CNC Configuration Guide

3040 4-Axis CNC as used by

DAMOGRAN LABS

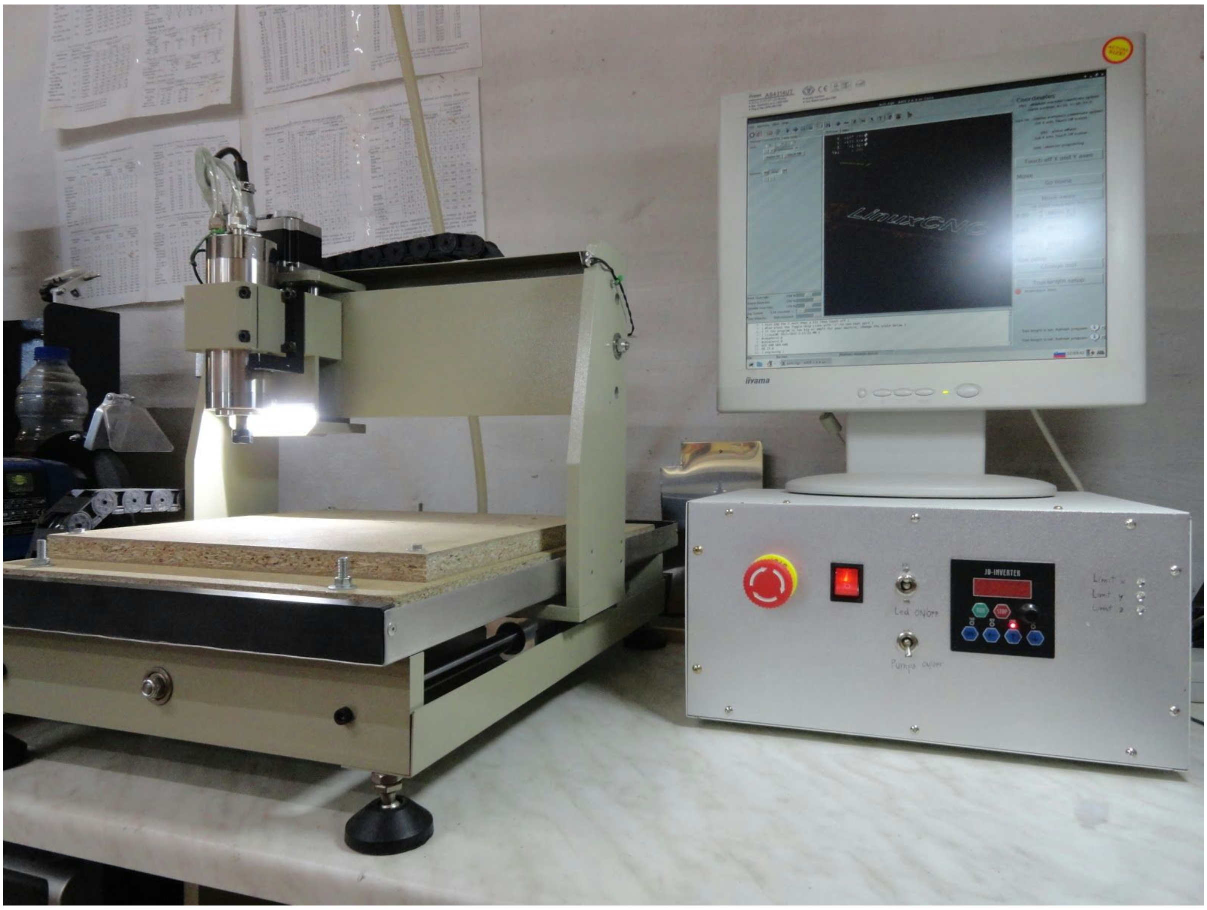


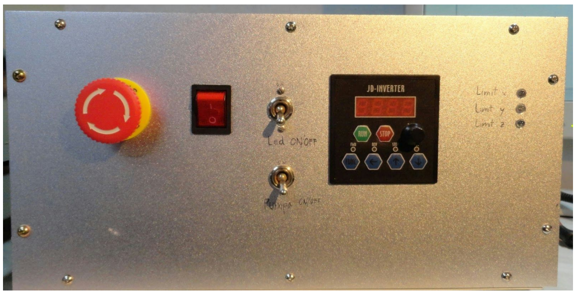
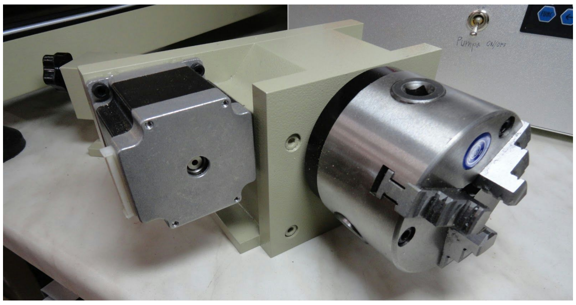
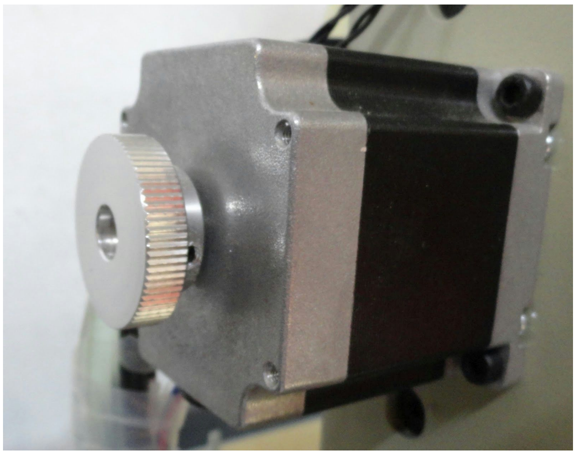
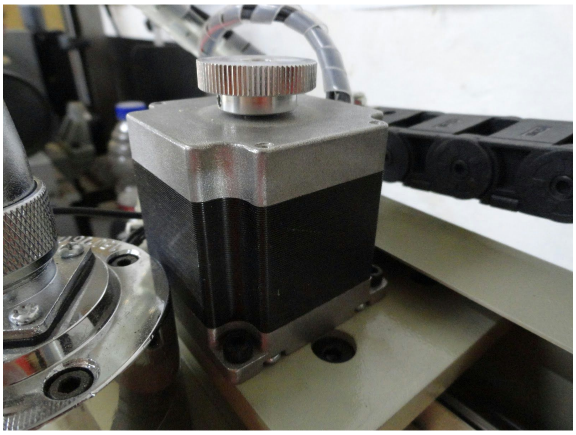
Table of contents

Hardware

This document applies to all similar CNC routers and 3D printers. Here is our CNC named Cene:

1. **CNC model:** Chinese 3040 / Donga TS 3040; 4 axis
2. **Table size:** 30x40 cm; 70mm max feeding height
3. **Stepper motors:** Nema 23; 57BYGH56-401, 2.2" (closest match found online)
    a. Bipolar, 2 phase
    b. 200 steps/revolution; 1.8°/step
    c. Outer dimensions: 56x50x55mm
    d. Rcoil = 0.9 ohm
    e. Ic = 2.8 A
4. **Spindle motor:** Jiandia JDFD-S5
    a. 24 000 RPM
    b. AC 230V/50Hz, 0.8kW VFD
    c. Water cooled, 60W immersion pump
    d. Nut model: ER11-A
5. **Drive unit:** Ball screws, 4mm/revolution (SFU01204-4)
6. **Communication interface:** parallel port (TX13963 HWJ130613B)
7. **Software:** LinuxCNC

We added a number of improvements (see below).

Limit Switches

We installed **6 limit switches**, one pair for combined home (min) and limit (max) position, for X, Y and Z axis. Each pair is connected in parallel directly to parallel connection board:

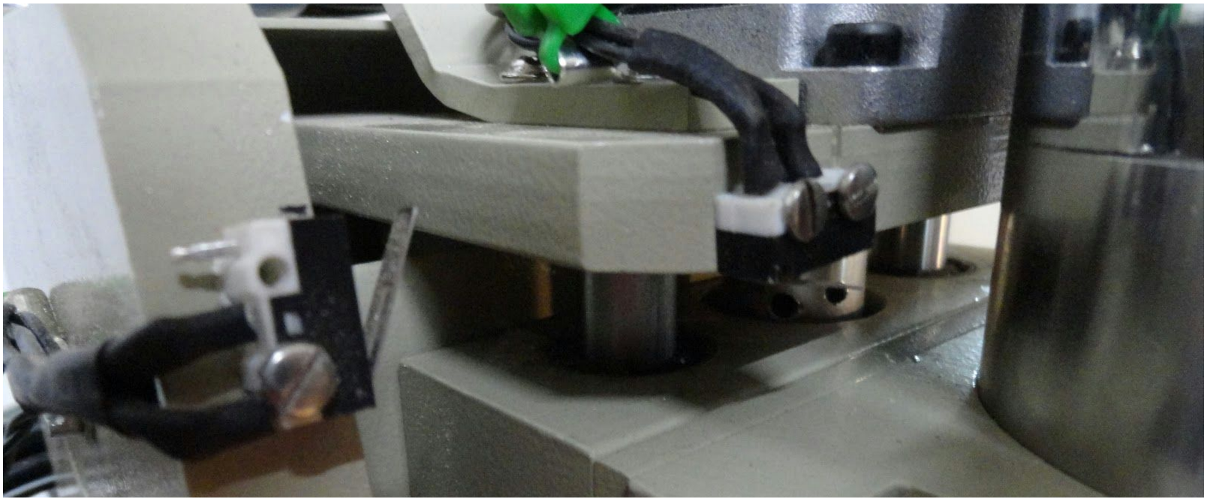
- X limit switches to pin: 11
- Y limit switches to pin: 12
- Z limit switches to pin: 13

