GrblGru - Operating Instructions

Version 3.44

toe@home

January 29, 2020

Contents

1 Before you start reading, you should note the following ...

There is certainly a lot more to improve on ***GrblGru*** . Also there might be be some hidden bug, and some things will only work if you do it in a specific order.
I just do it all at my free time because I enjoy programming. Please do not sting me if it does not work out. Let me know, and I'll try to fix it and make the next version a little better.

Ideas, suggestions for improvement and also criticism are always welcome.

Have fun with ***GrblGru***

2 What is *GrblGru* ?

2.1 The name

Grbl is the name of a free great open source software that can interpret G-code and use it to convert pulse and direction information to control the stepper motor power amps. Because ***GrblGru*** heads to GrB in control mode, the name ***GrblGru*** is the logical continuation of the 'stupid' name Grbl. No-one really knows what it means. The pronunciation is also not clearly defined. I prefer 'GrabbelGru' and because I describe myself often enough, I use the written short form a simple 'GG'.

2.2 The Basic functions

GrblGru is ...

- a 3D simulator for milling and lathe
- a GCode Sender forr **Grbl**, **TinyG**, **Mega-5X** and **g2core**
- a CAM program for milling and lathe work
- a Tool for learning CNC technology without needing a 'real' 'Machine

> I have noticed only recently what immense advantages the use of virtual simulations for training purposes offers. It would mean a great deal to me if the program could be used, for example, to bring this great hobby closer to young people. If any changes or adjustments are required, I am willing to do so.

2.3 The Operating Modes

GrblGru has 2 different Operating Modes:

- **Simulation Mode**
  for Virtual 3D simulation of CNC machine control
- **Control Mode**
  for Real control of a CNC machine with **Grbl**, **TinyG**, **Mega-5X** oder **g2core** operating system.

2.4 Simulation Operation

This operating mode is the original idea of the program. It was primarily intended to check existing NC programs on a virtual machine and to show any collisions or problems.
This type of early 'virtual machine' has prevailed in the industry more and more in recent years because it saves valuable machine time detecting possible errors in advance.

When ***GrblGru*** is started, the mode by default is set to the 'Simulation'. This gives the following situation.

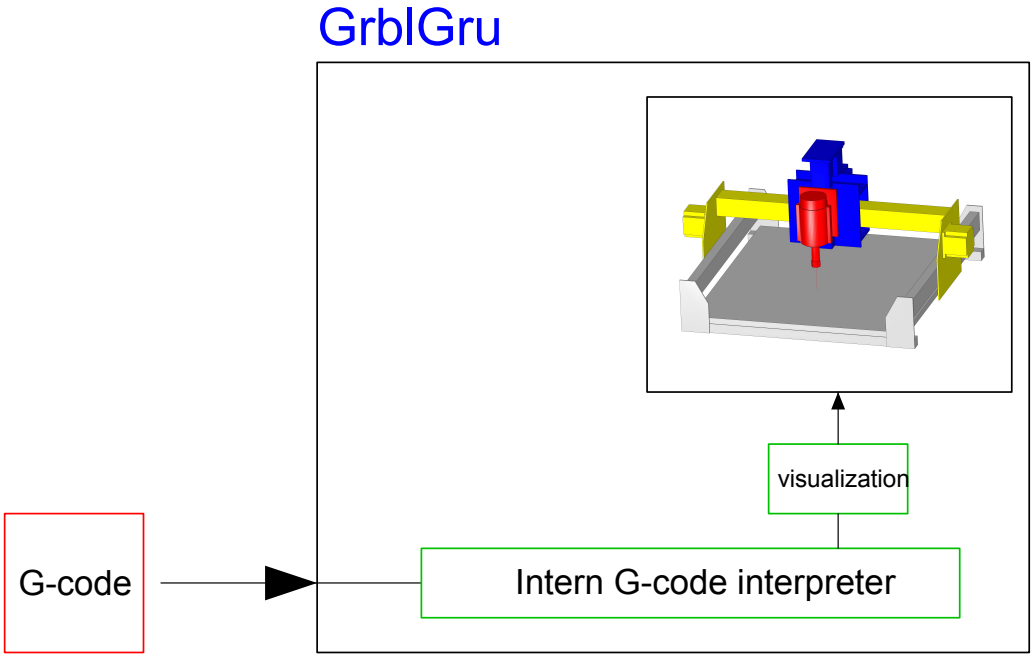


Figure 1: ***GrblGru*** in Simulation mode

There is no connection to a real controller. The NC program to be examined is loaded and the NC commands are processed by the internal NC interpreter. The loaded 3D model is animated accordingly.
NC = Numerical Control.

In the appendix you will find the currently available 3D models of the milling machines and lathe tools. So if someone is a little familiar with 3D and interested in visualizing his home-made machine, I'm happy to help..

A special feature of the program, however, is that the user can also import their own 3D models. As a result, it is possible to represent every question or lathe as a simple model in ***GrblGru*** .

2.5 Control Operation

In this mode, ***GrblGru*** can connect to an Arduino UNO, a TinyG, an Arduino Mega 2560, or an Arduino Due controller. The connection is made via the USB interface, which transfers both the NC commands to the controller and the information about the current axis positions back to ***GrblGru*** . ***GrblGru*** uses this information to control the 3D model so that the 3D model is synchronized to the 'real' axes.

> ***GrblGru*** automatically supplies the respective processor **not** with the necessary operating software. The user must therefore ensure that the processor is flashed once with the appropriate operating software beforehand.
> This has the advantage that it is usually possible to update the processor at any time without ***GrblGru*** having to change it.

The main difference of these four controllers lies in the number of possible axes. For example, TinyG can also control a Rotary axis, often called the Fourth axis, in addition to the three linear axes. In addition, 2.5 A drivers are already mounted on this circuit board, to which you can connect your stepper motors. For larger motors, however, the signals for controlling external drivers are also available.

At present the control of the G2core software on an Arduino DUE is still under construction. This system provides 6 axes (linear and rotary), making it the ideal control for my 5-axis dreammachine that I'll have at some point. :) The Arduino Uno can only control 3 Axes, generally XY and Z. Some people will repurpose one of the axes to control an accessory rotary axis but it is not a true 4th axis since the repurposed linear axis is rendered nonfunctional.

An imaginative software designer has made a special GRBL Modification specifically for the Arduino Mega 2560 board (with or without a RAMPS board) which can provide control of 5 axes (and possibly a 6th eventually). This is called Mega-5X. The RAMPS board has places for small stepper drivers to mount so the motors can be directly connected. If you use the Mega 2560 alone, there is a MEGA-5X Pinout available to connect this board to External Stepper Drivers.

When ***GrblGru*** is started, use the Controller option to select Simulation, **Grbl**, **TinyG**, **G2core**, or **GRBL-Mega-5x**. This will give you one of the following situations.

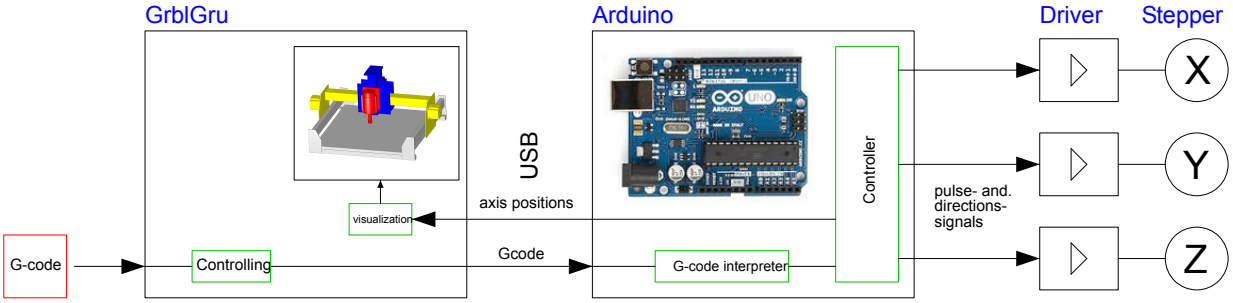


Figure 2: ***GrblGru*** in control mode with Arduino UNO (3-axis)

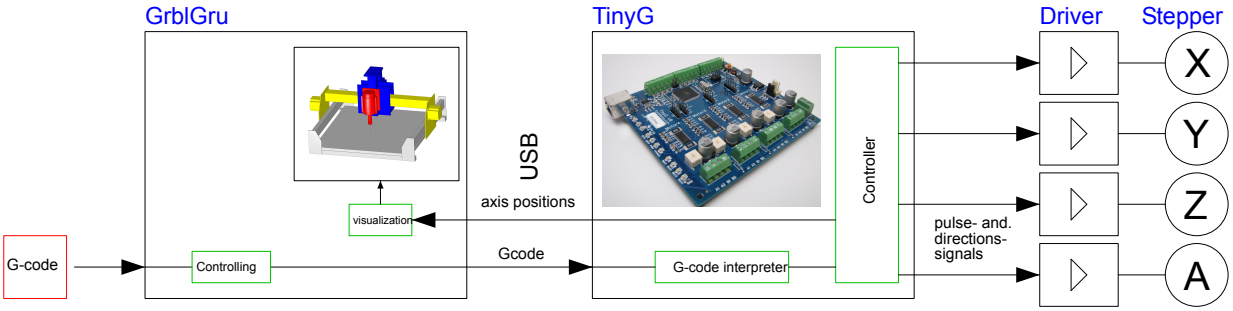


Figure 3: ***GrblGru*** in control mode with Arduino DUE (5-axis)

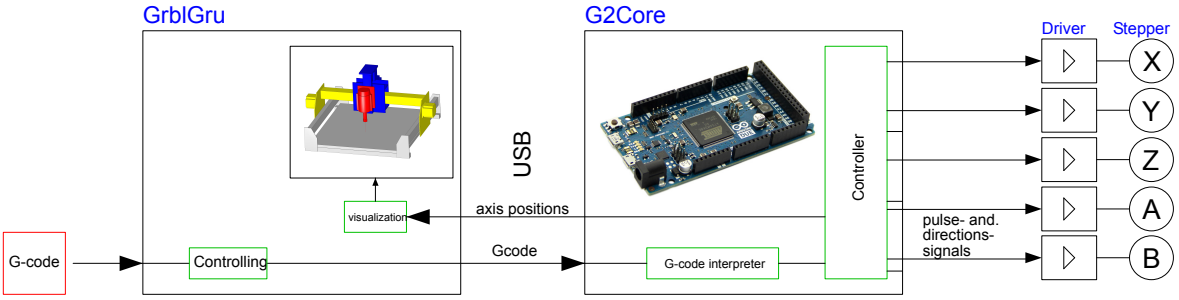


Figure 4: ***GrblGru*** in control mode with Arduino Mega-5X or Arduino DUE (5-Axis)

Additional information about Grbl and Co::

- **Grbl**
  https://github.com/grbl/grbl/wiki
- **TinyG**
  https://github.com/synthetos/TinyG/wiki
- **g2core**
  https://github.com/synthetos/g2/wiki/What-is-g2core
- **Mega-5X**
  https://github.com/fra589/grbl-Mega-5X

3 Download and Installation

3.1 Hardware and Software Requirements

A few words about computer requirements..

GrblGru runs on a PC with a Windows operating system from Win XP up to Windows 10. As a basic component, Windows Framework 4.0 is required. Check that it has been installed and if not, make sure to get it before trying ***GrblGru*** .

A faster computer is particularly advantageous in graphics activities and will be a bit more fun. My personal preference consists of a fast desktop computer for developing and simulating the models, and an old computer in the basement to control the CNC machine.

3.2 Where can I get the program from ?

Answer. From my new website:

http://GrblGru.com

The site is currently still under construction. At the moment you can only download the released (release) and the current test version (beta).

In the release version, there should no longer be any major errors. In contrast, I use the beta version to make my current status available to some 'brave' testers who have volunteered themselves.
More information will be made available as time allows.

3.3 Installation

The setup is done by simply starting the installer file. The installation time is only a few seconds.

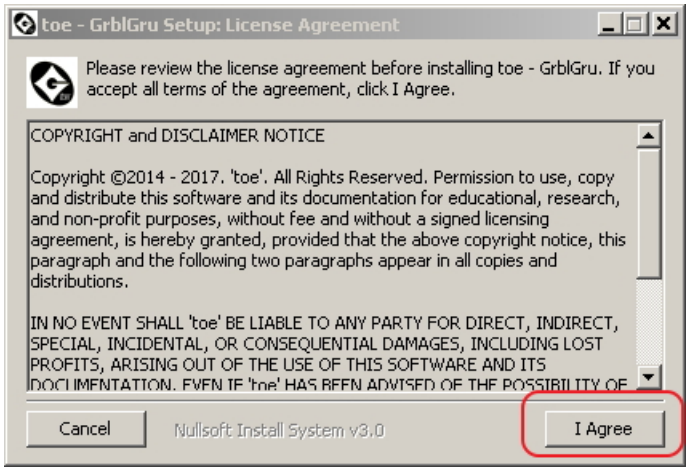


Figure 5: The dialog at starting the setup. The disclaimer must be confirmed.

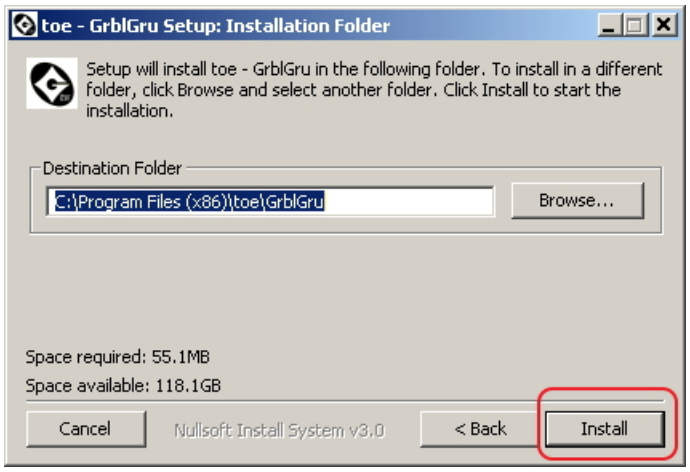


Figure 6: Query the destination folder. I recommend keeping the suggested path

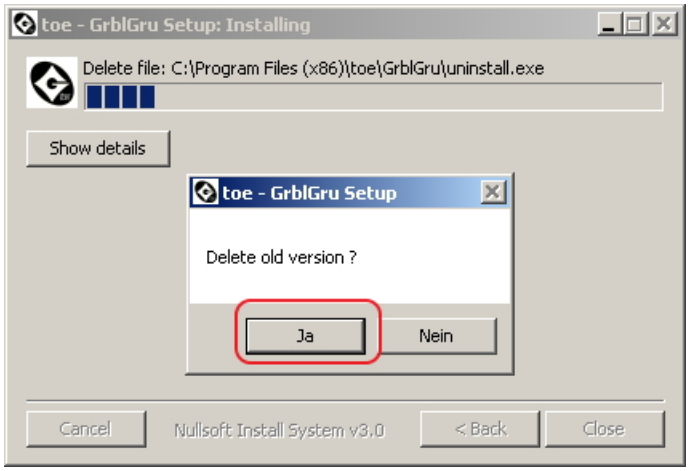


Figure 7: Answering this question with 'Yes' deletes the working directory under ProgramData before installation, which is equivalent to a basic installation. For users who are not knowledgeable, this is recommended.

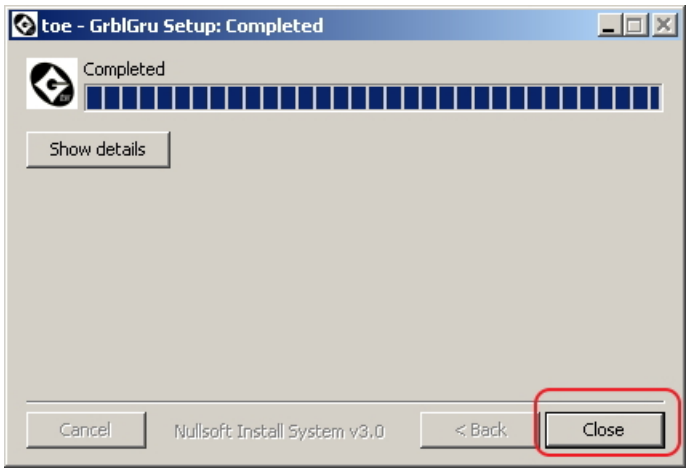


Figure 8: The display after successful setup

The installer will automatically create a desktop shortcut to launch ***GrblGru*** .

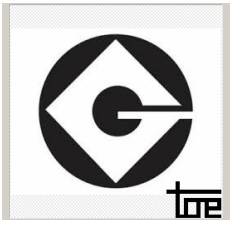


Figure 9: ***GrblGru*** shortcut

4 Projects to Try

I would not like, as in most manuals, to start explaining each button of the program individually. That would certainly be a lot of boring stuff that no one would read.

So I suggest that I introduce you to some small projects that are self-contained. If you understand the mindset of a program, you can often help in unknown situations. I hope that I manage to show you how ***GrblGru*** 'ticks'. Using ***GrblGru*** with guidance, rather than just reading, can lead to faster learning.'.

In chapter 4 I would also like to introduce more details on certain topics.

4.1 Project 1, Cut out the Moose

In this first simple project, I would like to show how to cut a small figure, in this case a moose (elk, elch), out of a 3mm thick plywood board. So let's go!

Figure 10: The moose or elch (the MDF variety)

4.1.1 Load Geometry

First of all, we need the geometrical data of the model. This is obtained from a DXF or SVG file. Under `C:\ProgramData\GrblGru\ExampleData` you will find a lot of example files, including the file Elch.dxf. To load this file we go to the Menu and under the item File - Import DXF, find and select the file Elch.dxf in the displayed window. [1]

And there it is, the first big surprise. The Moose is too big! Do not worry, our geometry can easily be scaled. For this we activate the button '2D Scaling' on the toolbar and set a factor of Fx = 0.3. If the box is checked to the right, the factor Fy is automatically set to 0.3 as well. Close that Window.

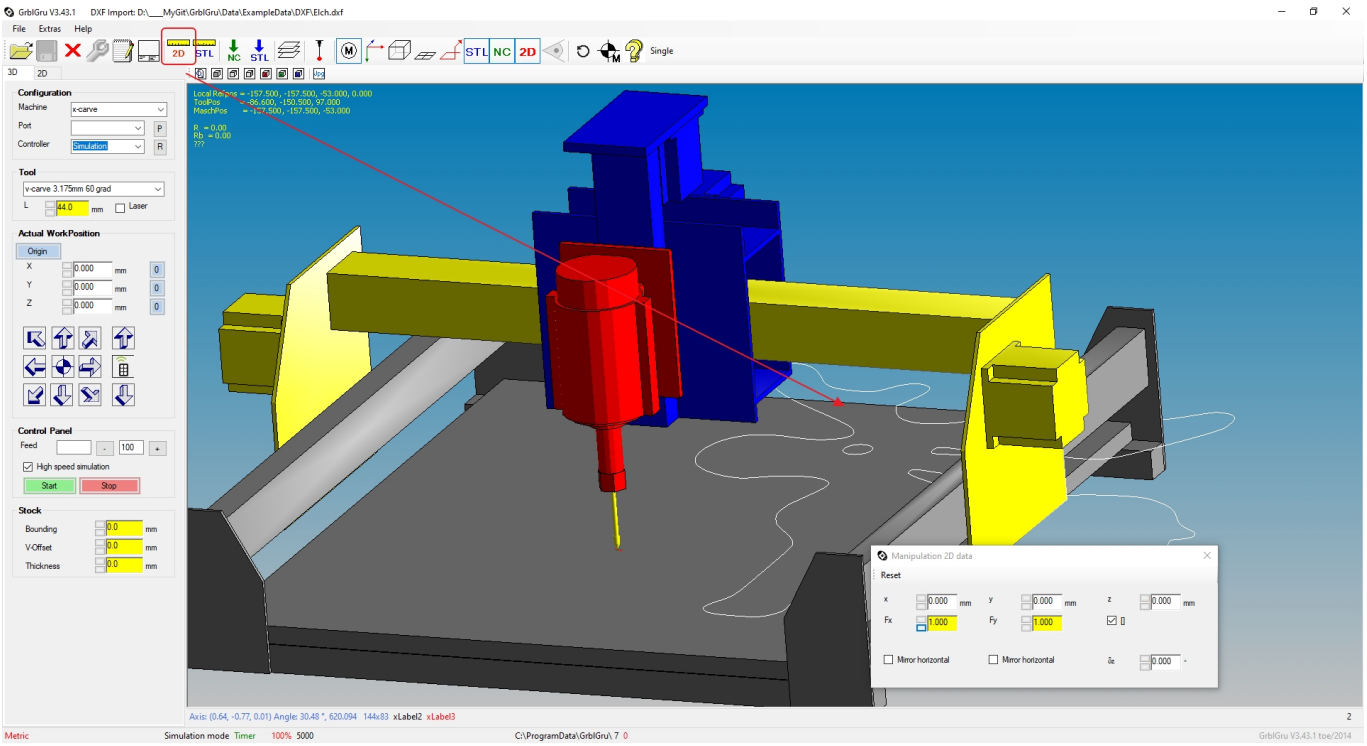


Figure 11: The display immediately after loading - Big Moose surprise

[1] See also chapter 'Loading geometry data'

4.1.2 Jobs and Job Templates

Next, ***GrblGru*** has to know what it should do with the geometry data. For example, which tool should be used, what depth to cut, which feeds and speeds should be used, etc. I call these things a Job in ***GrblGru*** . As you can imagine, there are some data types that you have to re-enter each time. To simplify this data input there are Job Templates that can be created once and then used repeatedly by simply loading them. In this exercise we will create a Template with the fitting name 'Cut 3mm plywood', which we can use whenever we tackle similar projects.

It is particularly easy to create a Template if you already have similar templates. During installation some startup templates are already included which we can easily change. For this we change to the 2D-view, Press the 2D Button (top left of screen) and take a closer look at the tree display on the left side. We see the 5 main nodes:

- Tools
- DXF Templates
- Job Templates
- Drawings
- Jobs

With a click of the left mouse button on the small plus sign in front of the respective node name, the tree is opened. Opening the 'Job Template' node, we get the following view:

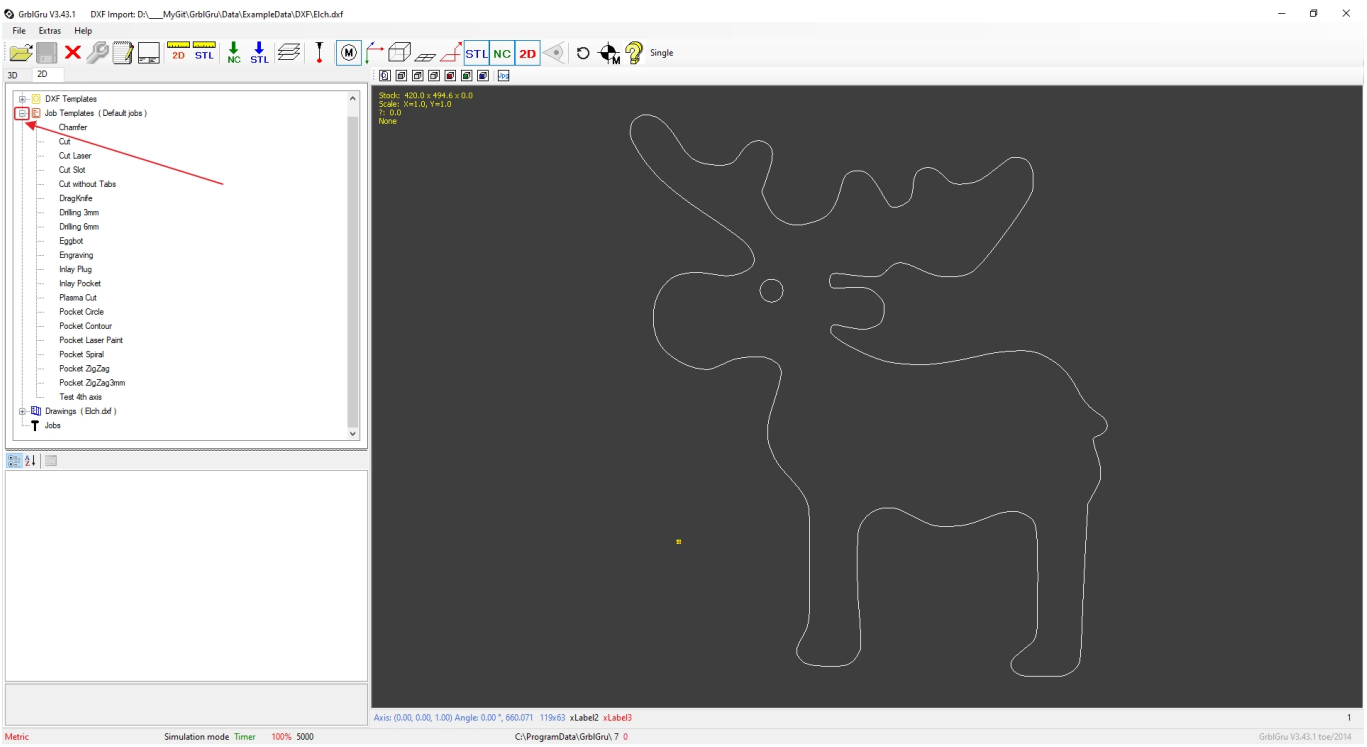


Figure 12: The Tree display in the 2D view

Next, let's use a copy of the 'Cut without tabs' template to create our template. For this we click with the Right mouse button on the 'Cut without tabs' template and select in the Context Menu 'Copy'. At the bottom of the list a New Template appears "Copy from: Cut without Tabs". Click with the left mouse button to look at the content in the properties window.

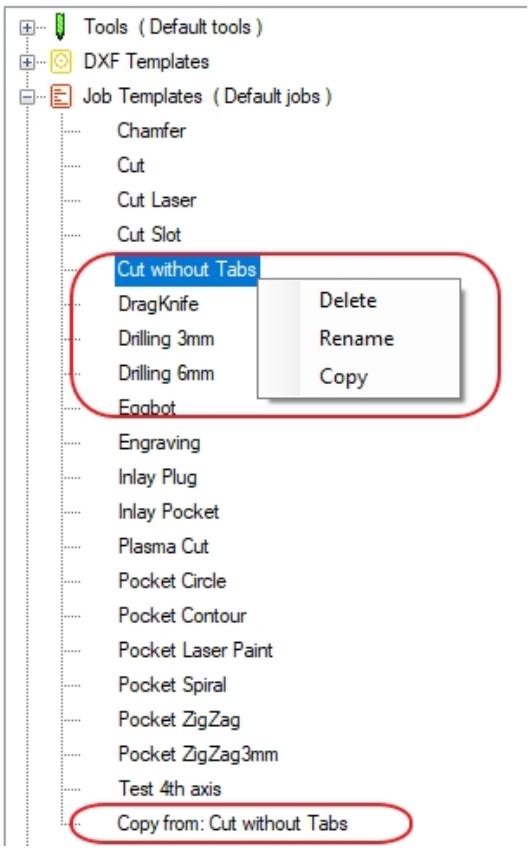


Figure 13: The display after successfully making a new Template.

Also accessible via the context menu is the function 'Rename', which we can use to give the template a descriptive name. In this case, maybe 'Cut 3mm Plywood'

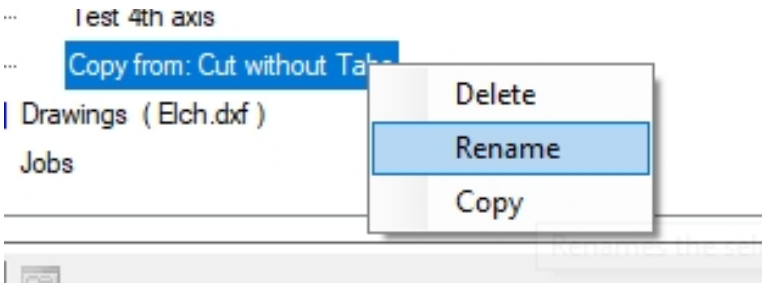


Figure 14: Rename the Template

Let's have a look at the properties window in the bottom part of the dialog. Here I would like to focus on the most important values for the moment. For the sake of simplicity, close the chapters 'General', 'Type of Tool' and 'Special' by clicking on the small arrow 'v' to the left of each heading.

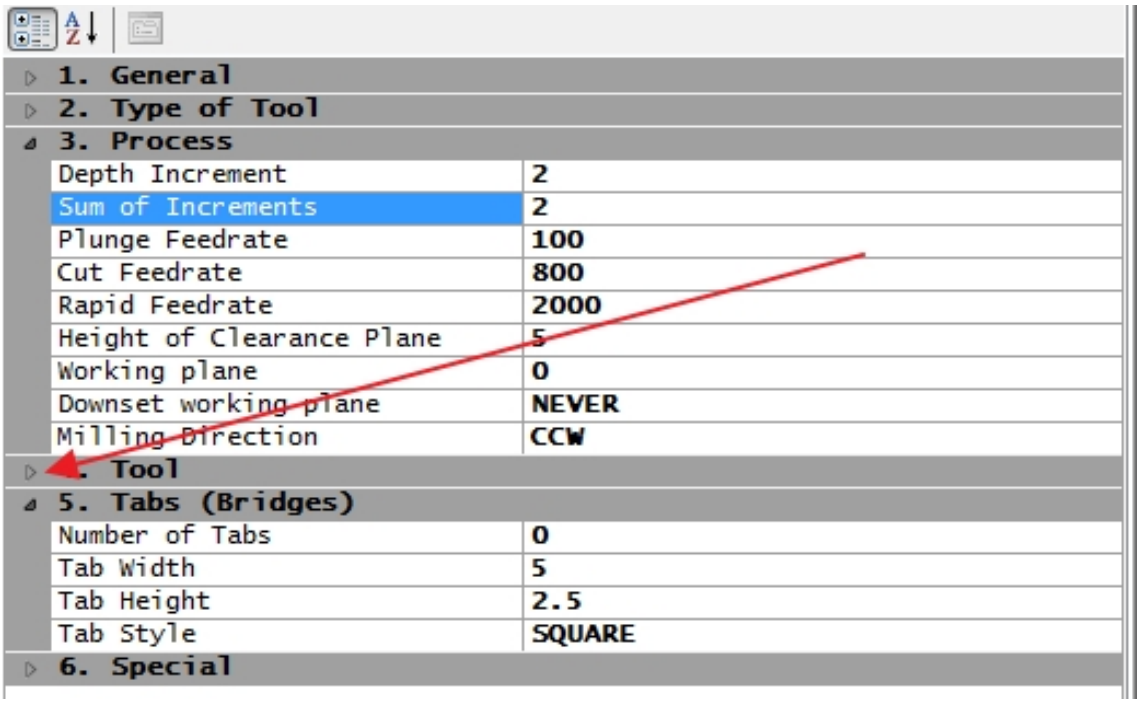


Figure 15: The Properties window

The following numerical values are not real values. They were arbitrarily chosen by me to represent certain facts. In particular, these feed values are a little imaginative and one finds everywhere a lot of (also different) opinions, recommendations and also concrete calculations.

- **Depth of Increment**
  Here we enter the depth that we allow the bit per pass. For example, enter 1.2mm. This Increment ensures that any differences in the height of the plywood and other small inaccuracies do not lead to the bit not completely cutting through the material at all points. (So that I do not 'shred' my spoil board so much, I often put a piece of cardboard under the workpiece).

- **Sum of Increments**
  Since we want to cut 3mm plywood, we enter as Sum of Increments 3.2mm. ***GrblGru*** calculates 3.2mm infeed total divided by 1.2mm infeed depth = two passes with 1.2mm infeed and a third pass with the rest, so 3.2mm - 2.4mm = 0.8mm

- **Plunge Feedrate**
  is the speed at which the cutting tool lowers into the work material.

- **Cut Feedrate**
  is the speed at which the tool forms the contour of cut.

- **Rapid Feedrate**
  is the Speed with which the tool travels outside the material in the 'safe' clearance plane.

- **Clearance Plane**
  is the Safe Height Above the workpiece to be cut. The cutter lifts off to the top and it safely moves from one contour to the next. Any movement at this level must be collision free!

- **Cut direction**
  Here you can choose whether to work in the conventional or in climb direction with the cutter. CCW or CW

- **Bit Radius Compensation**
  is choice whether the tool path should be corrected to follow the INSIDE, OUTSIDE or ON THE LINE (none) of the contour cut. This makes it possible to produce the same parts, even with different cutter diameters.

- **Type of Radius Compensation** the Tool type has an influence on the way the calculated contour is equidistant from the cut path to compensate for the different tool types. Choice of ROUND, SQUARE and NORMAL.

- **Tool Identification**
  The type of Cutting Tool to be used is selected here. I will go into that in the next chapter.

- **Number of Tabs** tabs are small connections between the cut out model and the base material. These prevent movement of the incompletely cut out part. They must be removed by hand with a sharp knife or small saw after completing the cutting process. They should therefore be as small as possible and be in easily accessible locations. For this exercise enter 3 Tabs.

- **Tab Width** eg 5mm

- **Tab Height** eg 2.5mm

- **Tab Style** eg SQUARE, ROUND, or TRIANGLE

4.1.3 The Tool Selection

As we have just seen, in a job you choose a tool name from a given list.

