bCNC: From DXF to G code

From jean-claude.feltes@education.lu

Though I love FreeCAD, I still struggle with its Path workbench. For most of my applications, FreeCAD is simply too complex and I get lost in the many windows of the GUI. So I came back to bCNC and gave it one more try. I like some of its features very much. Maybe I'll use it more often!

This document describes how I learned to do a (very) simple contour and pocket operation.
The DXF file for this was created in LibreCAD.

Github repository for bCNC: https://github.com/vlachoudis/bCNC

Wiki: https://github.com/vlachoudis/bCNC/wiki

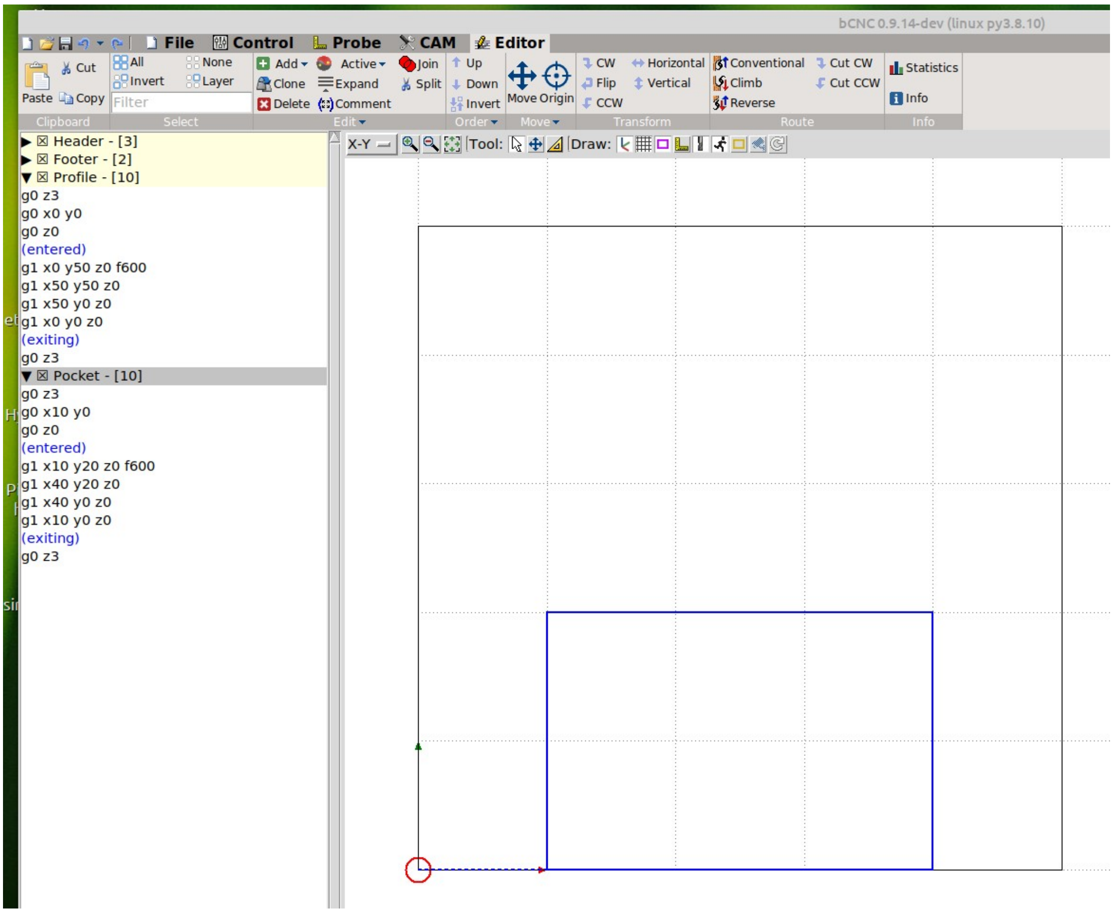
Start bCNC:

```
python3 -m bCNC
```

Remember: If anything goes wrong: <Ctrl>-Z = Undo

Opening a DXF file

File – open myfile.dxf



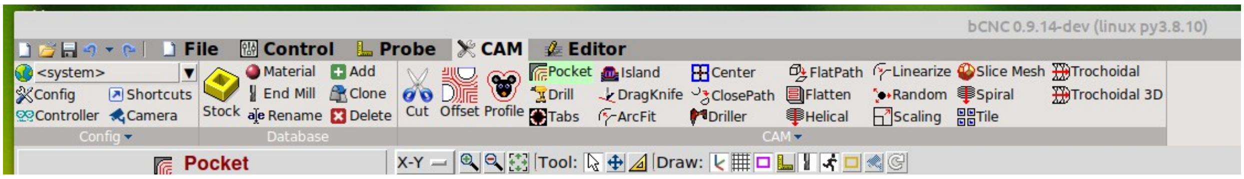