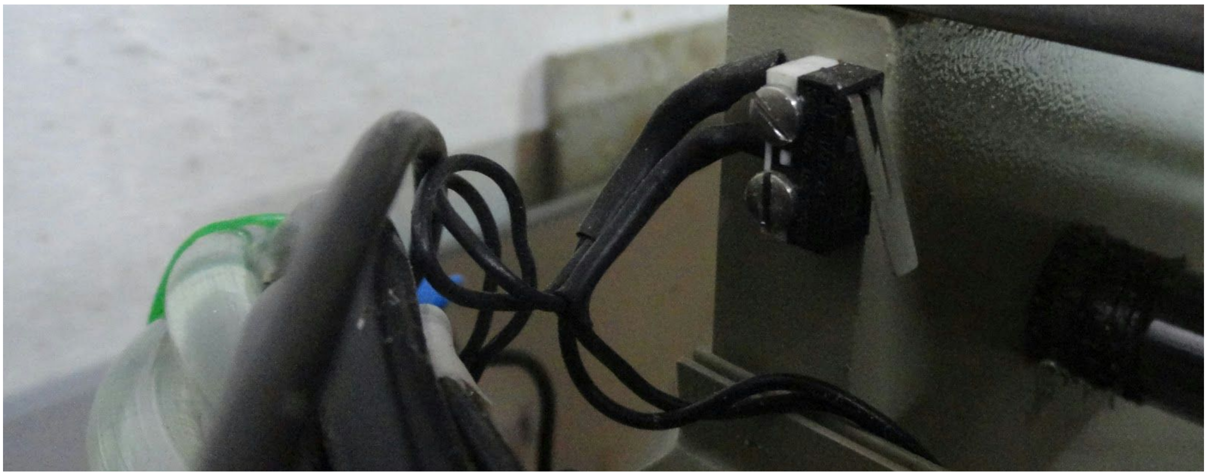
They are switching towards GND. No pull-up resistors are necessary - they are included in parallel connection board. All signal lines and common GND are added to additional four-pin connector on the back of CNC.

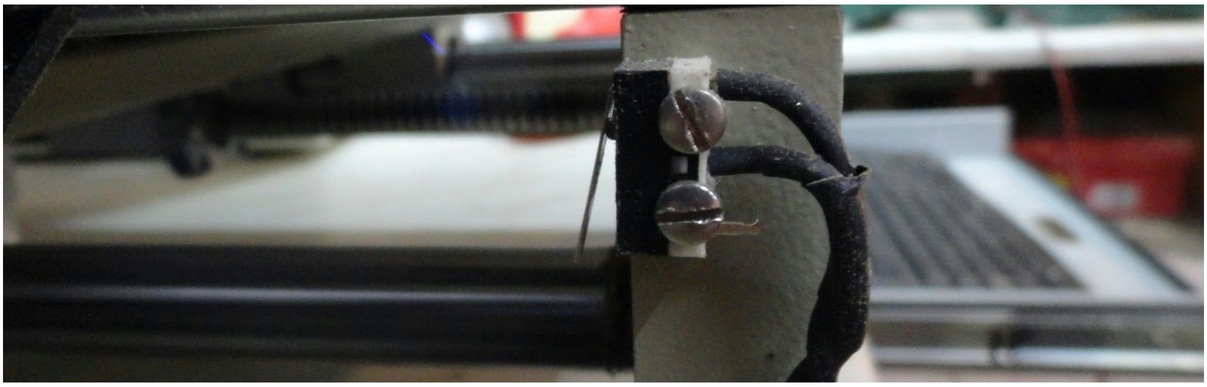
In software, these pins must be inverted. To add them manually in LinuxCNC, navigate to file: *`name_of_your_cnc`*`.`**`hal`** and add following lines:

```
net both-home-x     <= parport.0.pin-11-in-not
net both-home-y     <= parport.0.pin-12-in-not
net both-home-z     <= parport.0.pin-13-in-not
```

More about connecting limit switches can be found here (LinuxCNC Documentation).







There's also an LED on the front plate, indicating which of the switches are triggered. This is mostly useless since the switches only trip for a fraction of a second during homing.

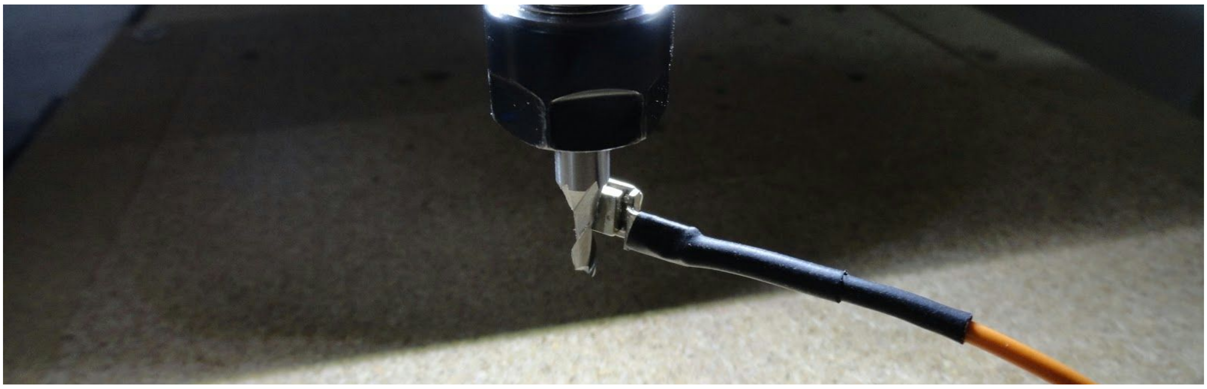
Tool Length Probe

We made a coin-shaped tool length probe with a spring inside to prevent any possible tool breakage. The other electrical pole is a magnet, that is placed on the machine/mill.
Its height is 8.00 mm. Setting tool length/offset with the probe is dead simple and very accurate.
Similar to limit switches, probe is connected to pin 14 and must switch towards GND.
In file file: *`name_of_your_cnc`*`.`**`hal`** add following line:

```
net probe-in         <= parport.0.pin-15-in-not
```

Signal is again inverted.

If you make your own, make sure to correct the height in *`routines/tool_length_offset`* file. The spring inside proved to be a *tool saver* in a number of different situations, so it is highly recommended.



Pump Switch & LED Light

We added a switch for spindle motor coolant pump and LED lights under motor directly to the front plate. It's convenient and it's harder to forget to turn it on.

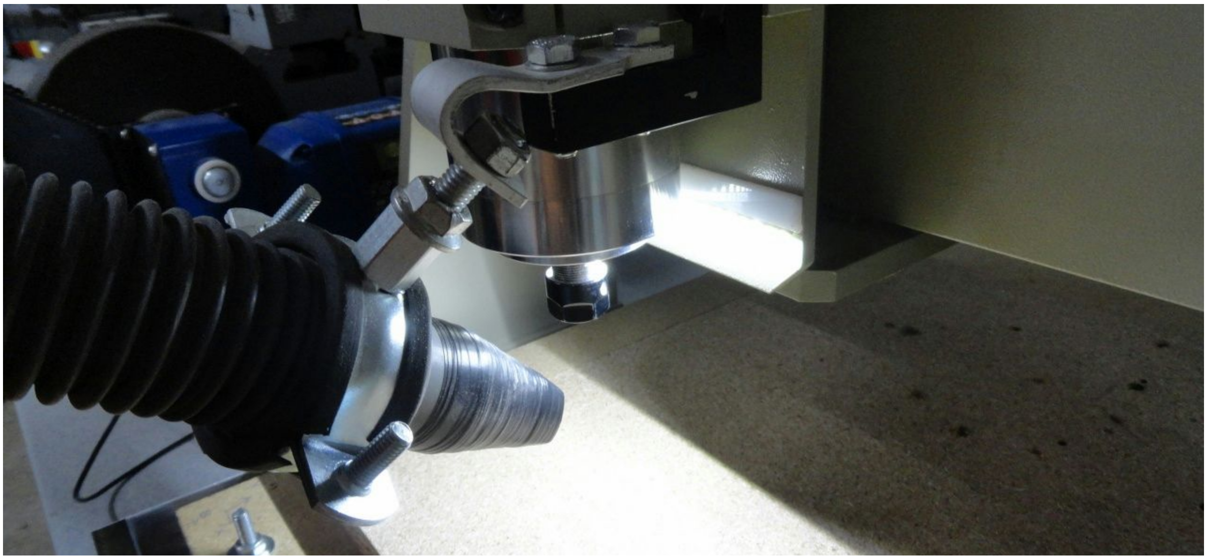
The pump is in a sealed container with water and anti-freeze to prevent mold growth and yes, freezing in our sometimes-very-cold workshop.