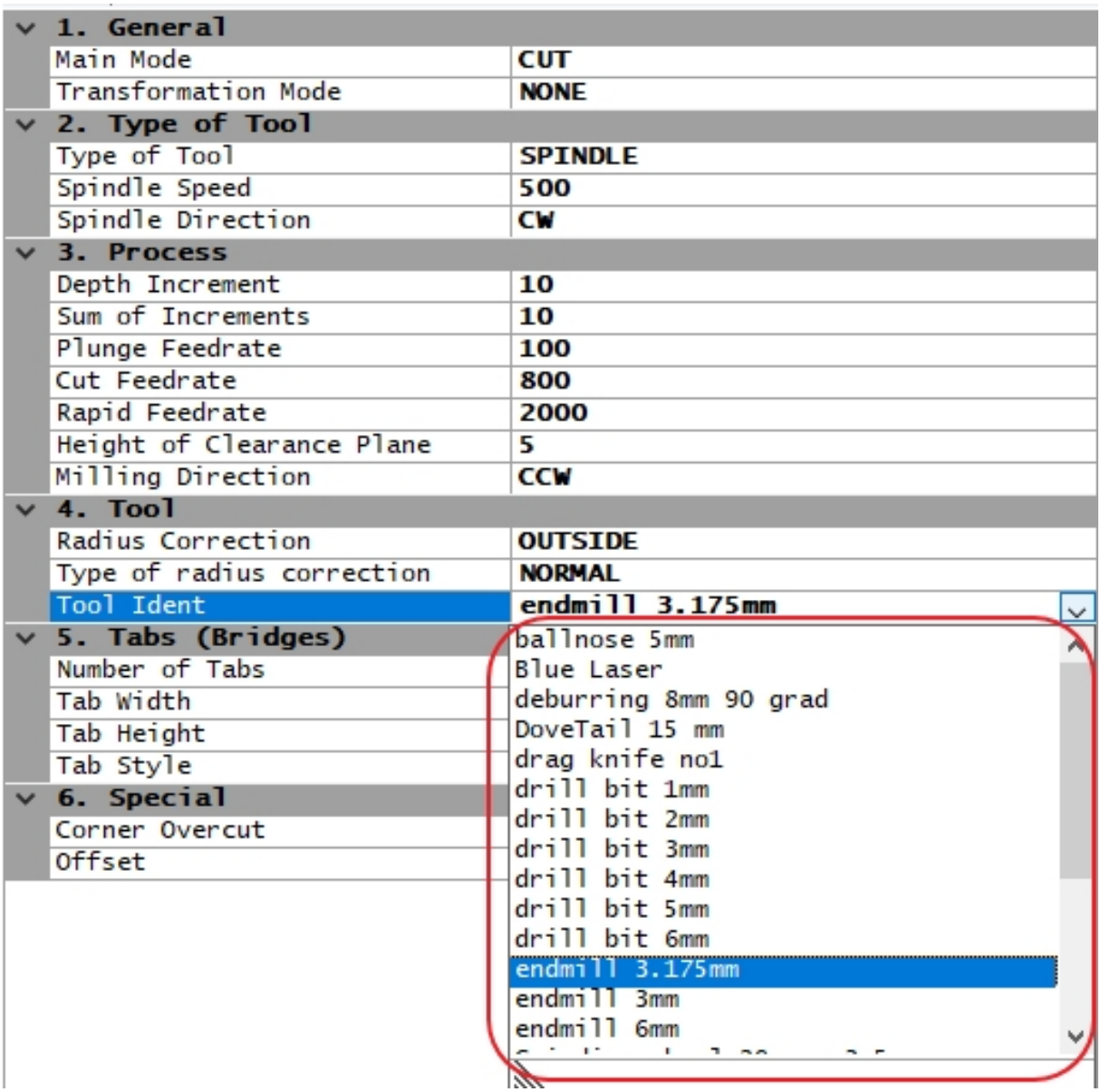


Figure 16: Tool selection list - Tools (Default tools)

The basic idea is to save all the geometric data of the tool under the tool name. The Tool Name contains the information about which tool type, tool diameter, cutting angle, etc.

GrblGru created the node 'Tools' in the tree view and added some tools as examples. In the properties window, a small sketch and a photo are displayed for each tool. Of course, you can also add your own tools, Copy, Rename, Save and Load. This is similar to what we've done in Job Templates. A click with the Right mouse button opens a context menu in which various actions can be selected.

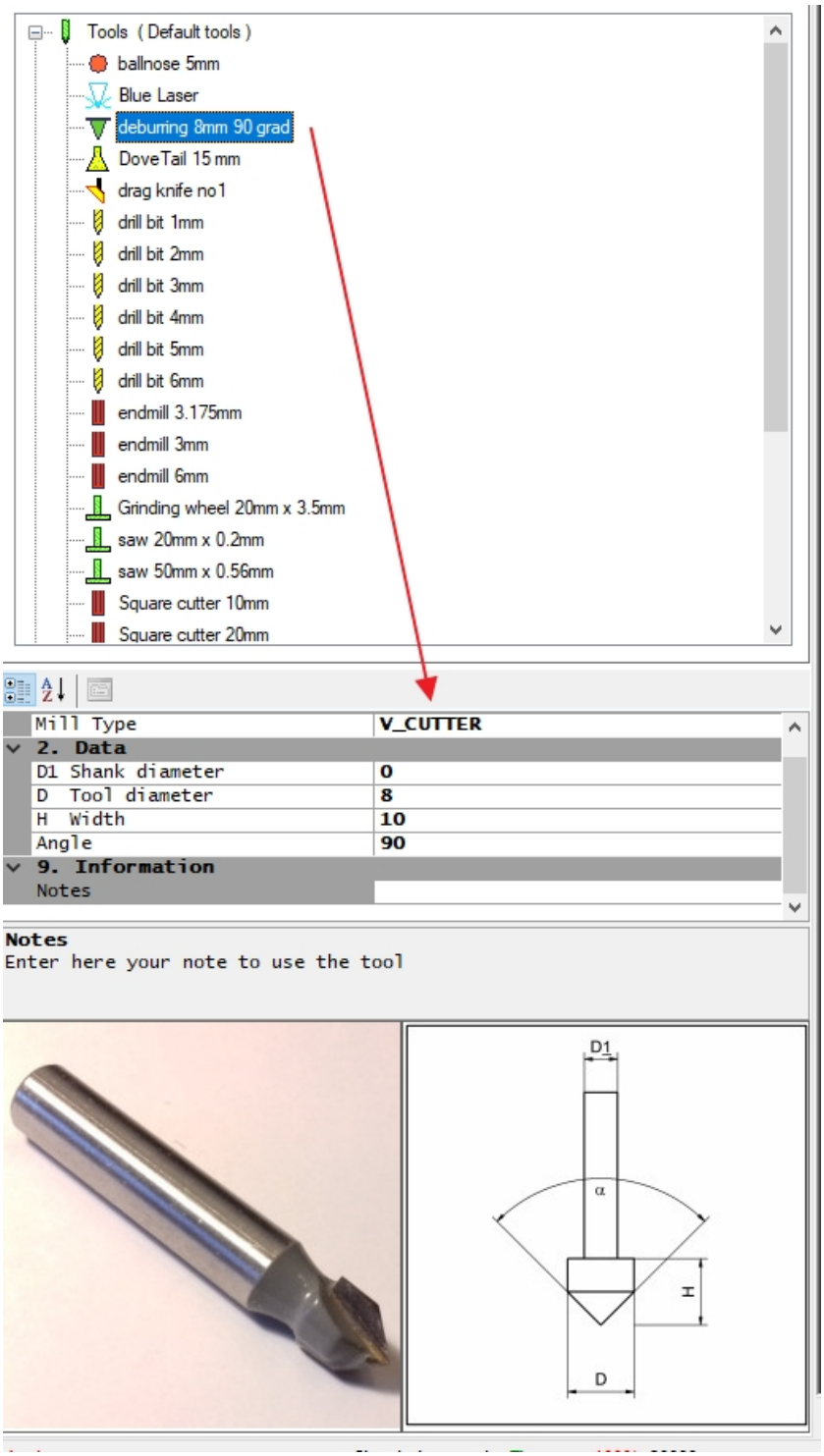


Figure 17: Existing tools in tree view

4.1.4 Elements of the 2D view

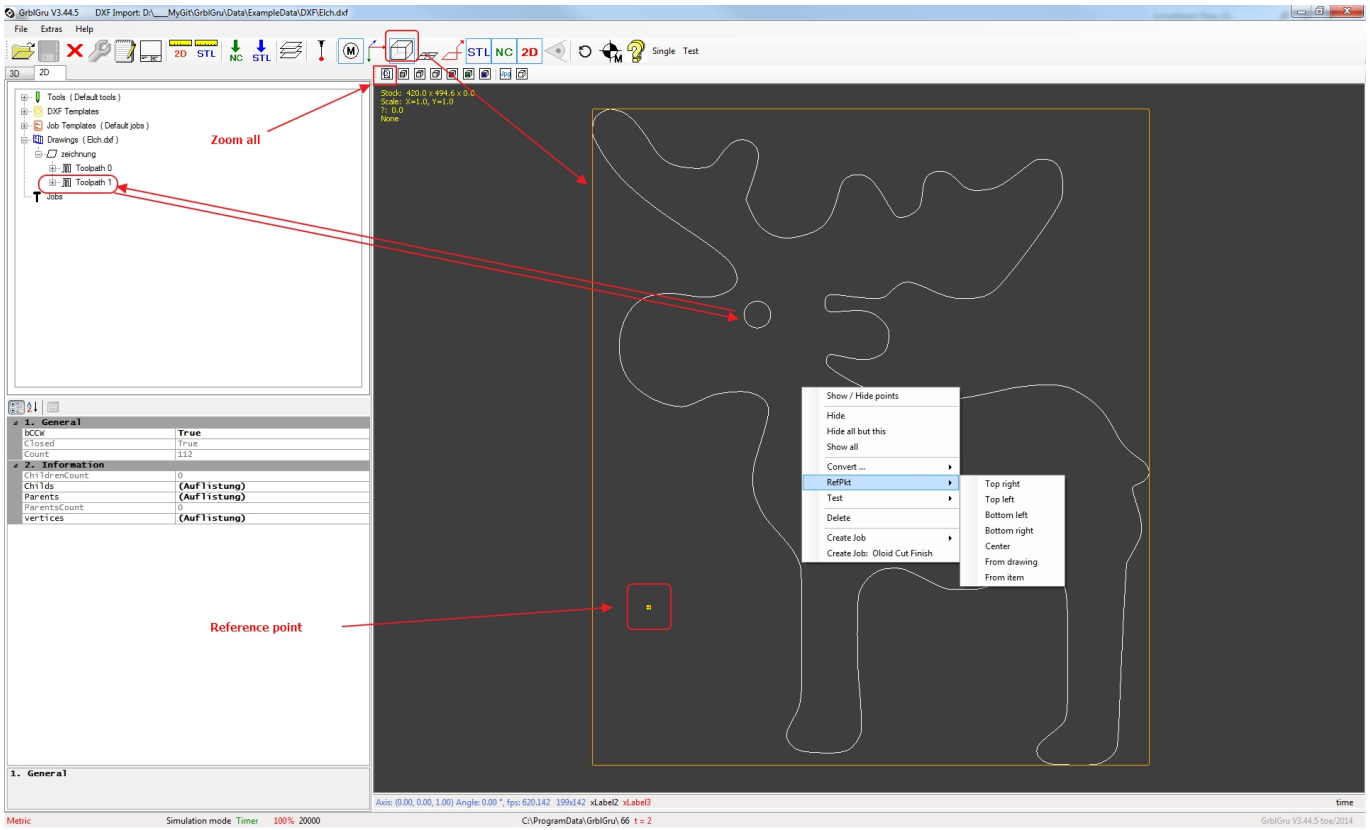


Figure 18: Elements of the 2D view

Let's take a quick look at what's going on in the 2D view. First in the tree view on the left choose Drawings and expand it. We can see that ***GrblGru*** detected 2 tool paths from the DXF file. When clicking on one of the tool paths, the respective path is marked Red in the graphics window. A click + ALT key on one of the lines in the graphics window reversely marks the respective path in the tree view..

The  button in the toolbar shows the dimensions of the drawing (brown rectangle). (If not visible click the Stock Box button on the top ribbon panel). These are also displayed at the top left (126.0 x 148.4 x 0.0). You determine the minimum amount that your work piece should have. The scaling is also visible here X=0.3, Y=0.3..

Another important point is the location of the Reference Point (Red+Green Cross). This is the position where the cutter begins its trajectory. It can be selected via the "context menu ? that opens with a click of the right mouse button. It can be moved under RefPkt tab eg Top Right, Center, etc.

Also very important is the  button on the far left of the small ribbon toolbar above the graphics window. It puts the graphic in screen-filling view. So if you can not see anything, this button is lways the first choice. This also applies to the 3D window. The other buttons show other views.

4.1.5 Assign a Job to the Geometry

Ok, now that we've created a Job Template we need to create a real Job by combining a Geometry to a Job Template. To do this, we click on the outline of the moose with the left mouse and press the ALT key. As a sign that it has now been selected, the outline is now highlighted in red.

A click with the right mouse button opens a Context menu in which we can select one of the existing Job Templates. Select Create Job, then Cut 3mm Plywood template with Left mouse clicks..

During this process, a Job is assigned to the selected Geometry Path. ***GrblGru*** now has all the information on how to edit this path.

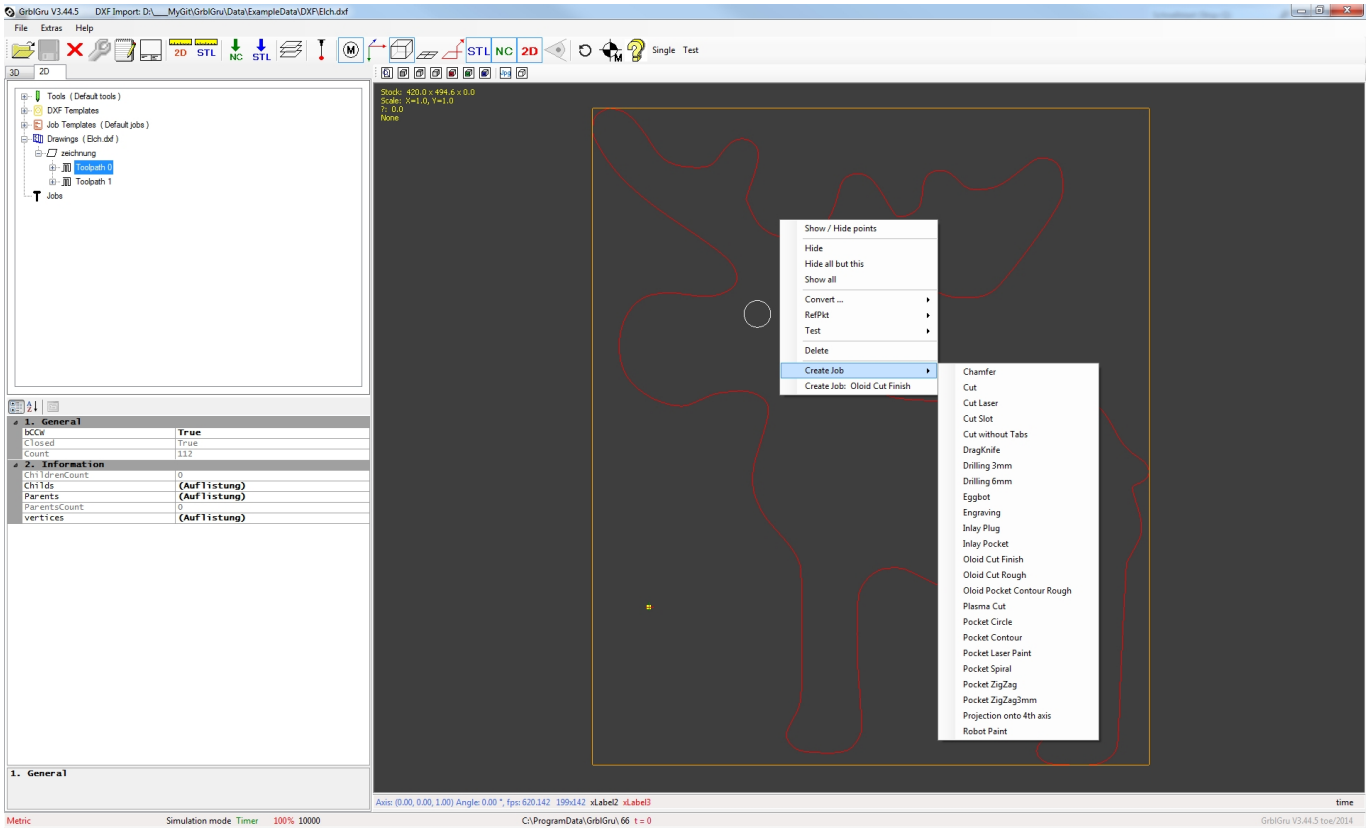


Figure 19: Create a Job

The result is a New entry in the Tree view under the 'Jobs' node called Job Toolpath 0. In addition, the graphic shows the Path of the Tool (green outline) and the Positions of the Tabs (brown rectangles) and Starting Point (blue triangle).

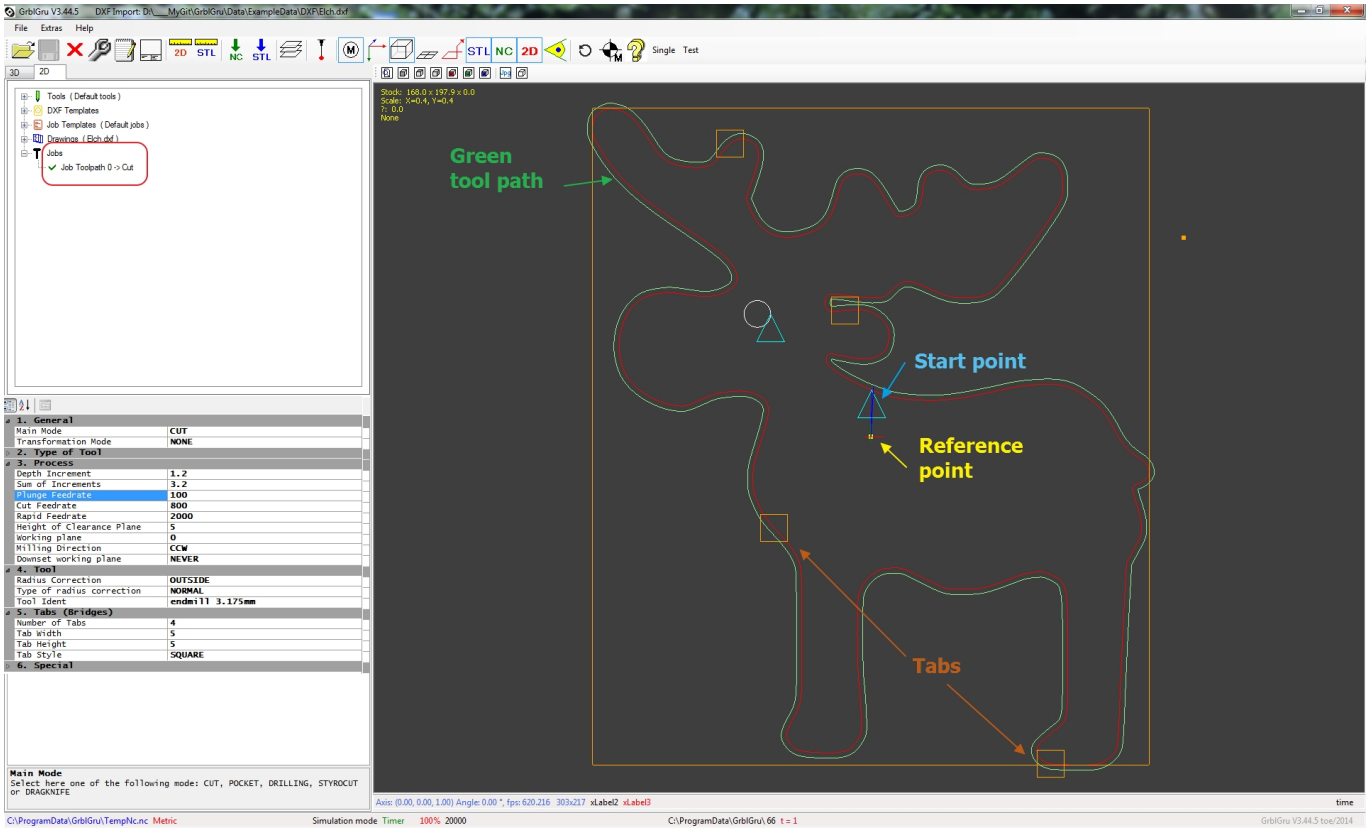


Figure 20: After creating the job

4.1.6 Starting points and Tabs position

Tabs and Starting points are arbitrarily positioned for irregular geometries. You can easily Move Tabs to any other position by pressing the left mouse button, hold down, dragging and releasing. It makes sense to choose places that you can easily reach later to remove the bars with a knife, a saw or a file. Of course, they should be distributed in such a way that they give the part the necessary support. Try moving the Tabs around now.

4.1.7 The Second Job

Now that we have successfully created the first Job, we can now tackle the next Job. To do this, we click on the Eye of the moose with 'left mouse button + ALT key' and then create a Job again with Right Click, Create Job, and use the Job Template 'Cut 3mm Plywood'. In the tree view, this second Job Toolpath appears under the Job node. Notice the light Blue triangle Start Point. Left Click on the new Job Toolpath 1.
?
In the Property window we must change the parameter 'Cutter radius correction' to INSIDE, so that the cutter only cuts the Inside of the eye. We don't need Tabs in this case, so we can change 'number of Tabs' to 0.

By clicking with the right mouse button on one of the Job Toolpaths, we are provided with a Context menu, with which we can set individual Jobs Active or Passive (Inactive). The Cutting Sequence of individual Jobs can also be changed with the menu items 'Higher' or 'Lower'.

For example, it is advisable to first carry out all work inside the part to be cut out first and only to cut out the outside part at the end. This avoids the tabs having to absorb the cutting forces of the 'interior work'. Try selecting Job Toolpath 1 (the Eye), Right Click, select Up to do this Job first.

The Toolpath generated is now displayed as Green with the non cutting tool movements in dark Blue.

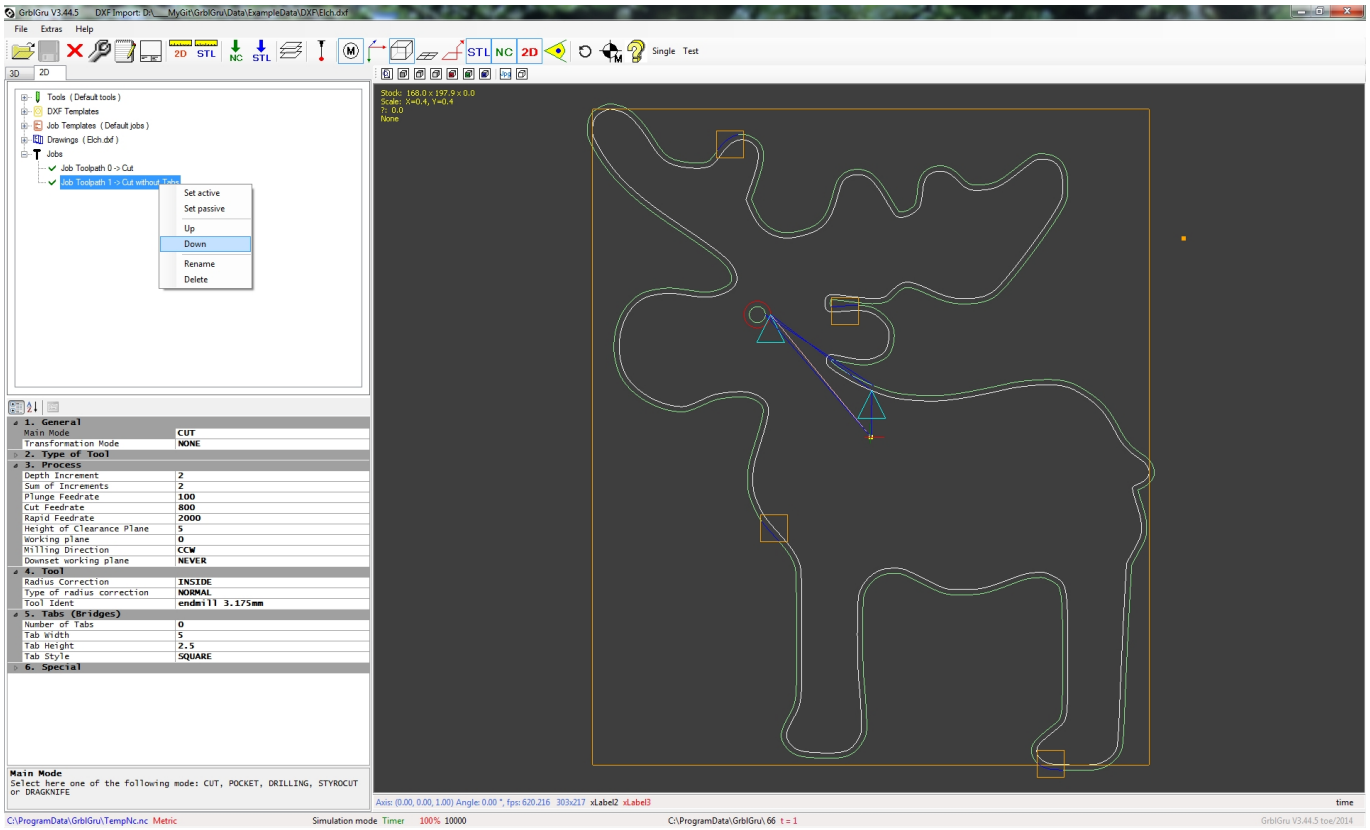


Figure 21: Moose with corrected tabs and created jobs

4.1.8 Dimensions of the component

At the top left of the 2D work area is described the dimensions of our component in the raw state. In our example that is the 3mm plywood board whose required Minimum dimensions were displayed in the 2D view at the top left (126.0 x 148.4). The 2D work is finished now..

Now return to the 3D view and take a look at the 'Stock' box with its 3 parameters. Click the 3D button.

Since it makes sense to make the base material a little bit larger for example, Bounding Box is Increased with the parameter **'Bounding'** enlarging corresponding amount on all 4 sides.

The parameter **'V Offset'** can be used to enter an allowance in Z-direction, eg . the height of a holding vice, a pad of cardboard, or equivalent.

Finally, the parameter **'Thickness'**, whose name indicates the material thickness. With the Stock Box button  in the toolbar the stock material can be shown or hidden.

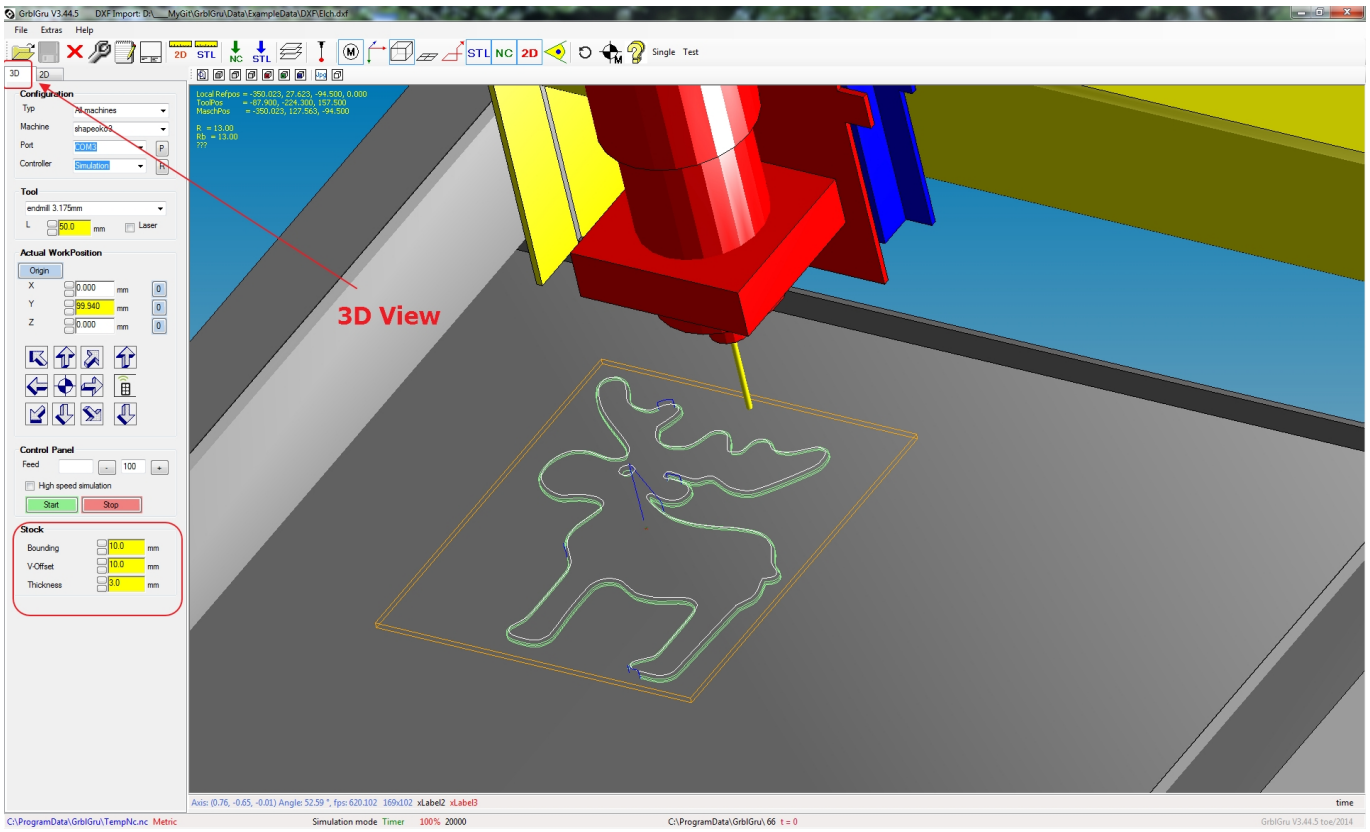
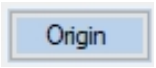


Figure 22: Display of the workpiece in the 3D view

4.1.9 Start Position Marker

Since I do not use limit switches on my machines I can only introduce my way of doing things here. The procedure using a homing+limit switch reference is correspondingly different..

As a rule, I work in the front left corner of my machine. So I fix my work piece there and position my XY axes so that the cutter tip is located just above my planned starting point (bottom left of the work piece). Then I carefully lower the Z-axis until the cutter just touches the component. Often I simply turn off the supply voltage of the motors briefly and turn the Z-axis by hand a little lower. A sheet of paper (about 1 / 10mm) serves as a good distance check between cutter and work piece. A Touch Plate could also be used..

Next I press the '**Origin**' button. 

This completes the process both in the graphic and the work piece. The reference point of the virtual loaded geometry (and thus also the workpiece) is relocated to the tool tip and the Z-axis is positioned so that the lowest point of the cutter touches the top of the workpiece..

This is why I always set the Reference Point of the geometry to 'lower left'. As a result I have the maximum positive X and Y distance available.

Historical Note:
GRBL comes from the early days of CNC machining. The default Homing point is at the Rear Right with the Z axis all the way up. This places the entire work volume in Negative Space. This makes a kind of sense since milling Removes material from the top down - a Subtractive Process.

When 3D Printing came along much later the homing point was set at the Bottom Front Left with the Z axis all the way down. This places the work volume all in Positive Space. The printer Adds material from the bottom up - an Additive Process.

It is simpler to just position the origin described at the top of this page. No Homing needed, no limit switches and no confusion.

4.1.10 The Tool Path

To control the individual tool paths more accurately you can click the button ‘Show Machine’  to turn the machine image off. In order to increase the contrast, the background colour is also changed.

The Green line showing the Tool Path is switched on / off with the NC button.  This represents the path of the tool. In combination with the SHIFT key the movements on the Safety Plane can also be made visible. The Tabs positions are shown as small notches in the Green Line.

The display of the White Geometry lines is controlled by the 2D button. 

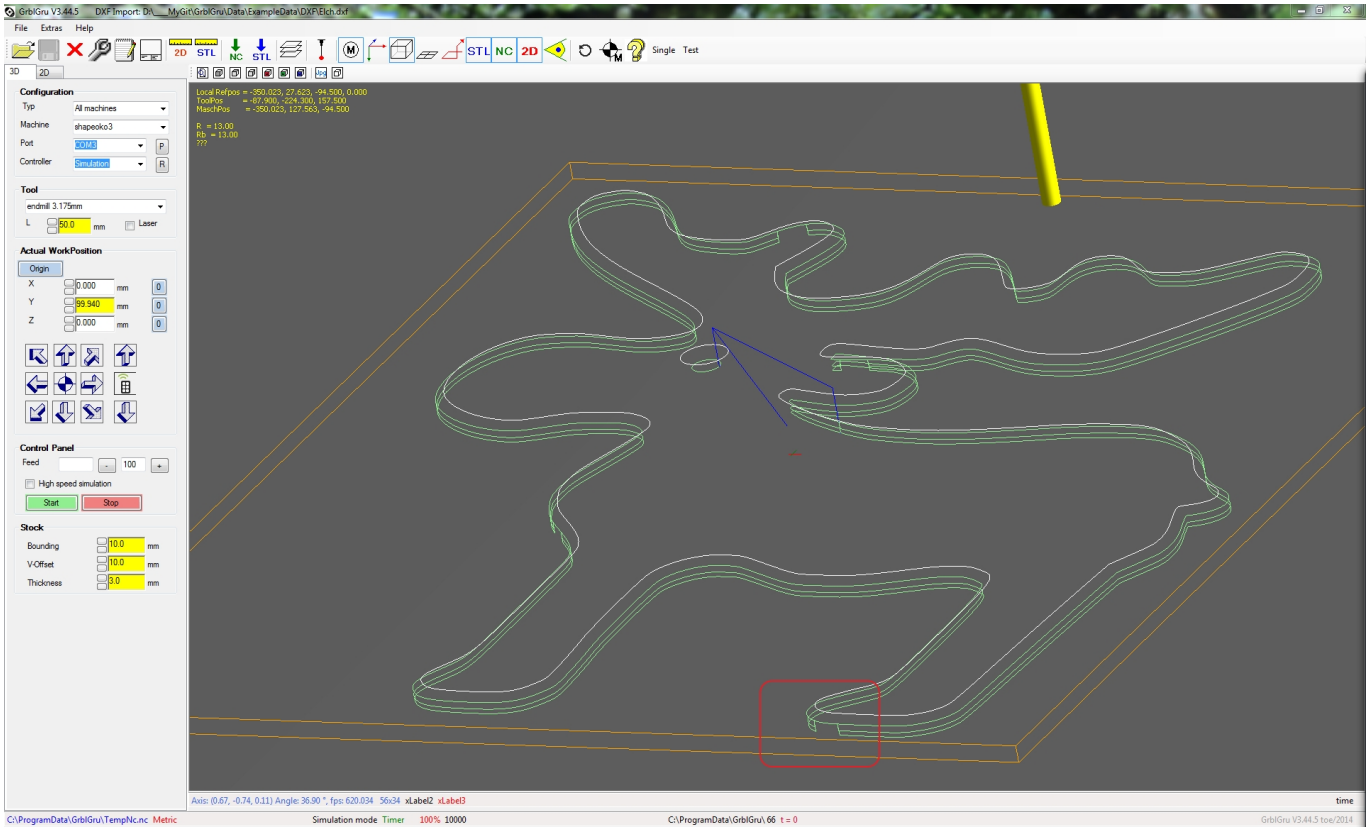


Figure 23: Visible in the marked rectangle is one of the tabs..

4.1.11 Preview

The Preview (Eye button)  on the toolbar can be used to show what the result will look like before the job is run.

