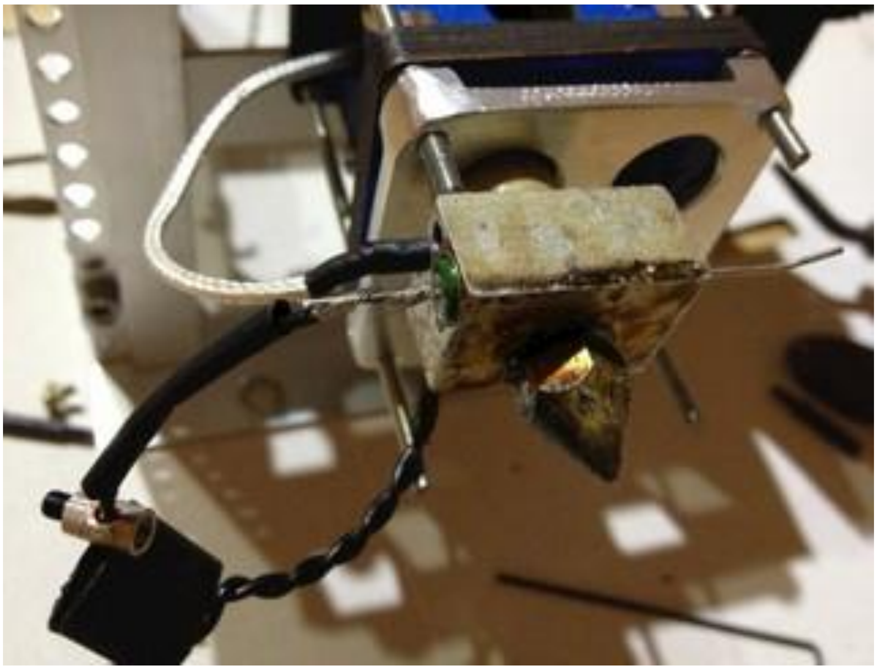
**Heating element**

As heating element I used one of the resistors as provided from Gneslab. It is a 6R8 Ohm 7W 5% resistor. In order to get it in the heating block I drilled the hole larger until it fitted. If I remember it correct I drilled up to 6.4 or 6.5 mm before it fitted.

At first I used solder to fixate the wiring to the resistor. I doubted if this would do the job as the hot end will be set to about 210-240 degrees Celsius. And the solder will start to melt around 200 degrees Celsius.

As I expected the solder did not hold above 200 degrees. But now with the resistor in the heating block the heating is sufficient and up to 210 - 230 degrees in about 4,5 to 5 min.

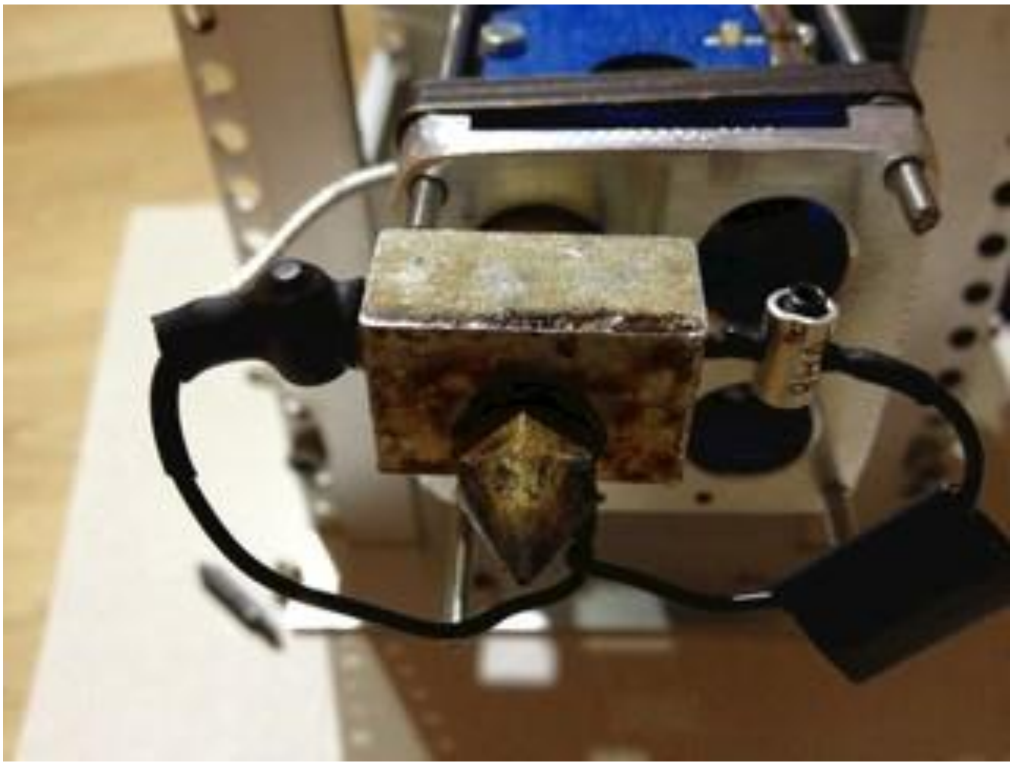
After this check on the heating time I started thinking on how to create the connection between the heating resistor and the wiring to the electronics. I have giving this quite some thought and ended up with a screw-based kind of connecting as I was not intending on welding it with TIG or what so ever. The next pictures show what I did.



I found that the screw terminals that can be found on DIN-Rail modules contained some parts that I could use for this. But you can also use a small metal tube and create some thread in it. If you next drill a small hole completely trough it should do the job.