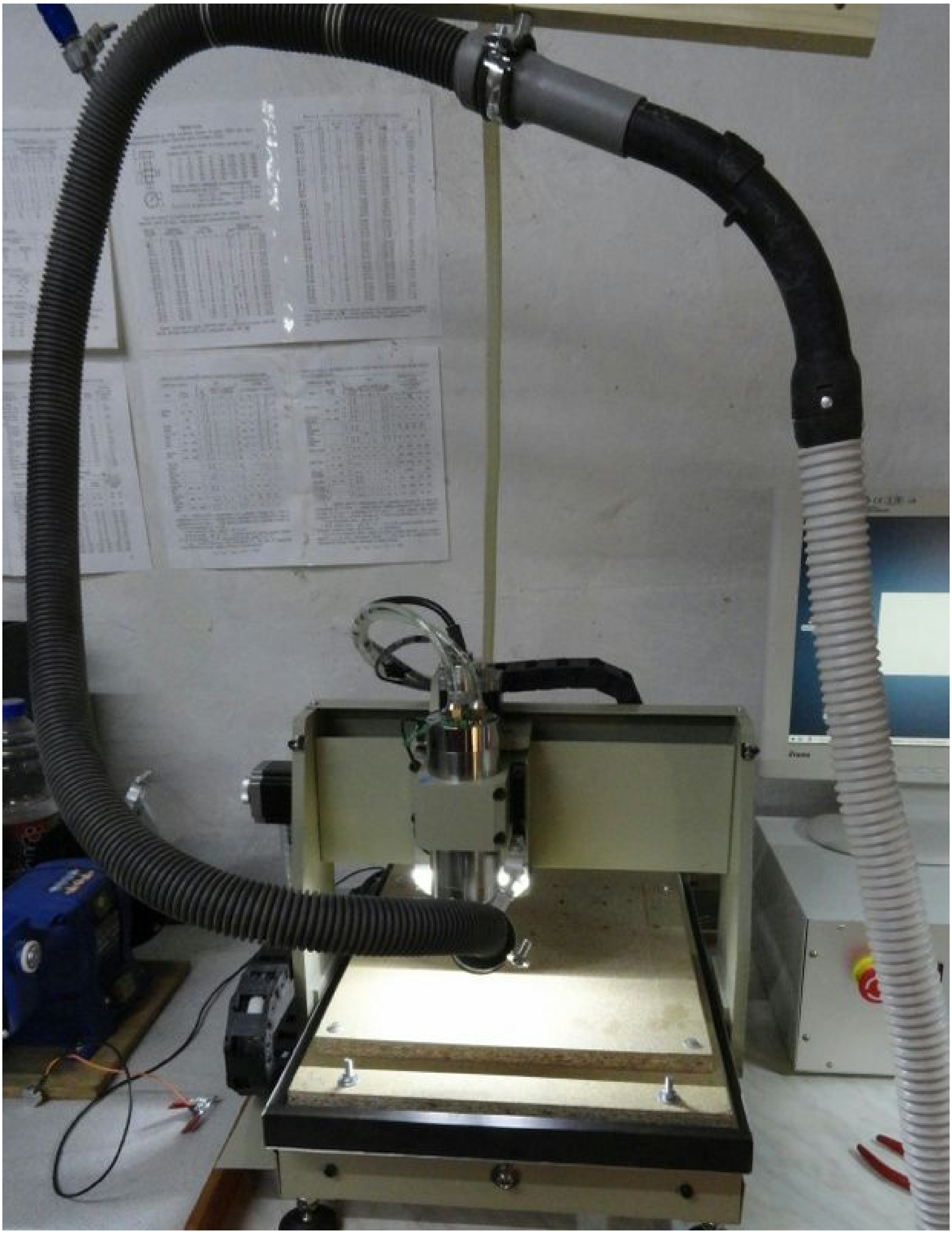
LED light and tool length probe share another four pin connector on the back of CNC, while the pump occupies an aditional connector since it is 230V.

Vacuum - Chip Removal

An industrial vacuum is attached to spindle motor and positioned as close as possible. Tube should be as short as possible, to achive the largest suction power and as soft as possible, to ensure its stiffness wouldn't disrupt axis movement.

A must have for **wood** working.