```
▶ ☒ Header - [3]
▶ ☒ Footer - [2]
▼ ☒ Profile - [10]
g0 z3
g0 x0 y0
g0 z0
(entered)
g1 x0 y50 z0 f600
g1 x50 y50 z0
g1 x50 y0 z0
g1 x0 y0 z0
(exiting)
g0 z3
▼ ☒ Pocket - [10]
g0 z3
g0 x10 y0
g0 z0
(entered)
g1 x10 y20 z0 f600
g1 x40 y20 z0
g1 x40 y0 z0
g1 x10 y0 z0
(exiting)
g0 z3
```

The Editor tab shows that there is already G code that has been produced.
It is structured into sub programs that correspond to the graphic elements of the drawing.
Here the subprograms have already been renamed to Profile and Pocket. (Double klick on the titles)

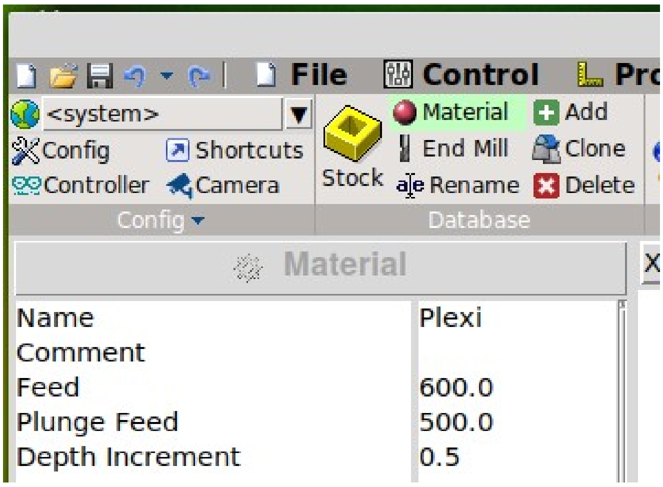
When clicking a sub program or a G code line, the corresponding element in the drawing is highlighted, and vice versa.

Setting up tools and material

- Switch to the CAM tab



- Define material (Klick Add button)