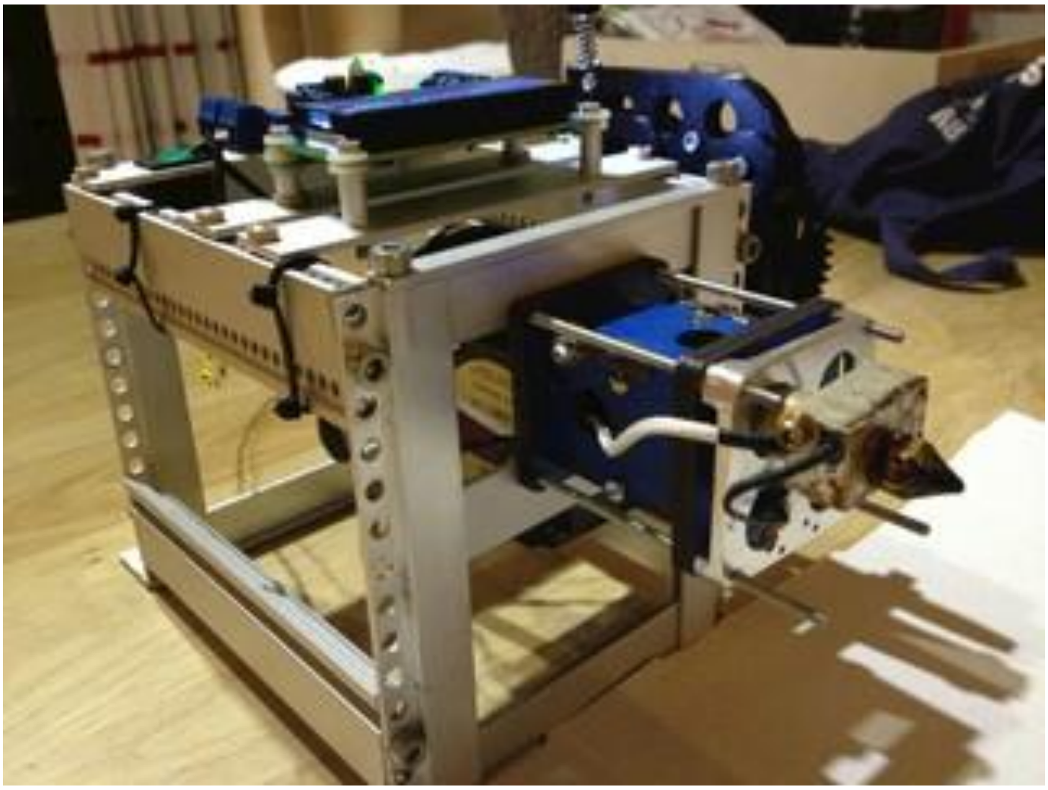
I used two of these clamps, four pieces of electronic crimping insulation of two sizes to make the connection.  First I twisted the stripped wiring around the resistor leads as can be seen in the picture above. Next the smallest insulation was crimped around this. Over this connection the small metal clamp was put and the screw was tightened. This connection looked pretty good and strong.

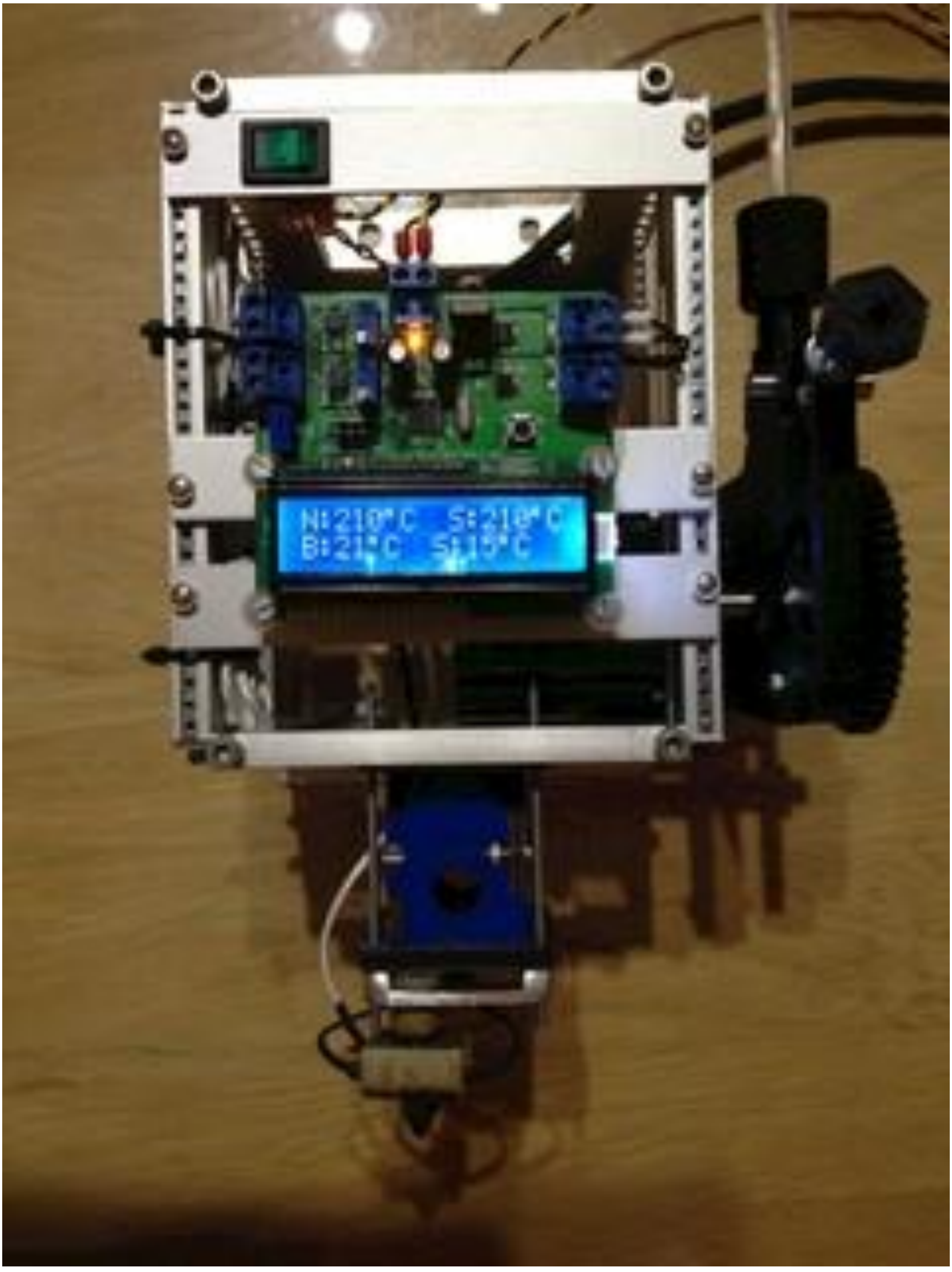
Next the larger crimping insulation was put over the whole connection and the same procedure was followed to do the other side of the resistor.