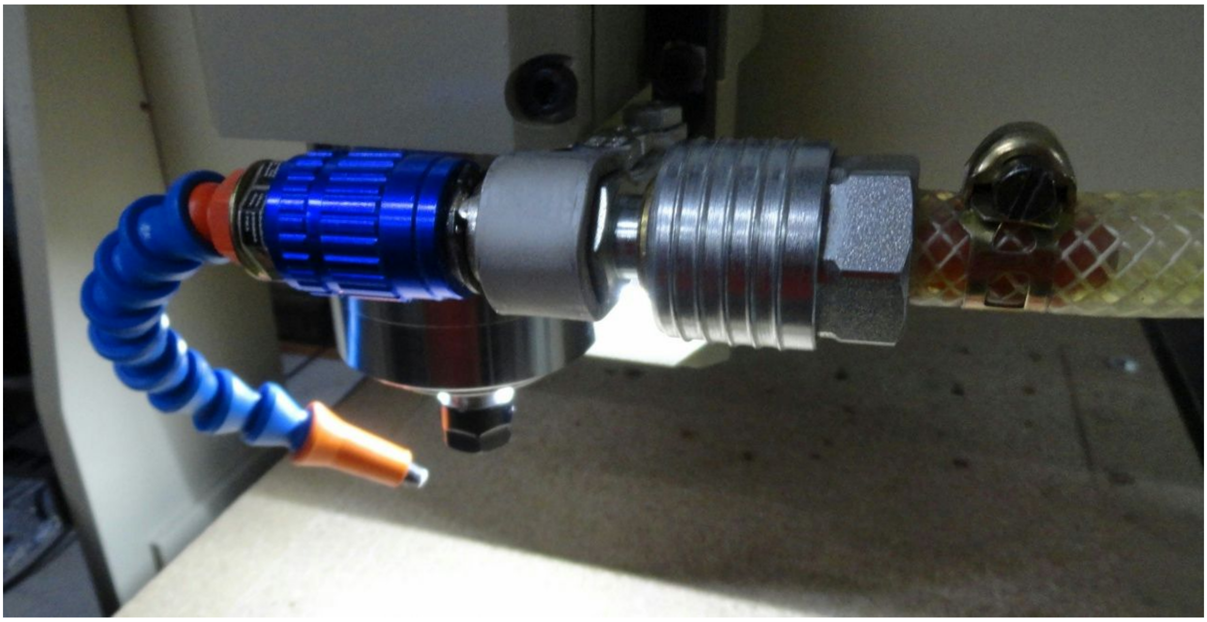


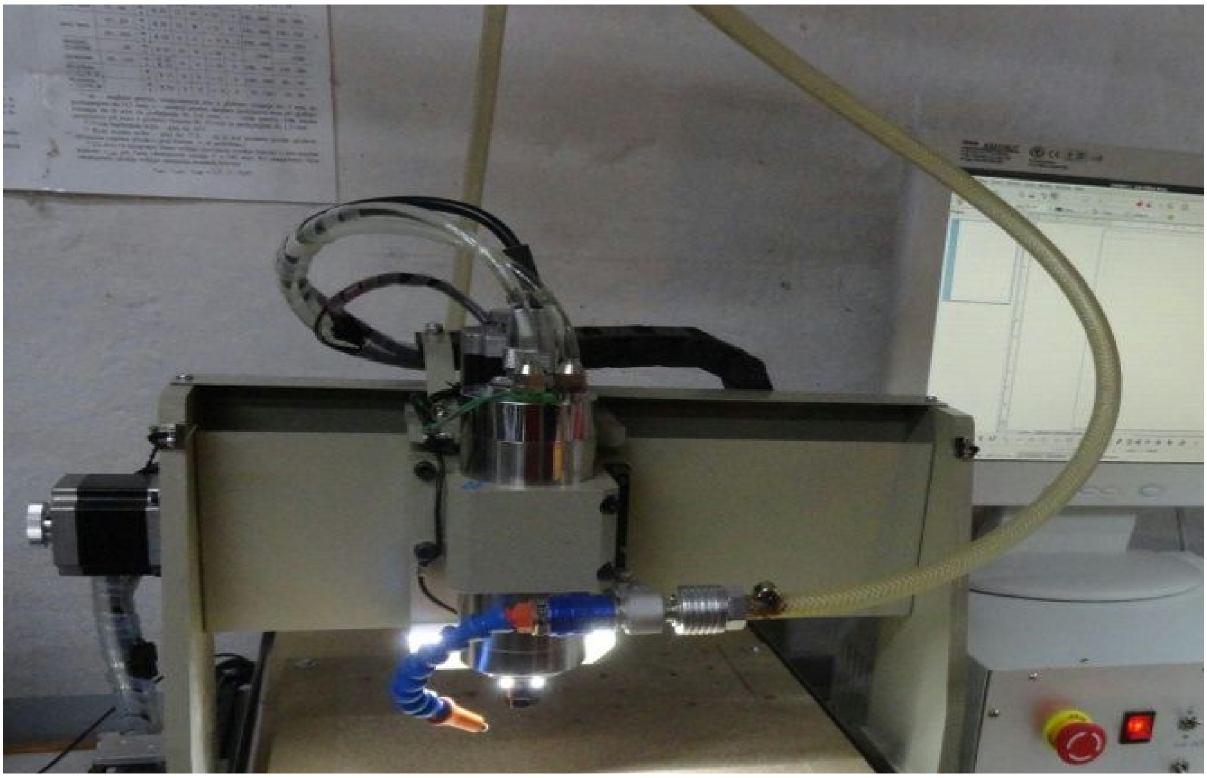


Tool Cooling, Lubrication, Blowing

A flexible hose for jet or mist spraying water/oil/emulsion is installed in the same manner as vacuum tube. A large tray under the aluminium profiles bed is installed - it collects coolant and chips, filters them and returns liquid to special cooling pump.
Currently we don't use any water/oil/emulsion - we blow away chips with high pressure air flow (from a workshop compressor).
Again, the pipes and tubes should be as flexible as possible.

A must have for **metal (aluminium)** working.



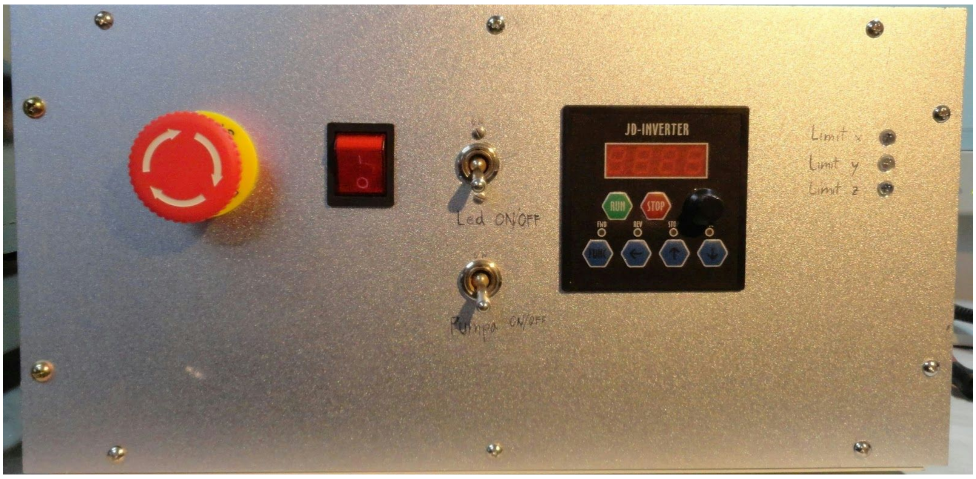


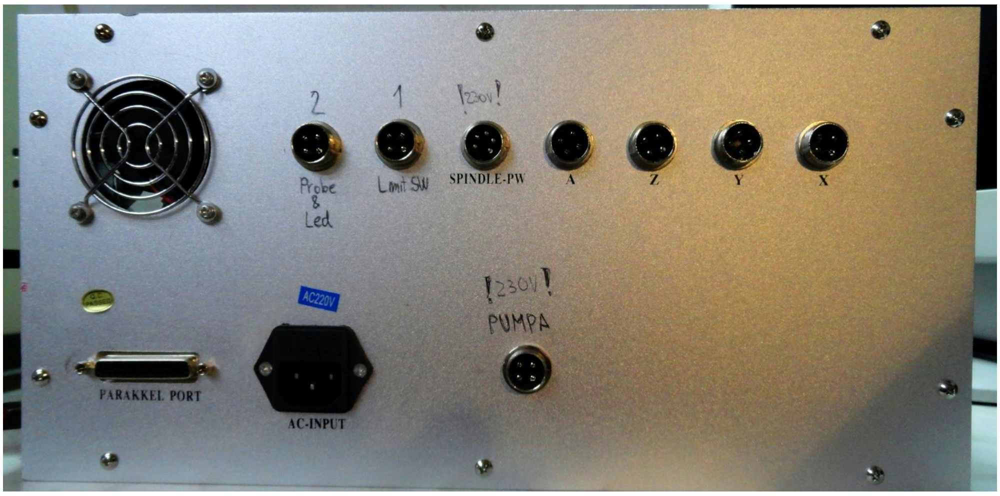
Motor Cooling System

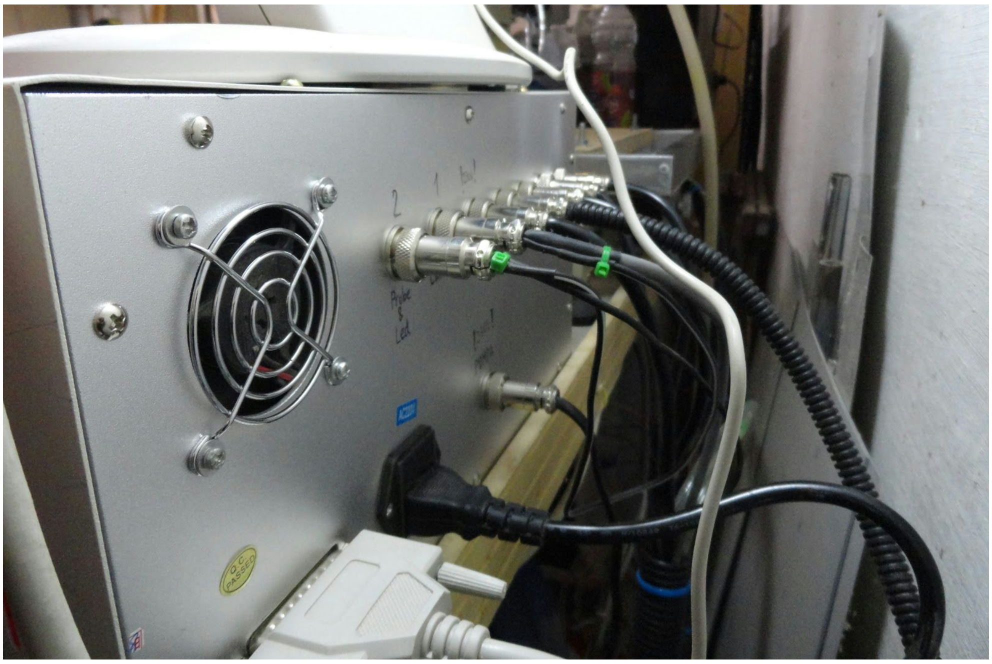
TODO. The sealed container for pump doesn't dissipate enough heat so we must insert a radiator somewhere on the coolant piping. It will be cooled either passively or actively using the exhaust of the vacuum for chip removal.

All the above additional equipment is needed for a slightly more serious milling, say aluminium parts. None of these is needed for PCB milling or engraving.

Control Unit

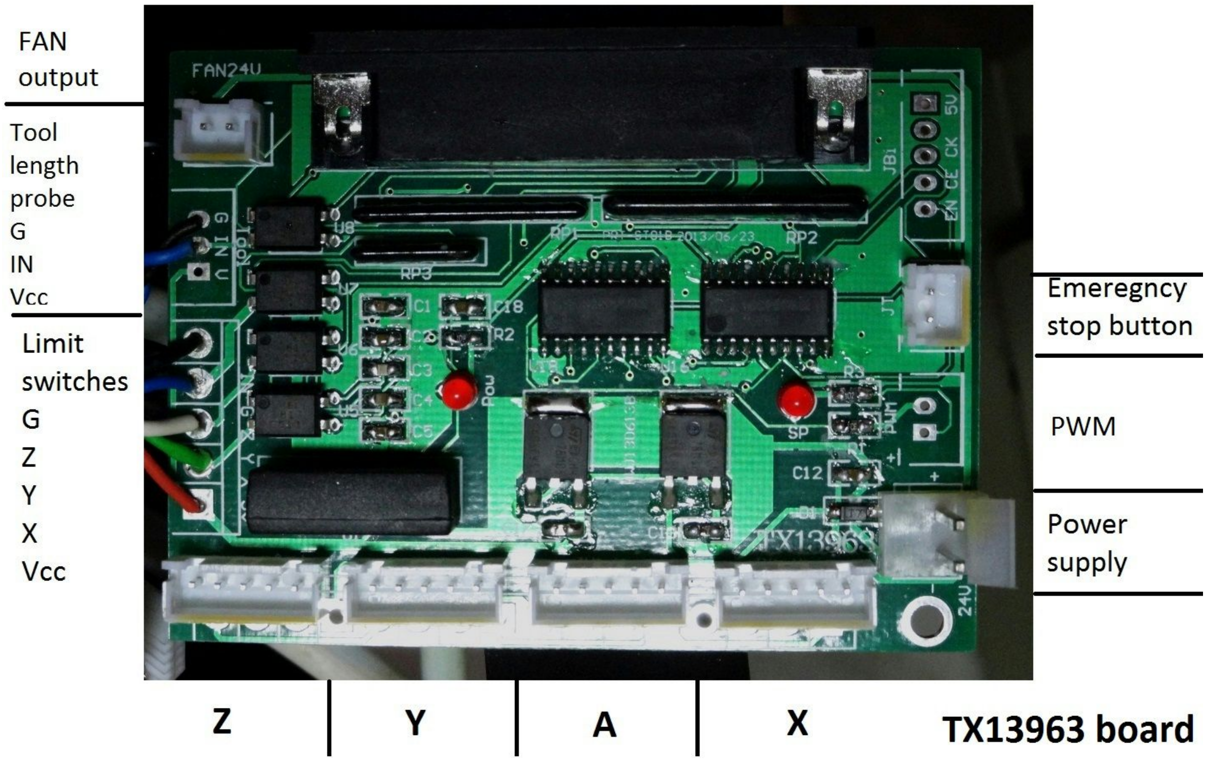






Parallel Port

Control unit is connected to PC with standard, old parallel (printer) port. In order to connect limit switches and tool length probe, we don't need any external components. Just connect them to prepared connectors as shown below:



There are another unused pins which could be used to control PWM motors (and other features), but we didn't implement them.

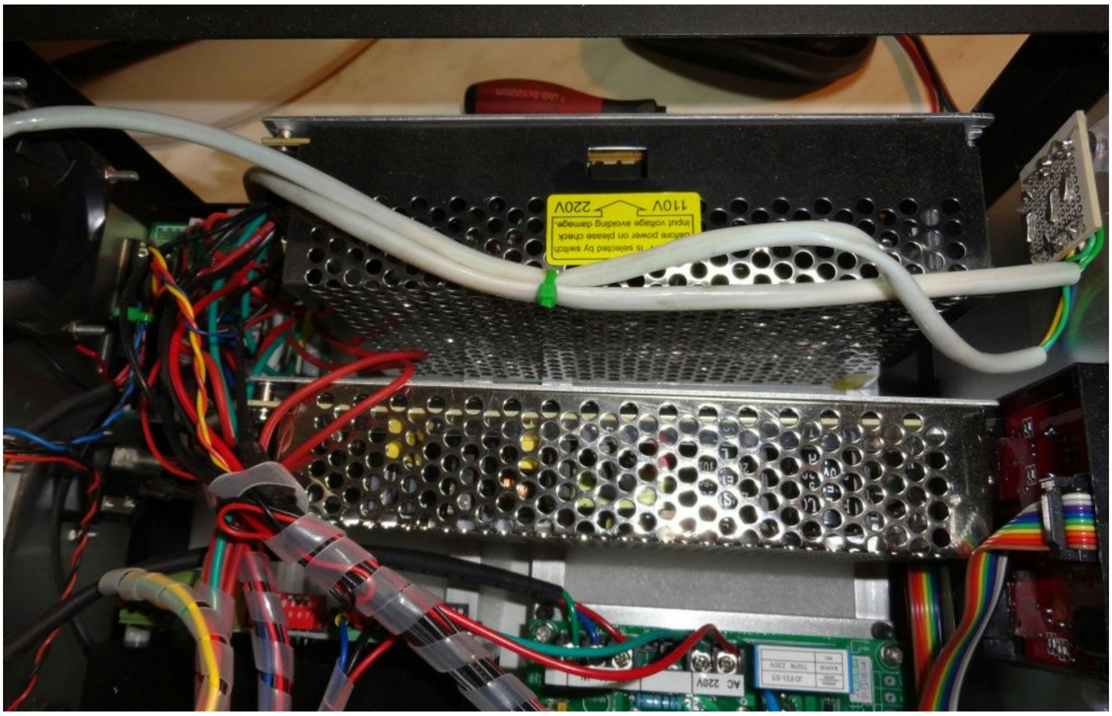
Power Supply

A quick calculation on stepper motor current consumption tells us, that installed power supply unit **(24V, 6A)** isn't sufficient: each motor can drain up to 2.7 A. If we use all axes (X, Y and Z) in one moment motors could require up to 3*2.7 A = 8.1 A. Also, power supply should have at least some reserve (1A - 2A) to handle short current peaks without voltage collapsing.

We installed another 24V, 10A PSU from ebay, and split stepper motor power supply lines:

- X and Y to first PSU (24V, 10A)
- Z and A to second PSU (24V, 6A)

Ground (GND) must be connected together, preferably in single common point.
The same for earth terminal.



Stepper Driver Board

There are 7 dip switches onboard to select stepping mode, decay mode and current (torque) mode. 7th switch is unused.

M1	**M2**	**DCY1**	**DCY2**	**TQ1**	**TQ2**	NC
1	**2**	**3**	**4**	**5**	**6**	**7**

- Note that M1 and M2 have pull-up resistors, while DCY1, DCY2, TQ1 and TQ2 have pull-down resistors.
- There is a whole section (*Improvements*) below for those with a little knowledge about eletronic. Seems like these board have quite a few errors, but somehow still manage to do their job.

- Changing these switches while running isn't recommended. Turn off, set switches and turn on should do it.

Excitation Settings (microstepping)

You can use M1 and M2 pin settings to configure four different excitation settings. Increasing microstepping has different effects:

- higher microstepping value increases resolution (e.g. 200 steps/rev → 400 steps/rev; 1.8°→ 0.9)
- higher microstepping makes motor rotation more fluid. Motors are quieter.
- higher microstepping drastically decreases torque. Can be a problem for Z axis (or other), where stepper motor must hold/pull up the entire Z axis assembly.
- higher microstepping means lower speed. More (micro-)steps are required for the same distance so speed must be increased - but the computer's real-time abilities set a limit on that (BASE_PERIOD in *`name_of_your_cnc`*`.`***`ini`***)

There are a lot of online articles about side effects of microstepping. We recommend reading at least these two:

- Microstepping Tutorial
- Microstepping: Myths and Realities

M1	M2	Mode (TBTB6560)	Microstep resolution	Setting
ON	ON	2-phase	Full step (no microstepping)	default
OFF	ON	1-2-phase	Half step (½ microstepping)	
OFF	OFF	2W1-2-phase	Either step (⅛ microstepping)	**recommended**
ON	OFF	4W1-2-phase	Sixteenth step (1/16 microstepping)	

With recomended setting (⅛ microstepping), we get:

- 200 steps/rev → 1600 steps/rev
- 1.8°/step → 0.225°/step
- smoother rotation (and quieter)
- very low torque (about 20% of full step mode).

After changing these switches you must change *SCALE* values in controlling software(`.ini`). Do not forget to take into account screew/ball ring drive ratio:

Motor: 200 steps/rev
Ball screw: 4 mm/rev
Microstepping: ⅛; 200 steps/rev → 1600 steps/rev
Scale (steps/1mm): 1600/4 = **400 steps/mm**

In LinuxCNC, navigate to file: *`name_of_your_cnc`*`.`***`ini,`*** search for `SCALE =` and correct value according to your microstepping setup, in our case:

`SCALE = 400`

Repeat this for all axes.

Current Decay Mode

This is a copy-paste from a forum:

In fast decay mode you allow current from the motor to go directly to the PSU. This allows the magnetic field to collapse quickly. The motor will coast afterwards. The advantage will manifest as faster possible motor speeds.

In slow decay mode you feed the current from the motor back into itself. The magnetic field collapses slowly in this case. The motor will 'dynamically' brake. This is sometimes a desirable attribute, but it will limit the motors speed.

Since we recommend ⅛ microstepping, this drastically lowers torque. When motors must make rapid clockwise and counter-clockwise moves, fast decay mode is desired, but, in order to hold the motor in its position (step) steadily, slower decay mode is desired. We stick somewhere between both of them, setting the current decay mode to 50%.

DCY1	DCY2	Mode	Setting
OFF	OFF	Fast mode (0%)	default
ON	OFF	25%	
OFF	ON	50%	**recommended**
ON	ON	Slow mode (100%)	

Don’t be afraid to change and test this switches.

Torque Settings

Torque is in proportional relation to current through each stepper winding. Reference current is predefined with onboard resistor and driving unit (TB6560AHQ). As datasheet states, excitation voltage is 0.5V, while our onboard ceramic resistor is 0R22, resulting in 2.27A. Our stepper motor can handle currents up to 2.8A, so highest possible current (torque) is recommended.

TQ1	TQ2	Mode	Setting
OFF	OFF	100%	**recommended**
ON	OFF	75%	default
OFF	ON	50%	
ON	ON	20% (weak excitation)	

Overall our recommended settings are:

M1	**M2**	**DCY1**	**DCY2**	**TQ1**	**TQ2**	NC
1	**2**	**3**	**4**	**5**	**6**	7
OFF	**OFF**	**OFF**	**ON**	**OFF**	**OFF**	X

Improvements

Thanks to another enthusiastic CNC hobbyist Vincent Delacotte, who reverse-engineered the stepper driver board, there is a long list of improvements/corrections that can be made. It is weird that this boards work at all. Anyway don't fix what isn't broken - do it only if you really can't sleep without fixing these problems :)

1. **Step down regulator.**
   Currently there is LM2576S-3.3 step down regulator, that has an output voltage of 3.3V. TB6560 datasheet states that VDD - power supply voltage for logic should be between 4.5V and 5.5V.
   Bottom line, LM2576S-3.3 should be replaced with **LM2576S-5** or similar 5V step down regulator.
   Not tested yet, the output coil and capacitor should probably also be modified.