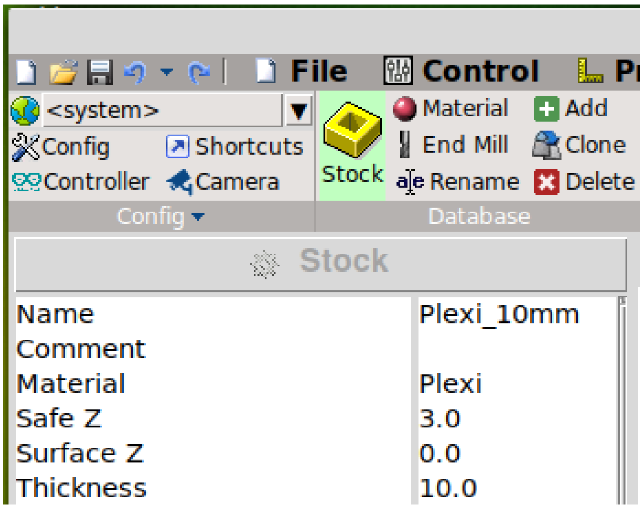


Material

Name	Plexi
Comment	
Feed	600.0
Plunge Feed	500.0
Depth Increment	0.5

- Define tool End mill the same way

- Define stock



Stock

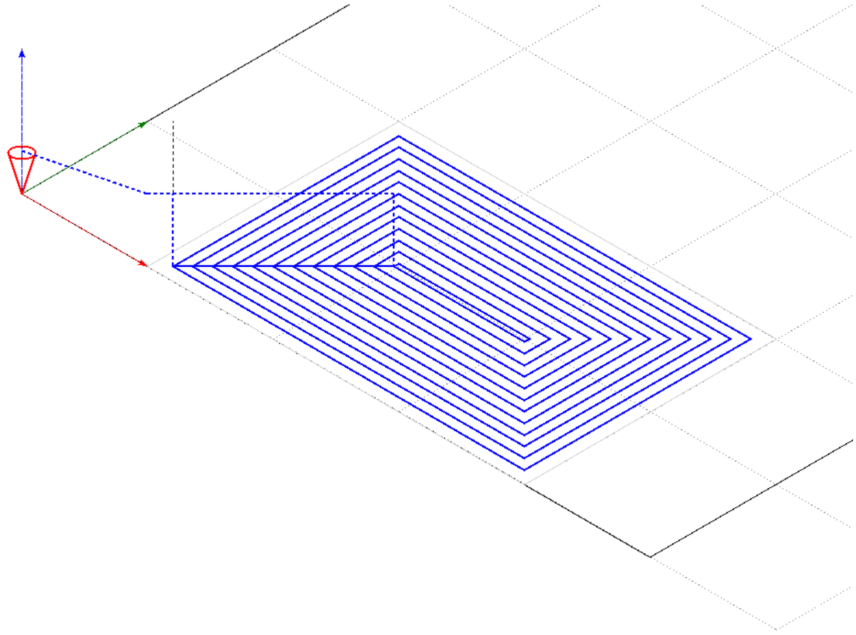
Name	Plexi_10mm
Comment	
Material	Plexi
Safe Z	3.0
Surface Z	0.0
Thickness	10.0

The pocket

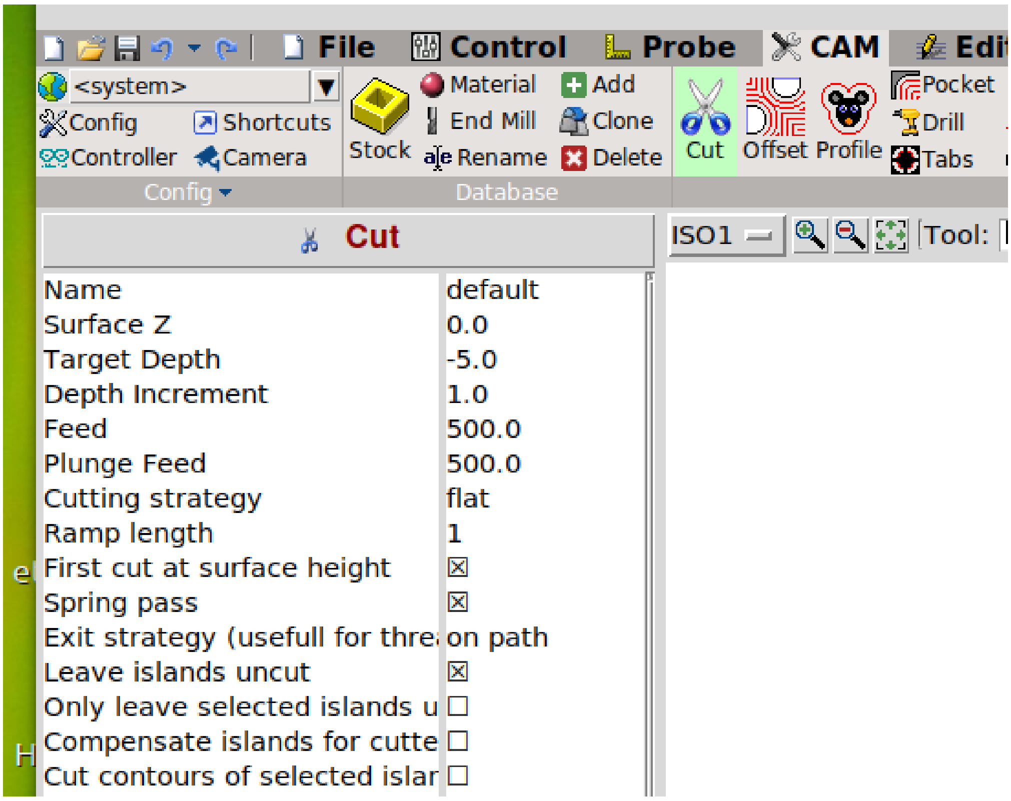
It is important to know about the special philosopy of bCNC:
The operation is divided into **2 parts: Pocket and Cut.**