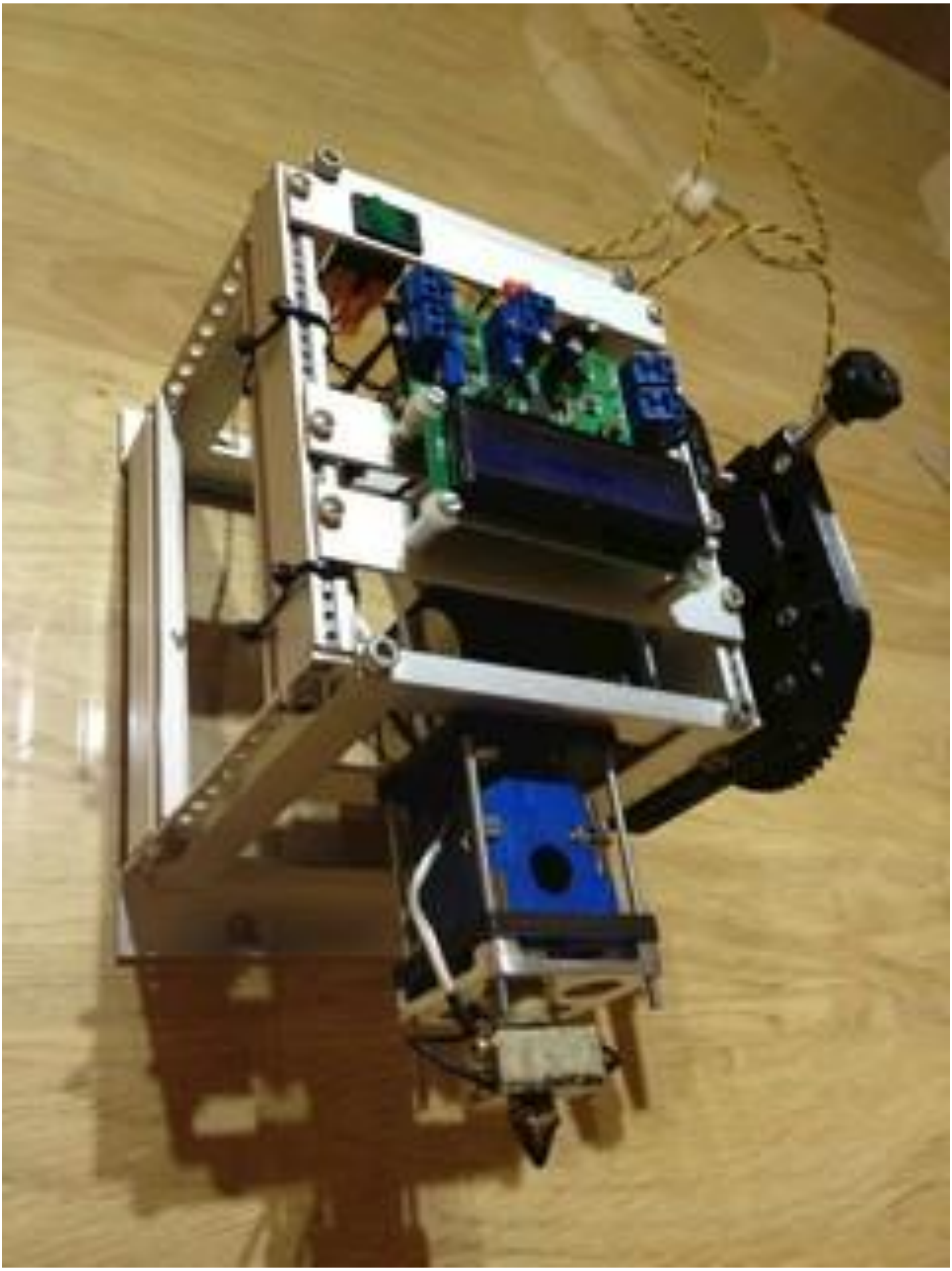


After this I ended up wit a solid connection between the wiring and resistor as shown above. Next is the mounting of the hot end to the module.

Finally I also mounted the hot end on the 3d printing module and added an on/off switch. It is starting to look like something now:

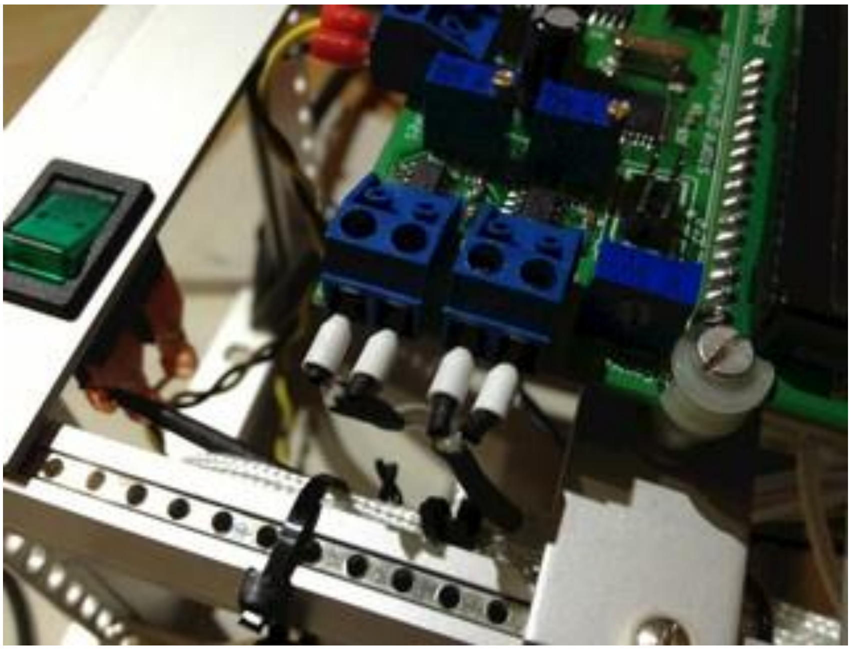



N:210°C S:210°C
B:21°C S:15°C



**Thermocouples**

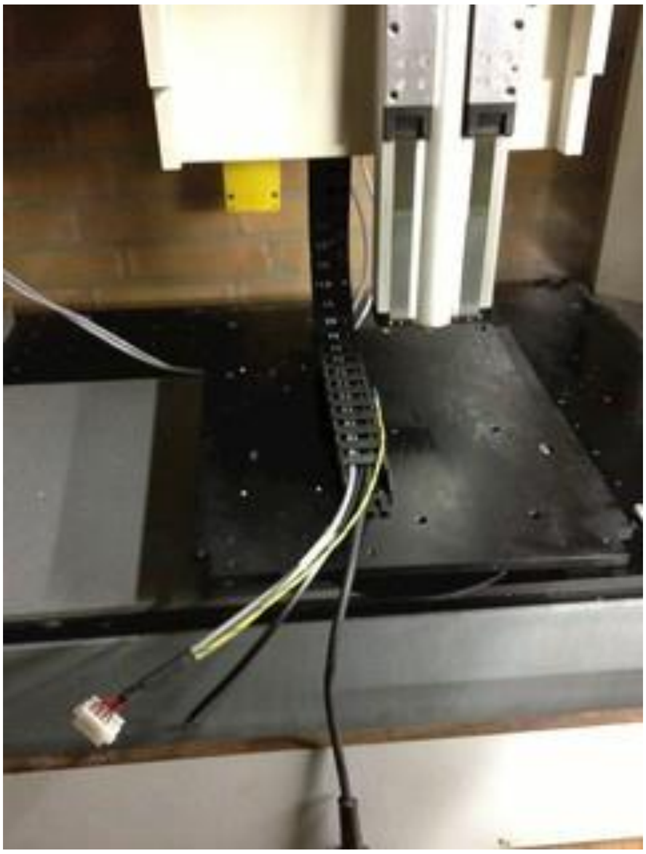
Used the thermocouples from the Gnexlab store. First I simply put them in the hole in the heating block but I found that this might not be the best option to mount them. I isolated the wirings and used wire-ends to get the screw terminals:



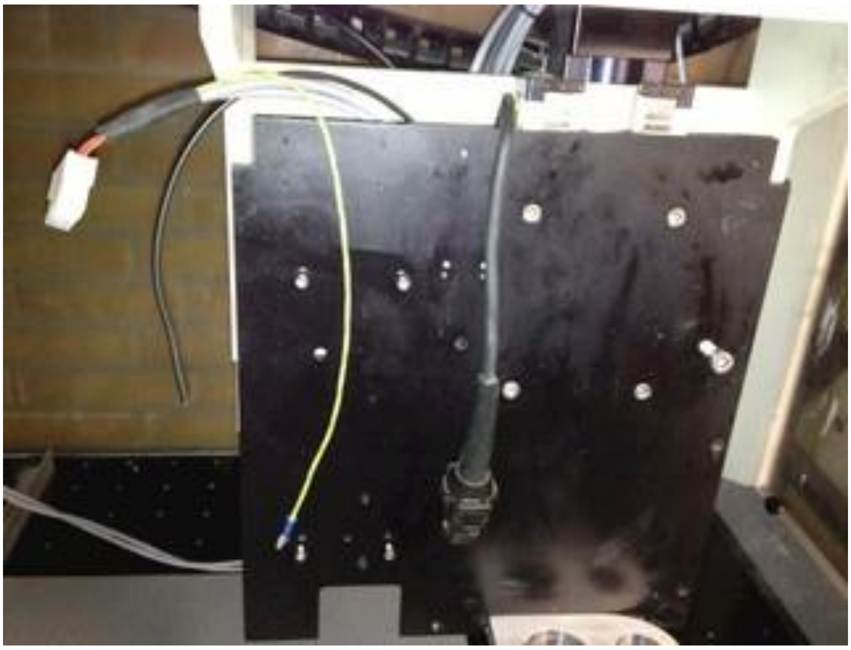
At current I mounted the thermocouple with a screw:

**Mounting the module to the machine**

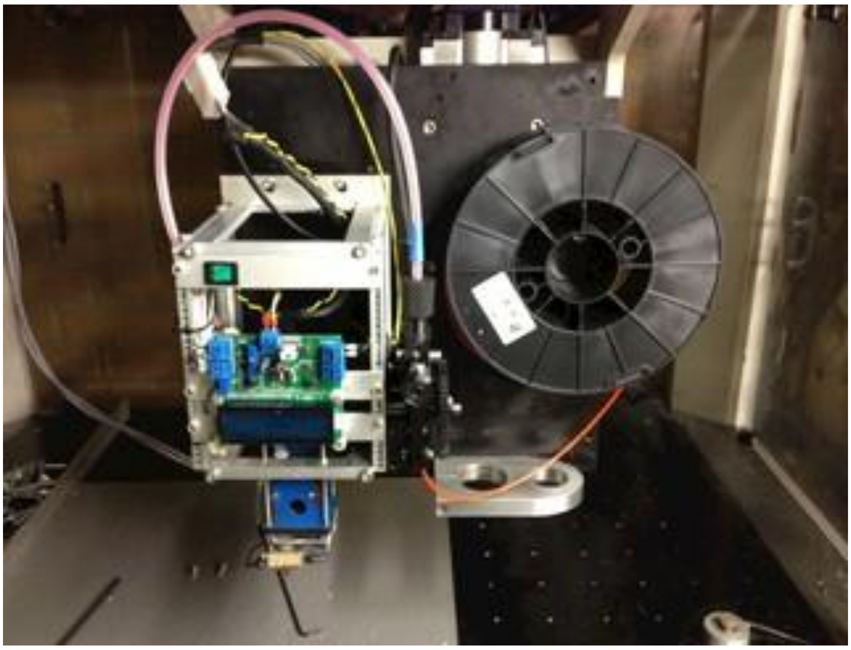
Working on the routing of the cabling I remembered taking this cable out a couple of years ago. The motor I am using was original mounted near the tool of the machine. So this came in practical when re-routing the cabling. All glands etc. where already available so it was just a matter of feeding the cabling through (1 for the motor and 1 for the 12V power) and checking how the movement was.



And indeed the wiring ended up just fine again.