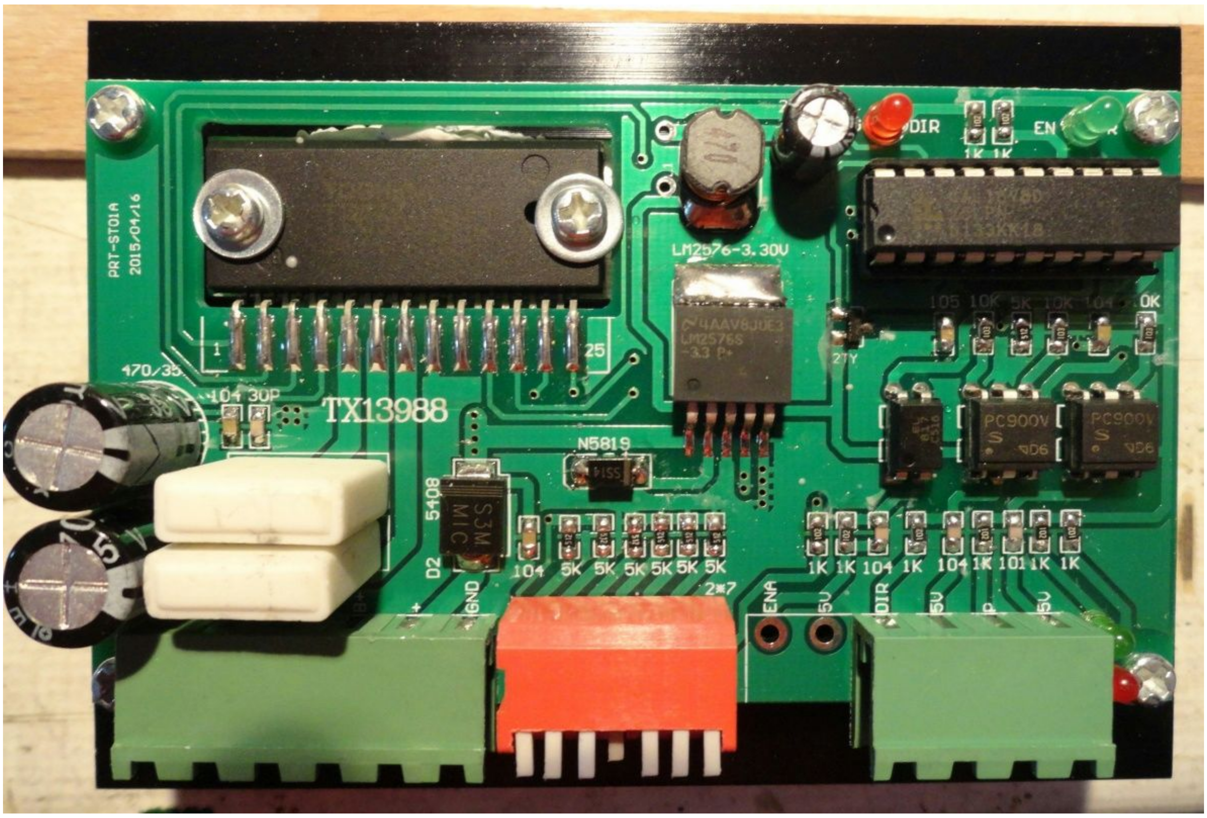
2. **Photocoupler parts.**
   Pull up resistors at PC900V photocoupler output side are currently 10KΩ. For a steeper rising edge, this resistance should be lowered (according to datasheet) to **280Ω**.
   Also, 10nF capacitor should be changed to **100nF.** Input resistor (1kΩ) could be lowered to **100Ω.**

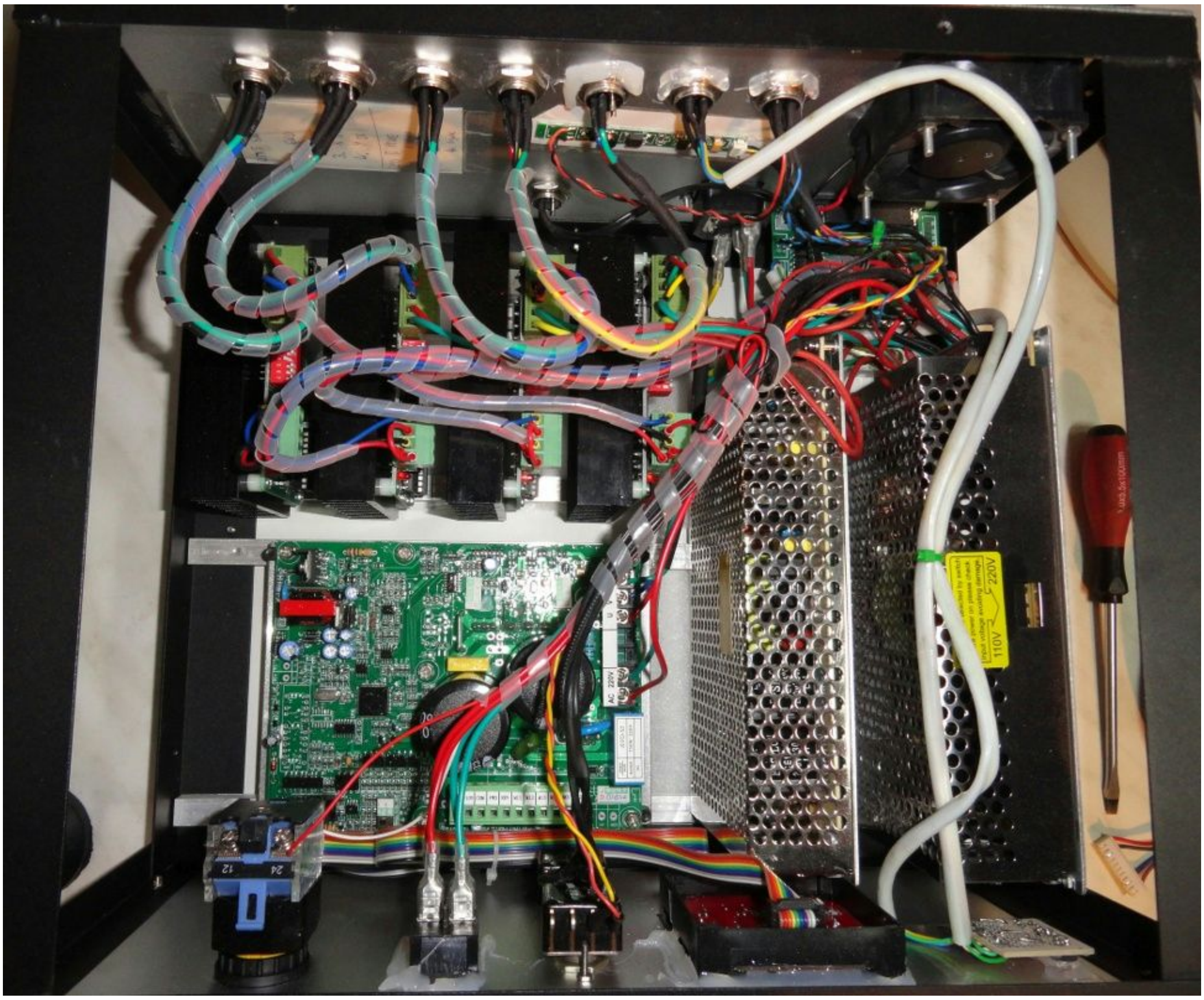
3. **Ceramic resistors**
   Those define reference current for stepper motors. Settings on torque switches define actual current in percentage of this reference current. Our NEMA23 stepper motors can handle up to 2.8A. Current resistors are 0R22. To set maximum current these resistors can be changed:
   Vref = 0.5V
   Iref = 2.8A
   Rref = Vref/Iref = **0R18.** Power dissipation on each resistor is: 1.39W.
   Both resistors on one driver should be changed to equal value.

4. **Power up timing** *(Vincent Delacotte)*
   I also noticed, thanks to posts in some CNC forums, that the 24V timings at the VMA and VMB inputs violated the power-up sequence described in the datasheet and the application note.
   So, instead of designing a breadboard with a relay as I saw in one forum, I designed my own breadboard with a reset generator from maxim MAX705 I had in my drawer since 2006.
   This one is the heart of the sequencer. It manages as well the power-up sequence with the built-in 200ms reset generator, but also the power down sequence with the PFI Power fail Input which trips the built in comparator as soon as the power input drops below 16.3 V.
   PFO output is tied to the MR Manual Reset input which also triggers the nRESET output, thus shutting down the P-channel mosfet again during the power-down sequence.

nRESET is tied to the TB6560 nRESET input.
It works well with the breadboard and the program for my next weekend is to convert the flying breadboard design into a small stripboard with soldered components.