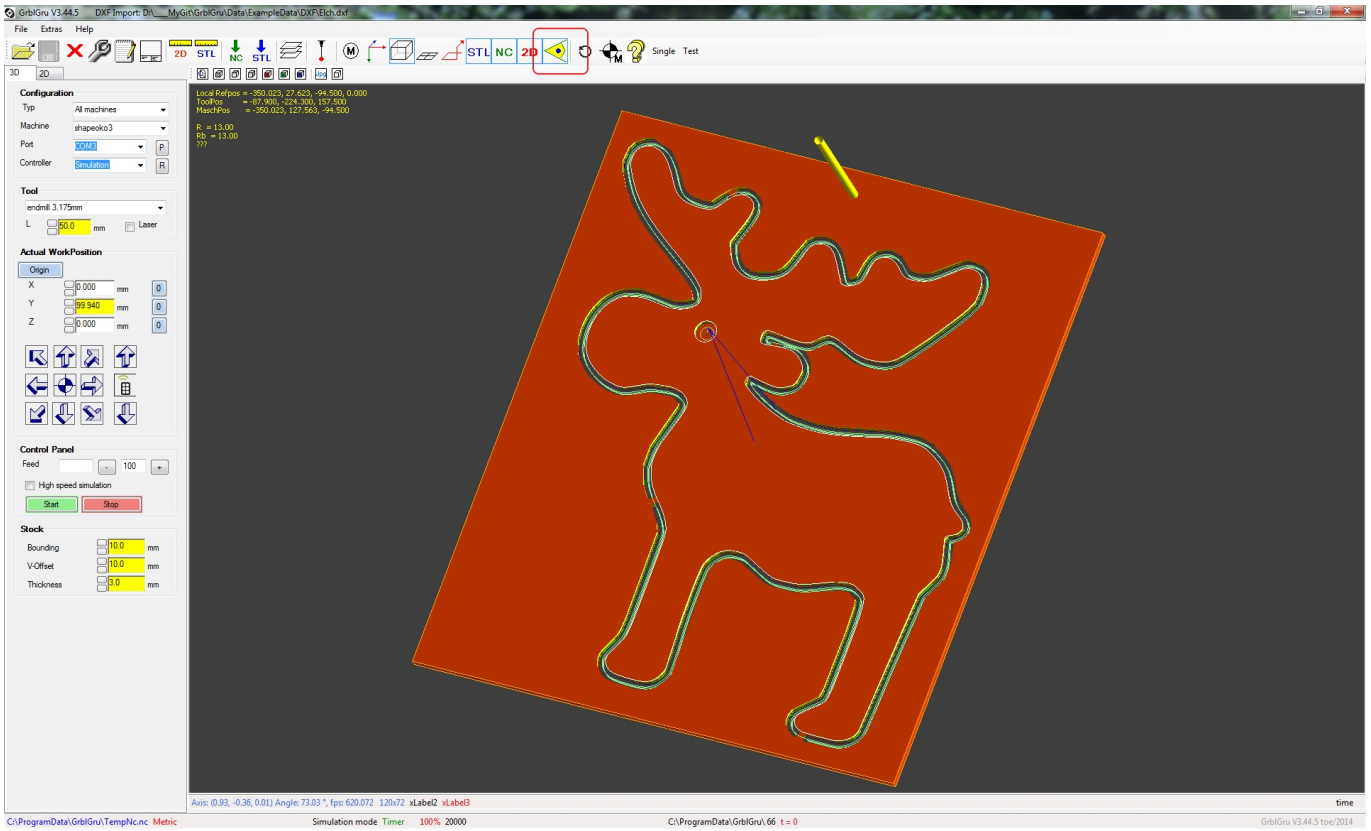


Figure 24: The preview shows what the work piece will look like

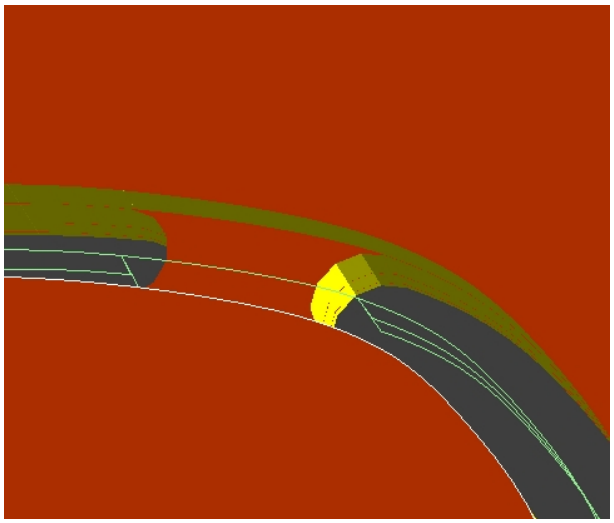


Figure 25: A tab enlarged

4.1.12 Save Project

At the end of the CAM work, what has been chosen so far should be saved as a Project. For thisyou find in the Menu the entry **'Save project as ...** '. Geometry and Jobs are stored in a Project File under your chosen name and can be reloaded at any time.

It is recommended to save the project as you go. This saves a lot of frustration in the event of a faulty operation or computer crash. It is also helpful to create an easily found folder Outside of the ***GrblGru*** folder to save your work in. This way it is not accidentally lost when ***GrblGru*** is upgraded..

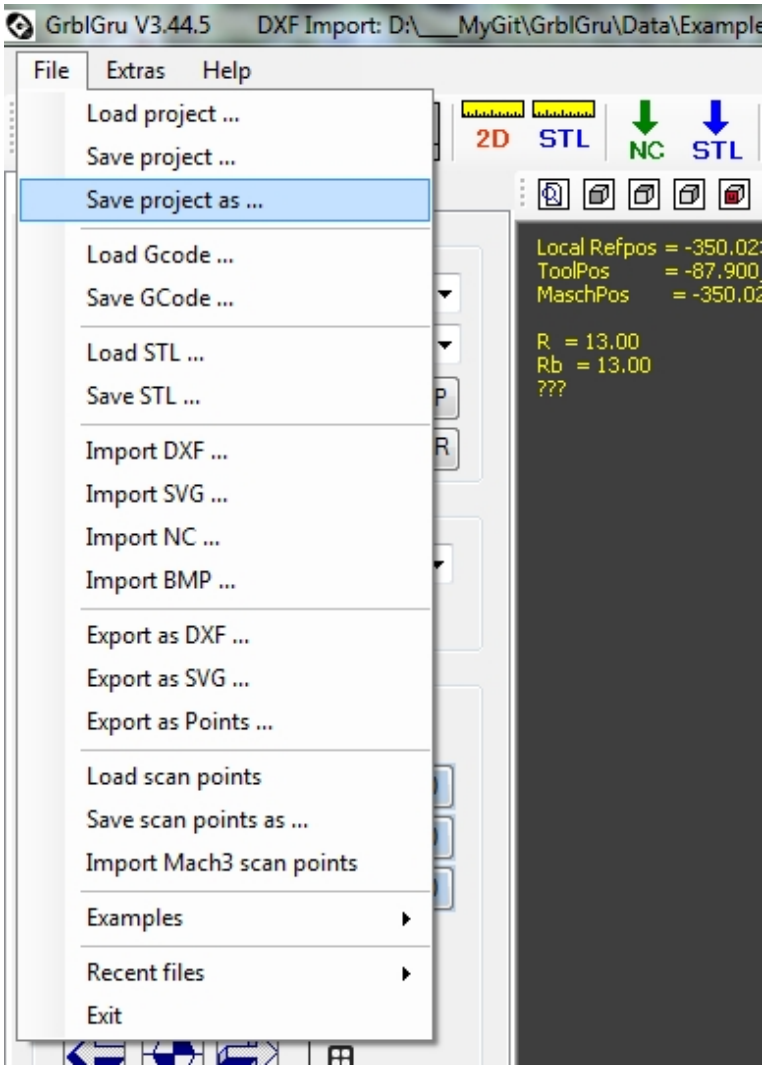


Figure 26: The menu item 'Save project as ...'

An Appropriate Location
Save your project, job templates, tool photos, anywhere outside the GrblGru directories eg. under `C:\MyGrblGruData`. This prevents your files from being accidentally overwritten during an update.

4.1.13 The Generated Gcode

If you also use ***GrblGru*** as a G-code Sender, you just press the 'Start' button and the G-code is generated by the CAM module. The GCode will be sent to the ***GrblGru*** Sender as the program executes in the Virtual model. You can watch the virtual tool move over the Geometry to make sure the program is executing as you thought it would. The model can be rotated and zoomed while the virtual work is progressing. The GCode will scroll up in the lower left window..

If you use another GCode sender program like Mach3 for control, you can also easily save the generated G-code for use later.

1) By pressing the Wrench Button in the tool bar the GCode is opened in a Text Editor. If you need to change some aspect of the GCode generated this is where is is done. The edited file can be saved in a location you specify.

2) By Pressing Shift+Wrench Button the 'Save As...' page appears preset to save as a NC (Numerical Control) file suitable for use later by ***GrblGru*** or other GCode Senders. Choose a suitable location to Save this file so it can easily be found later. Alternatively, you can just choose the file `C:\ProgramData\GrblGru\TempNc.nc` Location to Save the Copy

After your work is saved you may decide to work on another project. Press the Large Red X button to Clear the Workspace of any jobs.

4.2 Project 2: DXF template, the Coffee Cup Traye

In this project, I would like to show how to use and modify a DXF template to create a simple coffee cup tray from 6mm MDF material.

Figure 27:

4.2.1 Loading Geometry Data from DXF templates

This time instead of loading the necessary geometry from any DXF or SVG file, we use one of he DXF Templates. ***GrblGru*** periodically adds to the List of Templates. The following Templates are currently available::

- Rectangle
- Triangle
- Ellipse
- Breadboard
- Spur gear
- Box
- Bolt Circle Calculator
- Core Hole Making
- Puzzle Box

To choose a Template [2], left-click on the DXF Template Folder in the Tree Display to the Left of the graphic window. As a simple example Choose the DXF Template 'Rectangle', which creates a rectangle with rounded corners. The graphic window is updated immediately.

The Template Parameters Window appears below the Tree Display Window. As you enter the Height, Width, and Radius, ***GrblGru*** will generate the corresponding geometry in the graphic window. This can then be saved and used, or used as a part of another DXF model. Press the large Red X to clear the workspace.

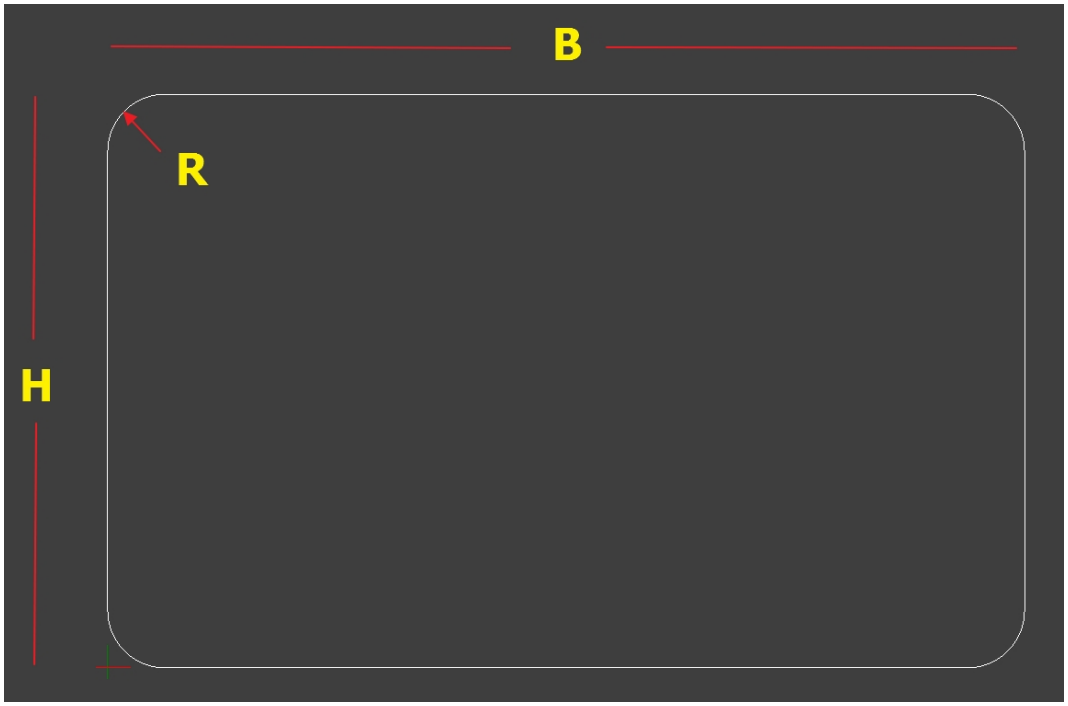


Figure 28: The simple DXF Template Rectangle. Specify Height, Width, and Radius of Corners

[2] see also chapter 'DXF templates'

4.2.2 The Box Template

For our Coffee Cup Tray example choose the Template 'Box'. With this Template it very easy to produce Finger Joint boxes. Once again, the Properties Window at the bottom left appears for input of the Height, Width, Depth and material Thickness of the box. In addition, you can also determine the Number and Size of Tails and Pins. These will interlock to form the box corners..

For each choice, changing the values in the property window immediately updates the graphical display. Try it out for fun..

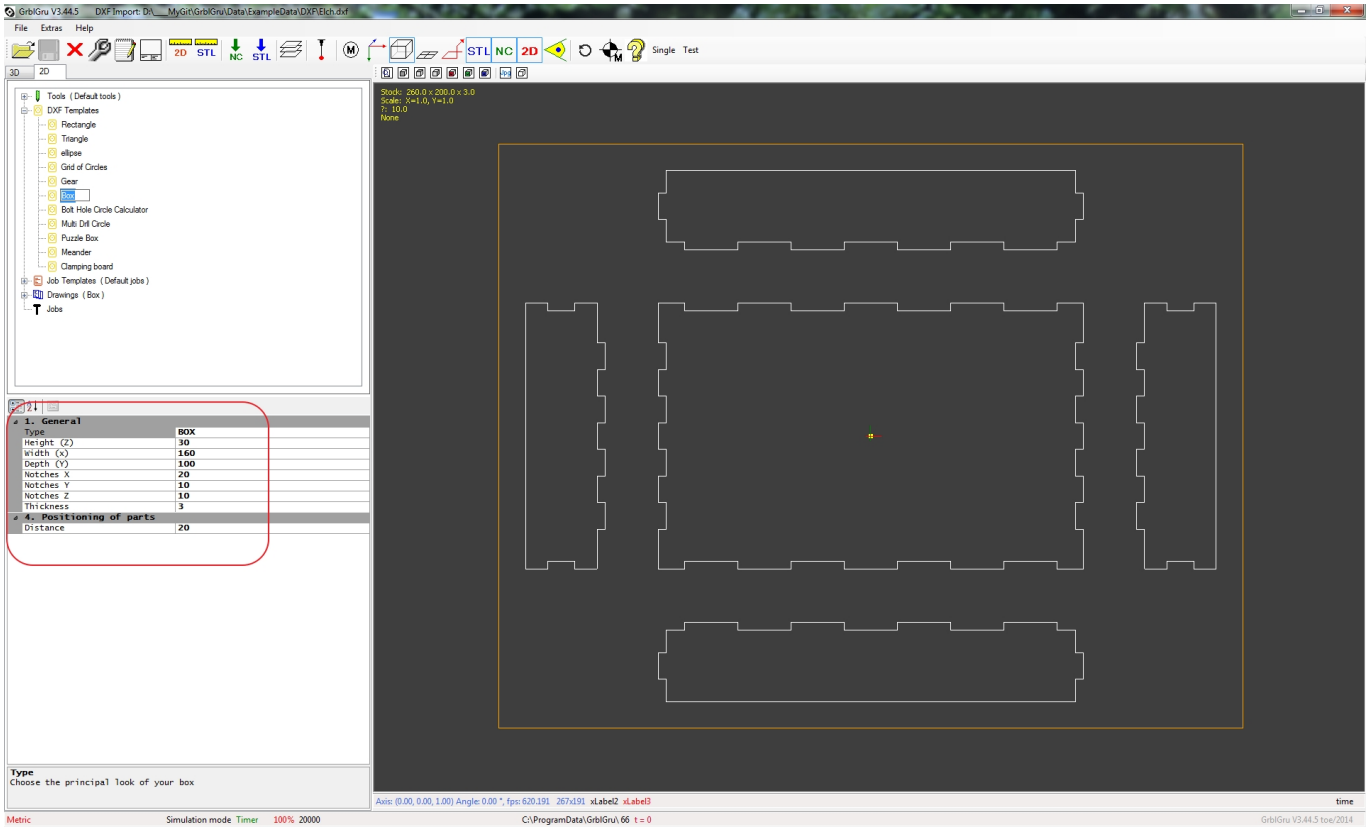


Figure 29: The DXF Template 'Box'

For anyone who wants to build a box instead of our coffee cup tray at this point the work is almost done. All you have to do is create a Job for all or some of the parts. This can be done via the Right Mouse Click Context Menu [3] as before with the Moose/Elk example. This Job will allow generation of the Toolpath GCode which can be Saved or Run.

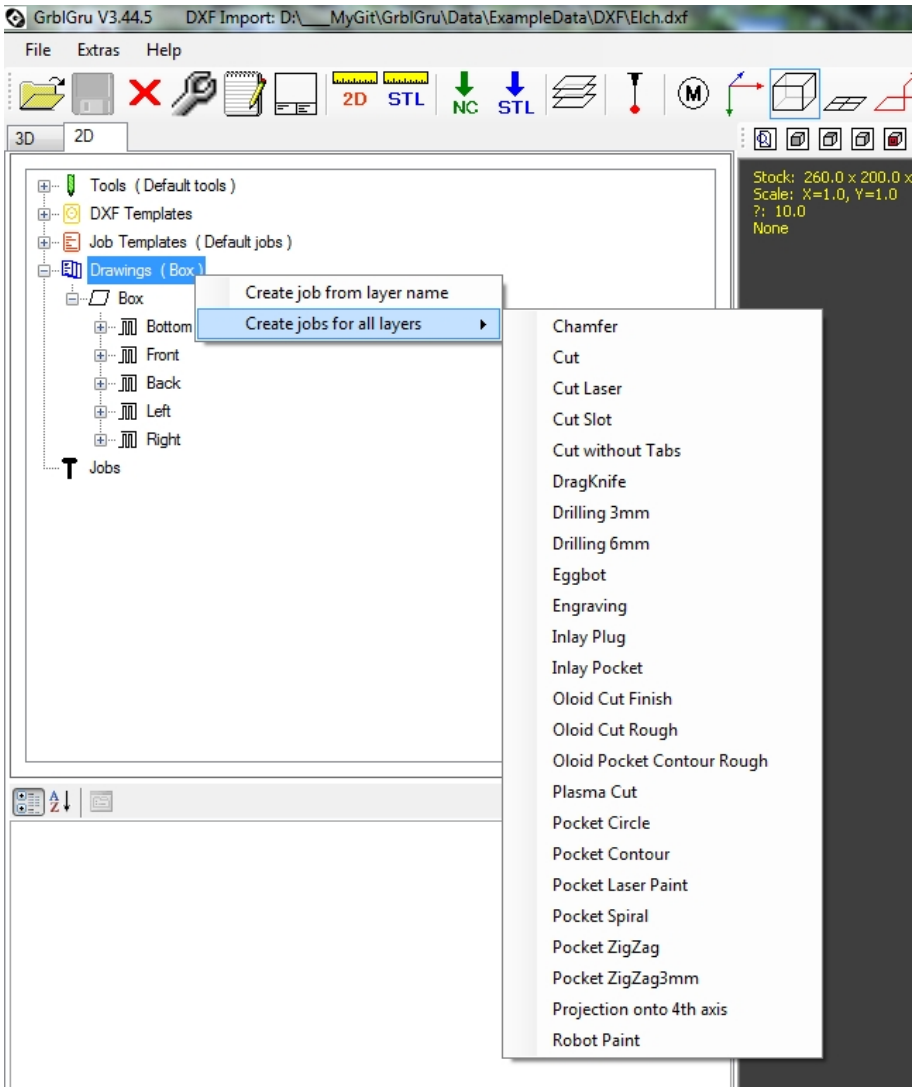


Figure 30: Create a job for all parts

[3] See also 'Various options for creating jobs'

The positions of the parts are chosen by ***GrblGru*** so that you can immediately see how they all come together. This is not optimized for space-saving in the milling process..

To optimize placement, you can Export the Parts as a DXF. Load them back into ***GrblGru*** or into the 2D Editor of your choice and position or change them as you like. They can then be saved as an edited DXF file and reloaded into ***GrblGru*** for further processing.

It is possible to select the parts individually (Left mouse click), or in suitable groups (Left mouse click + Ctrl) for processing with Jobs. To do this a context menu can be opened again by Right Clicking on a marked contour.

You can also Show or Hide any of the parts by selecting with Left click and then with the Right click Contest menu choosing Hide or Show. If you lose a part, choose Show All and it will reappear.

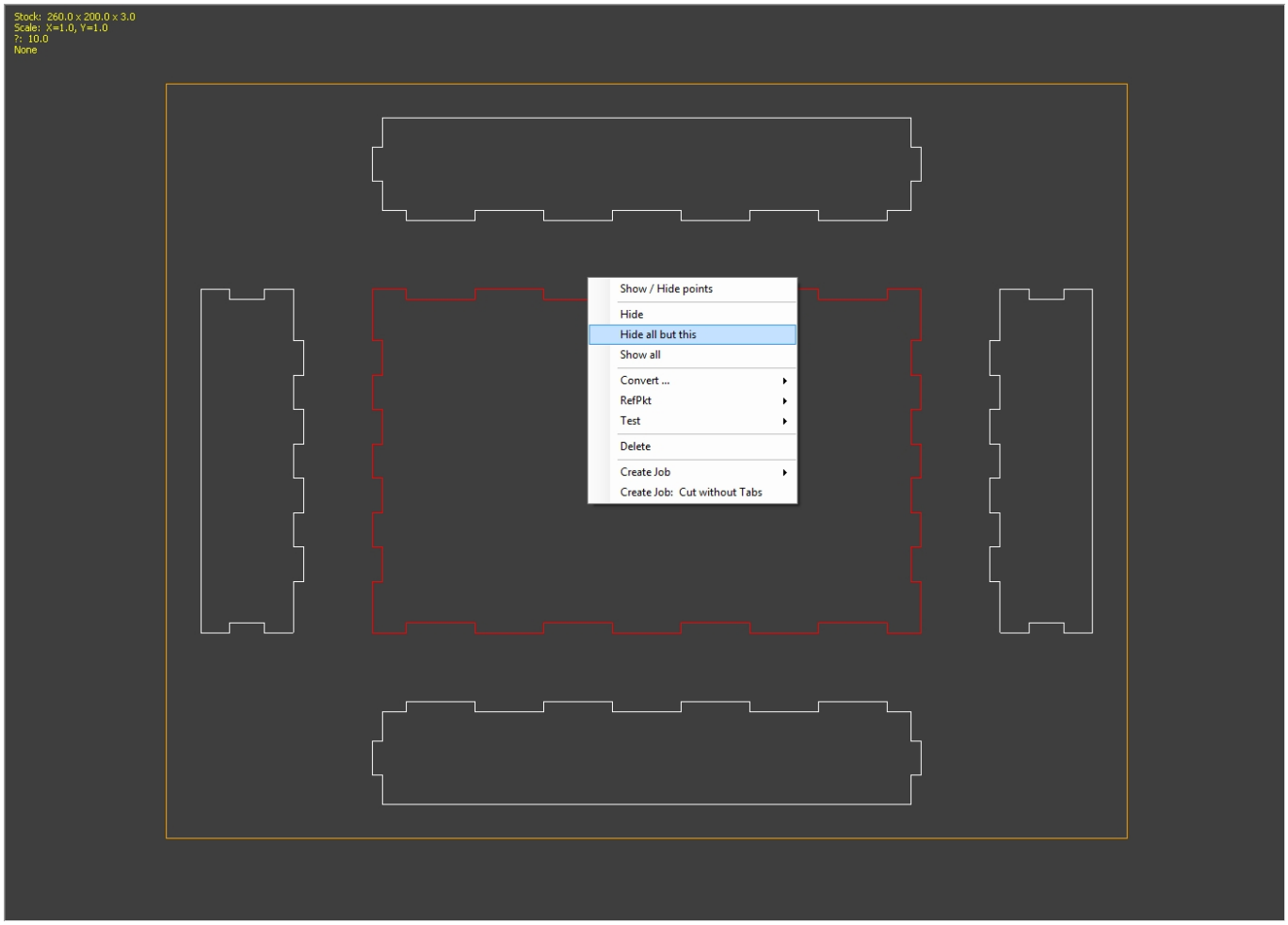


Figure 31: Context menu - choice to hide parts

4.2.3 Save Geometry Data

As already mentioned , ***GrblGru*** already has the geometry data internally available after activating a Template. This is why is it possible to Export them so easily as a DXF or SVG..

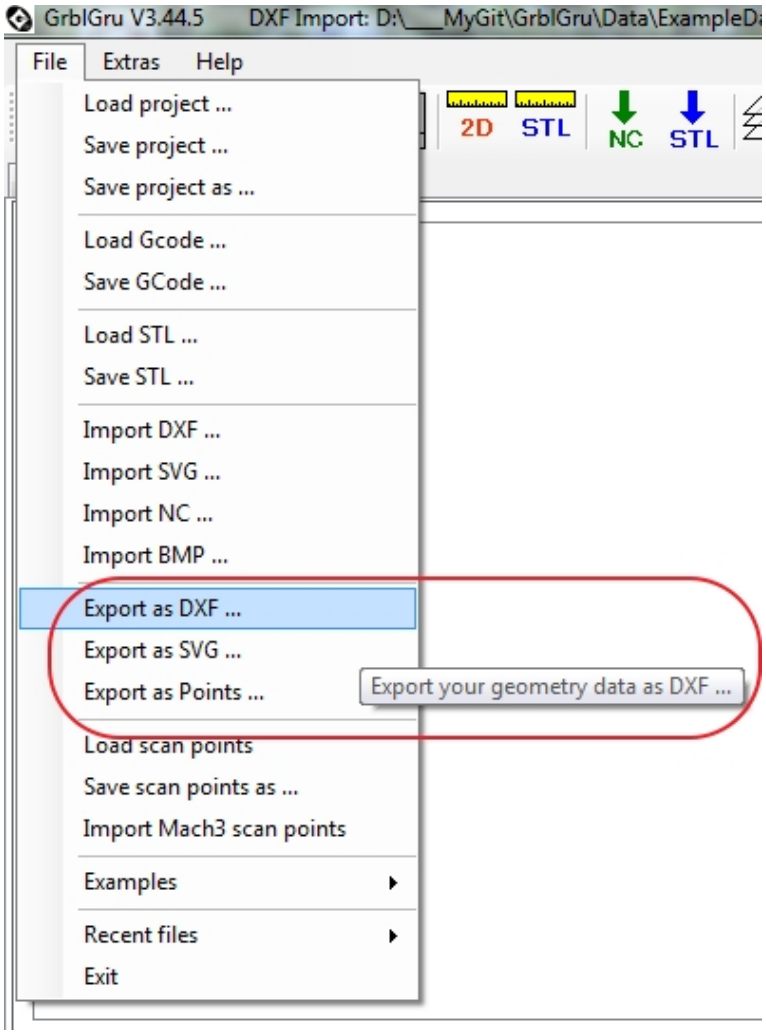


Figure 32: Menu choices for Exporting/Saving the geometry

4.2.4 Modification of the Geometrye

Saving the geometry data allows us to load that data into a 2D editor and modify it accordingly. In the following Figure you can see how you can do easily something else with the Box.

The redesigned Box result can be Exported as a DXF file and then Loaded back into ***GrblGru*** to create Toolpaths.

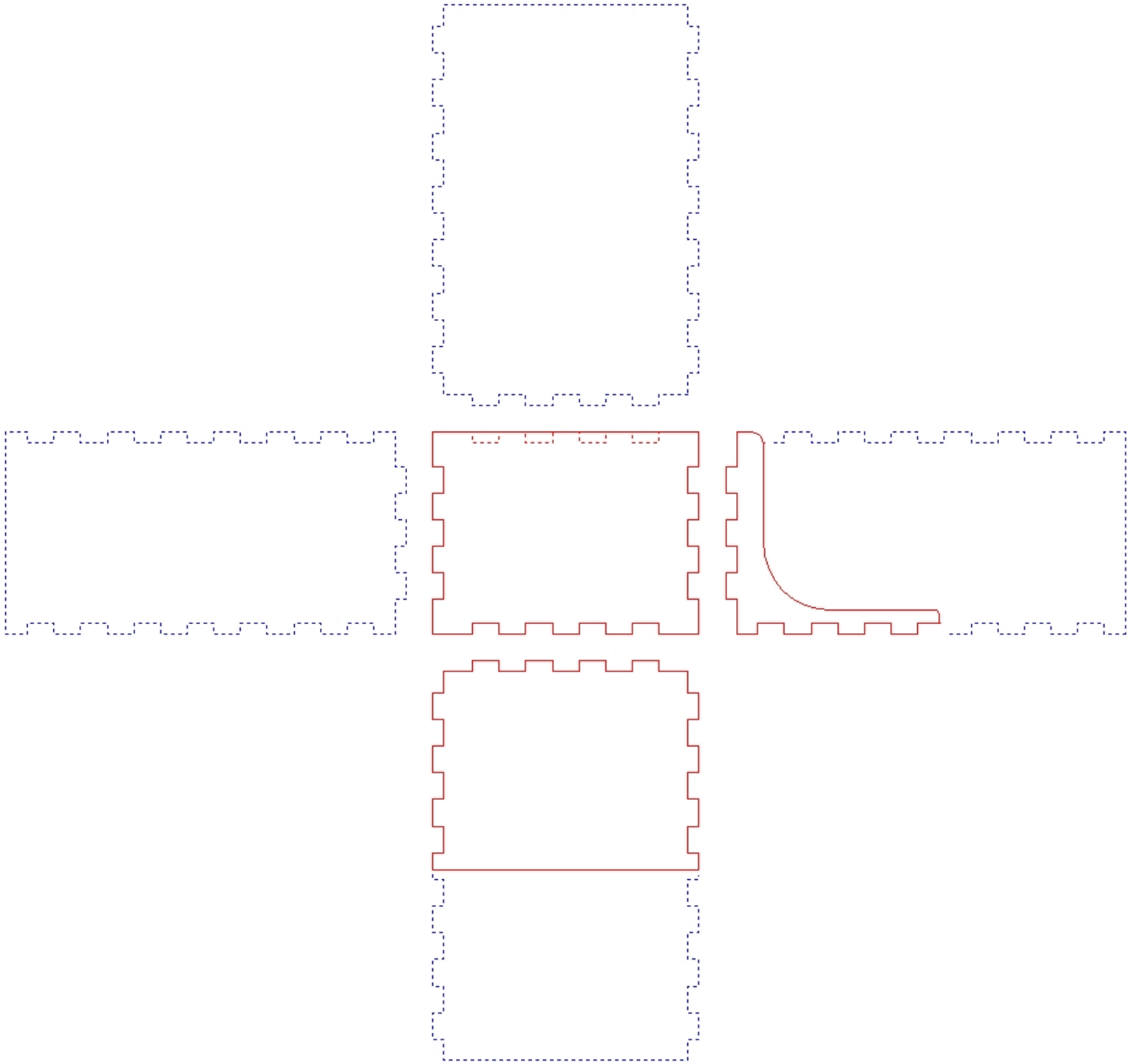


Figure 33: Redesign the Box in the 2D editor

If you want to make sure that the parts fit together correctly, you can test this model in a 3D CAD program like OpenSCAD. If this program is loaded on your computer in its default location, there is an option for it in Settings and it will appear as a convenient tab next to the 3D tab. It can then be launched from ***GrblGru*** .

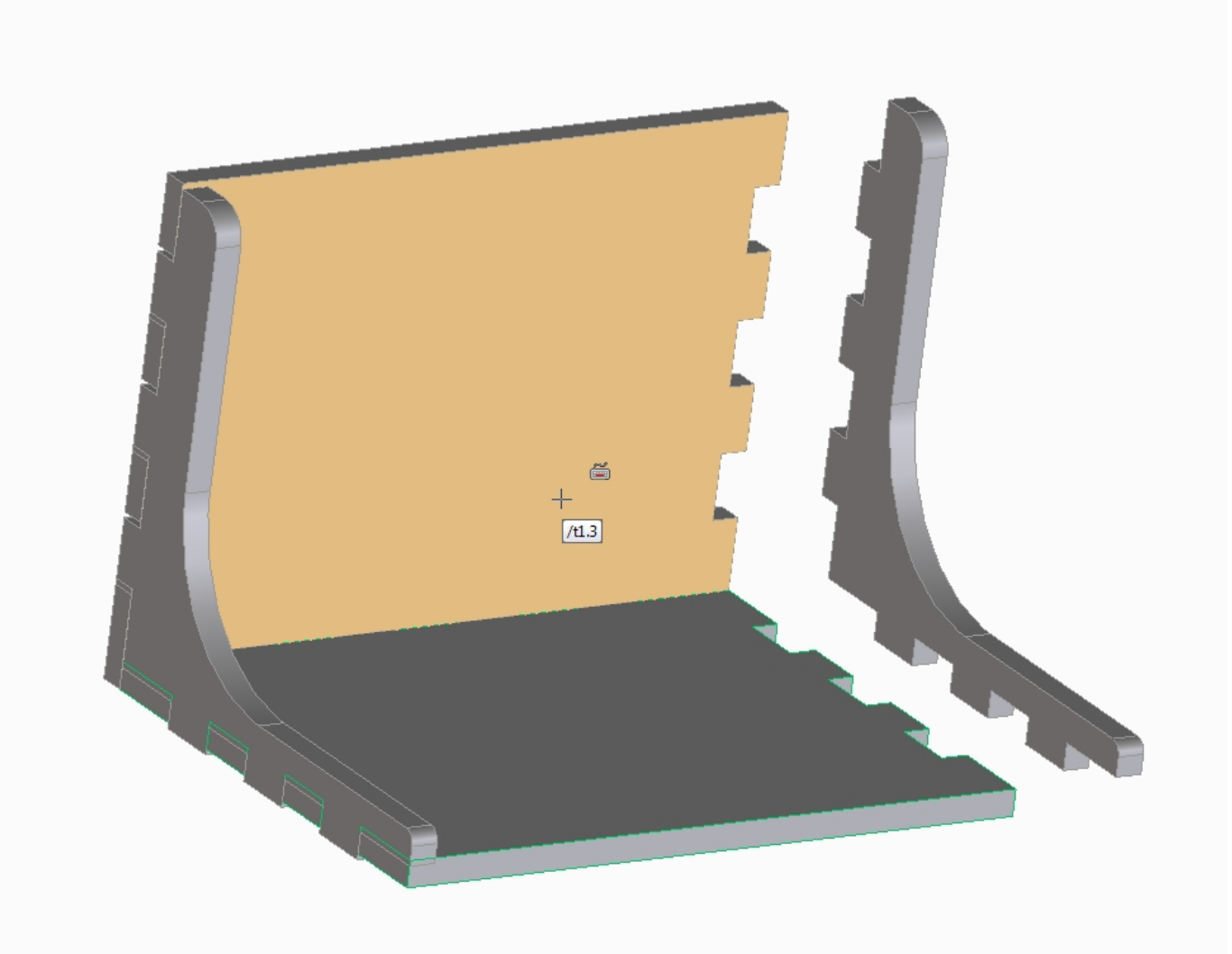


Figure 34: Check in a 3D program, ... it fits! :)

4.2.5 Create Job with ‘Corner Overcut’ Parameter

Now having made and saved the Box geometry data we are still missing an important process. All we have to do is create the Job again. Since this is a simple procedure we Select the DXF Template ‘Box’, specify the size parameters and Change the Parameter ’Corners Overcut’ in the Special section at the bottom of the Parameters window.