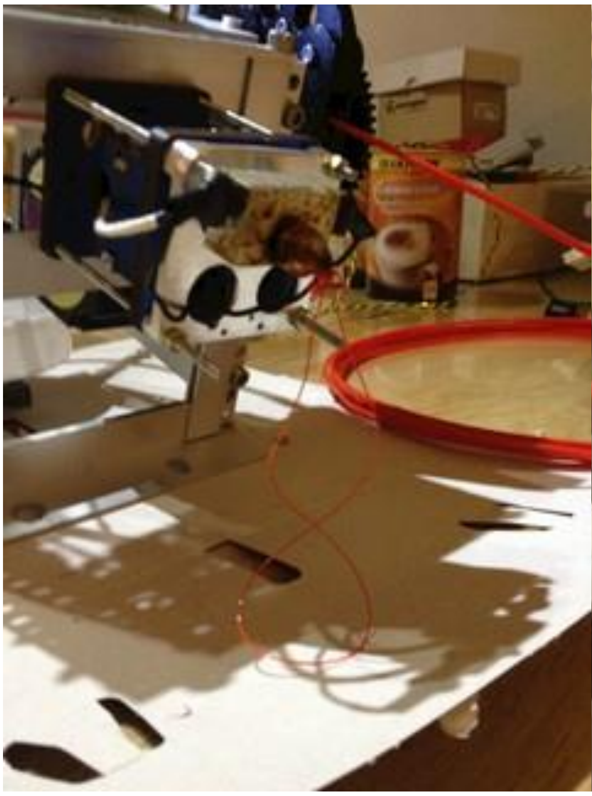
About the mounting: The 3d printing module is just bolted on the mounting plate with four M4 bolts. So this is nothing special or difficult.



So now the module is bolted onto the machine. I am not sure about the position for the reel. I guess we will find this out over time. I just put it on here for fun and to show it on the picture.

**Filament Melting Tests**

After mounting I did some filament melting tests. Some early tests are also given here. The first one is by moving the large gear wheel manual and this second one when the module was bolted onto the machine. Here the motor was turning and pushing the filament through the nozzle. This looked promising.

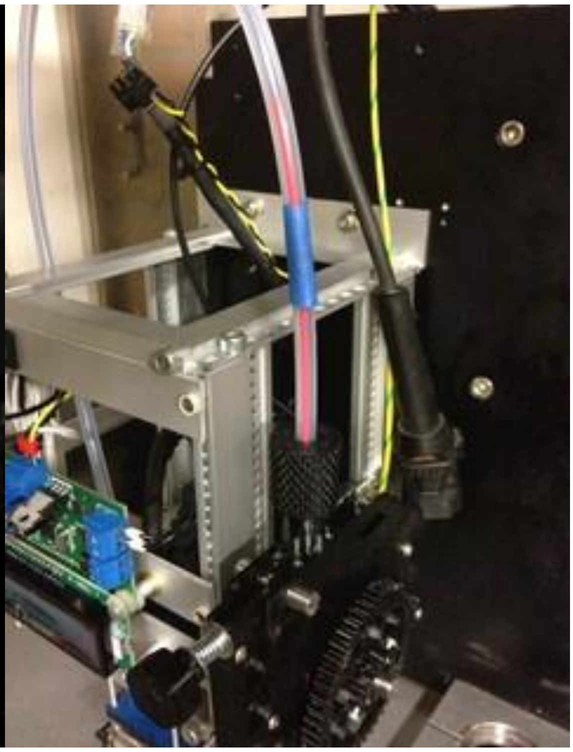



**Extrusion motor calibration check**

As I now think of it I actually checked the calibration of the extrusion motor before I did the melting test with the module mounted. In linuxcnc you can set a SCALE for each axis in the .ini file. I will come to the calculation of the scale a bit later. Anyway for calibration the check I did was as follows:

1. Move the filament out of the extruder, level with the outside of the bolt, in the Teflon hose. This is shown on the first picture.
2. This level I set to the zero of the A-axis. By using the touch off button in linuxcnc (on the first tab of graphical interface control)
3. I made sure the speed was not to high for the A axis movement.
4. Went to the second tab in the GUI. And here I typed G0 A100 to move the motor to position 100mm (= 10 cm). This caused the (red) filament to move up in the Teflon hose (or bend tube).
5. I measured the distance and in my case it was 100.005 mm so I left it. If the distance differs from 100 mm you need to calculate the adjustment factor and adapt your scale in the .ini file and try again. You can adapt the scale with a calculation as:

*NEW_SCALE = desired Distance / moved distance * SCALE*

Ok so this seems fair but what about the original start scale? For this I calculated the following. I had the next known parameters:

| | | |
|---|---|---|
| Microsteps: | 1 | as I setup this motor without them. |
| Motor steps per revolution (=360 degrees) : | 1000 | for my 110V ac stepper motor |
| Motor Gear teeth: | 8 | |
| Extruder wheel Gear teeth: | 49 | |
| Diameter of the filament driver wheel/bolt | 20 mm | This is the metal part with the filament-grip |

So I figured every complete turn of the larger wheel the filament bolt will turn, causing a movement of 2∏r where r is 10 mm. So each turn of the large wheel will give me 62.83 mm movement. I assumed there will be no slipping of the filament and the movement relation for this is 1:1.

The ratio between the large gear and motor gear is 49/8 = 6.125 meaning the small gear and thus the motor needs to turn 6.125 times for the large wheel to make a complete turn. And this is in my case (1000 pulses per turn =) 6125 steps.

Therefore my setup needs 6125 steps per 62.83 mm. As I wanted the scale to be setup in mm I found that the system needs 6125 steps/62.83 mm = 97.48 steps per mm. And thus I set the SCALE to this value in the systems .ini file.