First, select pocket outline in Gcode window or drawing, then klick Pocket and select tool (2mm endmill)

The surface cut is already to be seen:



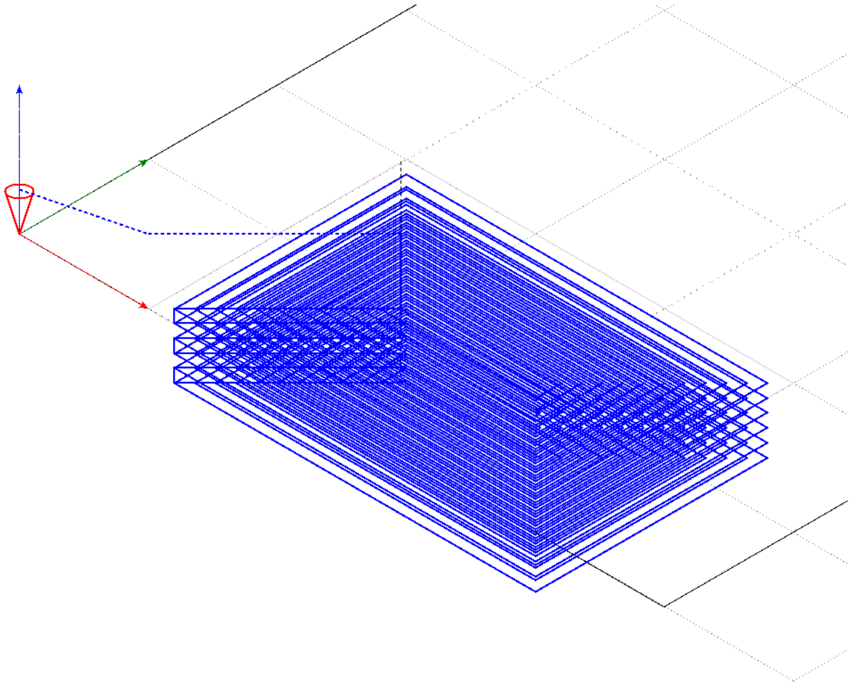
Now klick the “Cut” button and select the options, where most important are Surface Z, Target Depth and Depth increment:



Cut

Name	default
Surface Z	0.0
Target Depth	-5.0
Depth Increment	1.0
Feed	500.0
Plunge Feed	500.0
Cutting strategy	flat
Ramp length	1
First cut at surface height	☒
Spring pass	☒
Exit strategy (usefull for thre	on path
Leave islands uncut	☒
Only leave selected islands u	☐
Compensate islands for cutte	☐
Cut contours of selected islar	☐

Klick on the “Cut” button above the settings you just entered and watch the path:



Read the explanations in the help window to better understand the options. Some of them would get a better result than the ones I chose.

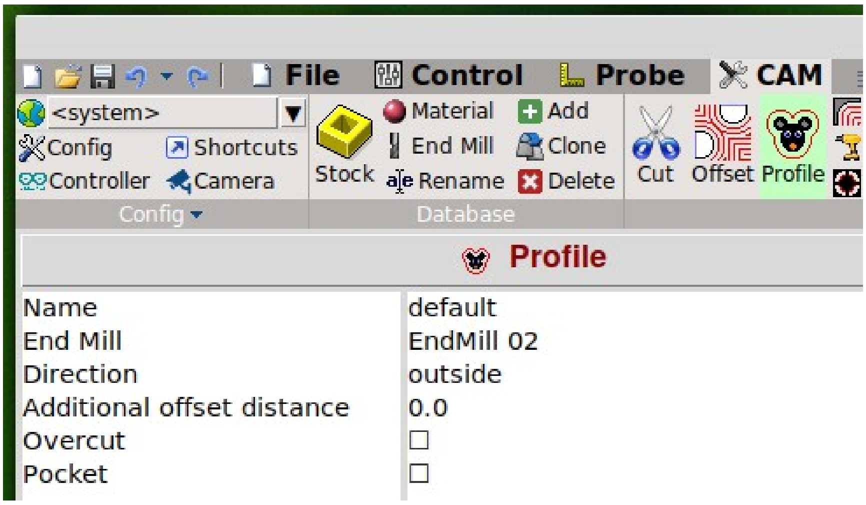
In the Editor Tab, you can now watch the produced G code, that has significantly increased in size.