This parameter ensures that inside corners are slightly Over Cut so that the teeth fit together. This is necessary because you are trying to cut square corners with a round bit. This is also called Dog Boning because of the shape of the cuts when done.

Tab Height	2.5
Tab Style	SQUARE
6. Special	
Corner Overcut	No
Offset	0

Figure 35: The parameter ’Corner Overcut’ in the Cut Job

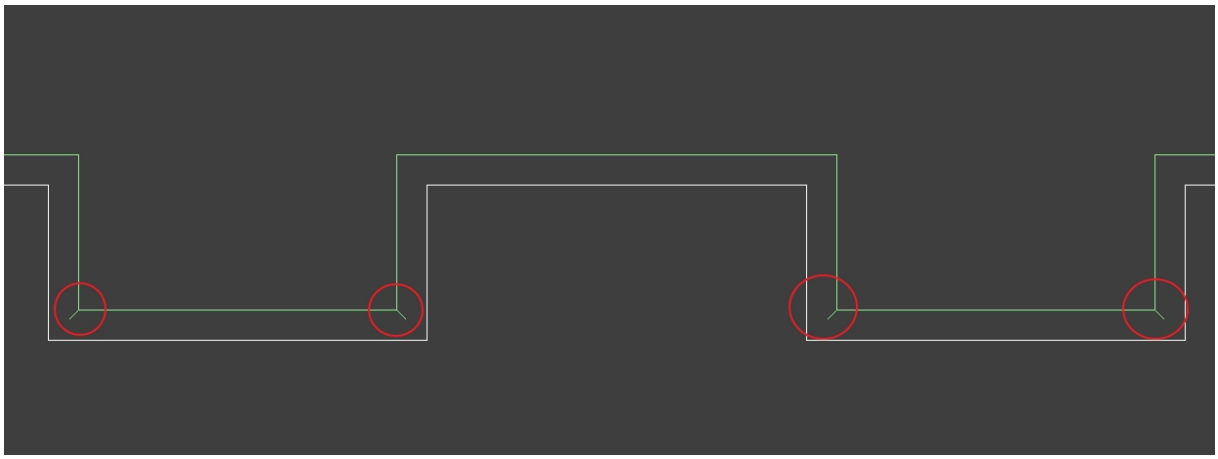


Figure 36: The effect in the tool path display

One can clearly see the effect of the OverCut corners in Preview (press the Eye button). You can see that during milling the tool will cut a little deeper into the corner. As a result, the fingers of the joint will fit completely together. There will be a slight gap at each corner when the box is completed..

For people who do not want to use this feature, there is an alternative .... file each joint by hand :)

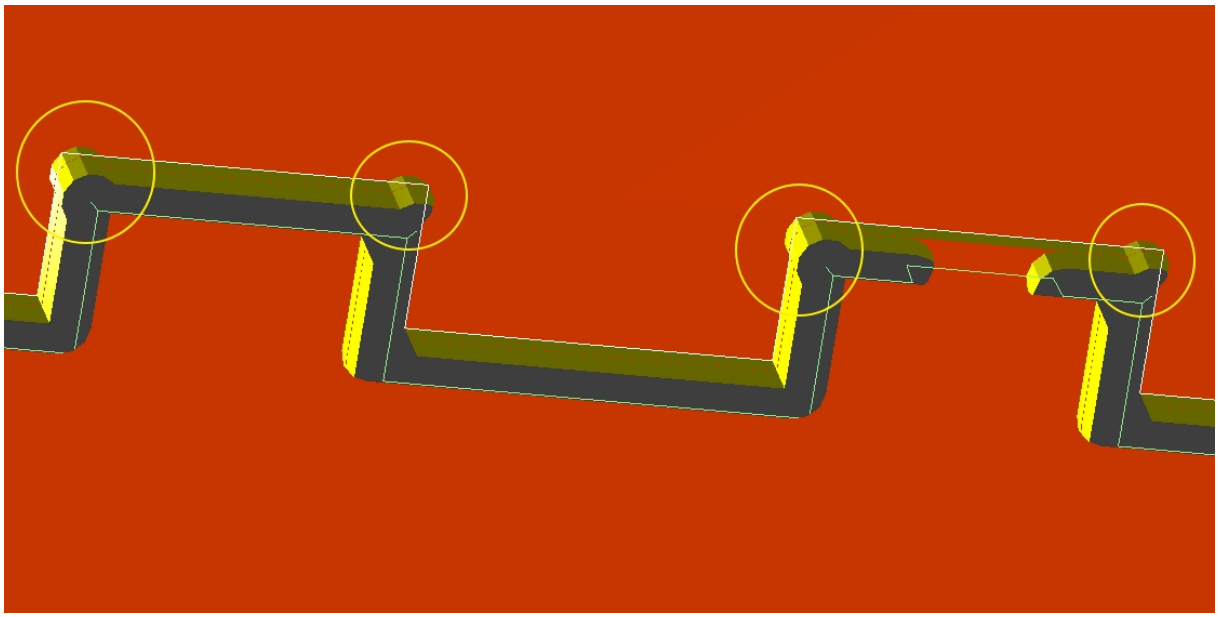


Figure 37: Check in the 3D program, ... it fits :)

4.3 Project 3: Bitmaps Drilling, the Smiley Face brooch

The possibilities of a CNC of drilling simple hole templates is easily overlooked. The practical benefits of the small project presented here can certainly be called into question, but it describes all the necessary settings that you need for drilling other patterns (even larger) like a Spoil board. This also demonstrates importing and using a Bitmap.. :)

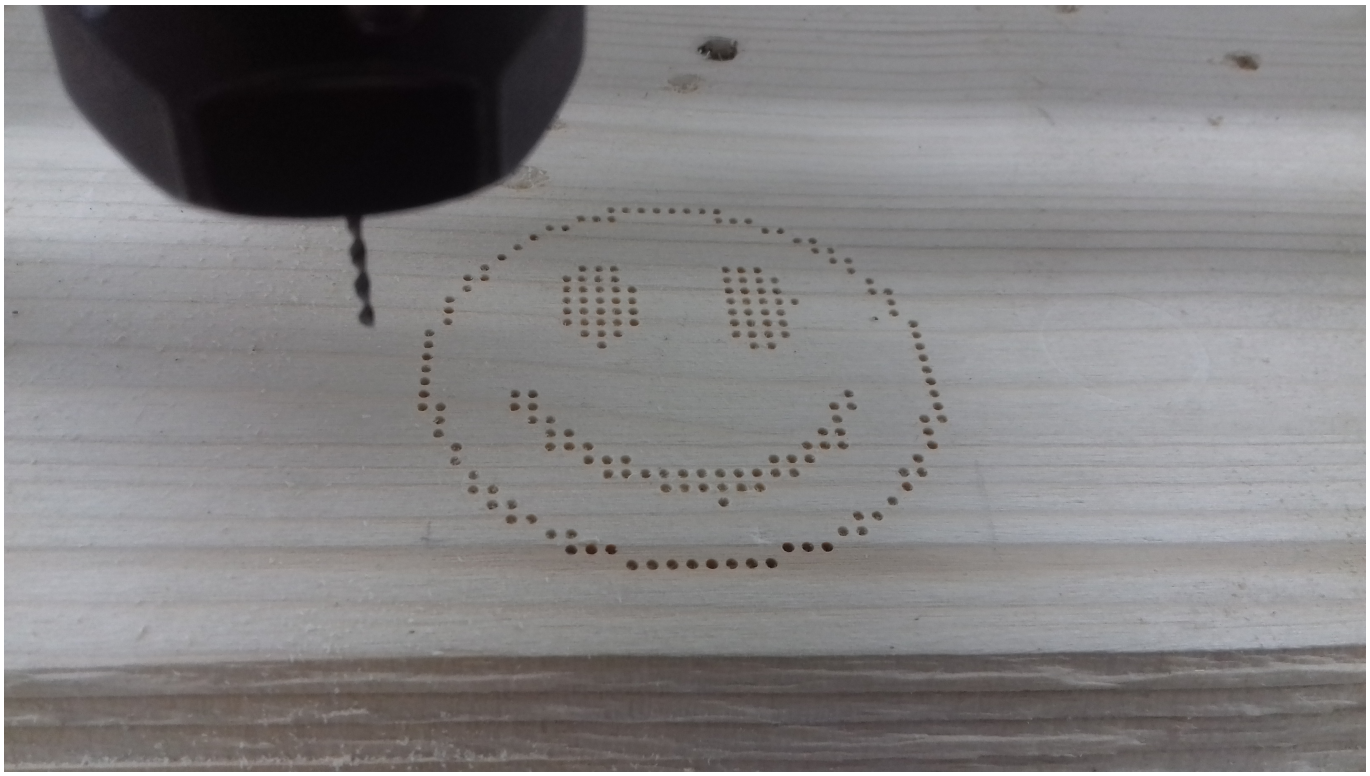


Figure 38: A simple 'Smiley Face pattern'

4.3.1 Drilling

The basis of a hole is a circle. The diameter of the circle does not matter. The Drilling will always take place in the Centre of the selected circle..

To create a Drill Template with multiple holes, you will need a DXF or SVG drawing of what you want to drill that contains Circles and select them for drilling..

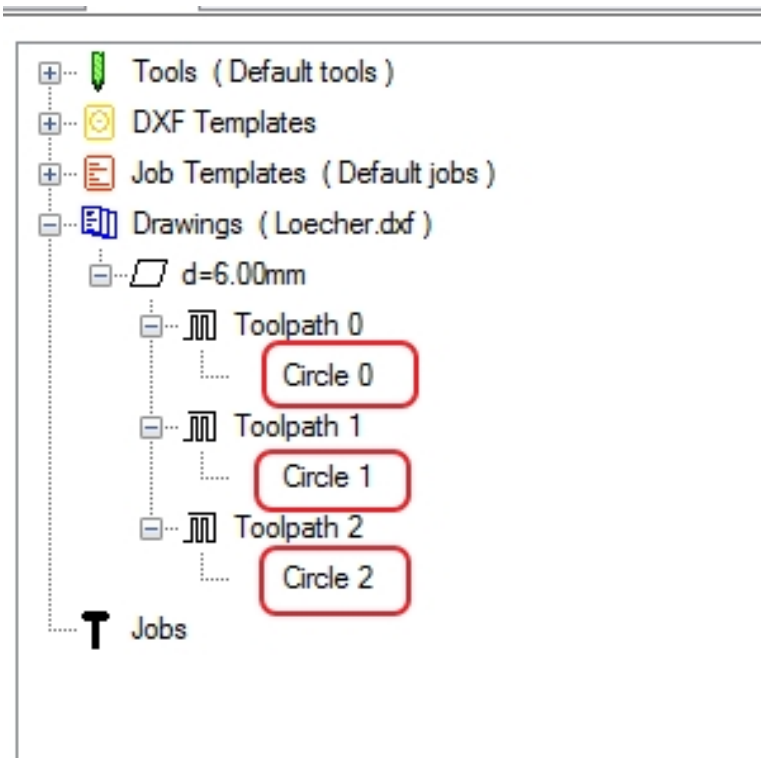


Figure 39: Drawing Expansion in the Tree of the Drawings that can be used to drill.

4.3.2 Loading Bitmaps

However, there is a special feature that we want to use for this project. When importing Bitmaps, either via the menu functions or with the aid of Drag & Drop, white pixels are interpreted as 1mm circles. While choosing Import, Press the SHIFT key to load all the non-white pixels as circles. The example Smiley.bmp is located in the folder `C:\ProgramData\GrblGru\ExampleData\BMP`

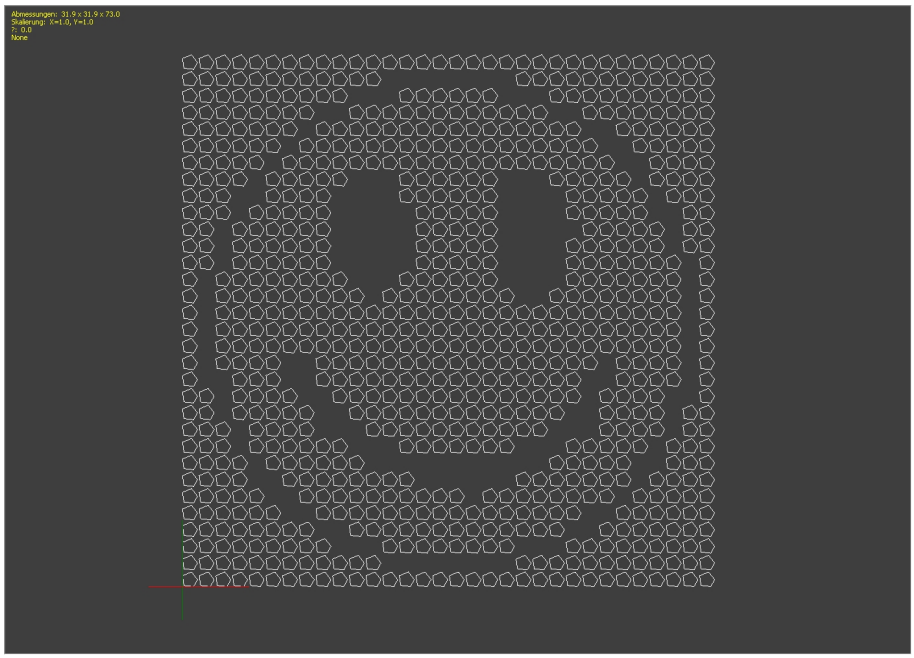


Figure 40: The BMP imported **without** Shift

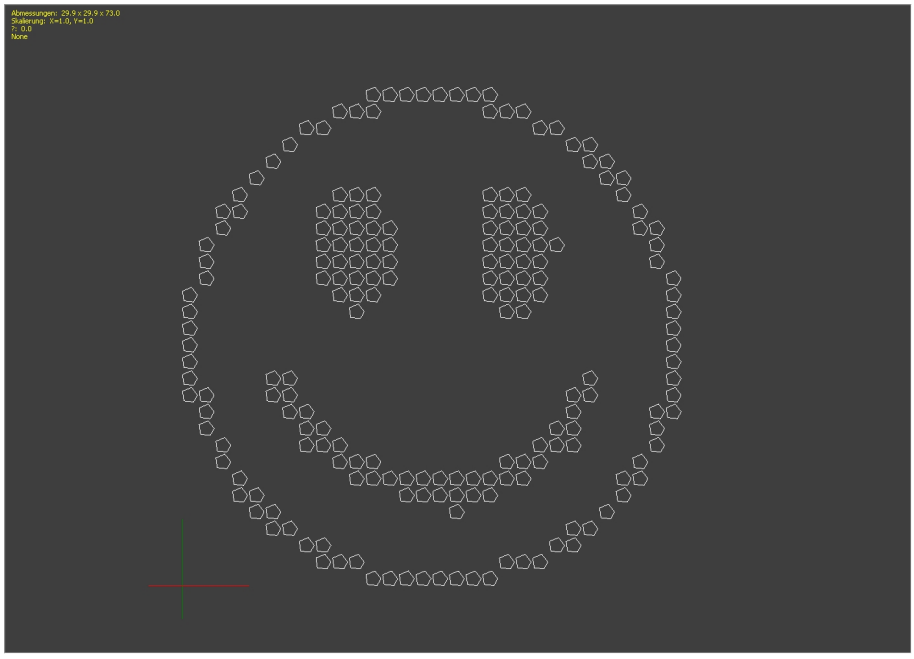


Figure 41: The BMP imported **with** Shift

If you want to Enlarge or Reduce the distances between the holes or work with a drilling tool other than a 1mm drill, you can do it all with the Scale function 

.

For editing icons or BMPs, I recommend the free program IcoFX.

4.3.3 The Operating Mode 'DRILL'

To create a Job you can use the ready-made Job Template 'Drilling', the essential feature of which is the 'DRILL' mode.

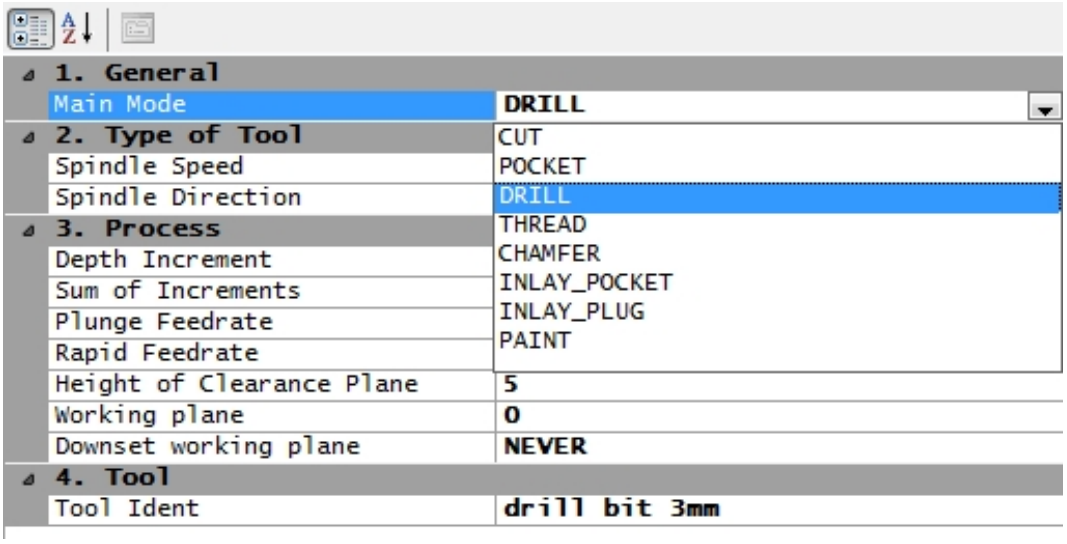


Figure 42: Operating mode 'DRILL'

The most important parameters besides the speeds are the Infeed depth (depth of the single hole) and the Infeed sum (end depth of the hole). The remaining parameters are self-explanatory .The procedure is to drill to a certain depth and then pull the drill out of the hole to remove the chips. Thereafter, the process is repeated until the final depth is reached. This is also called Pecking (like a bird).

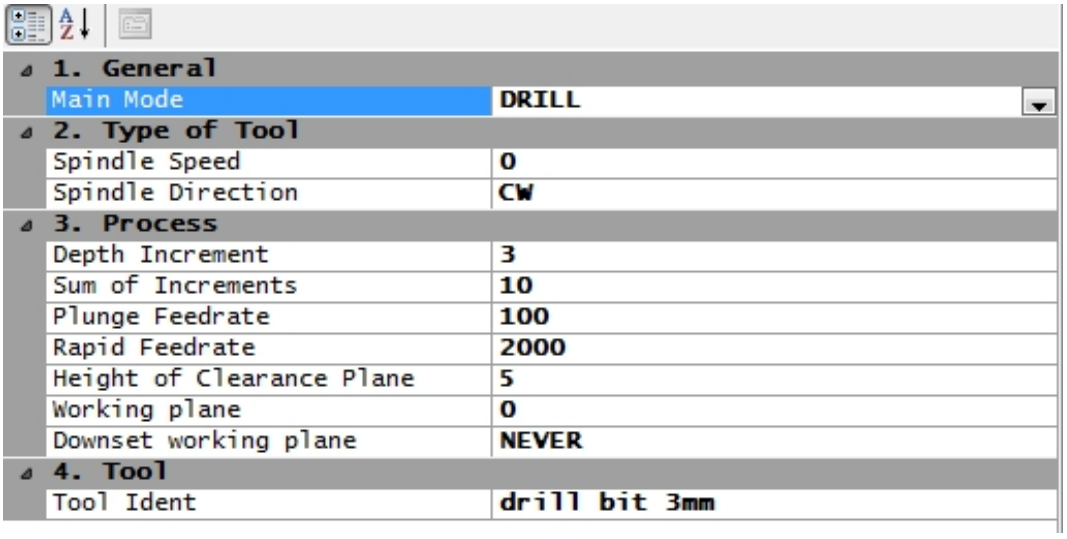


Figure 43: Parameters of the operating mode drilling

Finally, here is the 3D view of the process. The display of the tool paths can be combined with pressing the NC button. In combination with the Shift key, the paths of the tool are also displayed on the safety level.

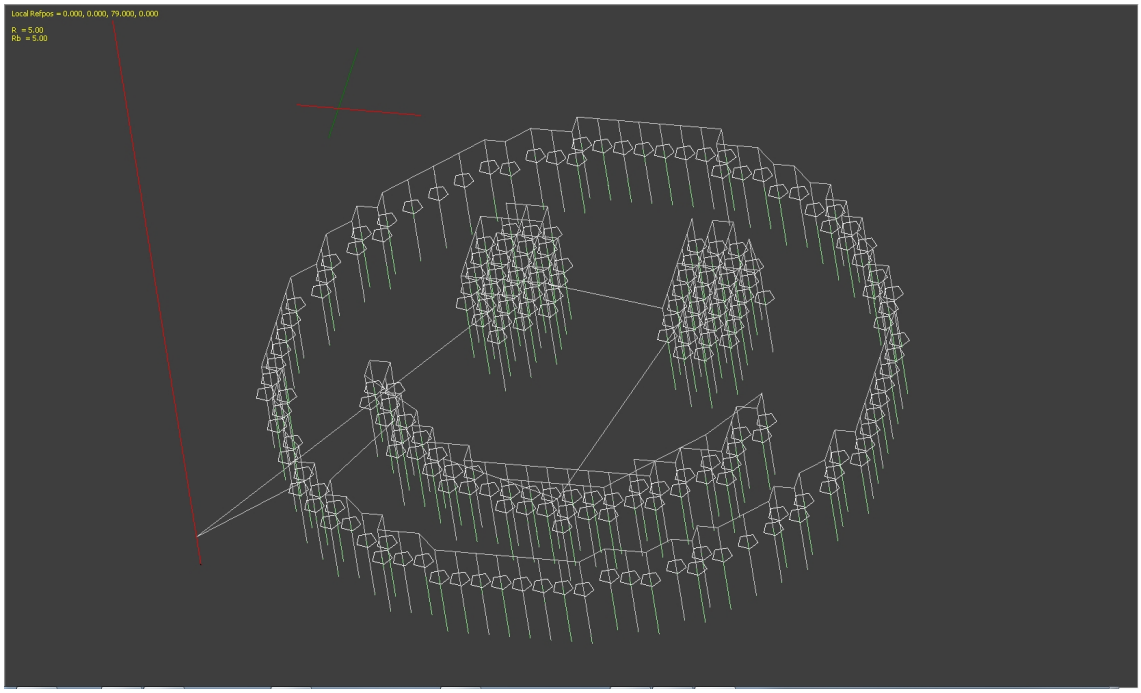


Figure 44: Drilling in 3D view

4.4 Project 4: Projection onto a rotary axis, the Viking bowl

4.4.1 Prerequisites

The prerequisite for successful milling of a projected DXF or SVG sketch is the exact indication of the relative position of the probe origin to the tool origin. !

> During the measurement, the probe and tool move simultaneously. To ensure that the tool does not collide anywhere, it is best to remove it during measurement.

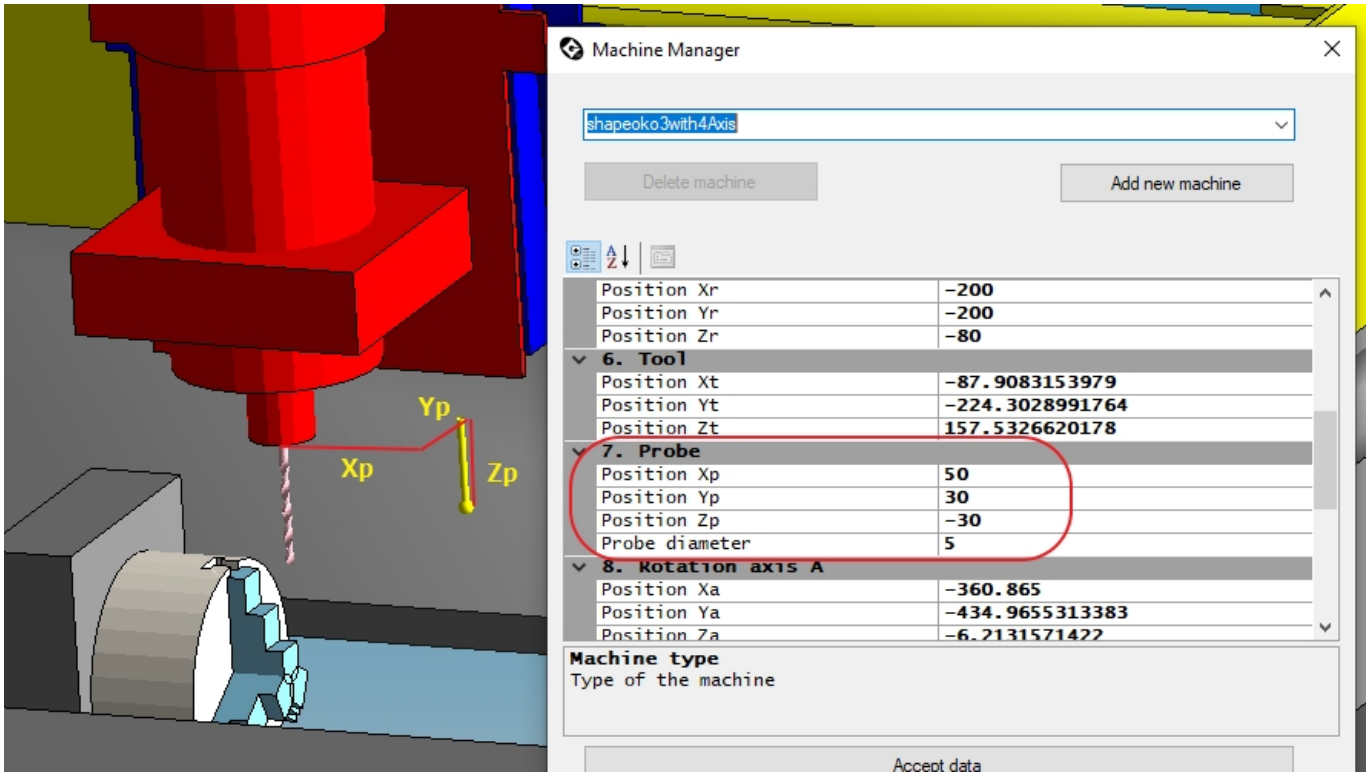


Figure 45: Offset of the probe

1

4.4.2 Scanning the workpiece

1. Position the probe over the starting point. The height above the component is not important. But the probe must be positioned exactly above the rotation axis. Accept this position with the 'Origin' key and start the measurement.

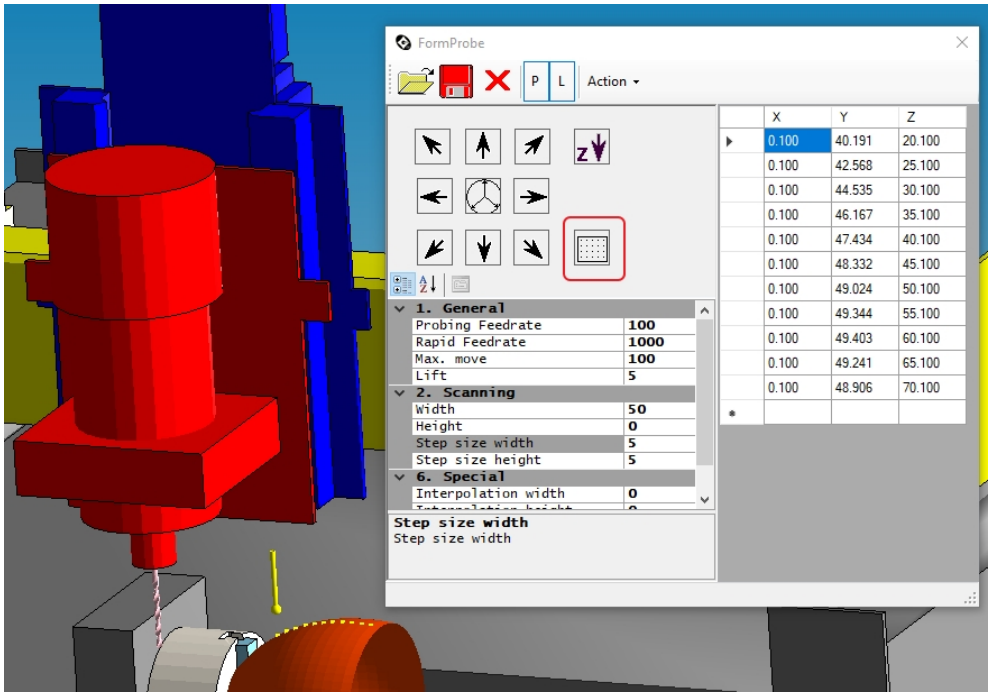


Figure 46: Scanning

2. After the menu item "Create STL" has been executed, a new STL object is created from the existing measurement points, which can then be used for the projection.

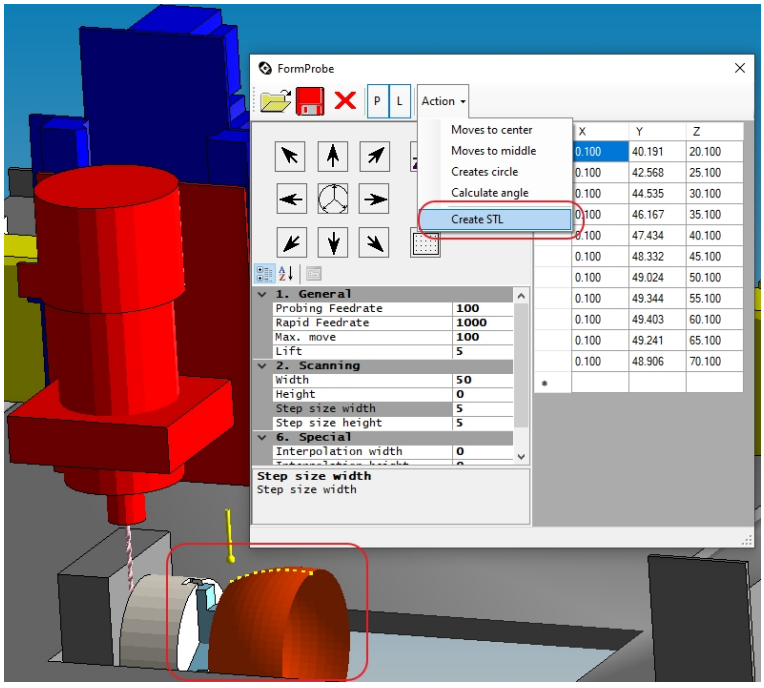


Figure 47: The new STL object from the measuring points

4.4.3 Load, scale and position sketch

1. Remove the probe if necessary so that it does not interfere during milling. End the measurement dialog and position the tool at the same point where the probe was previously located. The height is not important.

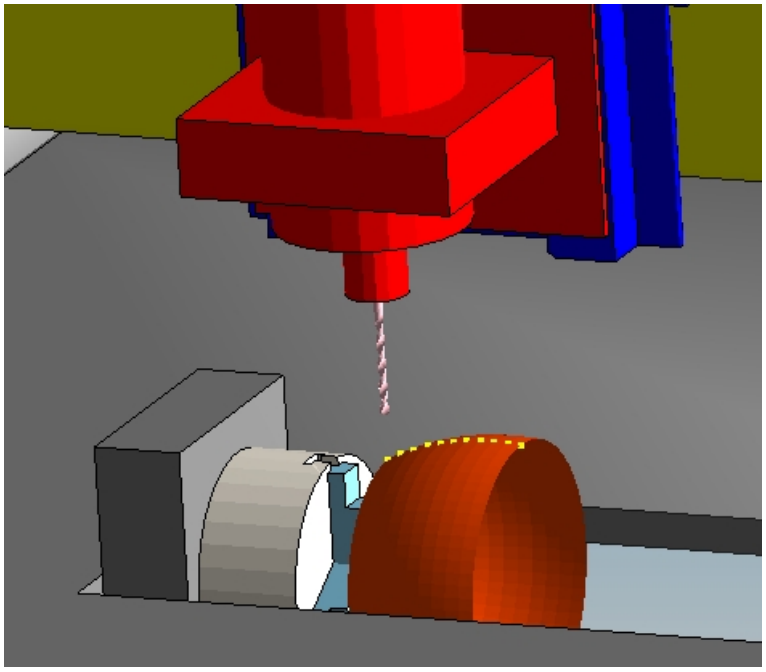


Figure 48: Tool at starting point

2. Now load a DXF or SVG sketch and position and scale it **ONLY** with the 2D data manipulation dialog. The axes X,Y, and Z must **NOT** be changed.

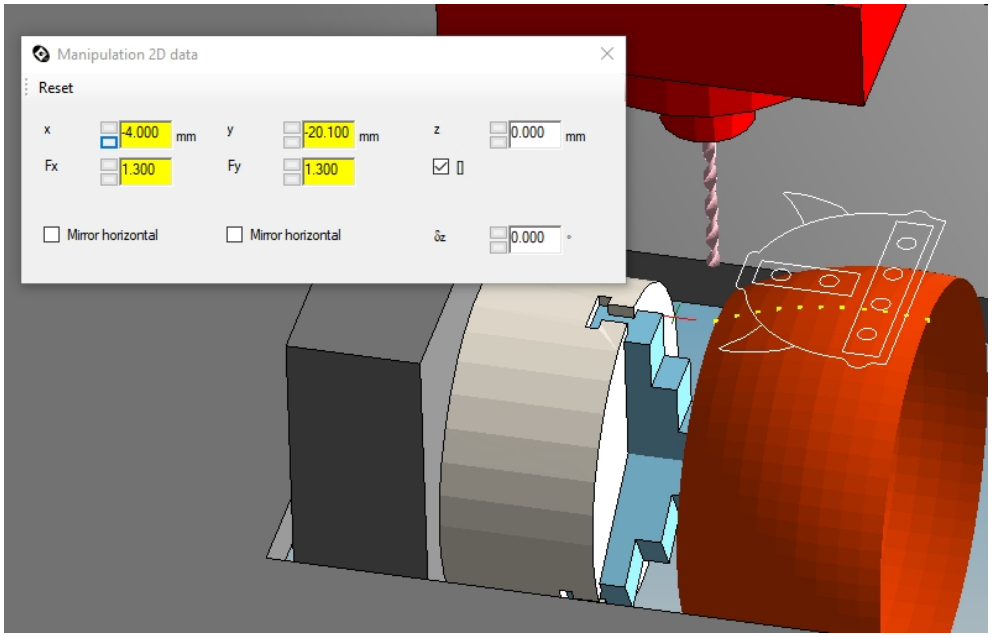


Figure 49: Scaling and positioning the sketch

For better positioning you can also hide the machine parts with the toolbar button 'M' and look at the workpiece from above (small toolbar button).