Other values may be changed according to parts datasheets. Anyway, as it was said before, if it works, it works.





Software

We decided to use LinuxCNC as the controlling software. It's open-source, well documented and it works. Here's how we did it and how:

1. Find an old computer
2. Do a fresh install of the newest LinuxCNC and do a latency test, if it's too high, discard this computer and go back to step 1 (really, there's very little you can do in that case). There are number of discussions online about this latency problem.
3. Configure it. This took us a while but we will spare you all the work and just give you the configuration files. There are two different folders, `Cene` (that's the name of our machine) and `Cene4`. The first is a classical 3-axis mill, the other is with added A axis. Put both directories in `$/configs/`, then run LinuxCNC and choose the preferred machine. See chapter below (*Configuration*)
4. (Optional) Add some custom buttons and routines. We added:
    1. **Touch off X and Y axis:** touch off X and Y coordinates with one click.
    2. **Go home:** moves to home position – used for changing tools
    3. **Move away:** moves to the other end, useful for mounting and cleaning workpiece
    4. **Tool length setup:** touching off the Z-axis using the custom made tool length probe.

Configuration

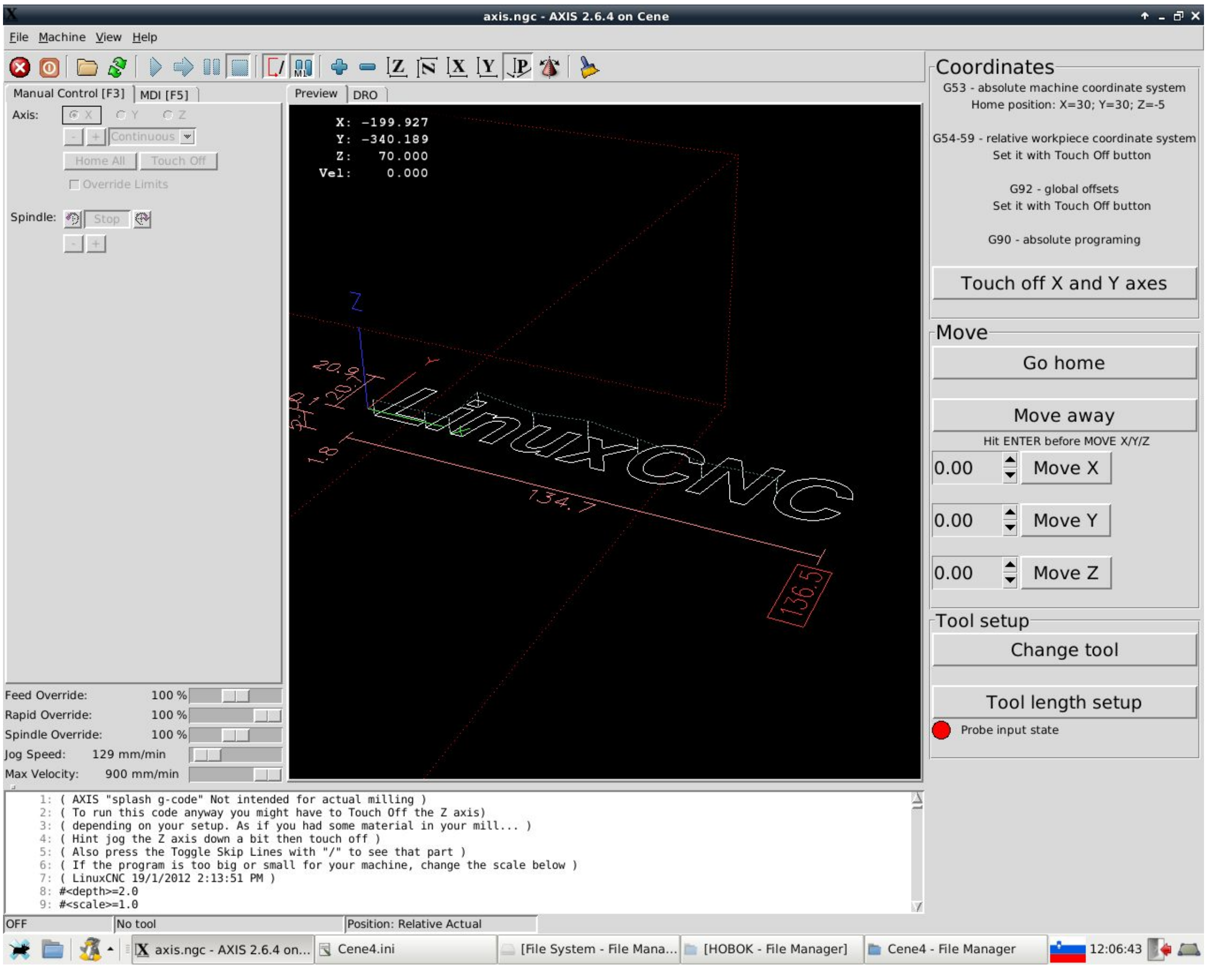
We don't know why, but we just couldn't generate correct code with Stepper Configuration Wizard. Anyway, LinuxCNC is well documented, so you can do it on your own.

There are few files you realy should know about if you add limit switches and other software buttons:

- *name_of_your_cnc*.***ini*** (in our case: `Cene.ini`)
  Base file - axis values, custom mdi files, custom hal file include, ...
- *name_of_your_cnc*.***hal*** (in our case: `Cene.hal`)
  Parallel port pinout and timing.
- *custom_postgui*.***hal***
  ALink between default and custom HAL layer. This file includes all signals from buttons, that are mapped to base HAL layer.
- *custompanel*.***xml***
  Add custom panel on the top of the LinuxCNC GUI. This file defines look of buttons and other data displayed on the right side of GUI.

We've included all configuration files for our 3 and 4-axis Cene.

The best way to learn how to modify our custom panel is simply to edit current configuration and of course, google it. We've found a lot of help on forums.



```
X: -199.927
Y: -340.189
Z:   70.000
Vel:   0.000
```

G53 - absolute machine coordinate system
Home position: X=30; Y=30; Z=-5

G54-59 - relative workpiece coordinate system
Set it with Touch Off button

G92 - global offsets
Set it with Touch Off button

G90 - absolute programing

Hit ENTER before MOVE X/Y/Z

0.00

0.00

0.00

Feed Override: 100 %
Rapid Override: 100 %
Spindle Override: 100 %
Jog Speed: 129 mm/min
Max Velocity: 900 mm/min

```
( AXIS "splash g-code" Not intended for actual milling )
( To run this code anyway you might have to Touch Off the Z axis)
( depending on your setup. As if you had some material in your mill... )
( Hint jog the Z axis down a bit then touch off )
( Also press the Toggle Skip Lines with "/" to see that part )
( If the program is too big or small for your machine, change the scale below )
( LinuxCNC 19/1/2012 2:13:51 PM )
#<depth>=2.0
#<scale>=1.0
```

OFF | No tool | Position: Relative Actual

axis.ngc - AXIS 2.6.4 on...