SCALE = 97.48

As I found during the calibration this calculation is just right and usable for first setup. Here is the part of my linuxcnc .ini file:

```
[AXIS_3]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 360.0
MAX_ACCELERATION = 360.0
STEPGEN_MAXACCEL = 450
SCALE = 97.48
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -9999.0
MAX_LIMIT = 9999.0
HOME_OFFSET = 0.000000
HOME_SEARCH_VEL = 0
HOME_LATCH_VEL = 0
HOME_SEQUENCE = 3
```

**Software setup**

For controlling the machine I have been using linuxcnc for some years now and I am quite happy with it. The only disadvantage I see in my setup is that the computer is limiting the speed of the machine due to the motor pulses it has to send over the parallel port. But for the former aluminium and wood work this has never been an issue. Maybe this will be of importance in 3d printing. I guess I will find out over time. Of course I could install a Mesa card but this is not really the way for me at the moment. I am not going to give instructions on how to install linuxcnc as this is very well documented. See [www.linuxcnc.org](http://www.linuxcnc.org) for info.

Cura

Beside linuxcnc I am used with the Ultimaker Cura software and this can be installed on Mac, Linux and Windows computers. Therefore I tried the Cura installation for the linux machine as described here:

[http://wiki.ultimaker.com/Software_setup_guide#Linux_Dependencies](http://wiki.ultimaker.com/Software_setup_guide#Linux_Dependencies)

First I found that I needed a root access for this but that the default sudo command or su didn't work on this distribution due to a 'special' linuxcnc rights user. In this distribution the root level has no password and therefore commands like sudo, su –l cannot work. The first thing you need to do is set the password for the root level. This can be done with the commands:

> **$ sudo passwd root**
> When prompted for a password, give your user password
> When prompted for the root password, give something suitable
> Now enter
> **$ sudo passwd -u root**
> You will get a reply like 'password expiry info updated' or similar
> Now enter
> **$ su root**
> Enter the root password when prompted
> You should now have a root prompt **'#'** and you are ready to go.

If you like you can read more about this here:

[http://www.linuxcnc.org/index.php/english/forum/9-installing-linuxcnc/21331-root-password](http://www.linuxcnc.org/index.php/english/forum/9-installing-linuxcnc/21331-root-password)

Okay so from the ultimaker wiki I basically checked the provided commands:

```
sudo apt-get install python-setuptools python-numpy python-scipy
python-wxgtk2.8 python-wxtools wx2.8-i18n libwxgtk2.8-dev libgtk2.0-
dev sudo apt-get install python-setuptools python-numpy python-scipy
python-wxgtk2.8 python-wxtools wx2.8-i18n libwxgtk2.8-dev libgtk2.0-
dev.
```

I installed all the packages here by using the package manager in the Ubuntu distribution as this gives me a better feeling about the package compatibilities. Anyway that is just a personal feeling. There is however a comment on this page that you need the

```
python-profiler
```

package and this is correct without it I got messages and could not start Cura. So you better not forget this one. Next the routine tells you to install a package named pip. I found a bit trouble with this as I installed the pip package from the package manager but it didn't work. This one did not know the 'install' command. I quickly realised that Cura is very python based and tried the python-pip package. This is the correct one. So make sure to install python-pip and not the regular pip.

The next command the installation routine tells you to do is (with the new python-pip). Make sure you change to the root user (su root) or the PyOpenGL_accelerate will not succeed:

```
pip install PyOpenGL PyOpenGL_accelerate pyserial numpy
```

The wiki also tells you to get the following pip package:

https://github.com/GreatFruitOmsk/Power

I got it but didn't know what to do with it. I found some additional information on installing this here so check it if you like:

http://fablabamiens.org/index.php?title=Ultimakers

Cura does not seem to miss it anyhow. Finally you need Cura. I used version 12.12 and the linux download you can get it here:

http://software.ultimaker.com

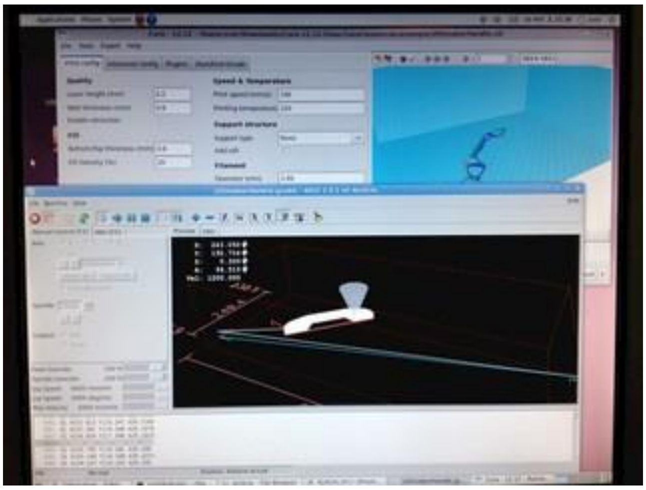
I unpacked it in my downloads folder with the command:

```
tar zxvf Cura-12.12-linux.tar.gz
```

and started it with:

```
cd Cura-12.12-linux
./Cura.sh tar
```

Now the very un-linux like thing was that it actually worked!! ☺


Quality
Fill
Speed & Temperature
Support structure
Filament

Openscad

As the installation of Cura went well I also tried installing Openscad as this is another package I like to use now and then. Info about Openscad can be found here:

http://www.openscad.org

There are also installation instructions on this site for linux Ubuntu distributions. It states:

```
$ sudo add-apt-repository ppa:chrysn/openscad
$ sudo apt-get update
$ sudo apt-get install openscad
```

If I remember it correct I followed the first two commands and then installed Openscad via the package manager. And also this package is running on the computer. Very cool ☺

Python script for opening Cura gcode files

As mentioned I am using a python script to open the gcode output files from Cura. Here I show the script and what it does. It is not high level coding but it does the job. Lets start with the script:

```
import wx
import sys

def get_path(wildcard):
        app = wx.App(None)
        style = wx.FD_OPEN | wx.FD_FILE_MUST_EXIST
        dialog = wx.FileDialog(None, 'Open', wildcard=wildcard, style=style)
        if dialog.ShowModal()==wx.ID_OK:
                path = dialog.GetPath()
        else:
                path = None
        dialog.Destroy()
        return path

def dellines(fname):
        f = open(fname,"r+")
        x = f.readlines()
        for line in x:
                if 'CURA_PROFILE_STRING' in line:
                        pass
                elif 'M1' in line:
                        print(";" + line),
                elif 'M9' in line:
                        print (";" + line),
                elif 'M8' in line:
                        print (";" + line),
                elif 'G28' in line:
                        print (";" + line),
                elif 'G92 X0 Y0 Z0' in line:
                        print (";" + line),
                else:
                        line = line.replace('E', 'A')
                        print (line),
        print "M2"

fname = sys.argv[1]
if fname == "":
        fname = get_path('*.gcode')
if not fname == None:
        dellines(fname)
else:
        print ("; no valid file found!!")
        print "M2"
```

The goal was to have the script do the following:

- Open a file dialog to search for the .gcode file created by Cura.
- Replace all E commands for the extrusion axis to the A axis.
- Remove or mark out all the gcode commands linuxcnc cannot handle.
- Add codes needed to be able to use the output in linuxcnc.