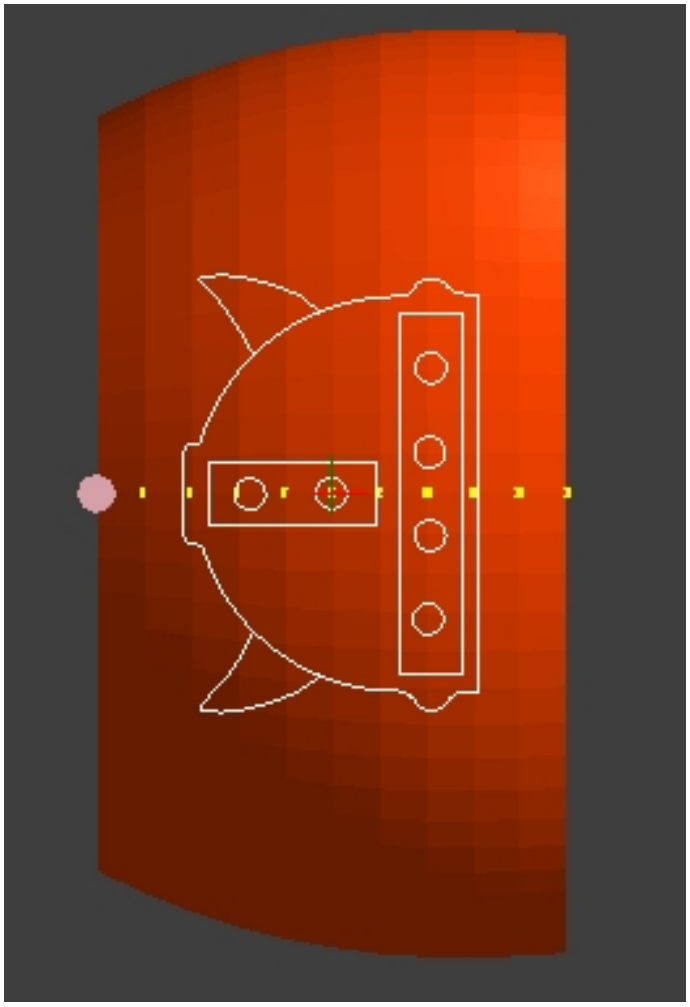


Figure 50: Top view

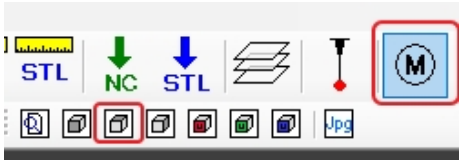


Figure 51: The buttons to look vertically from above

4.4.4 Create a job and start

3. Now go to the 2D view, select the paths to be milled and create a projection job.

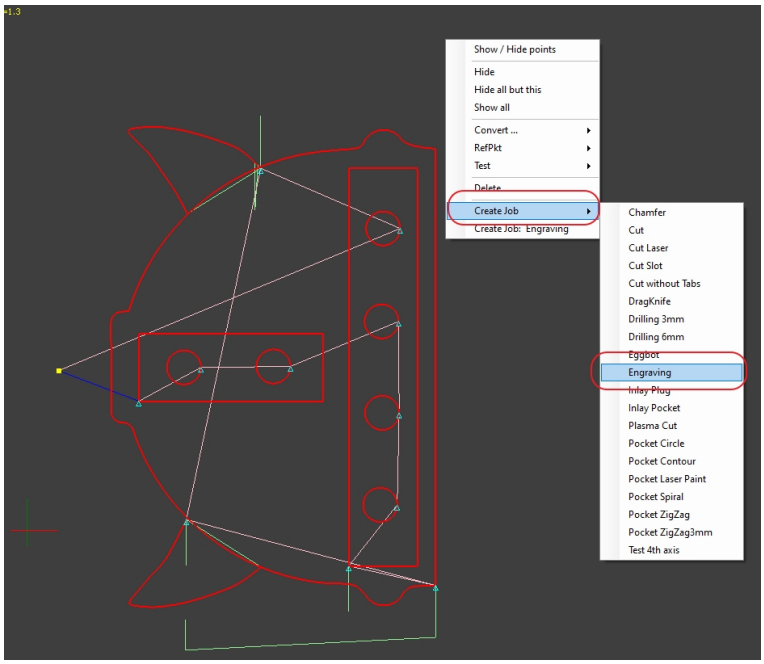


Figure 52: Job creation via right mouse click context menu

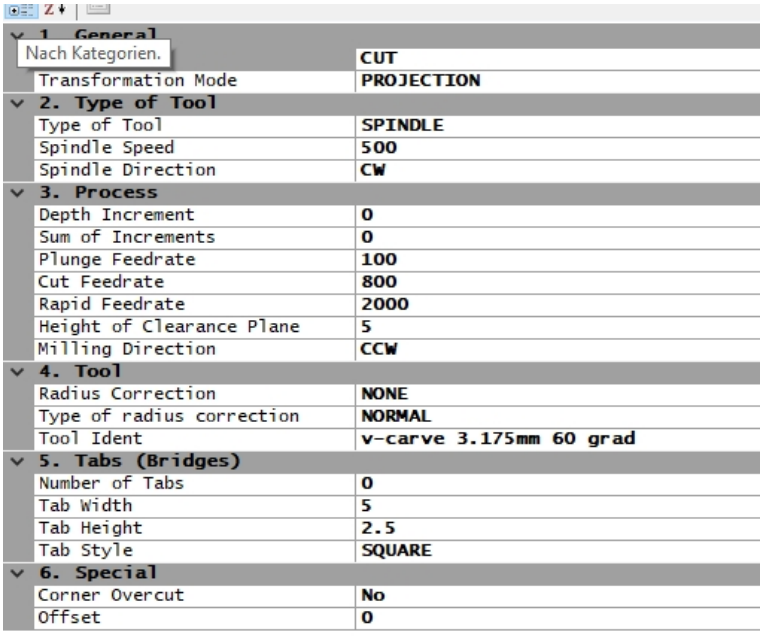


Figure 53: Example of a projection job

4. Back in the 3D view you can see how the green tool path was projected onto the workpiece.

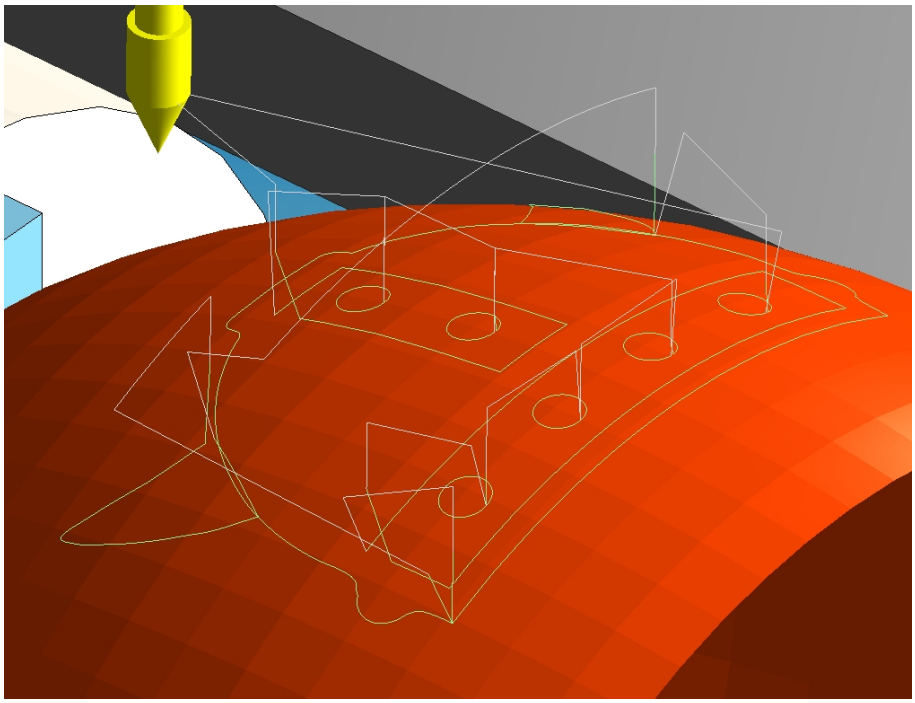


Figure 54: The tool path

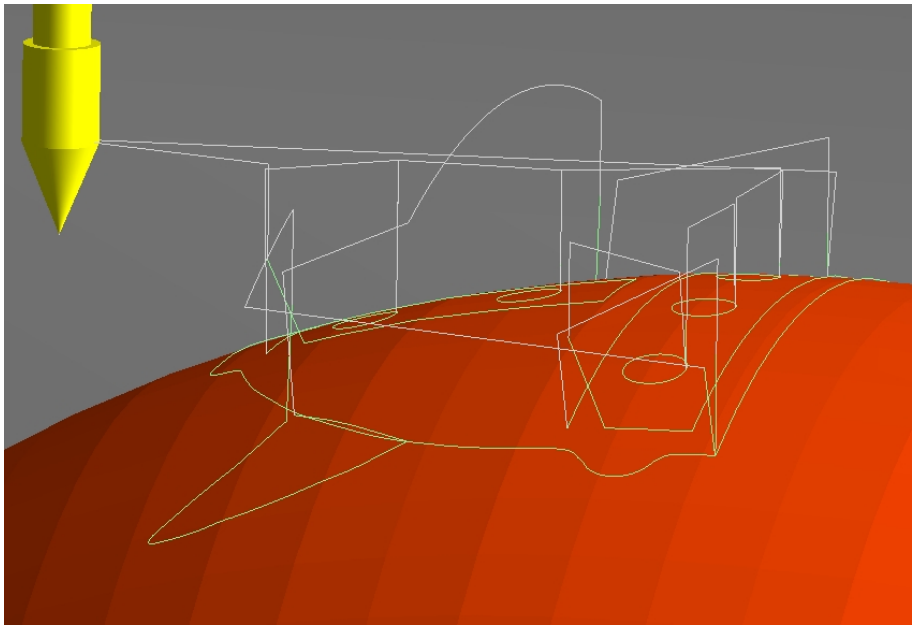


Figure 55: The tool path

All that remains is to switch on the spindle and press the Start button. :)

5 Details on other Program Operations topics

The following chapters are a disorderly collection of descriptions of specific topics.

5.1 3D graphics

To get directly to the point, for those who have never dealt with 3D graphics before, it is a little difficult in the beginning. But after a short period of practice, you will be much more comfortable. It's fun to master these graphics and you just see a lot more than 2-dimensions :)
Commands to control the 3D View

- **Show Entire Model,  , Front,  , Top,  and Side View  **
  These are on the small Toolbar Ribbon above the Graphics Window. The wireframe icons are there as well for the 2D view.

- **Zoom window**
  In the Graphics Window Press and Hold the Left mouse button and draw a Window from top / left to bottom / right. Release the button. The Graphics window will zoom into that area

- **Pan = Move Graphic**
  Hold down the middle mouse button and move the mouse..

- **Zoom = Increase / Decrease**
  Use the Mouse wheel

- **Orbit = Rotate the model in 3D**
  Hold the right mouse button and move the mouse

- **Selecting the Midpoint of the Graphic**
  A double-click with the left mouse button on a Node of the Graphic fixes this as the pivot point of the orbit function in the centre of the screen.

The View Commands above can be shown / hidden in the Graphics Window using the Large Yellow button  in the toolbar.

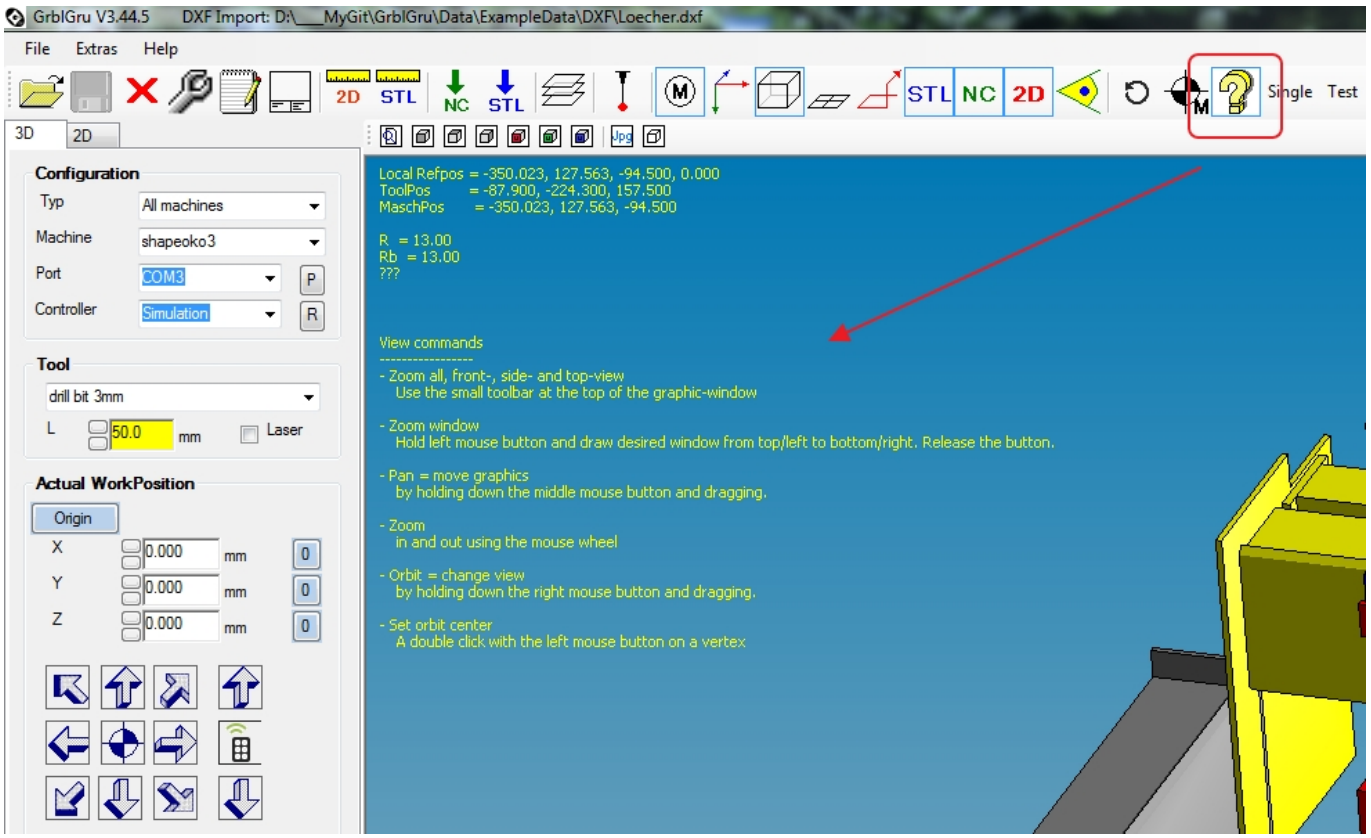


Figure 56: Quick help for the graphics functions

5.2 Options to Load Geometry data

The ***GrblGru*** CAM processor first needs 2D geometry data as the basis for its calculations. These can be read in both **DXF** and **SVG** formats. The Loading can be done in several different ways:

5.2.1 Via the Menu - Import DXF or Import SVG

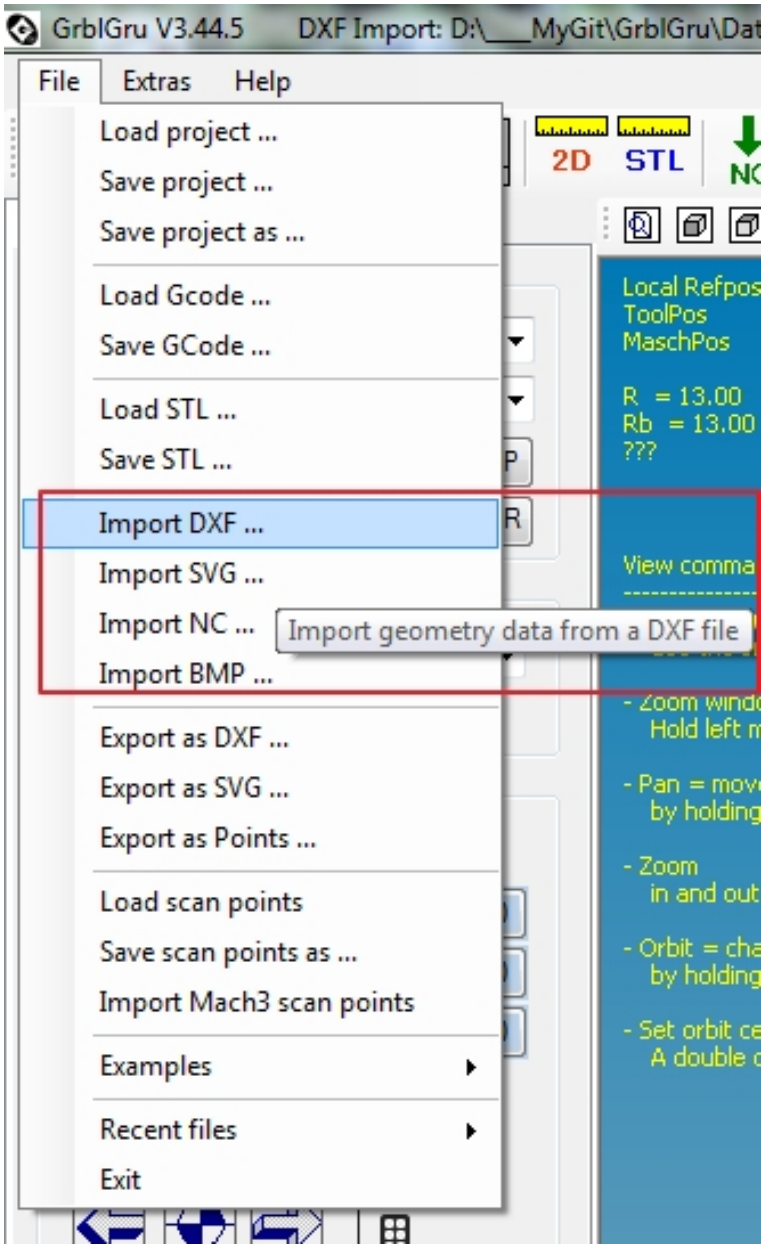


Figure 57: Loading the geometry data via the menu

5.2.2 With Drag & Drop

A very convenient way of reading is to simply load the files using drag & drop. Click on the file to be loaded in Explorer with the left mouse, drag it to the GrblGru graphics window and then release the mouse button..

5.2.3 By selecting Recent Files

there is the possibility that a current file is on this list to use.

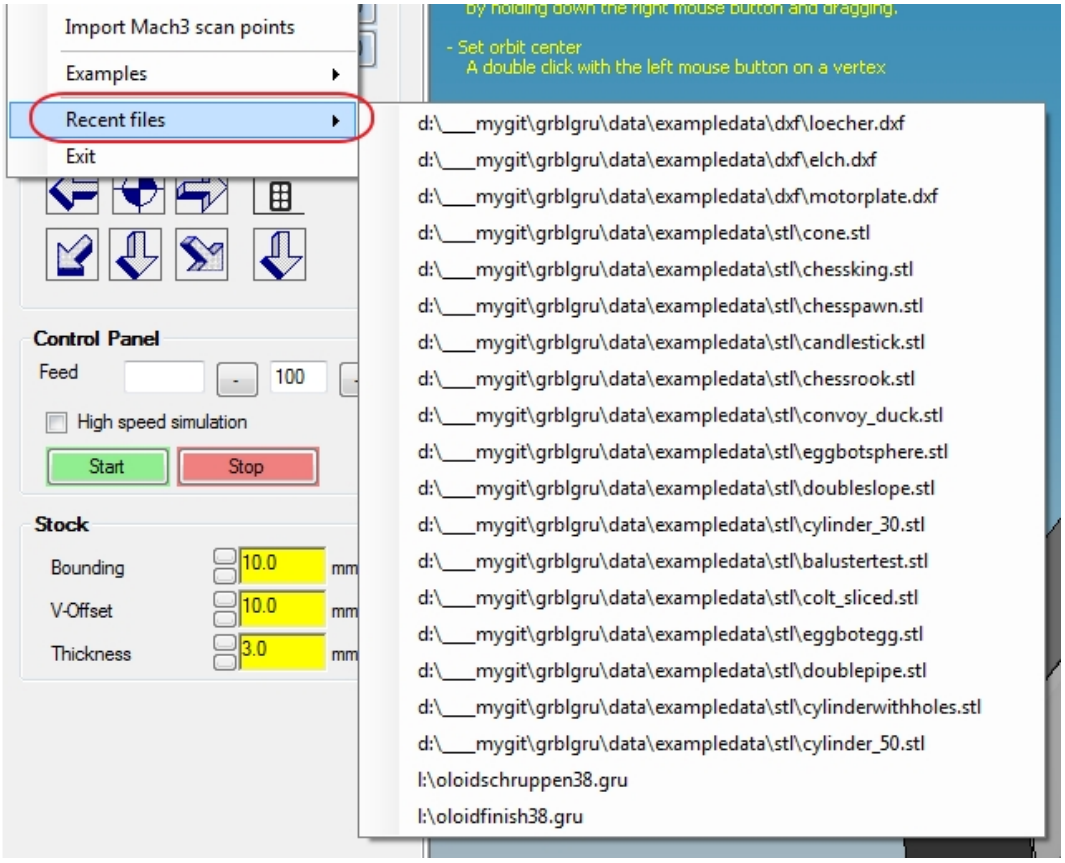


Figure 58: Loading the geometry data from the list of recently opened files

5.2.4 Load With 'Quick Load Last File' button

Last but not least there is the possibility for the last loaded data to reactivate it with the help of Hotkey buttons.

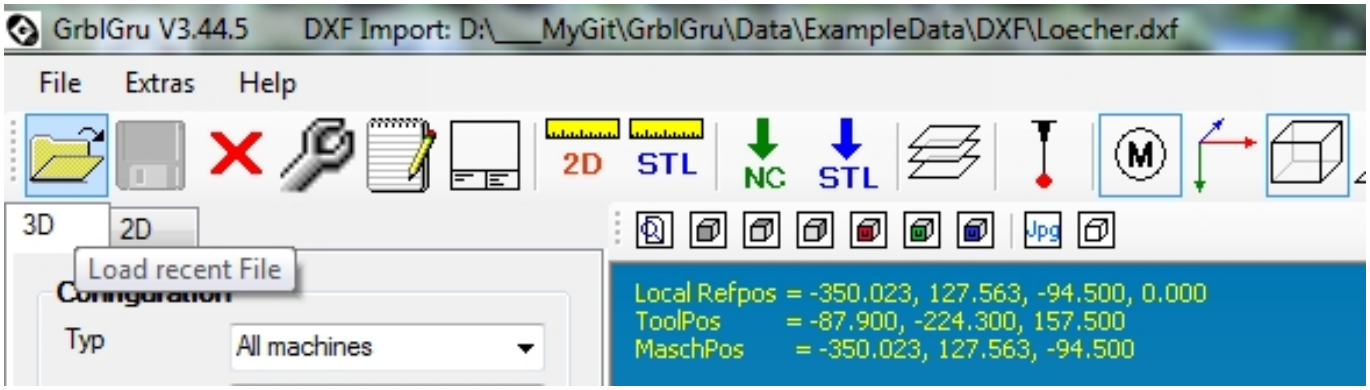


Figure 59: Loading the geometry data via 'Quick Load Last File' button

Unfortunately, there are a lot of different DXF and SVG formats. So it can happen occasionally that ***GrblGru*** does not completely load a file. For many DXF editors, you can specify different options or versions when saving. So if there are any problems, try a different setting first. If that does not lead to success, please send me your DXF or SVG file so I can see what it is.

GrblGru has only limited 2D editing functions. I think that most of users already have 'heir own favourite 2D editor that they are prepared to use and can use more effectively. For those who have not yet found an editor, I recommend the free program **Inkscape**.

5.3 Ways to Create Jobs

There are a number of ways to Create Jobs. Depending on what you are doing, one or the other method may be more suitable.

5.3.1 In the Tree View: Drawing

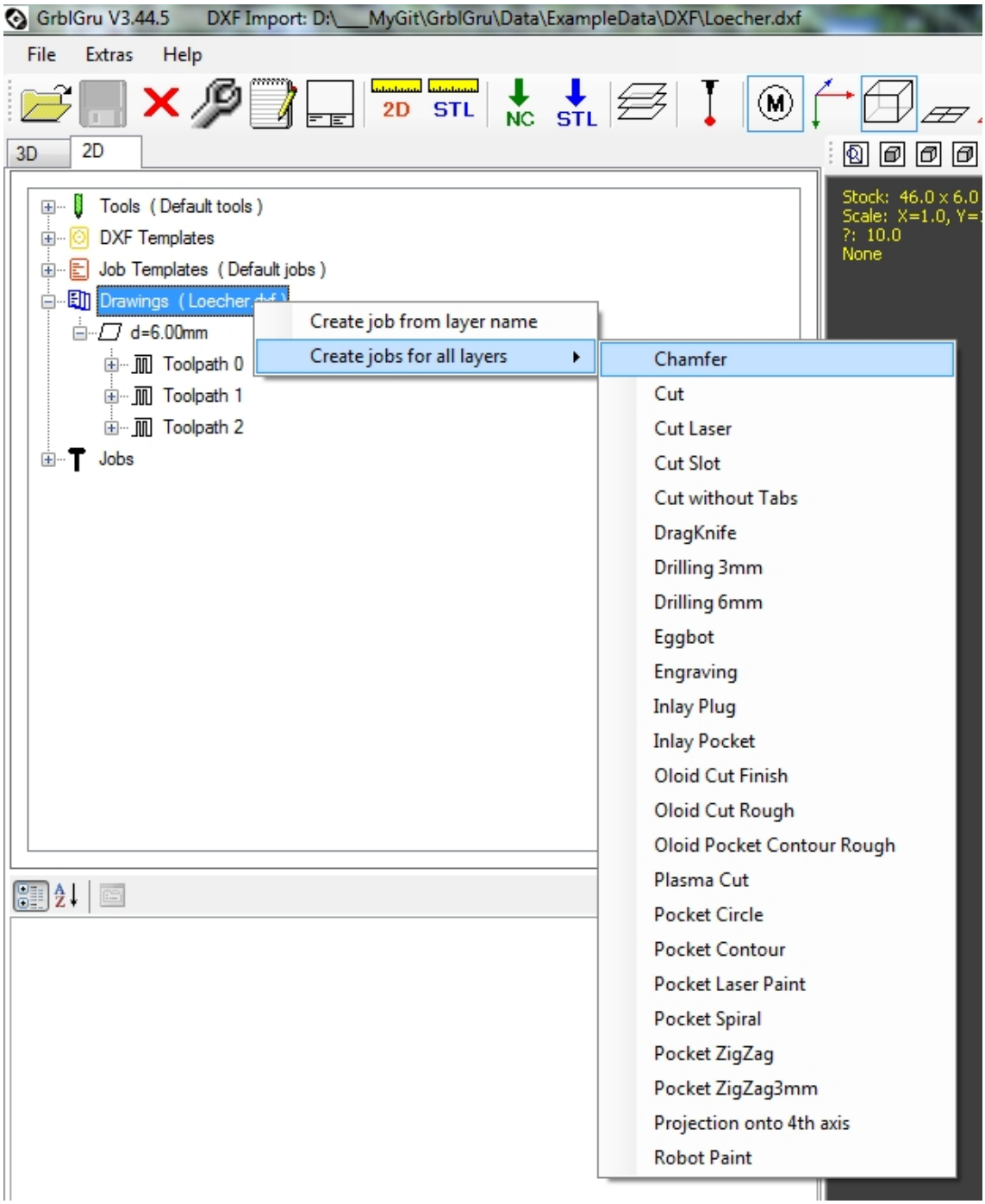


Figure 60: Shared job for the entire geometry data

Here all geometry data from all layers are provided with a Job. It is often the case if you have a Drawing with many small parts that just need to be cut out. Selecting the top one makes this easier.

5.3.2 In the Tree View: Toolpath

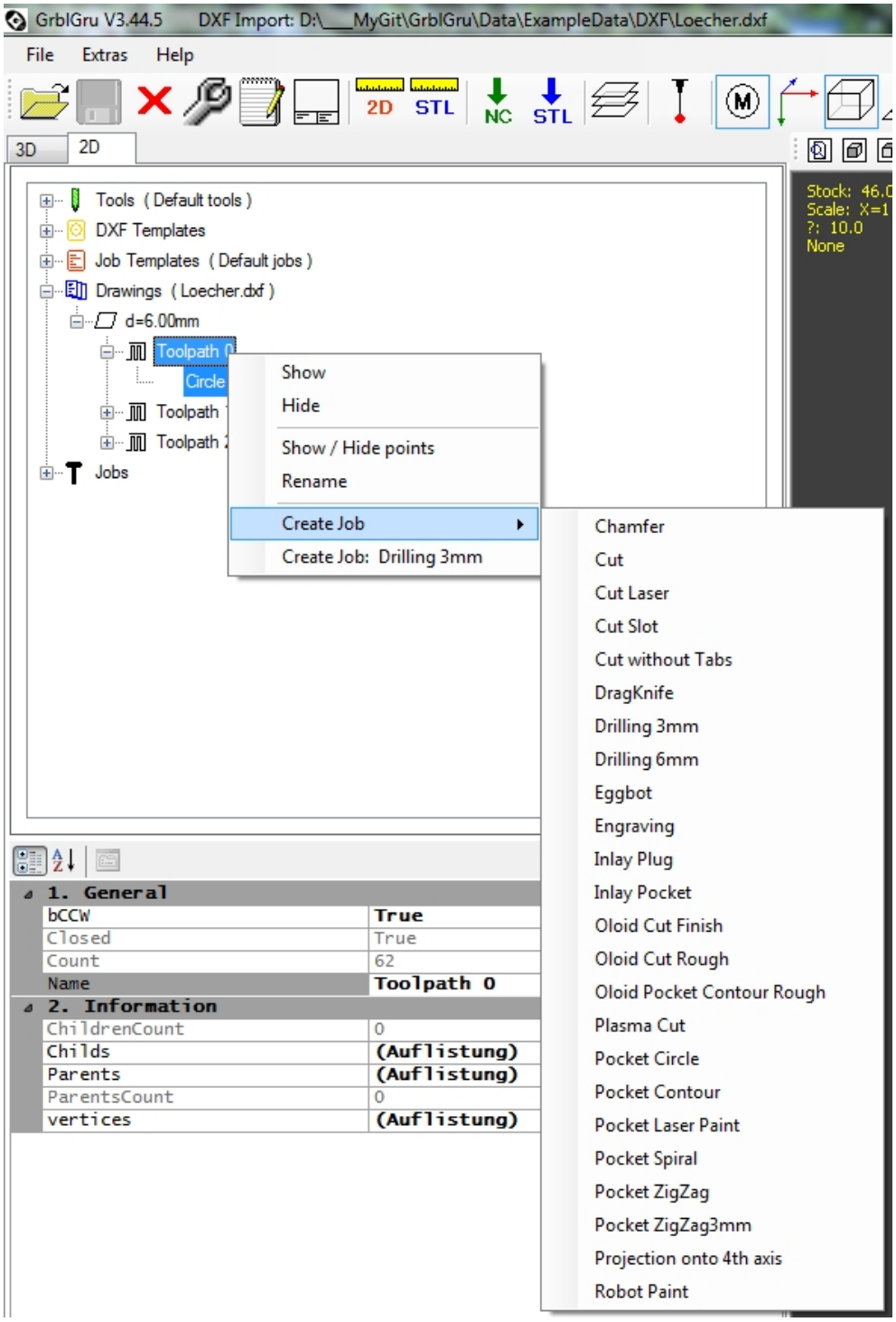


Figure 61: Job for geometry data of a layer

Here all geometry data from all layers are provided with a job. When clicking on the tool path, the corresponding area is highlighted in red in the graphics window.

5.3.3 In the Tree View: Layer Name

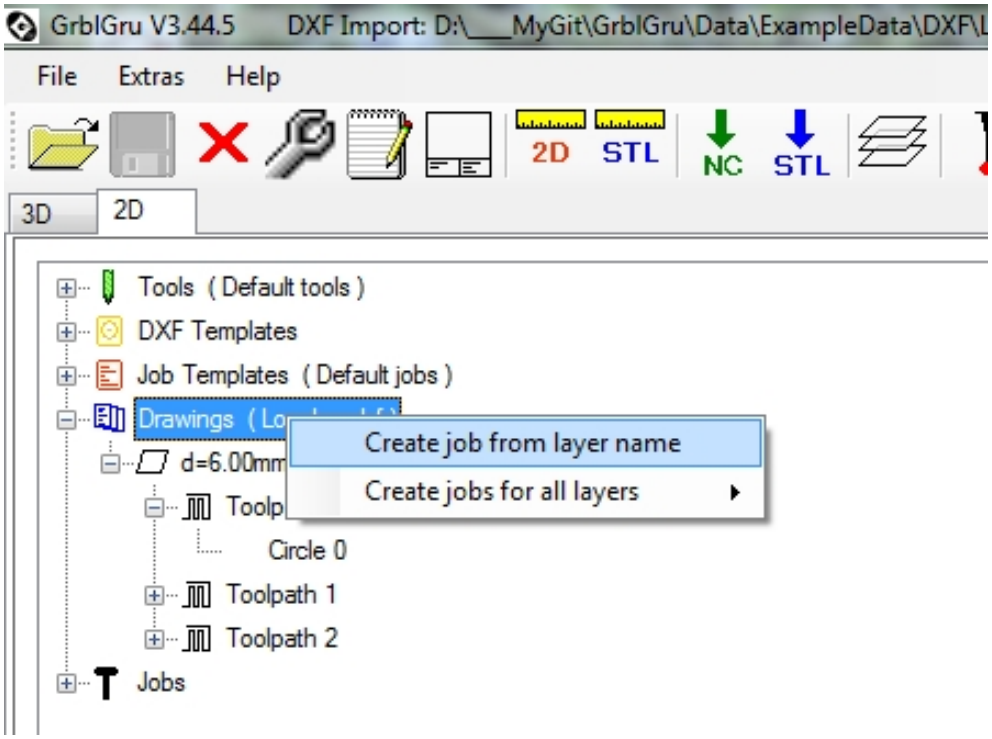


Figure 62: Job for geometry data of a Layers

An extremely effective method for lazy people (like me). Jobs are automatically assigned based on the Layer names used..

For example, suppose you have similar tasks to repeat a number of times; parts that need to be cut out, areas where pockets have to be cut or places to drill. For this you have created the templates 'Cut', 'Pocket' and 'Hole', which you now have to assign to the individual areas of the graphic. If you have to do these frequently, it makes sense to do this during design in the 2D editor. With all 2D editors there is the possibility to draw on different Layers and also to name them..

If you give the Names of your Templates to your Layers, GrblGru will automatically create corresponding Jobs from them.