The profile for the contour

A profile usually needs 3 operations:

- Define the contour and the tool
- Add holding tags
  (this can be omitted if you fix the workpiece with double sided adhesive tape).
- Define the Cut properties (like depth)

Select the contour in the editor. In the CAM tab, klick on Profile and select tool:



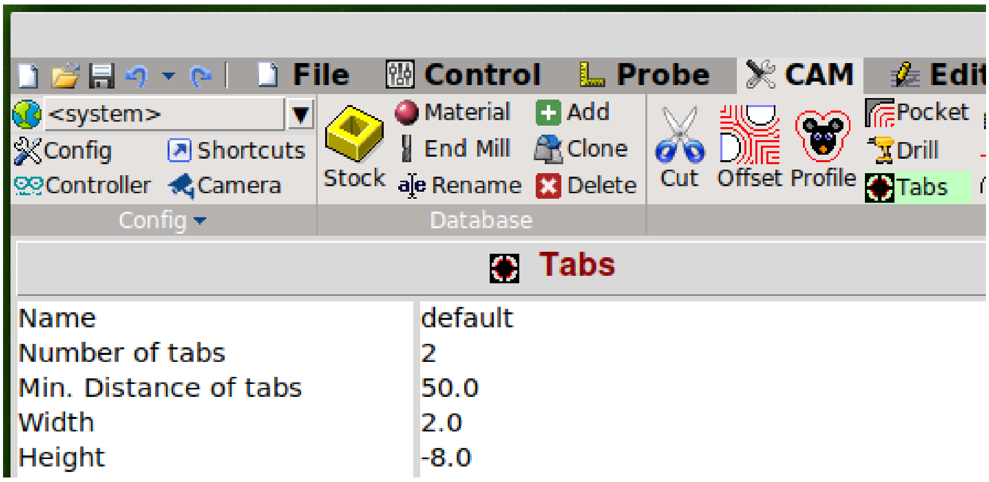
Profile

Name	default
End Mill	EndMill 02
Direction	outside
Additional offset distance	0.0
Overcut	□
Pocket	□

When clicking the big “Profile” button, you already see the profile contour on the surface.

Now it is a good idea to define holding tags (called Tabs in bCNC) that prevent the workpiece of flying through the room when it gets loose on the last cut.

Klick “Tabs” to the right of “Profile” and define the properties:



Tabs

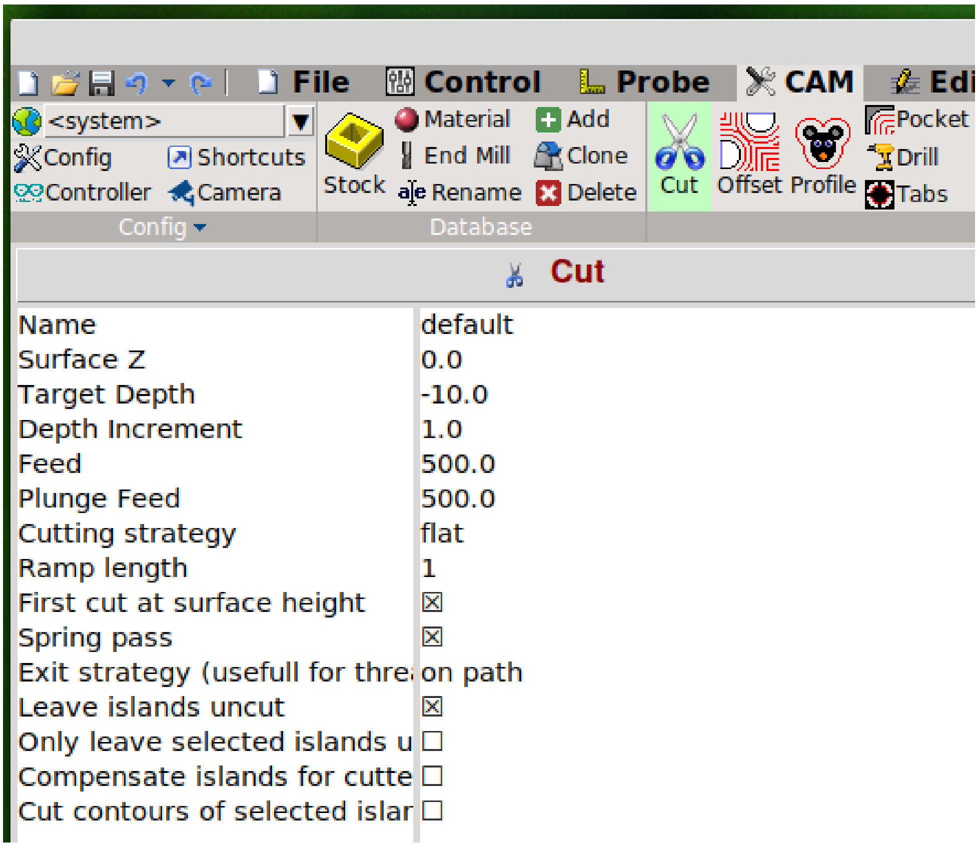
Name	default
Number of tabs	2
Min. Distance of tabs	50.0
Width	2.0
Height	-8.0