To open a file search dialog I found some examples and saw the wxtools package has the wx class library that I could use. Therefore it might be needed to install the python-wxtools package in order to use this script.

For the dialog I setup the function *get_path* that returns the filename or *'None'* in case not successful. Next I wrote the routine *dellines* to change the parts from the file so that it can be used in linuxcnc. I got the filename with a wildcard *.gcode as this is the file extension used by the Cura output.

The *dellines* routine loops through all the lines in the file and marks out the codes not usable for linuxcnc. It also adds a M2 command at the end. This is needed for linuxcnc in order to be able to open the file. It also removes a big line at the end that contains the Cura profile settings.

The script can also be started from the command-line with a filename as parameter. When not specified it will bring the file dialog up.

As in linuxcnc file extensions can be linked to specific scripts or programs I setup the .ini file to open all .gcode files with this python script. By doing so I gained two things:

1) The Cura output files can be opened directly from within linuxcnc.
2) No extra files are needed and I only need the .stl files and .gcode files.

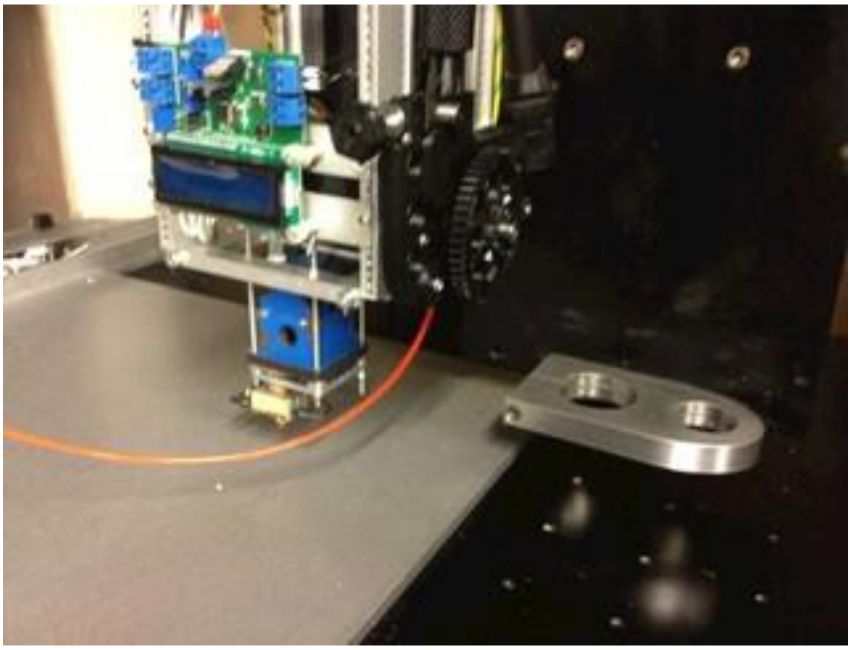
In order to do this I had to setup the file filter part in the .ini file to:

```
[FILTER]
PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image
PROGRAM_EXTENSION = .py Python Script
PROGRAM_EXTENSION = .gcode Cura Gcode
png = image-to-gcode
gif = image-to-gcode
jpg = image-to-gcode
py = python
gcode = python OpenCuraFile.py
```

Also I had to place the *OpenCuraFile.py* file in the */configs/mysystem/* folder from linuxcnc.

So by now I had a system which could control a 3d printer (linuxcnc), slice .stl models (Cura) and even create the models (Openscad). ☺

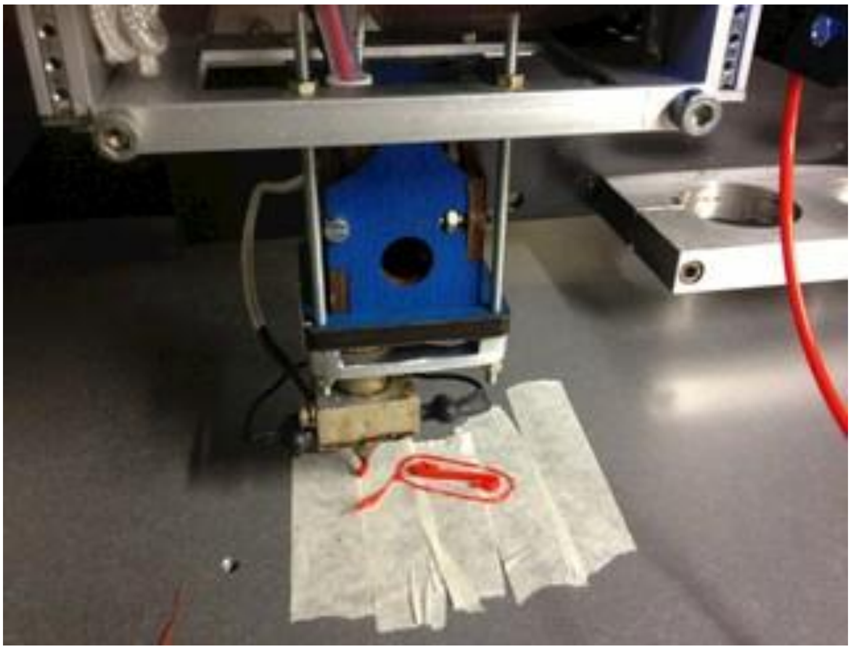
So about time to do some early dry testing:



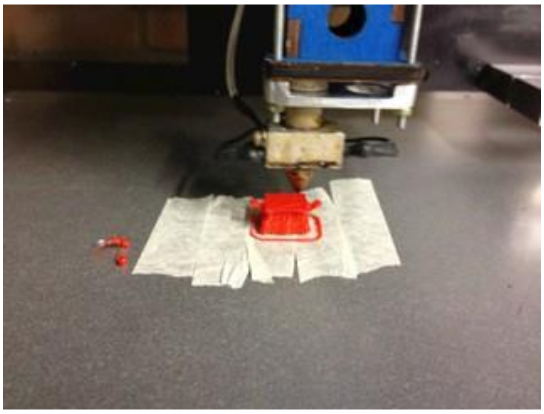
This did the job. At least the software moved all the axes correct and the z-heights looked fine. So the next thing would be some  real testprints!