5.3.4 In the Graphics Window: Pull up the window

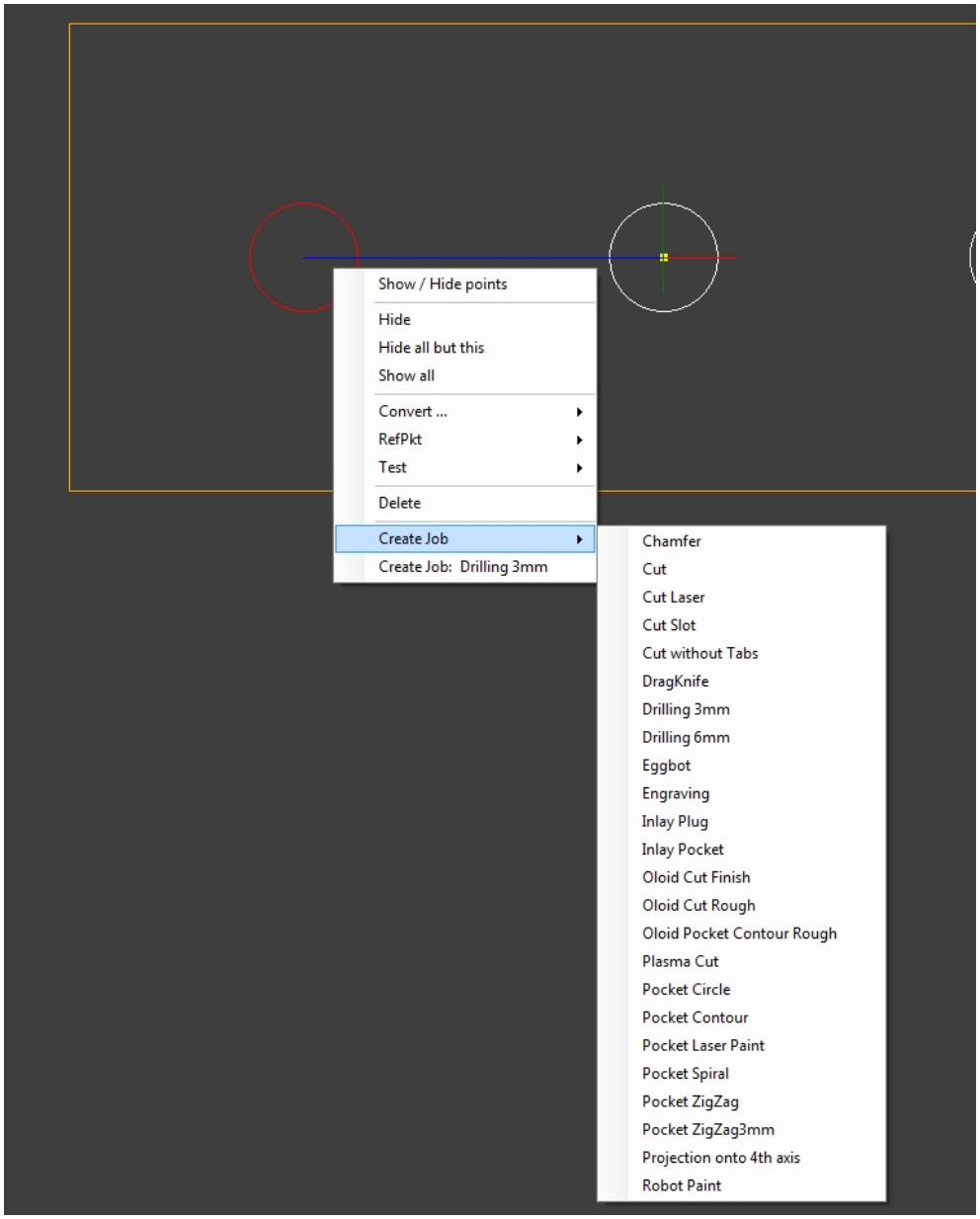


Figure 63: Mark an area with a window

With the Left mouse button you can create a window by Click-Hold and drawing from Upper left to Lower Right. The paths in it are highlighted in Red. A subsequent click with the Right mouse button opens the Context Menu for creating a Job.

5.3.5 In the Graphics Window: Click on Paths

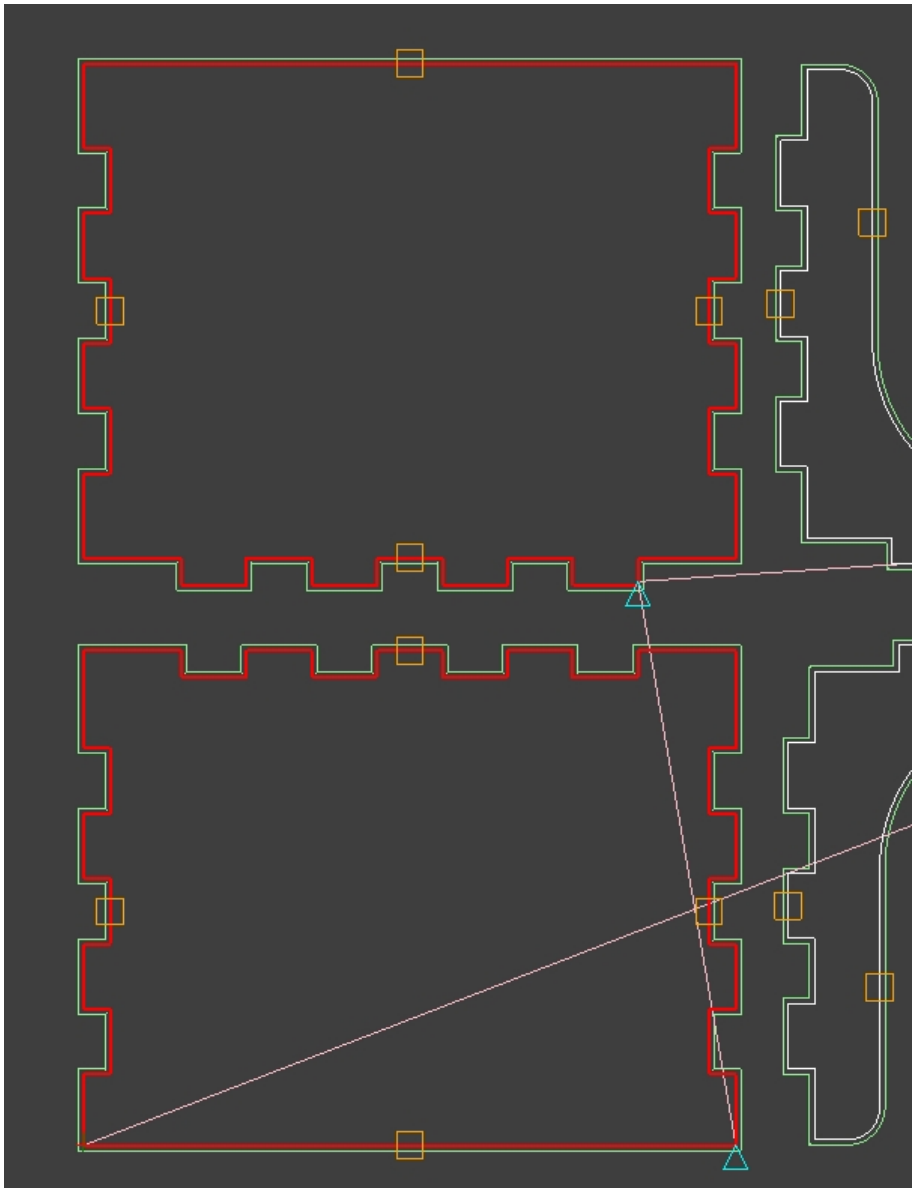


Figure 64: Direct Select Multiple Paths

With the left mouse button, paths can be selected directly. If several paths are to be selected, press and Hold the CTRL key while clicking the Left mouse button. Selected Paths are highlighted in Red. A subsequent click with the right mouse button opens the Context Menu for creating a Job.

5.3.6 Last Used Job

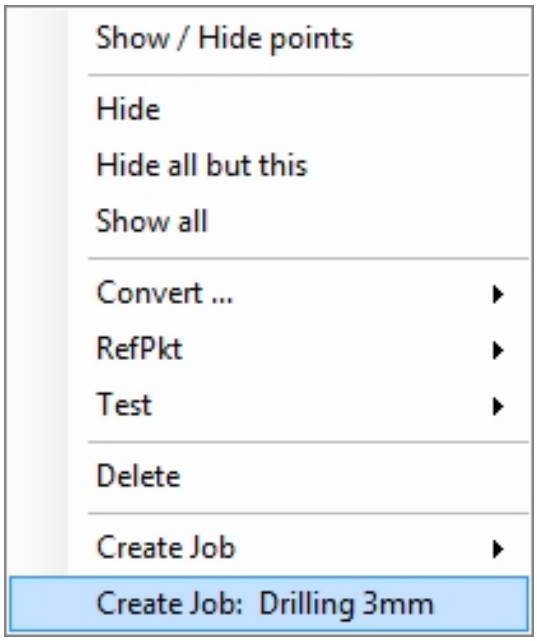


Figure 65: Quick selection of the last used job

In the dialog box of the Context Menu, the Last Job selection is made available on the last line as a Quick selection. This saves the selection in the second dialog if you can select the same job in succession.

5.4 Options to Mark Elements in the 2D view

In order to assign a process to the objects of a loaded DXF file (ie create a Job), you must be able to select individual objects. Basically, this can be done either in the tree view or in the 2D graphics. There is an interaction between these two. For example, if you mark a path in the graphic, the corresponding branch is also marked in the tree view. Likewise item marked in the tree view is also immediately highlighted in the graphic.

Before I get to the various possibilities of Marking, I would like to clarify the two terms.

- **Element**
  As an element, I refer to a line, arc, circle, arc, and so on.
- **Path**
  As a path I define a polyline as consisting of Contiguous or Joined Elements. This means that there is no gap between any 2 elements. When the end of a path is connected to the beginning, I'm talking about a closed path.

Selection in the graphics window is done by clicking with the left mouse button in conjunction with the 3 special keys SHIFT, CTRL and ALT.

- **No special key is pressed**
  Left Clicking on an Element marks the complete path in which the Element is located. The Marking of all other paths and elements will be excluded.
- **CTRL key + Left Click**
  A click on an item to move the marker of the complete path where the item is located. This means that the marking of a previously marked path will be deleted and an unmarked one will be marked.
  All previously marked paths keep their condition!
- **Alt key + Left Click**
  Here not the complete paths, but only the individual clicked elements are marked. The marking of all other paths and elements will be excluded.
- **ALT and CTRL key Left Click**
  Clicking on a marked item will change the state of the selected element. This means that the marking of a previously marked element is deleted and a non-marked element is now marked. All other previously marked elements keep their condition!

This may sound difficult at first glance, but it is not. For elements, use the ALT key, but not for paths. If you want to collect elements or paths you use CTRL. (as with marking in Explorer)

5.5 DXF templates

DXF templates are configurable generators of DXF drawings . They are very simple use e.g . The Rectangle Generator, which generates a drawing of a rounded rectangle after entering the Width , Height, and Radius parameters . But there are also, for example a Gear template, the Box and Clamping Board.

In each Template changing the values in the Property Sheet immediately makes a corresponding change of the graphical display. Experiment with this feature.

The following are the currently existing DXF templates are listed below :

- Rectangle
- Triangle
- Ellipse
- Breadboard
- Spur gear
- Box
- Bolt Circle Calculator
- Core Hole Making
- Puzzle Box
- Clamping Board

5.5.1 Rectangle

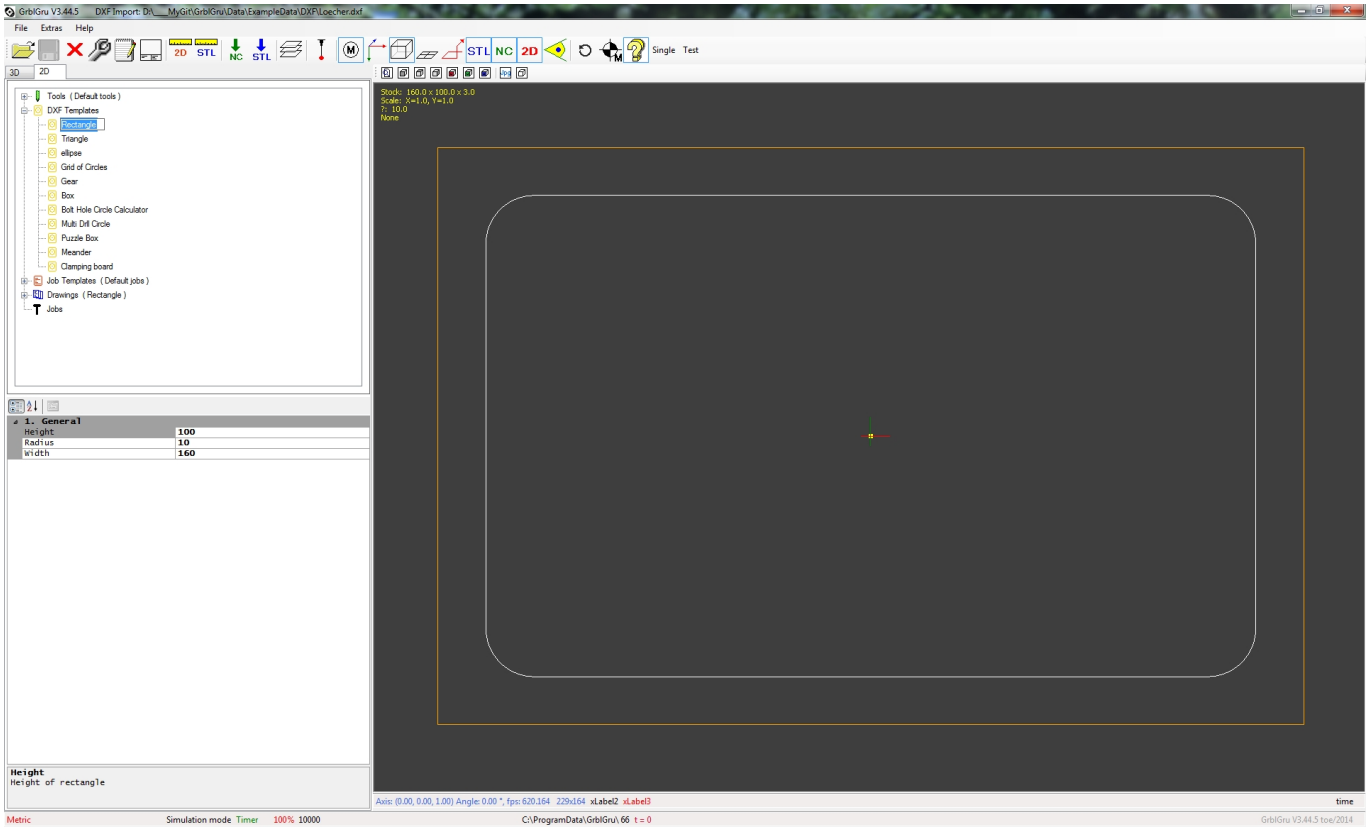


Figure 66: Template Rectangle

Simple template to make a Rectangle element or Box.

5.5.2 Triangle

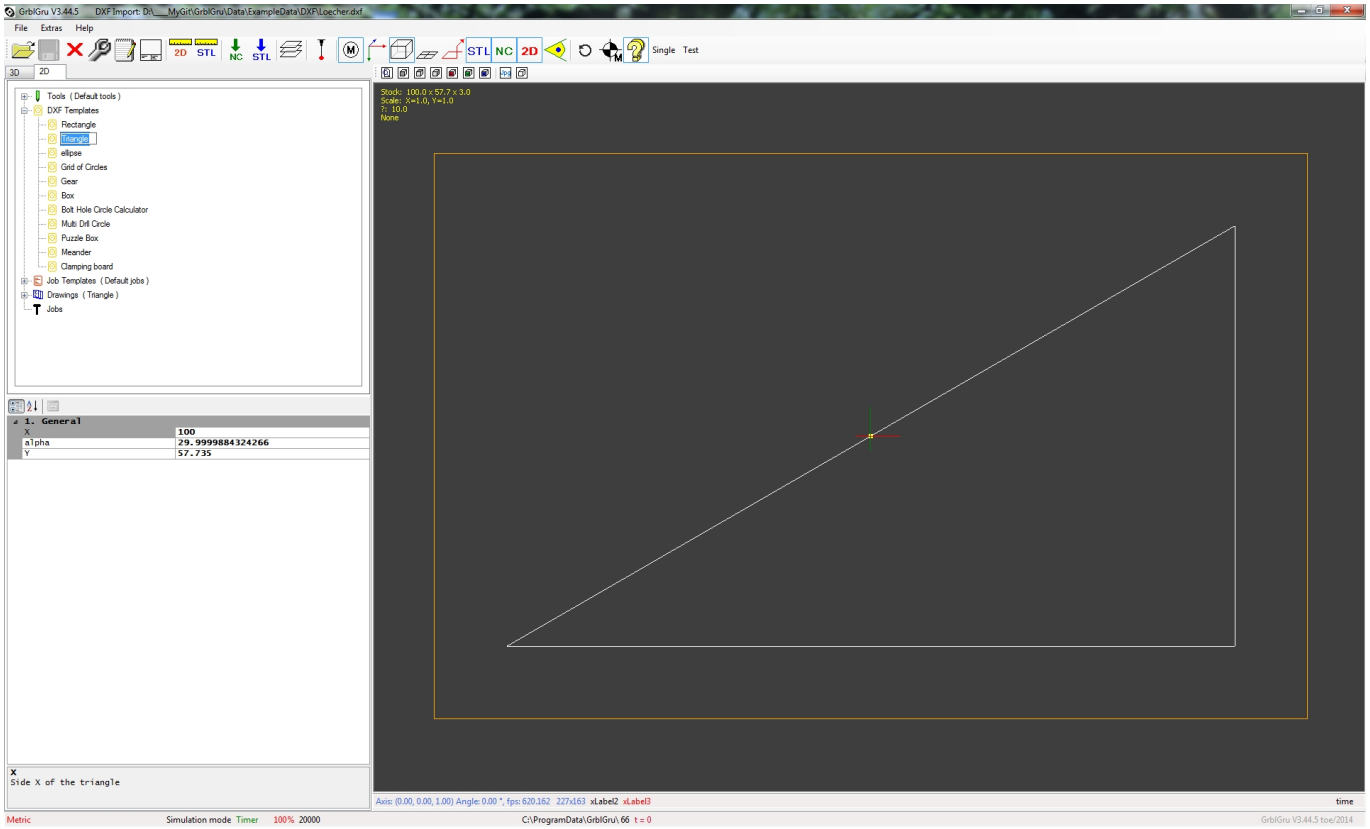


Figure 67: Template Triangle - Simple Triangle

Simple template to cut out a port or to mill a pocket

5.5.3 Ellipse

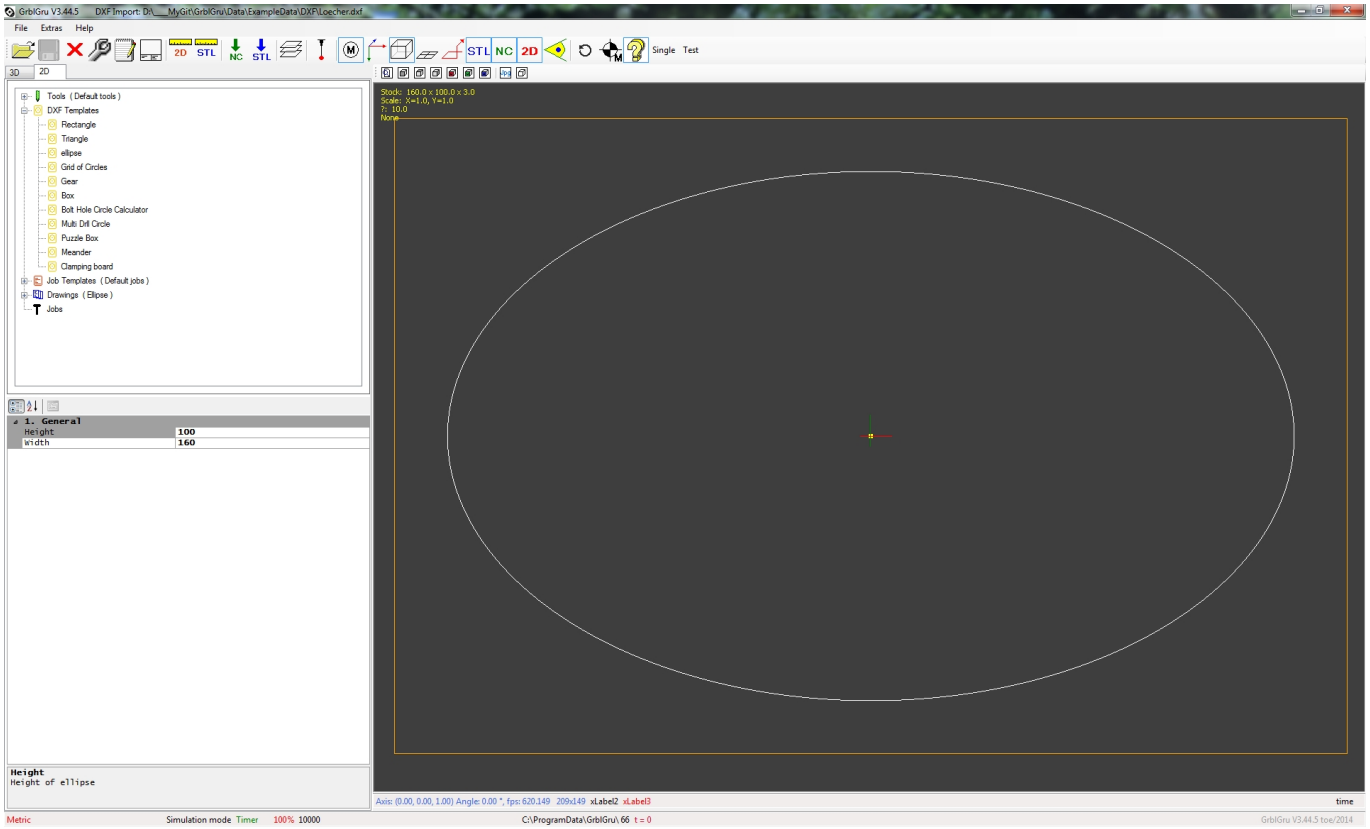


Figure 68: Template Ellipse

To create a door sign for example

5.5.4 Breadboard

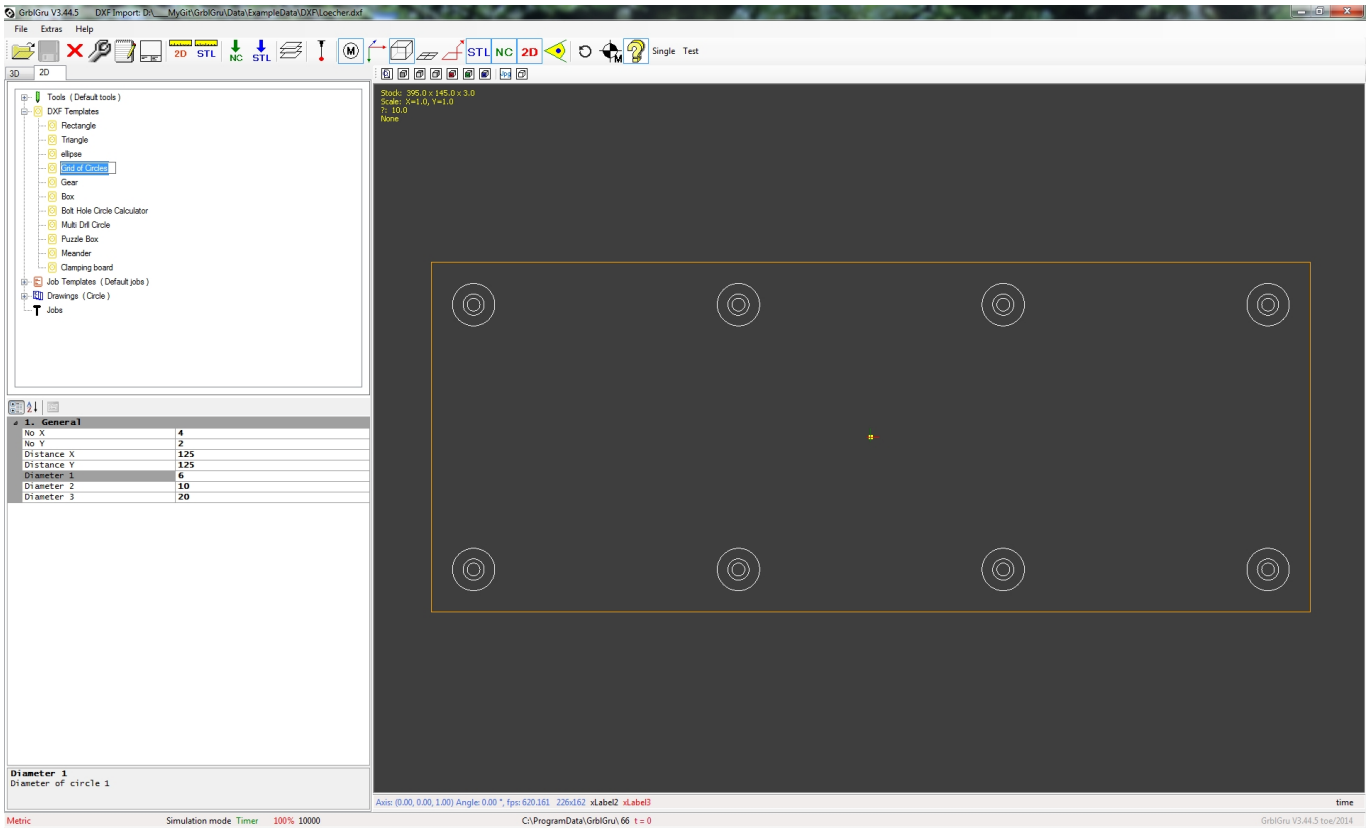


Figure 69: Hole pattern

Can used be to drill one or several nested holes. By entering values in the X and Y direction you make a grid of holes. This could be used to produce a spoil board for your CNC. It can also specify the diameters of up to three nested holes. This can make a bored hole, and at the same time the necessary recesses for a screw head. For each circle a depth and size can be input.

5.5.5 Spur gear

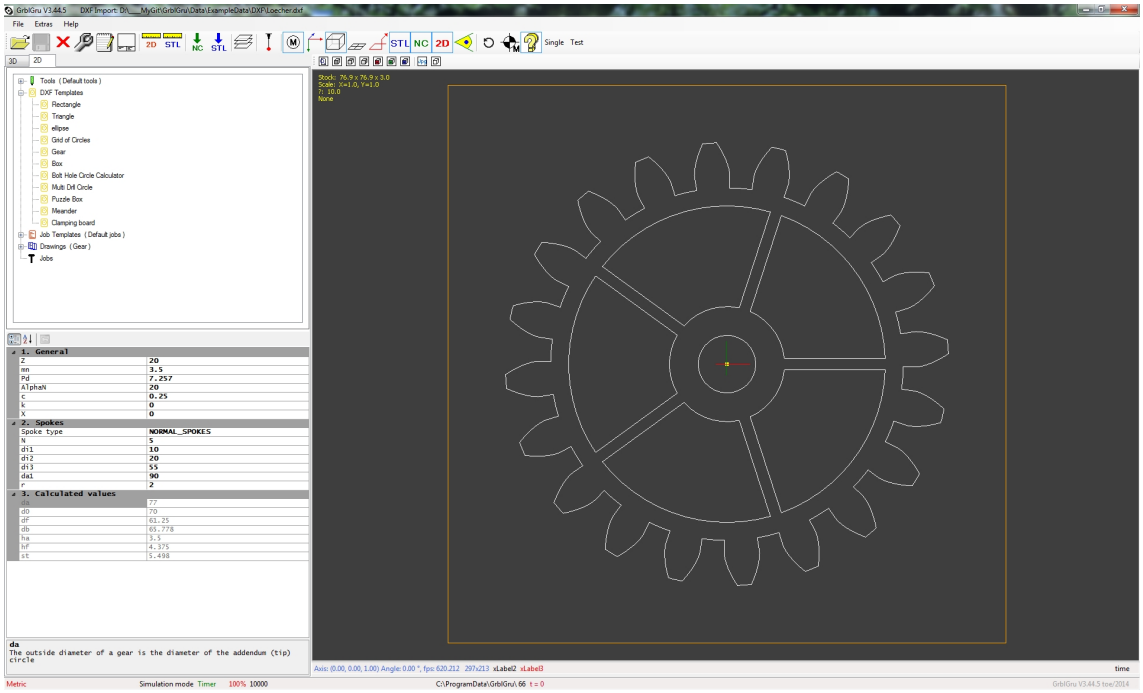


Figure 70: Template Spur Gear

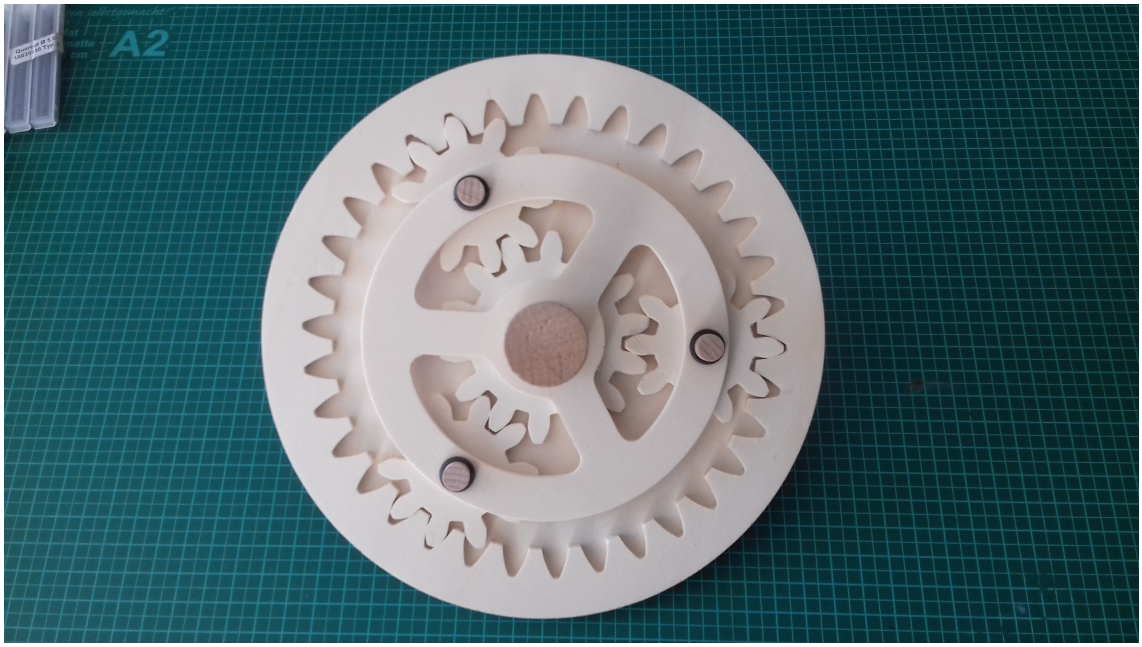


Figure 71: Planetary gear

Relatively complex template for the production of Spur Gears. You can interpret it with complete spur gears . Take time to make a demo planetary gear shown. For this purpose there is an video below `https://youtu.be/l9u2lKB3kMs` that shows you how with the help of beer cans. :)

5.5.6 Box

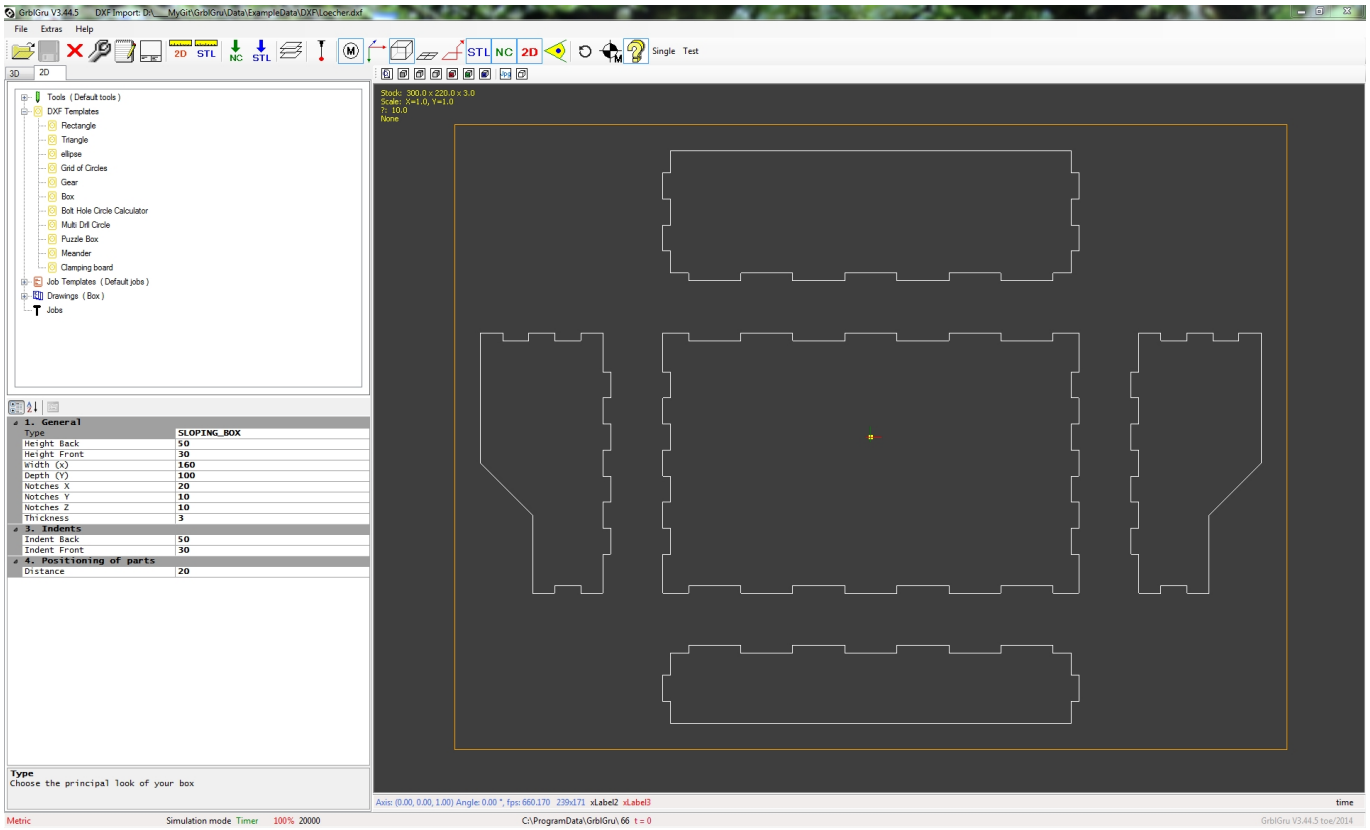


Figure 72: Template Box

THE Classic Template . Very helpful to design and make finger joint Boxes.