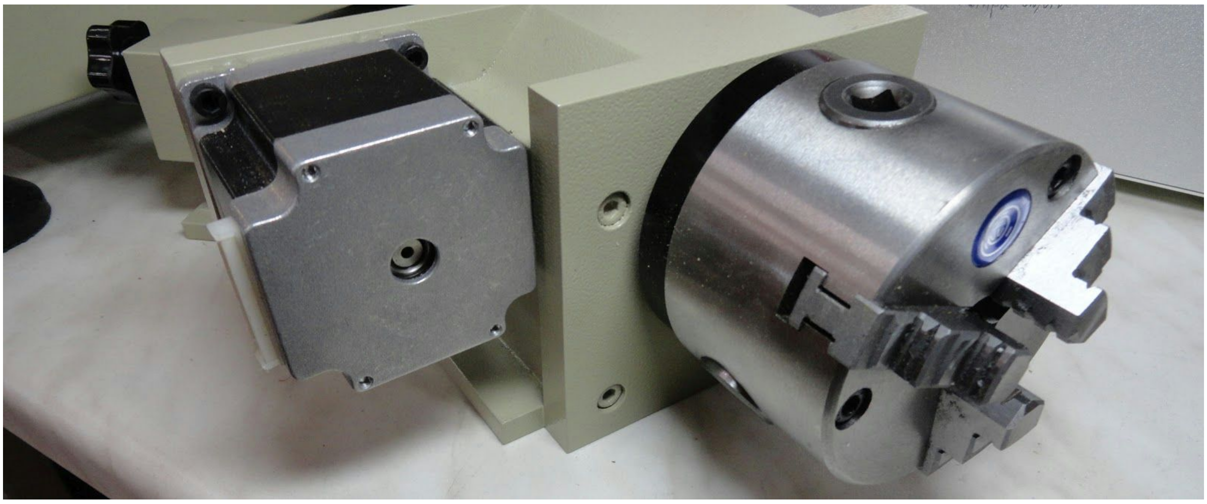
USB CNC

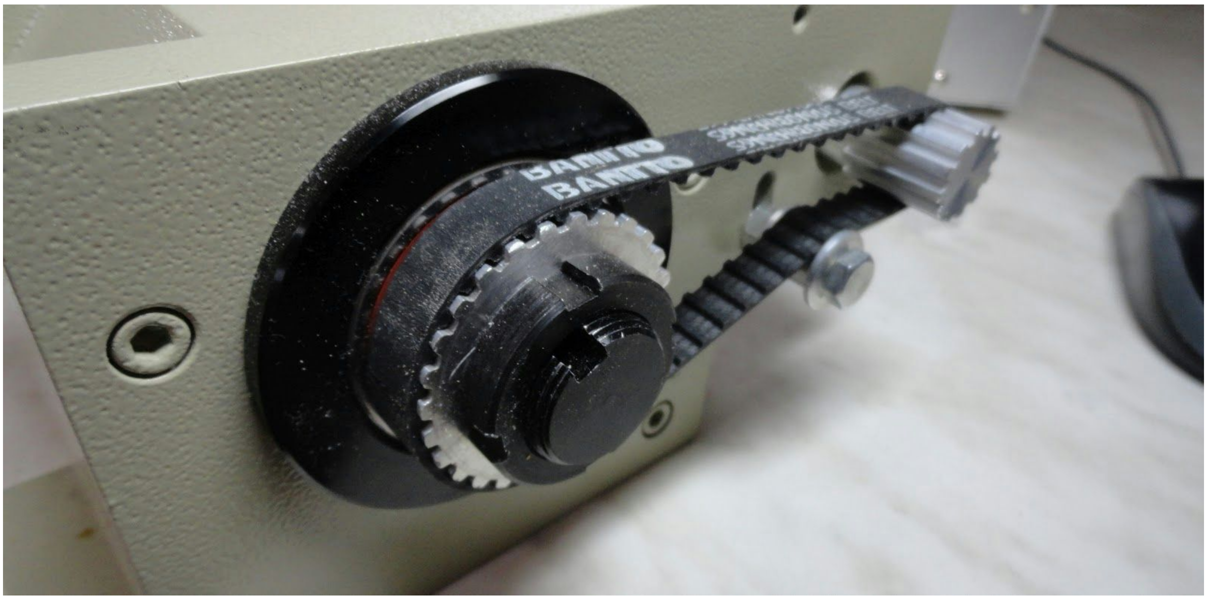
LinuxCNC currently doesn’t support USB connections. Anyway, there are some affordable hobby CNC USB hardware and software online. For instance, PlanetCNC (about 200€).

Working with A axis

We haven't done any serious milling with the 4th axis so far therefore its setup hasn't been tested yet, but here are a few things:

- Home position: the A axis has no home position, but you can touch it off just as the Z axis.
- Homing: when you click the *»Home All«* button, it will home the XYZ but not the A axis. You have to do it in the menu >*Homing* > *A axis.* If you don't do that, you won't be able to send MDI commands because LinuxCNC will say the machine is not homed.
- One millimeter means 1 degree for the A axis.





Other Thoughts

Stiffness, Speed And Materials

Face it, it is a cheap Chinese CNC. It is good for many things, but don’t expect it to be the same as 100.000€ professional CNC. Although build quality is quite good, machine isn’t as rigid as it should be, so harder materials can’t be milled.
The tricky part is to define appropriate drill/mill bit, spindle motor speed, milling width and milling speed. The key is to keep cutting forces as low as possible since higher forces will cause the machine to flex, thus breaking the tool.

We use this online tool to calculate speed and verify forces, and still manage to break bits. So far we have successfully milled wood, PCBs, phenolic resin board and aluminium, but the latter two are pretty much the limit of the machine (or maybe even a little beyond).

Special care must be taken when G-code is generated. For instance, if possible use trochoidal milling with very narrow and shallow cuts to keep forces low. Also use smaller tools (2-mm) and lower number of flutes (1-2). Lubrication helps with lowering cutting forces, but for aluminium, chip removal is even more important. If not removed properly, chips will stick to the tool, clog it and break it.

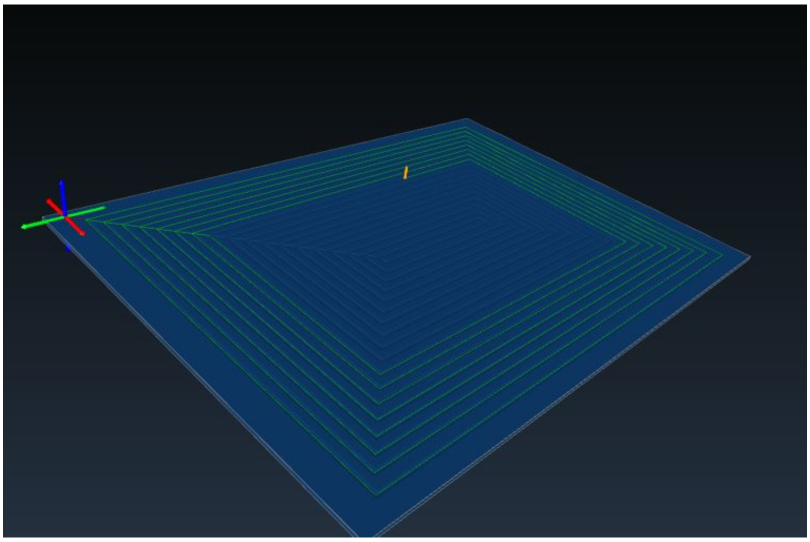
It is obvious that machine frame is very thick and sturdy but the linear shafts/bearings that support the whole frame are too thin to be rigid enough. One solution is to mill smaller parts if possible and always put them as close to one corner as possible. The frame is more rigid near corners.

Bed Flattening

Our Cene has quite large Z axis movement range, about 70mm. If we want to route PCBs, they must be raised with an additional wooden backplate. To avoid changing this backplate every time but still maintain a flat surface, we generated a scalable bed flattening G-code program.

Define rectangular dimensions, set tool offset and run it to flatten out surface. You can also change mill width and depth.

This file is named `bed_flattening.ngc`



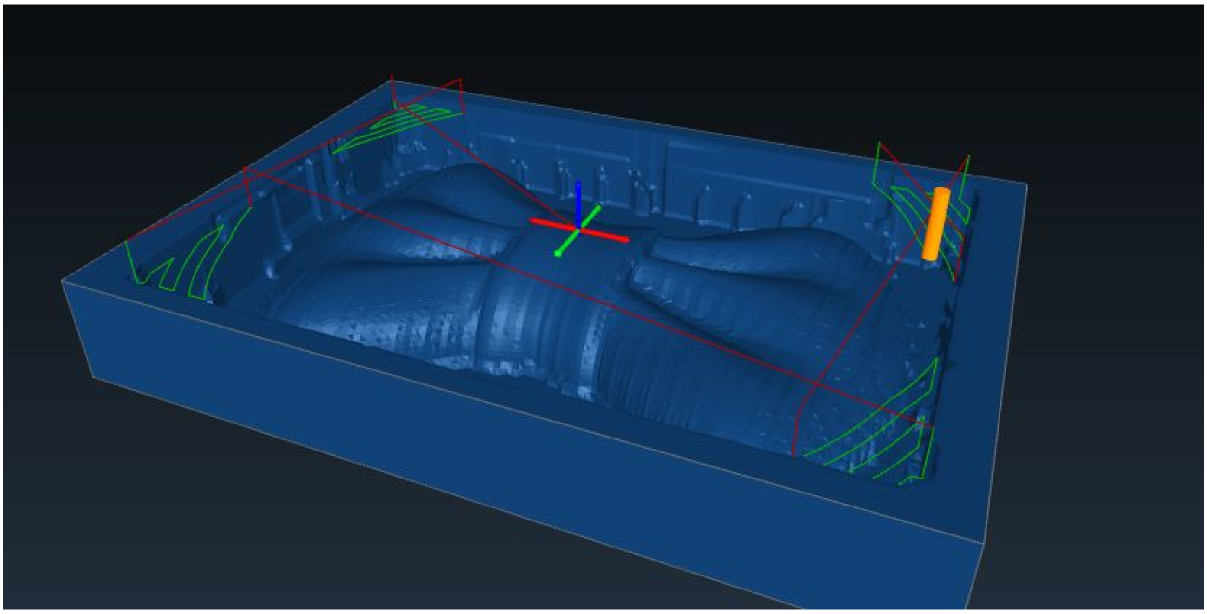
G-Code generation

This is a part where we just can't tell you much. There are tons of CAD/CAM programs out there but they are really complicated and expensive.. Be prepared to invest a lot of time in learning software and all the CNC glossary.

Included Files

There is:

- CNC configuration folder
- Datasheet folder (including all component datasheets and PCB schematics by Vincent Delacotte)
- example code:
    - bed flattening
    - wooden bow tie



Written by Damogran Labs
http://www.damogranlabs.com/

Special thanks to **Vincent Delacotte** for schematic drawings, research and thoughts.