The Height property is the depth to which the path is raised by the holding tag, in our example with maximum depth 10mm and tag height = -8, the result is a tag with 2mm height from the bottom.

Klick on the big “Tabs button”

The tabs are displayed as grey circles, not easy to see.

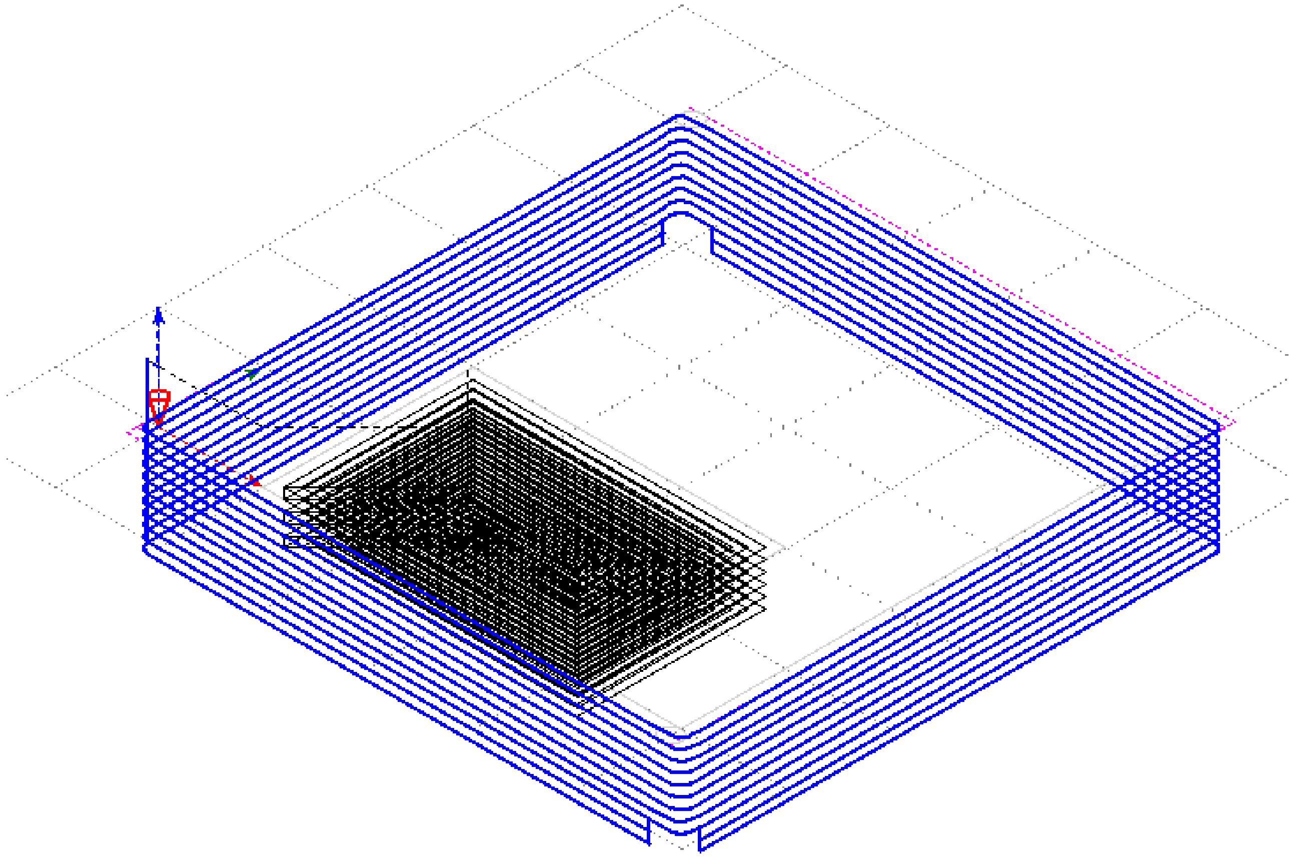
To do the final profile operation, klick “Cut” and define the parameters:



Cut

Name	default
Surface Z	0.0
Target Depth	-10.0
Depth Increment	1.0
Feed	500.0
Plunge Feed	500.0
Cutting strategy	flat
Ramp length	1
First cut at surface height	☒
Spring pass	☒
Exit strategy (usefull for thre	on path
Leave islands uncut	☒
Only leave selected islands u	□
Compensate islands for cutte	□
Cut contours of selected islar	□

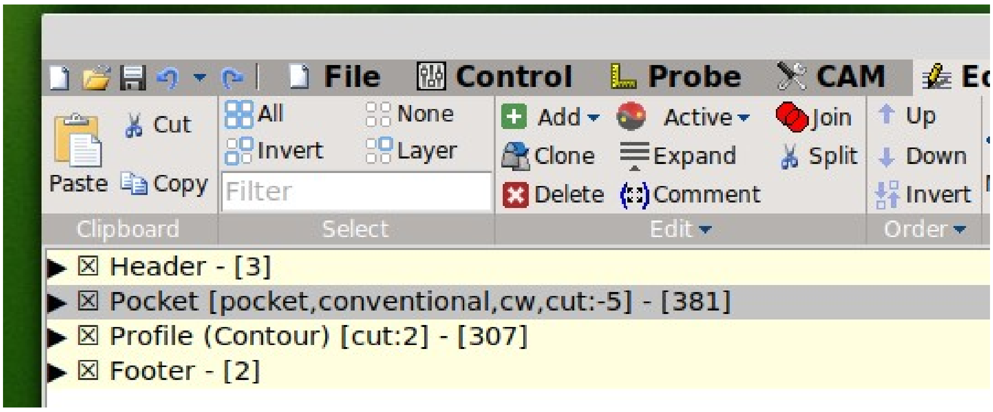
When you klick the big “Cut” button, the path can be seen:



Order the operations

The G code is executed in the order seen in the editor.

We have to process the pocket before the contour however. This can be done by selecting the code block and klicking the Up arrow:



▶ ☒ Header - [3]
▶ ☒ Pocket [pocket,conventional,cw,cut:-5] - [381]
▶ ☒ Profile (Contour) [cut:2] - [307]
▶ ☒ Footer - [2]

Milling with LinuxCNC

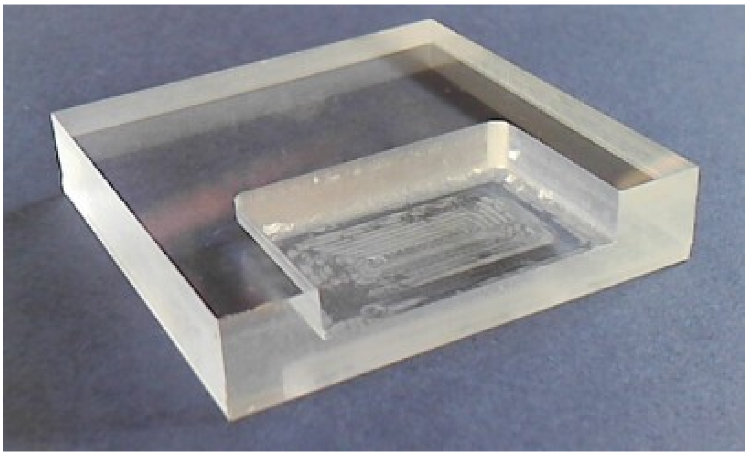
LinuxCNC had two complaints:

- Nested command in line …
  (caused by parenthesis in the block name [["Profile (Contour) [cut:2]")
- No program end

It was easy to solve these problems:

- Rename the problematic block
- Add “M2” at the end of the G code.

After this, the milling worked fine:



(This was done with a depth increment of 0.5mm and cooling by water spray, as acrylic has a very low melting point).