5.5.7 Bolt Circle Calculator

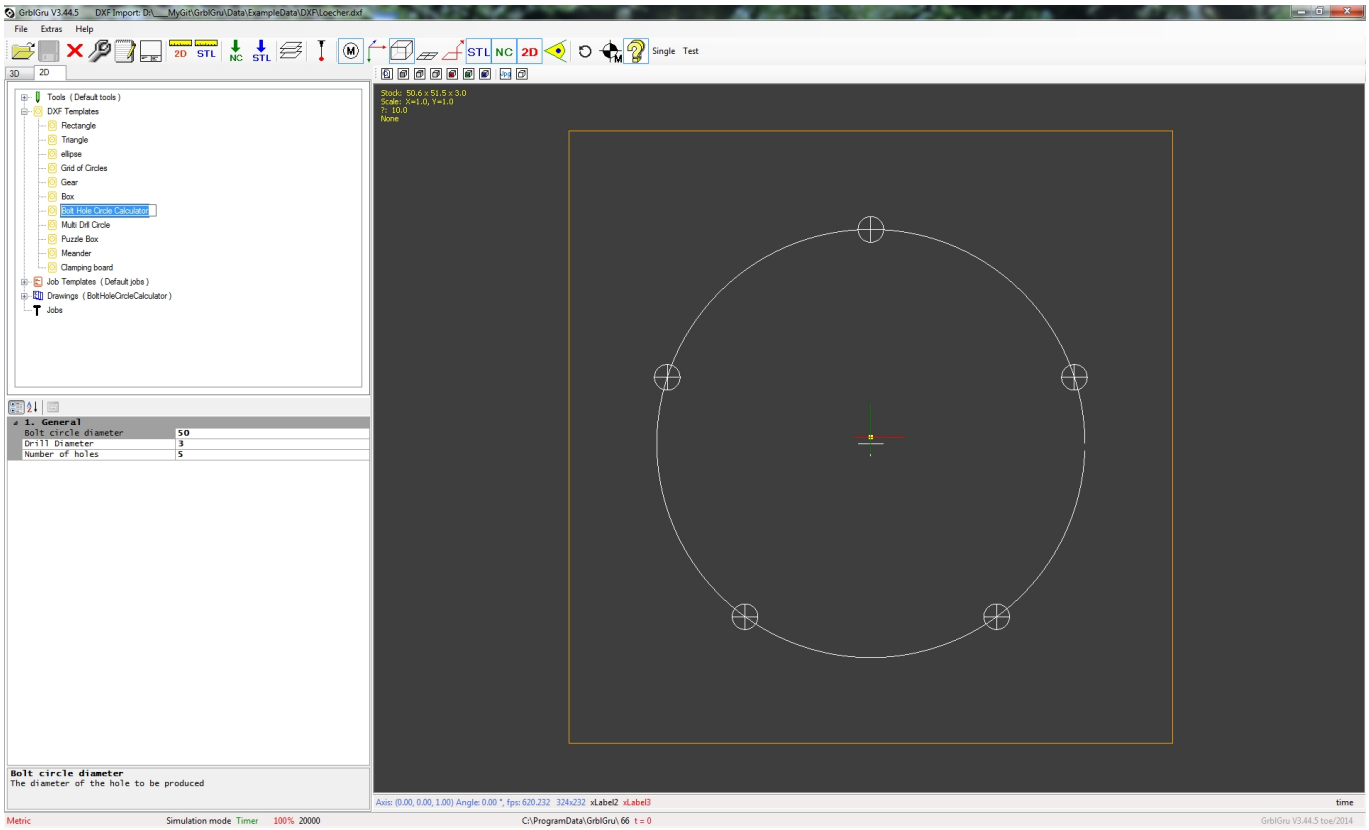


Figure 73: Template for Bolt Hole Circle calculato

Very practical eg. in the manufacture of different holes in the Model.

5.5.8 Multi Drill Circle

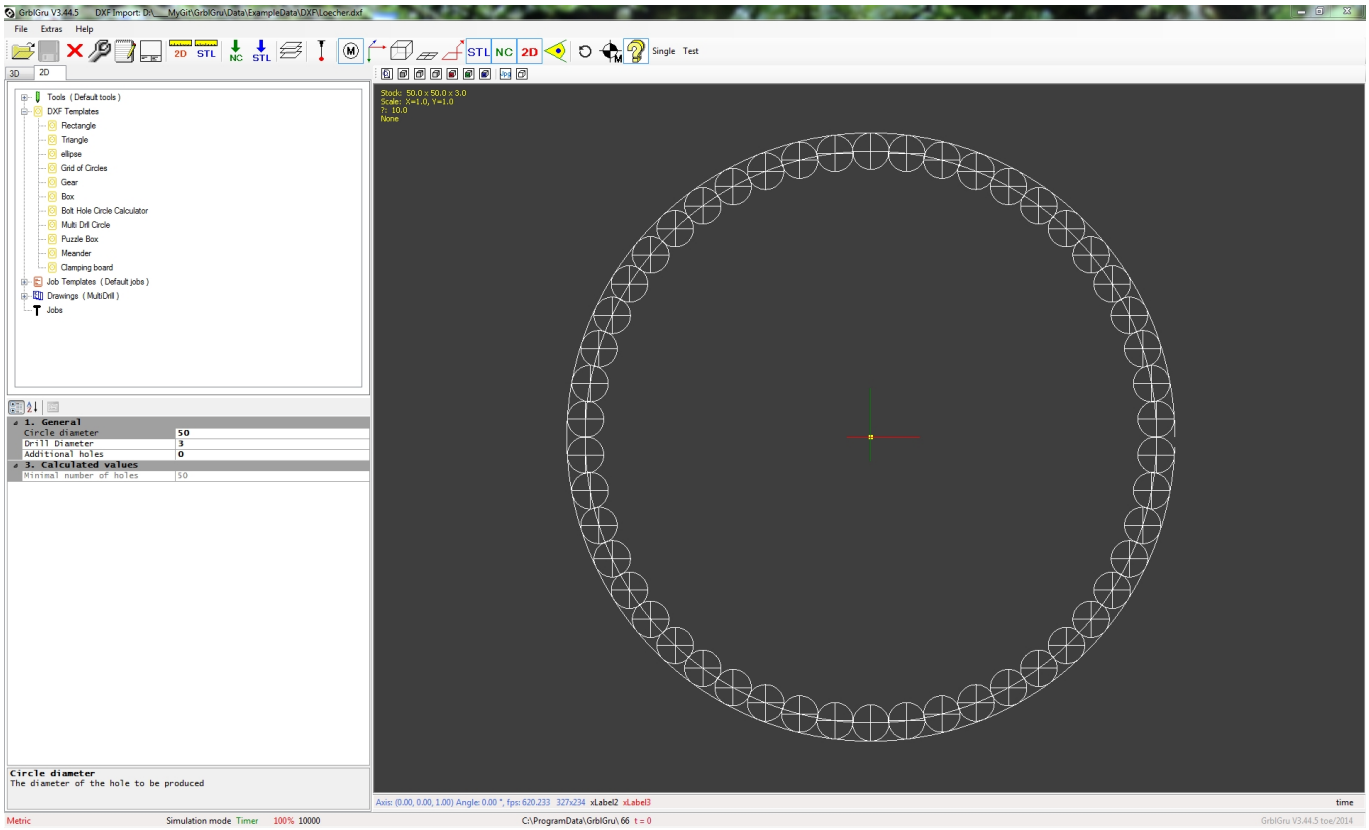


Figure 74: Template Core hole production

Can used for the positioning of small bore holes to help make a much larger core hole.

5.5.9 Puzzle Box

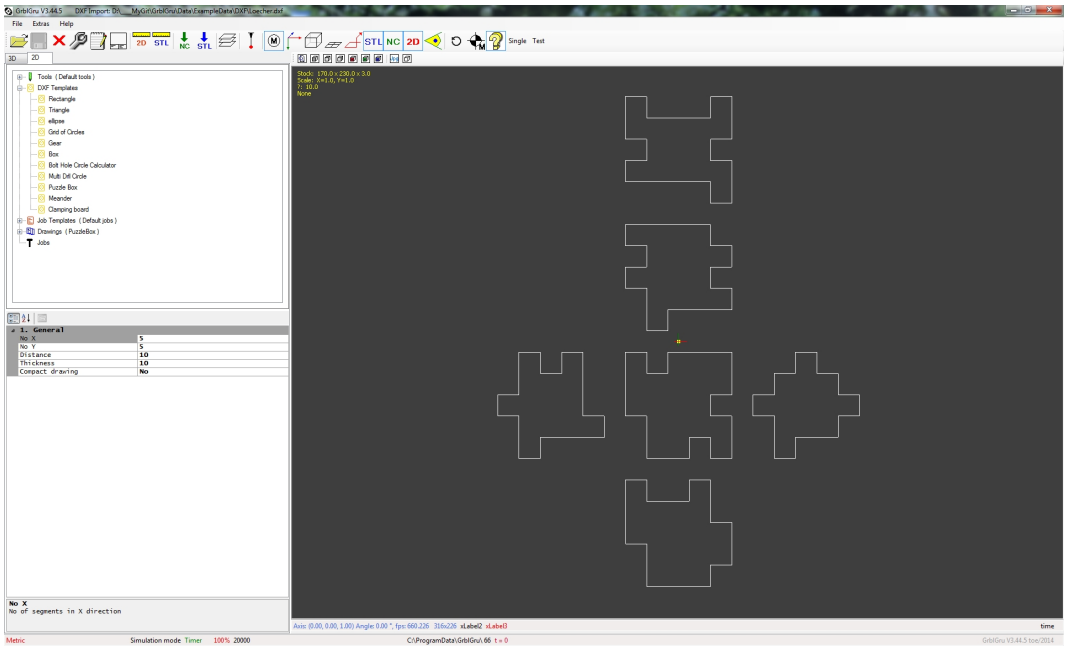


Figure 75: Template Puzzle Box

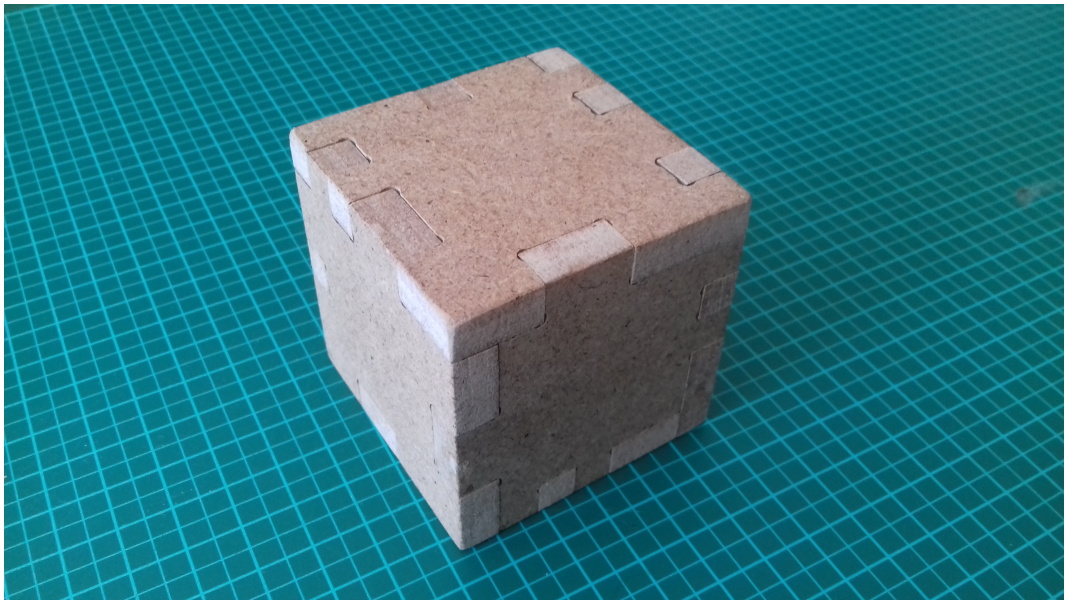


Figure 76: Test box made of 8mm MDF

My recommendation for a creative packaging of gifts . The generator is calculates a new puzzle pattern each time..

My tip: Make yourself a record of the solution and never mix parts of 2 boxes. They will never be assembled again? :)

There is also a short video of Puzzle Box: :) `https://youtu.be/214Bczo-8uw`

5.6 USEFUL Free programs to use with your CNC

I have here a small list of programs that I quiet conscience further recommend can :

- **GRBL**
  3-axis operating system for Arduino UNO. The CNC operating system for public use.
- **g2core**
  5-axis operating system for Arduino DUE
- **Gimp**
  A great program for editing of graphics of all kinds. It is worth getting to know all it?s functions
- **Inkscape**
  An editor for editing SVG files
- **OpenScad**
  A very useful 3D CAD program. Can be used with ***GrblGru***
- **7-Zip**
  A program for packing and unpacking files
- **F-Engrave**
  THE really good program for Engraving . There is nothing better !
- **Notepad++**
  A great GCode editor - Can display G code in color
- **IcoFX**
  A program for editing bitmaps and icons

5.7 Notepad++, a Editor for G-Codee

Press the  button on the toolbar and the generated G-code with the Windows Notepad Editor window opens. You can also specify your favourite Editor in settings, so ***GrblGru*** can use that to display the GCodes created. Notepad++ for example.

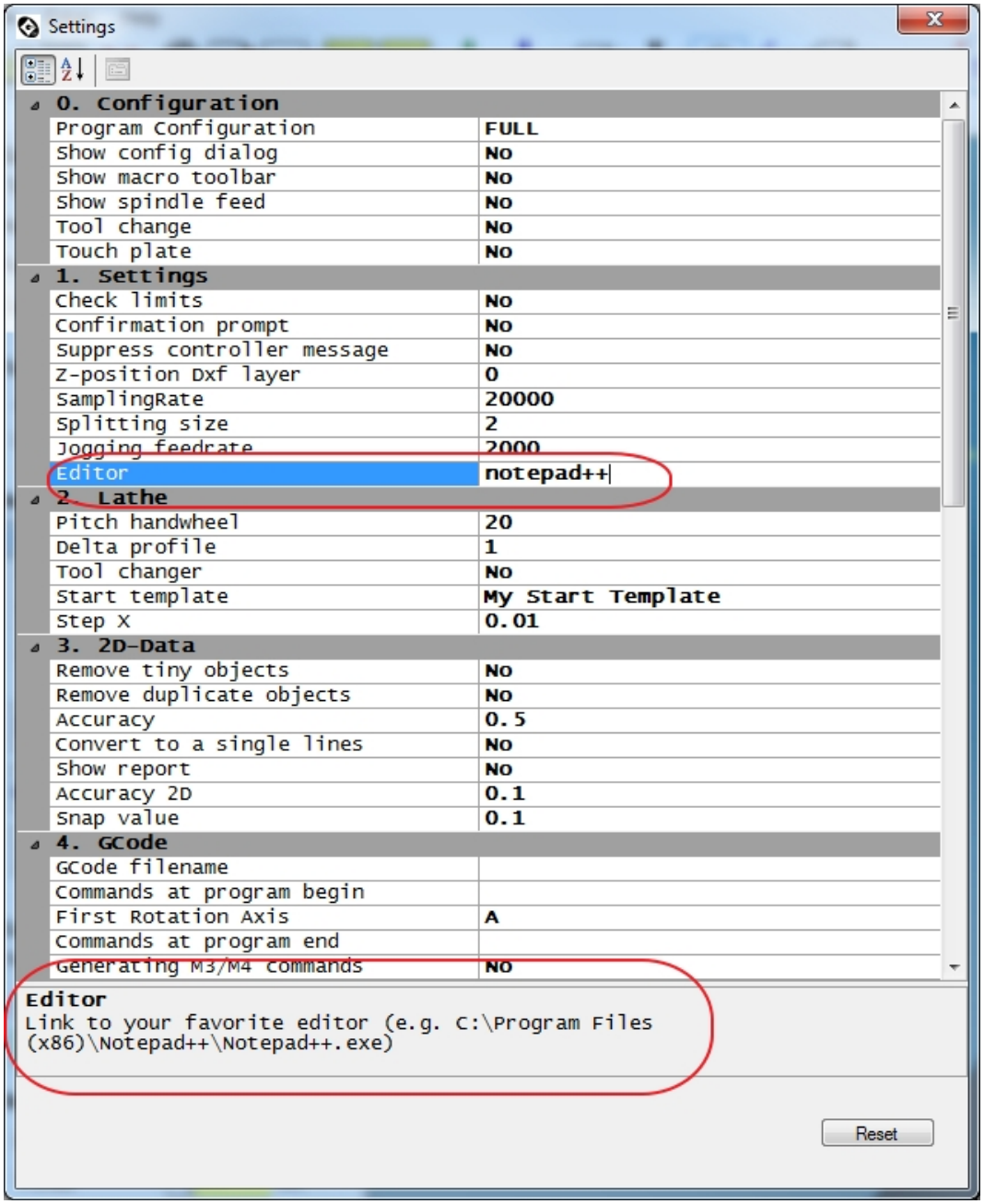


Figure 77: Entry in Settings under Extras on the Toolbar

Personally, I always use Notepad ++ because it gives you the ability to colour code certain key words.

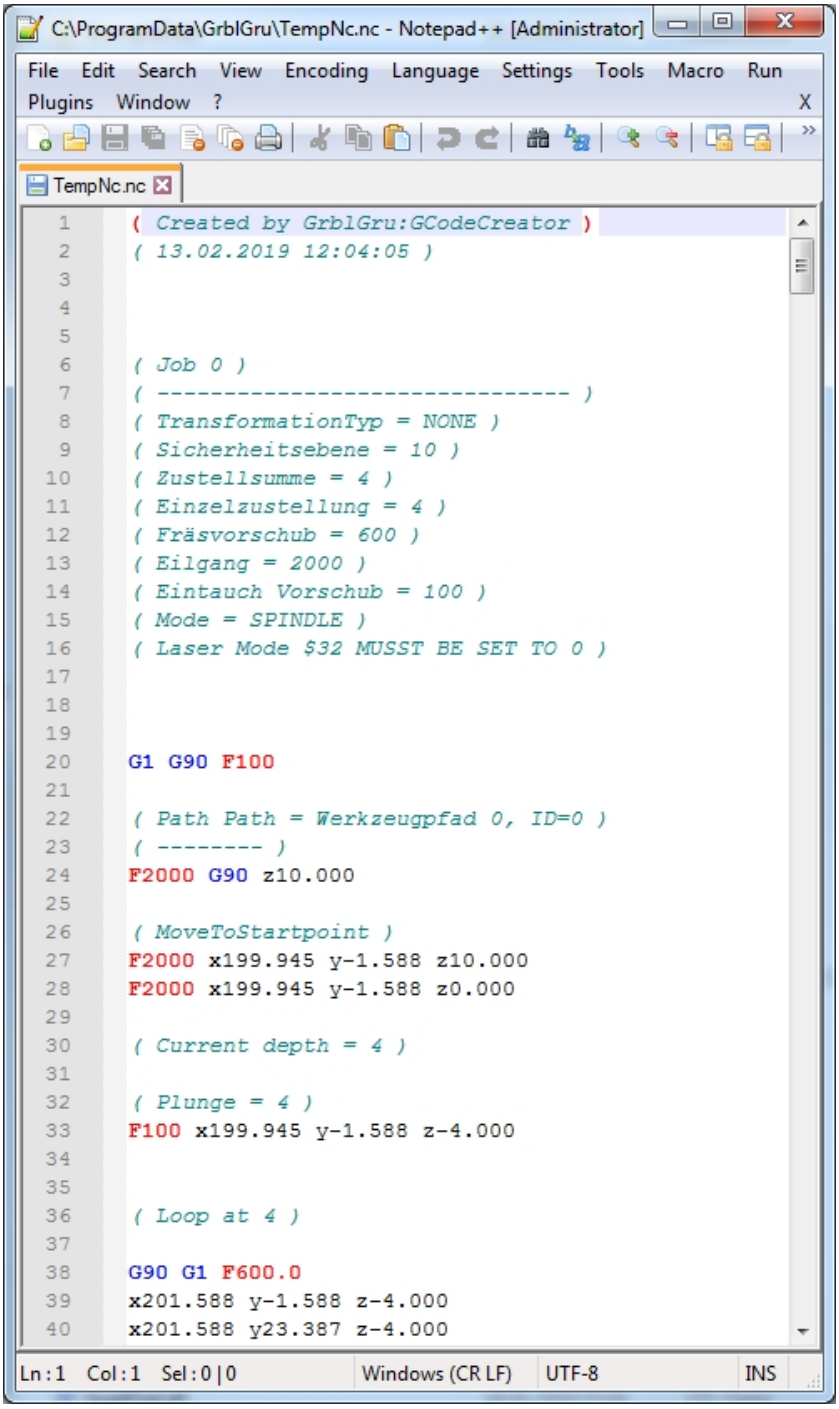


Figure 78: Display of a GCode file in Notepad ++

For the GGode generated by ***GrblGru*** I wrote a special custom language file. This file named ***GrblGru*** GCode.xml. It can be imported into Notepad ++ under the menu item Language-Import. When ***GrblGru*** is installed, the file is stored in the installation directory.
`C:\ProgramFiles(x86)\toe\GrblGru`

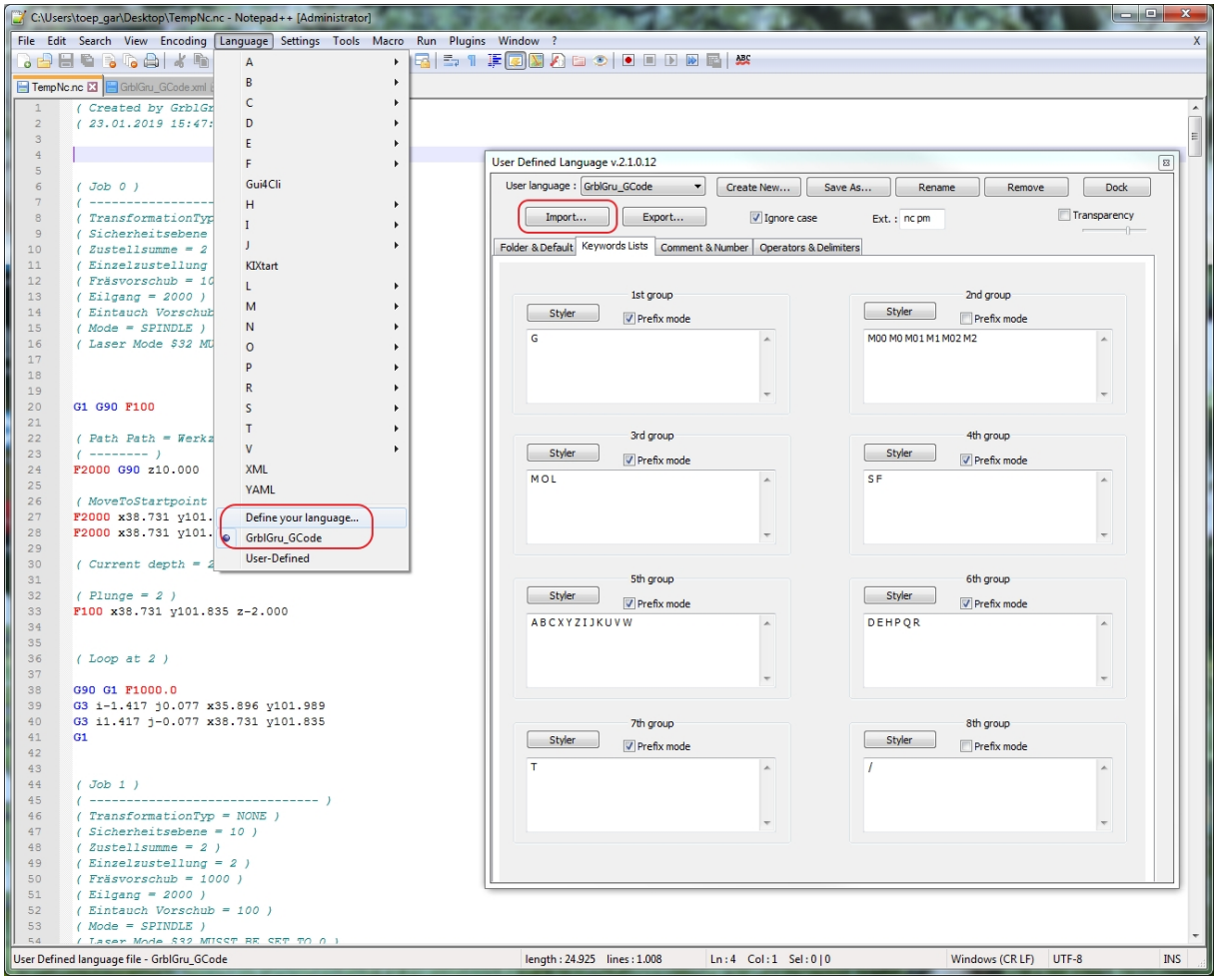


Figure 79: Importing the custom language file

GrblGru uses some files with a so-called Ini structure. This means that these files store their contents in chapters with keywords. To display these files, there is the prepared language 'Ini file?. So that Notepad ++ can open a file with the extension 'dat' immediately with the corresponding language, there is the possibility to fix a file extension eg '.dat' with one language. The necessary dialogue can be found in the Extras Menu under 'Settings-Style'

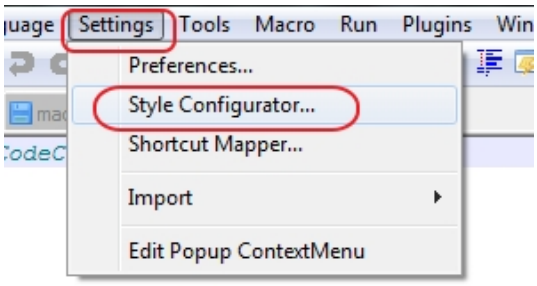


Figure 80: Access to dialogue

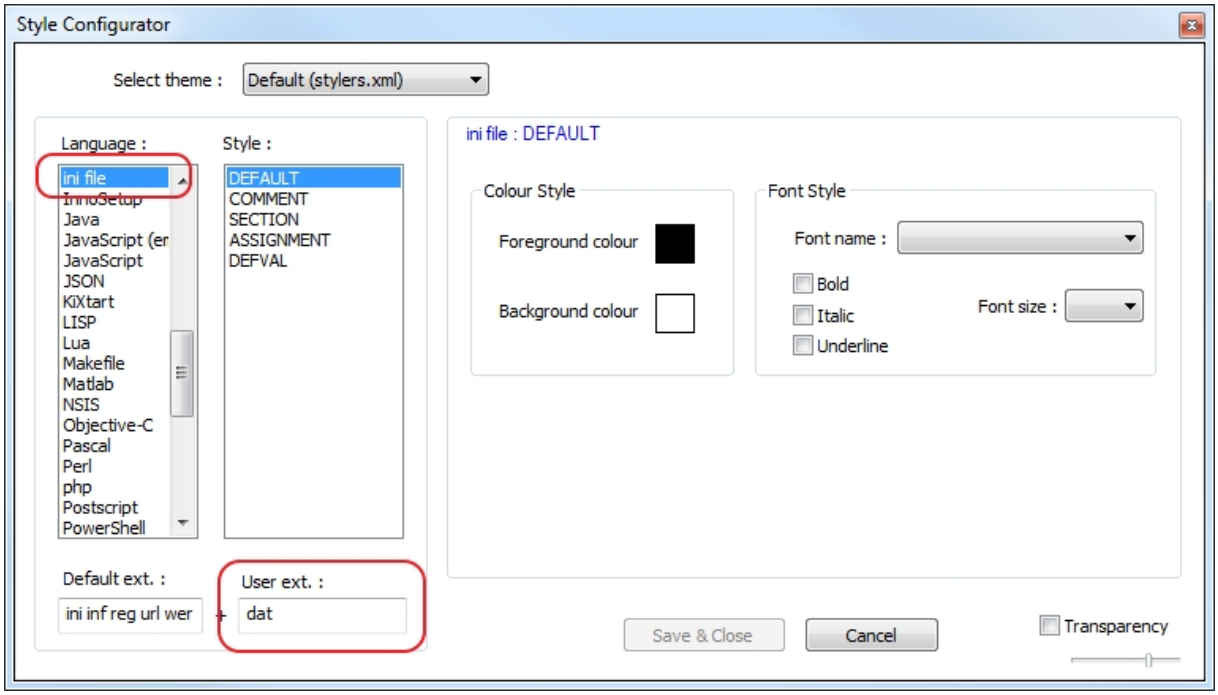


Figure 81: Assignment of a file extension to a language

6 Macros

There are situations in which you want to call up certain processes at the push of a button. For example, a hole can be drilled very carefully and with several chip removal operations because the material requires this. In this case, the best result is achieved if the GCode is created by hand, tailor-made for the required process.

So that this function is always available at the push of a button, ***GrblGru*** offers an easy way to create macros. To do this, the macro bar must first be switched on on the settings page.

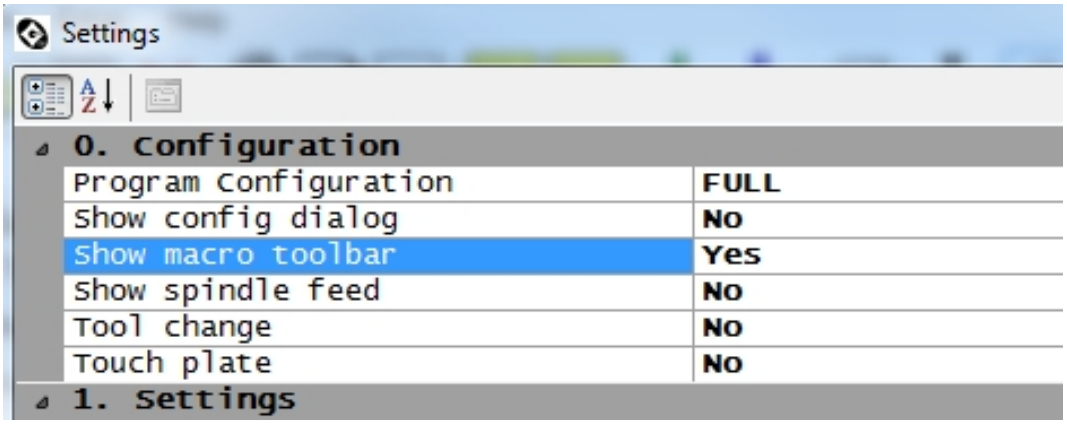


Figure 82: Switch on the Macrobar

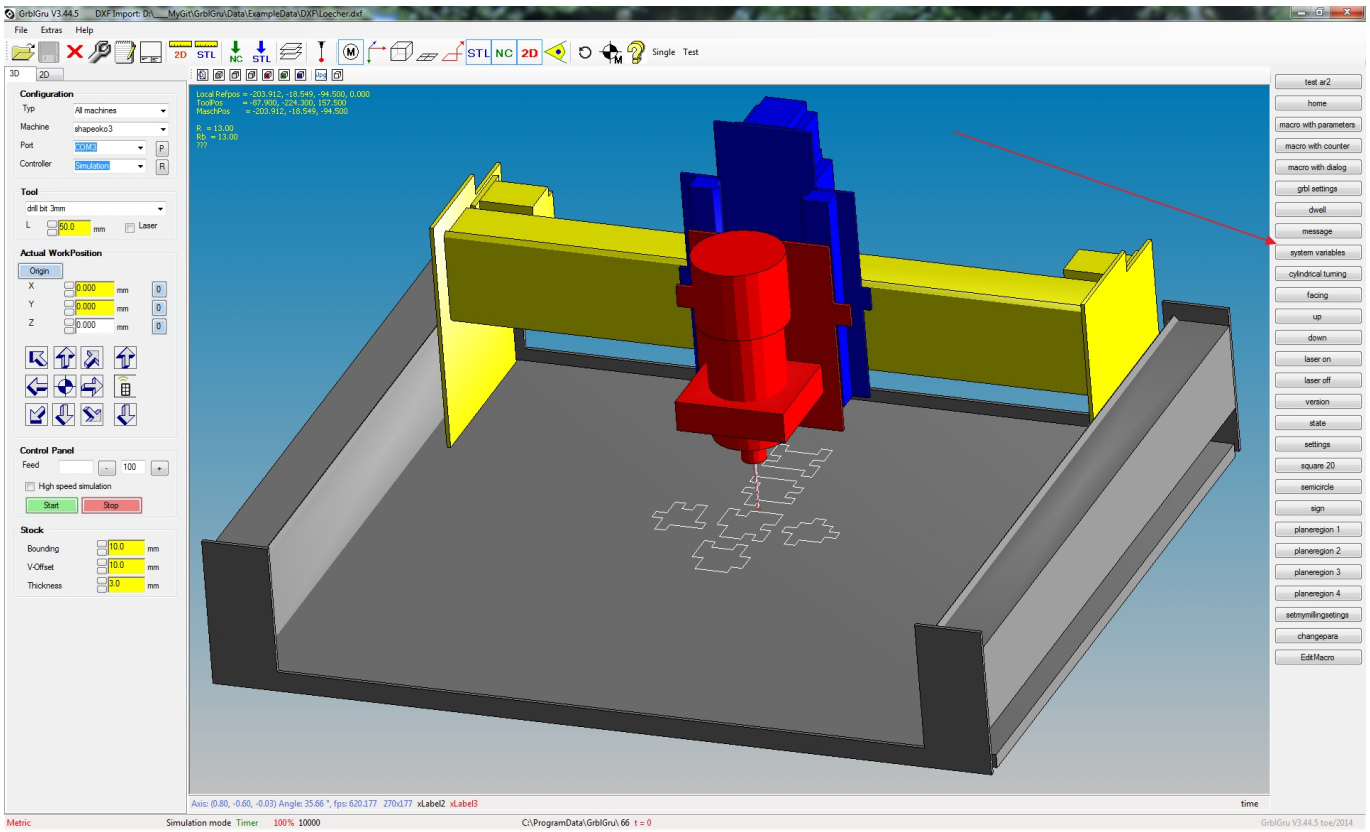


Figure 83: Macrobar at the right side

By pressing the lower button with the name 'EditMacro' the file Macro.dat is opened in the editor and can now be edited..