**Start printing!**

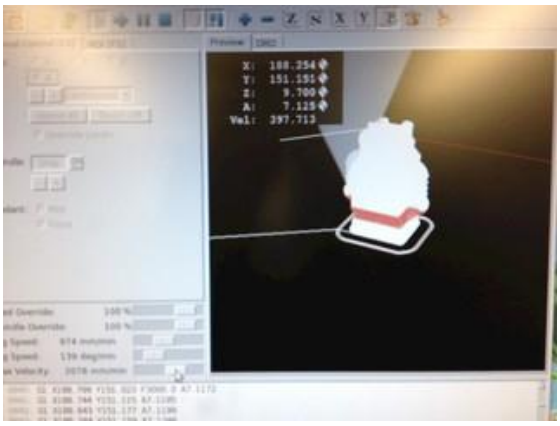
So here the first test prints. This went amazingly well:



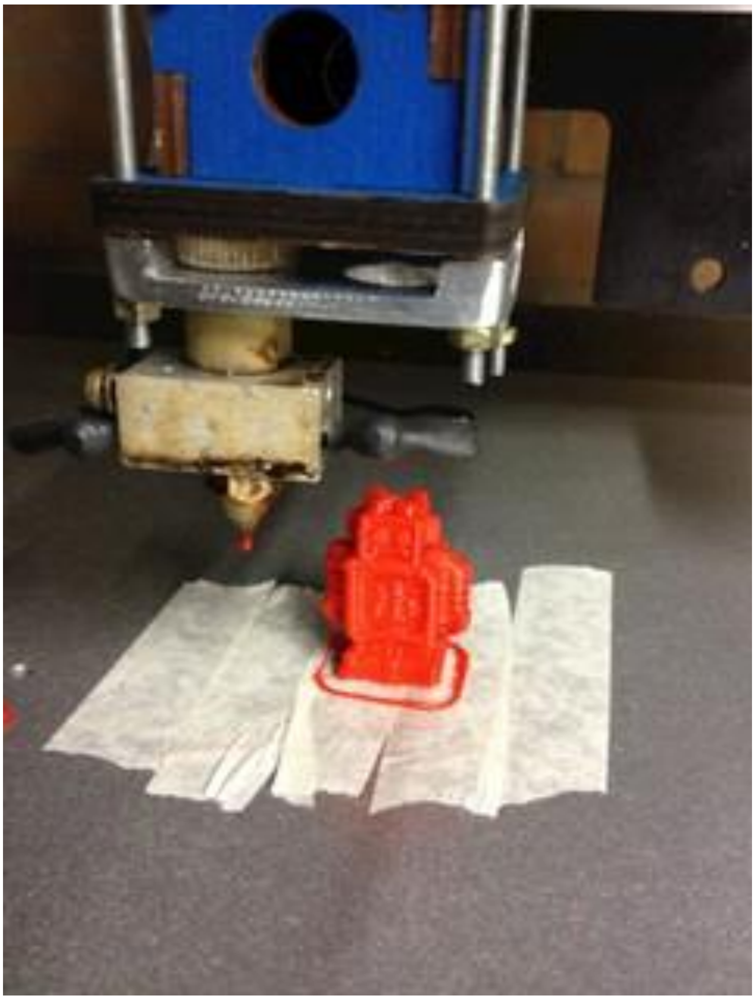
So I did  a larger print. I started the Robot-logo from the Cura examples. Okay the bed is not fixed yet and needs to be adjustable. Also I didn't have any captive tape but..



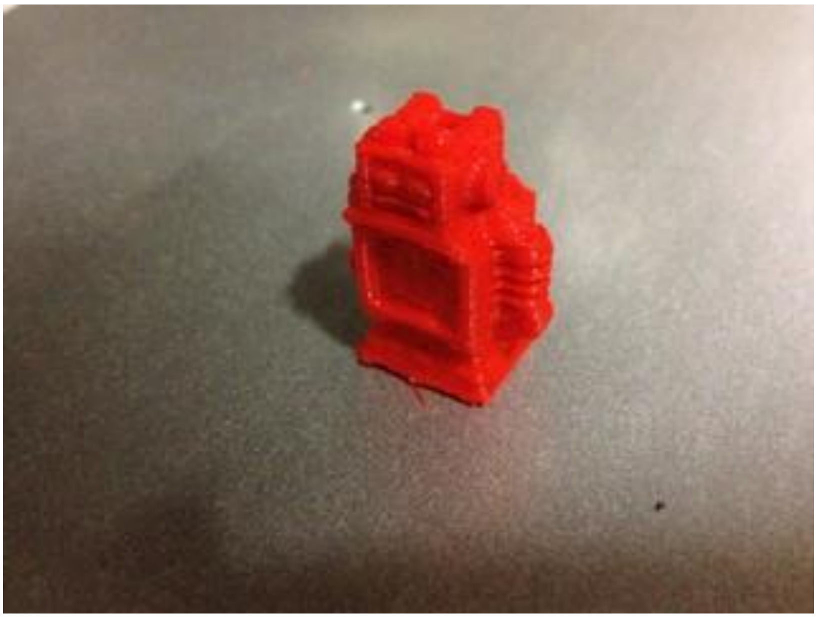
It printed!



Okay near to the head the little robot was pushed from the tape and I had to hold it for printing the little robot head but it finished the first robot..



Not too bad for the first tests…



**Things still need to do:**

- Optional: mount teflon hose from filament feeder to extrusion head. Only if problems occur.

- Mount print bed (non heated first); Build spring based mounting shoes. Idea is ready.

- Fine tune optimal machine speed with new computer

- Improve print quality /bug fixes