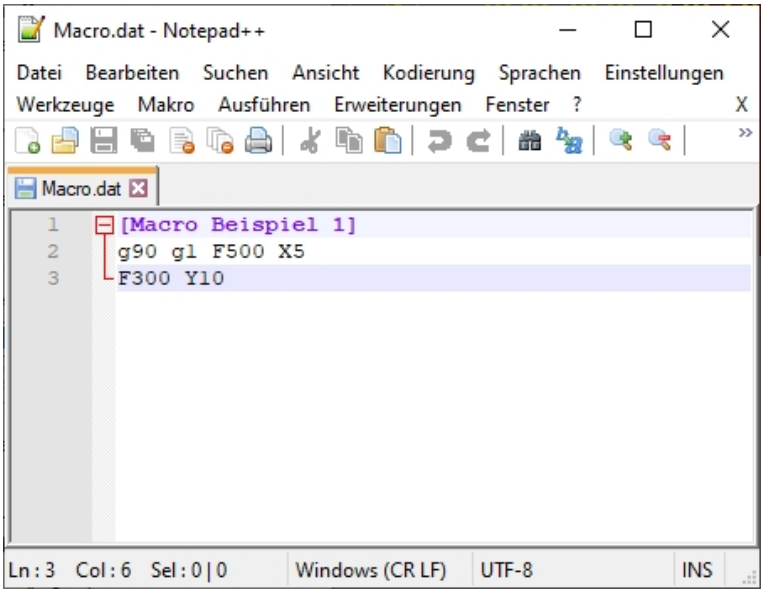


Figure 84: The open Editor

A macro always starts with the macro name in square brackets.
So e.g. [**Macro example 1**]. The name should be meaningful because it will be displayed on the button in the macro bar.

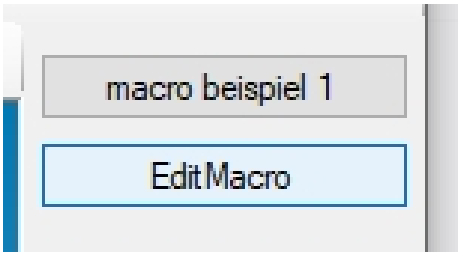


Figure 85: The buttons of the Macrobar

After that any number of lines with valid GCode syntax can follow.
After the input the editor is closed. After that ***GrblGru*** immediately updates the macro bar and the currently entered function can be called by pressing the corresponding button.
The macros are also available after a restart. You can create as many macros as fit on the screen.

6.1 Operating system Macros

Direct communication with the respective processor is to be mentioned as quite interesting application possibility of the macros. All operating systems Grbl, TinyG, Mega-5X and g2core offer the possibility to change movement commands and settings via commands. ***GrblGru*** provides the possibility of direct communication in the two controller windows.

So you can easily change several settings with the help of macros. The macro listed below sets e.g. all settings of **Grbl**.

[SetMyMillingSetings]
$0=10
$1=25
$2=0
$3=0
$4=0
$5=0
$6=0
$10=0
$11=0.010
$12=0.002
$13=0
$20=0
$21=0
$22=0
$23=0
$24=25.000
$25=500.000
$26=250
$27=1.000
$30=1000
$31=0
$32=0
$100=100.000
$101=100.000
$102=50.000
$110=500.000
$111=500.000
$112=500.000
$120=10.000
$121=10.000
$122=10.000
$130=200.000
$131=200.000
$132=200.000

The following macros may also be helpful

[Version]
$I

[State]
$G

[Settings]
$$

6.2 Advanced Macro functionality

The advanced ***GrblGru*** Macro language is very powerful. It allows the use of parameters, loops and conditional jumps. There is also the possibility to display parameter values in a dialog box. It is also possible to enter values in a dialog box.

Parameters must begin with a '#' e.g. #Length.

- **Parameter Assignments**
  #Length = 12.5
  #Length = #Diameter

- **Mathematical operations**
  #Jeff * 2
  #height / 2
  #Diameter + 2
  #Infeed - 2

- **Conditional jumps**
  if (#FeedRate > 50)
  {
  instructions ...
  }

- **Loops**
  while (#Counter > 0)
  {
  instructions ...
  }

Behavior control instructions

- **%Reload**
  The macros write the generated GCode to TempNc.nc exactly like a job does. If you want to use a macro, e.g. to switch only the laser on / off, the code previously in TempNc.nc will be overwritten. %reload saves the old code and writes it back after processing the macro.

- **%NoAutoRun**
  Normally macros are executed immediately with the keystroke. **'NoAutoRun'** causes that the GCode **not** is started directly. It must be started explicitly with the green 'Start' button.

Various dialog boxes are available for displaying and entering data:

- **%message**

  Example 1: %Message, This is a message without value

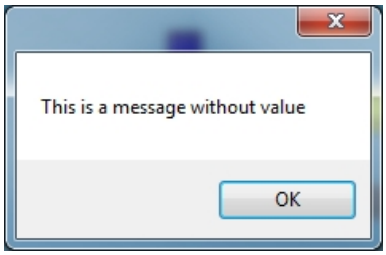


Figure 86:

  Example 2: %Message, Length = , #Length

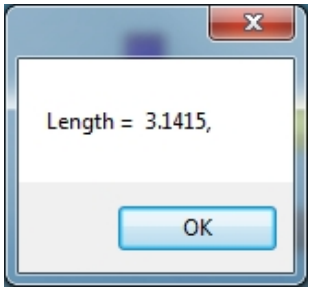


Figure 87:

- **%Input**

  %input, **cylindrical turning**, **Length of piece**, #Length, **Sum of Increments**, #SumOfIncrements, **Dept increment**, #DepthIncrement, **Cut feed rate**, #CutFeed, **Plunge feed rate**, #PlungeFeed, **Rapid feed rate**, #RapidFeed

  The command %Input exists for data input, which displays an input dialog in which values can be entered. These values are then assigned to the corresponding parameters and can be used in the macro.

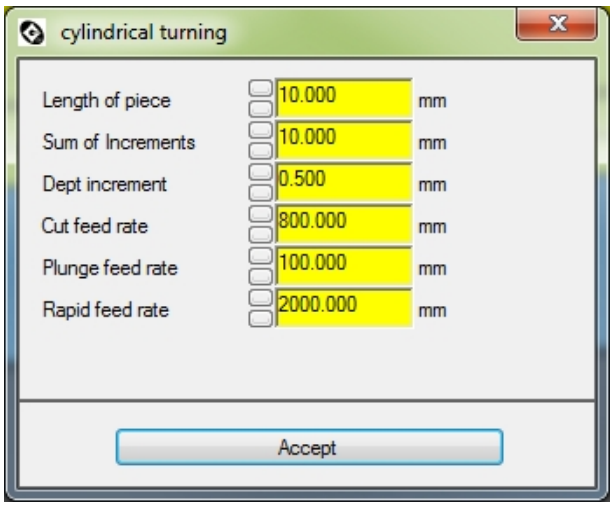


Figure 88: The input dialog

- **%variables**
  This command is used to display the system parameters and their contents.

  Example:

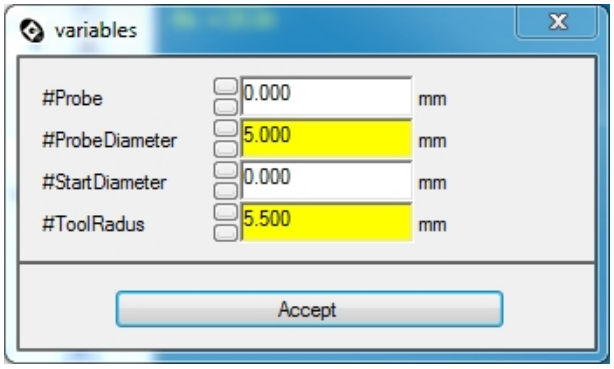


Figure 89: The system parameters

7 Administration of languages

GrblGru can manage many languages. Each language is saved in a separate file. The structure is very simple. Each row has 2 columns. The columns are separated by a ';' symbol. The 1st column contains the key used in the program. It always starts with a '#' or a '?'. The 2nd column contains the text in the respective national language.

```
#_author;   05.03.2019 / toe
#_language; Deutsch

#0configuration;    0. Konfiguration

#1general;  1. Allgemein
#1settings; 1. Einstellungen

#2axes; ?
#2data; 2. Daten
#2ddata;    Zeichnungen
#2info; 2. Information
#2lathe;    2. Drehbank
#3process;  3. Prozess
#2rough;    2. Schruppen
#2stl;  2. STL
#2stock;    2. Werkstück
#2scanning; 2. Messen
#2top;  2. Deckel
#2wheelspokes;  2. Speichen

#32ddata;   3. 2D-Daten
#3axeshelp; ?
#3calculatedvalues; 3. Berechnete Werte
#3finish;   3. Schlichten
#3indents;  3. Einrückungen
#3parameter;    3. Drehstahl-Parameter
#2spindlecontrol;   2. Art des Werkzeugs

#4extprofile;   4. Externes Profil
#4limit;    4. Begrenzungen
#4openscad; 4. OpenSCAD
#4positioning;  4. Teile Positionierung
#4tool; 4. Werkzeug

#5bridges;  5. Brücken (Stege)
#5colors;   5. Farben
#5iso;  5. Wendeschneidplatten
#5pockets;  5. Taschenräumung
#5refpkt;   5. Referenzpunkt
```

Figure 90: Part of Language German.txt

The order of lines within a language file is arbitrary. If you want to create a new language, it is best to copy the English language file English.txt and replace the English texts with the new texts. Then the file can be renamed accordingly when saved. GrblGru recognizes all existing language files at the start and offers them in the language selection dialog.

After a language has been edited, it makes sense to sort all the texts again alphabetically. This makes it much easier to find searched texts. This sorting can be done via the menu item Sort Text:

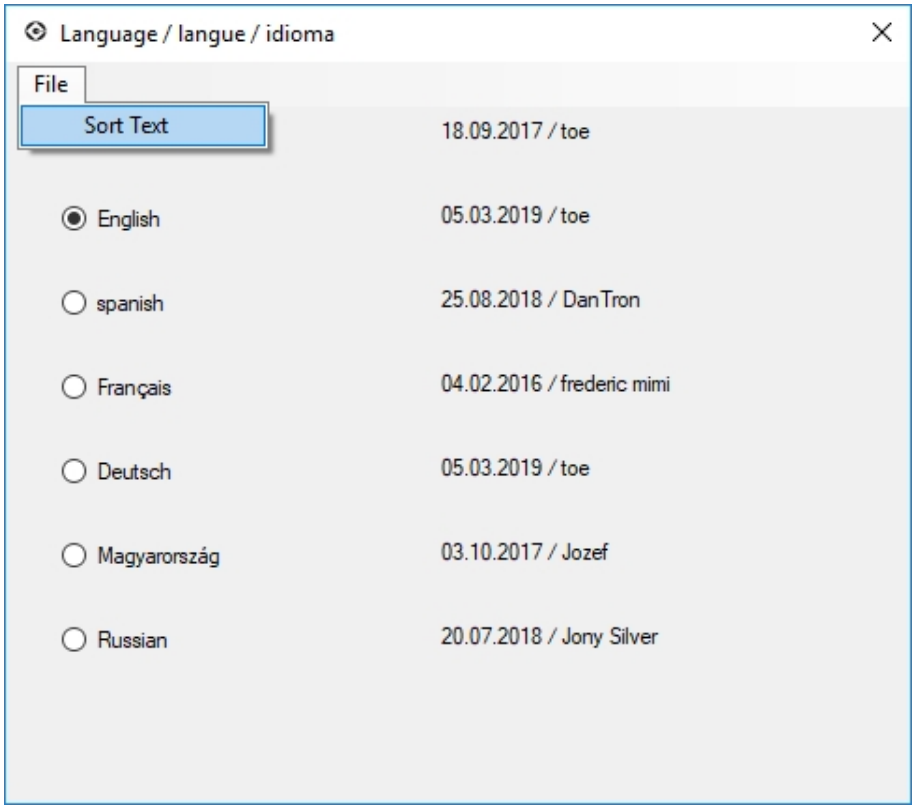


Figure 91: The Languages dialog

All language files are located in the folder `Program\DataGrblGru`

- Language_english.txt
- Language_espanol.txt
- Language_french.txt
- Language_german.txt
- Language_magyar.txt
- Language_PortuguesBrasileiro.txt
- Language_Russian.txt

8 How to Import your own 3D Machine Model

A special feature of ***GrblGru*** is the ability to import your own 3D models. These models are then used to graphically simulate the loaded G-code.

This chapter presents the necessary steps to import an existing 3D model.

8.1 Can *GrblGru* simulate my machine ?

If you can not find a corresponding model in the list of available machines, you can import your own 3D model. The condition for this, however, is that the type of the model already exists. The type determines the relations of the axes to each other. For example, whether the tool moves to the component, or the tool is fixed and moves the component to the tool. Is the Z-axis traversed when X is processed? Does an A-axis rotation affect the B-axis or not?

As an example for two completely different types, the following machines are listed.

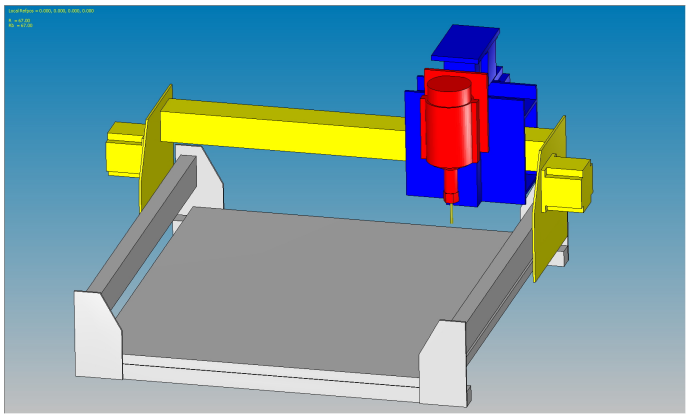


Figure 92: Example of 'ShapeOko' type

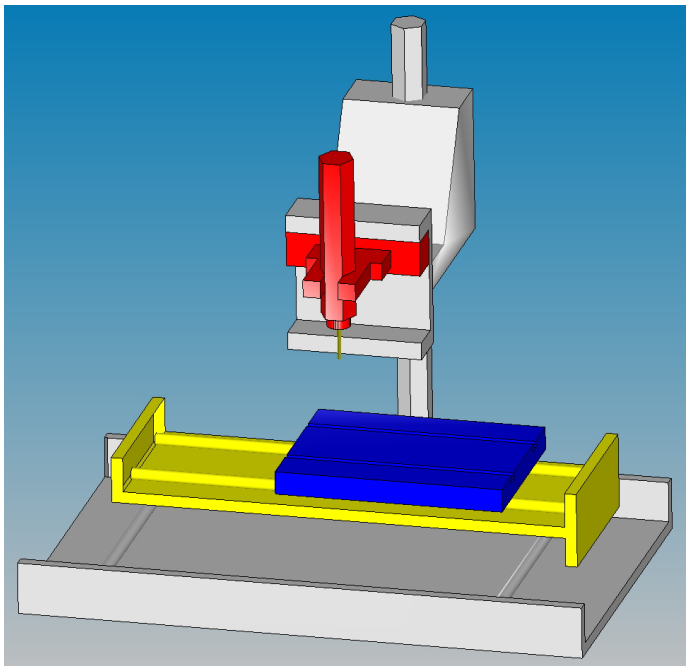


Figure 93: Example of 'Milly' type

I recommend looking at some of the existing models to make the differences clear. You can find a list of all machines under the menu item Extras / Machine Manager and can see the corresponding type under the entry Machine type.

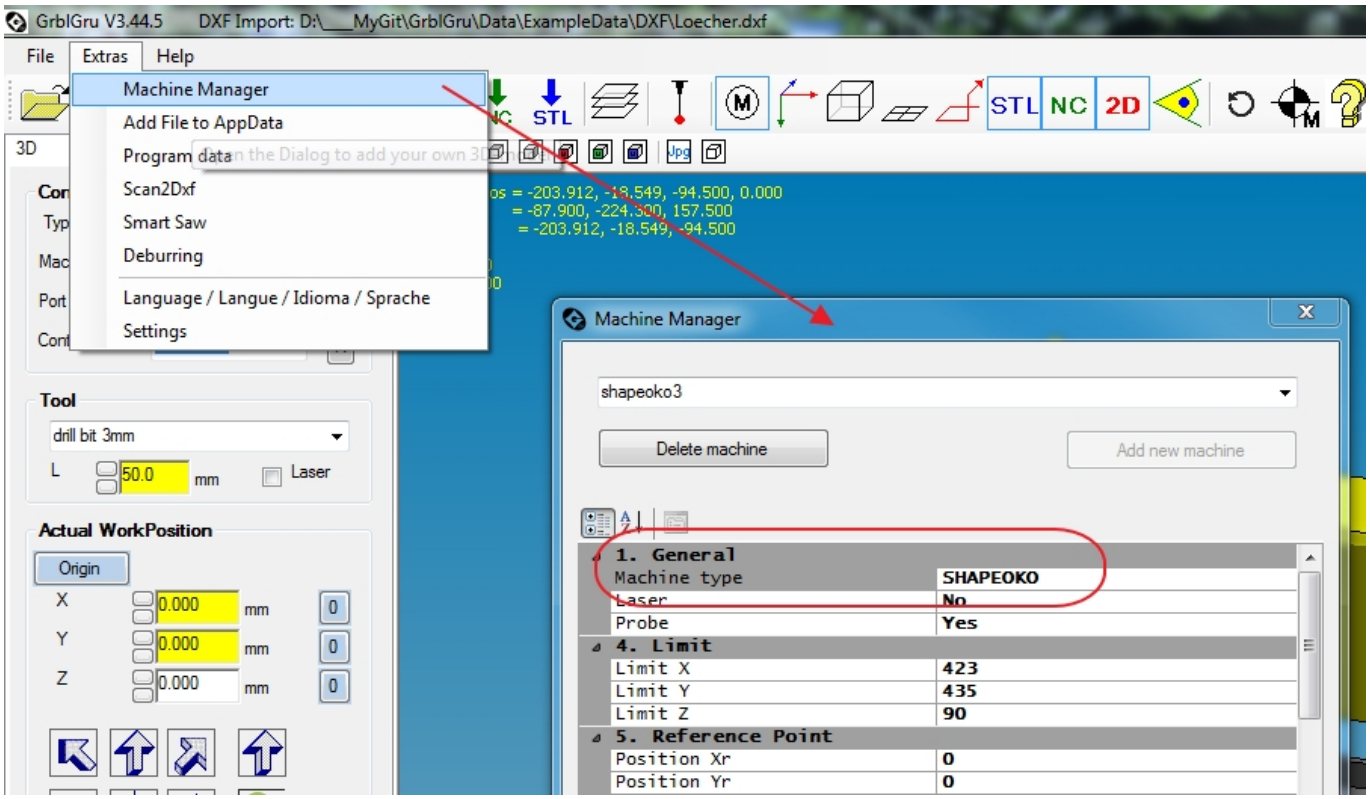


Figure 94: Enter machine type in the dialog of the machine administration

If you need a model whose type is not yet available, please let me know. I will then try to supplement the corresponding type.

The following types of machines have been realized so far:

- SHAPEOKO
- MILLY
- SPRITE
- NONY
- EGGBOT
- PRINTER3D
- SHAPEOKO-AX
- LASER
- FIVE-AXES-1
- FIVE-AXES-2
- FIVE-AXES-3
- DETTORRE
- LowRider
- MpCNC
- BUX

8.2 Construct the machine in CAD

First, of course, you have to construct the machine in CAD. I do not want to go into that on this point, because it essentially depends on the CAD system used. To create my previous models, I have always used the program CREO.
However for construction with CREO there is a little video below:
https://youtu.be/v5m8HKOdHG0

The only important thing in the design is that you align the machine coordinate system as follows.

> The X-axis points to the right, Y to the rear, Z to the top. The preferred Origin of all three axes lies at the Bottom Right Rear.

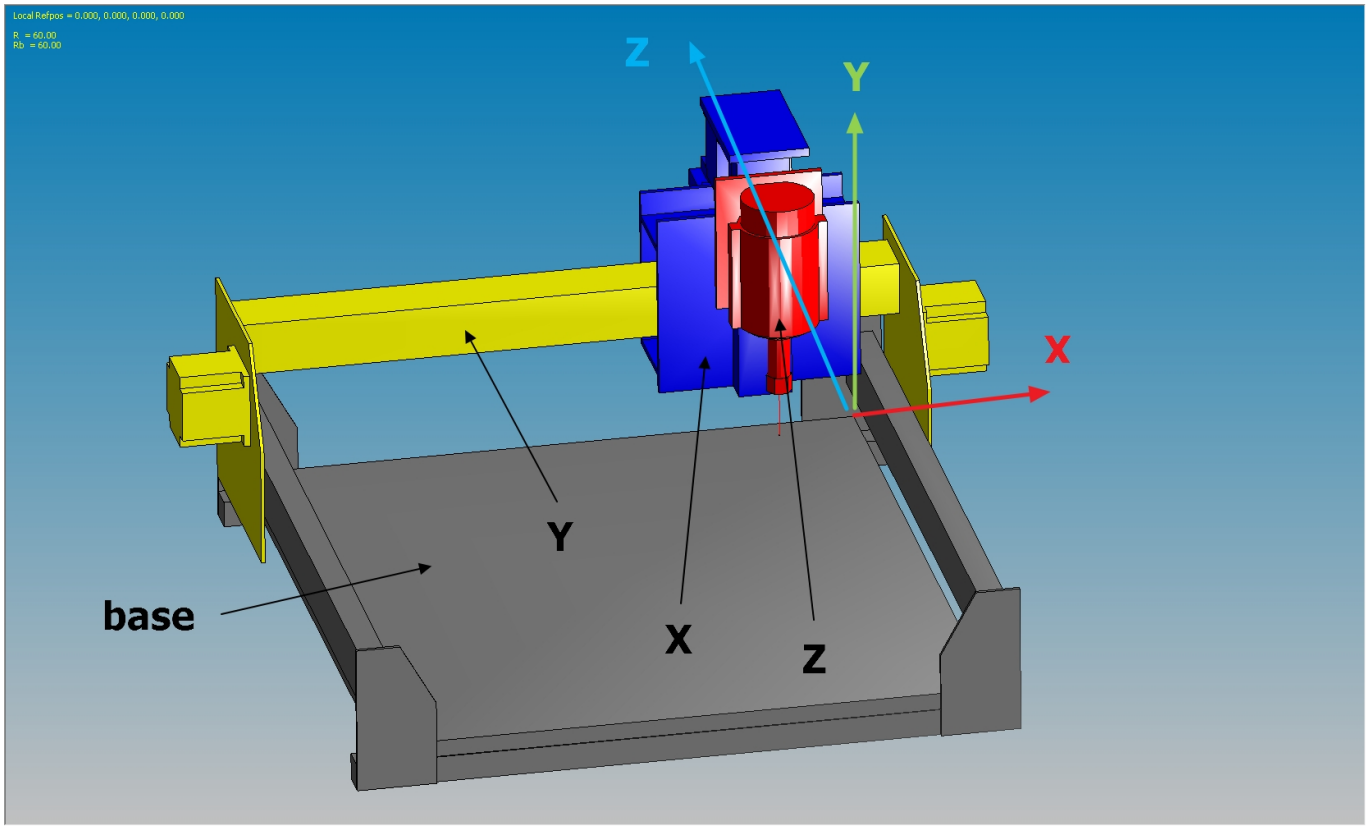


Figure 95: The machine coordinate system

8.3 Save Modules as STL

At the moment ***GrblGru*** can only read 3D data in STL format. For this purpose, the axis modules must be available in this format for the actual import.

Once the complete model has been constructed, the assemblies are assigned to the desired axes and stored in the STL files. The following file names are expected:

- **Base+.STL**
  Fixed Basis
- **X+.STL**
  X-axis;; linear axis
- **Y+.STL**
  Y-axis;; linear axis
- **Z+.STL**
  Z-axis;; linear axis
- **A+.STL**
  A-axis;; axis of rotation
- **B+.STL**
  B-axis;; axis of rotation

The '+' in the name causes the edges of the respective module to be traced and can also be omitted. All files are first stored in any directory.

8.4 Create a new machine in the Machine Manager

The next step is to start a new machine first. To do this, you can find the following dialog in the Menu under the item **Extras / Machine Manager**.

The easier entry of the machine data is facilitated by first selecting a machine of the type that you would like to create in the first line. As a result, many data are already preallocated.

Simply enter the name of your model in the first line and press the button **"Add new machine"**.

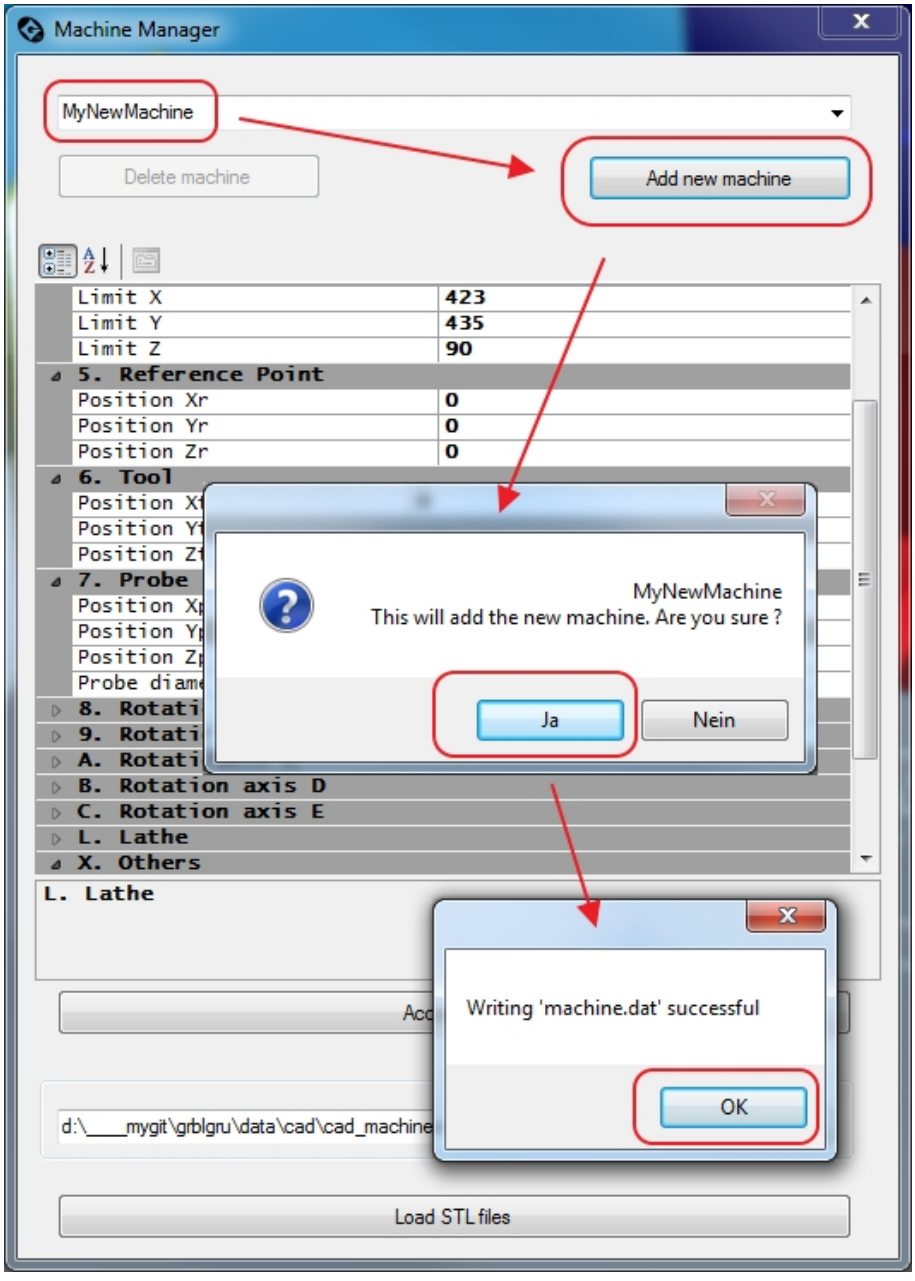


Figure 96: Dialog of the machine administration

8.5 Import STL files

After creating all STL files and creating the new machine, you can now load the files in the Machine Manager. To do this, select the directory with the STLs via the 'Browse' button and confirm the queries.

> Please DO NOT enter the path manually. That does not work !

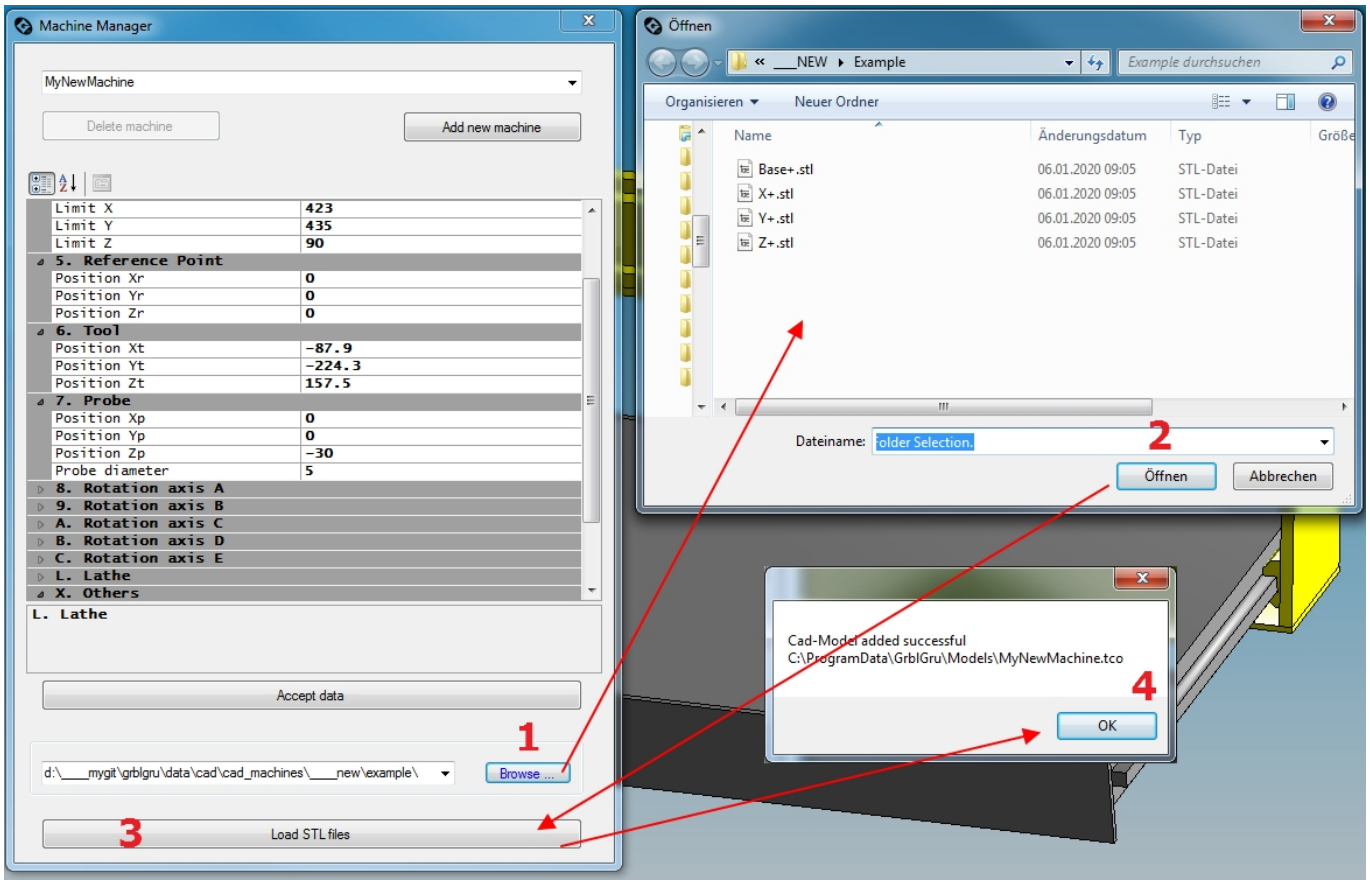


Figure 97: Import of STL files in the machine management

8.6 Reading out control values

The machine can now be selected and displayed in the program.

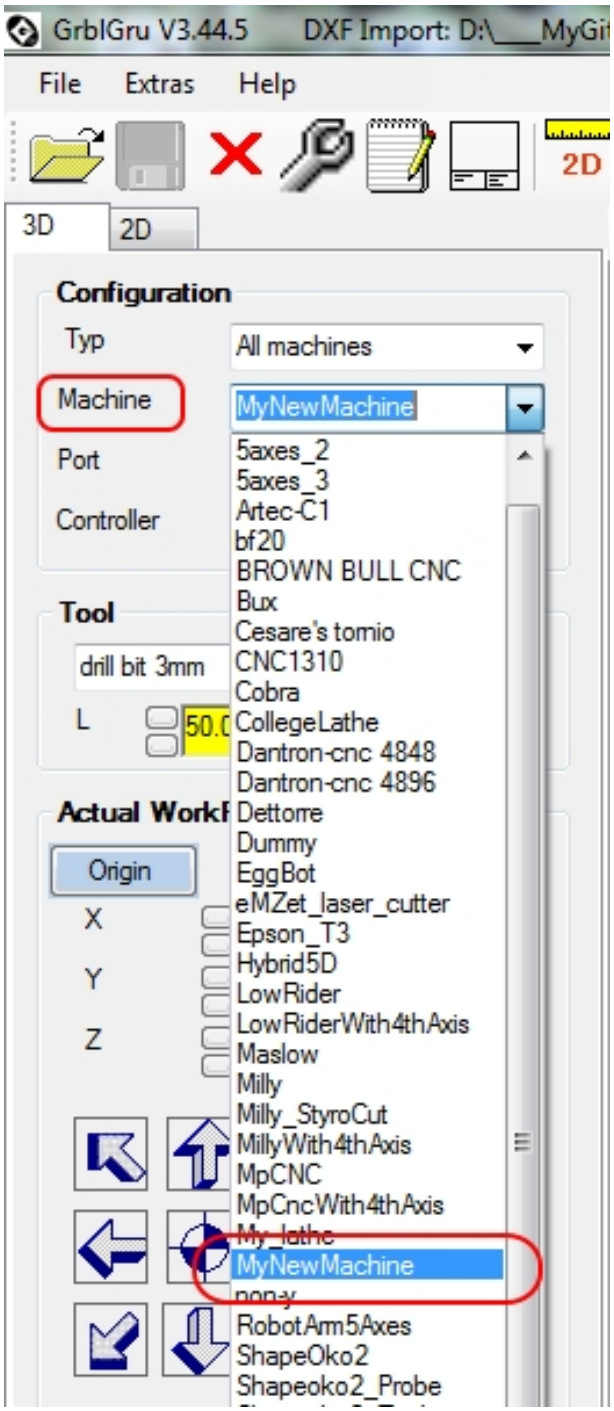


Figure 98: List of available machines

8.7 Tool Position

It is likely that the tool may not yet be displayed in the correct position. To do this, ***GrblGru*** needs the coordinates of the tool origin in the machine coordinate system.

These coordinates can be taken from the CAD directly at the design, or measured retroactively with ***GrblGru*** . To do this, click with the left mouse button and the pressed SHIFT key in succession on 3 points of the circle or polygon. The coordinates of the centre point are then displayed in the status bar and must be entered in the machine administration dialog. The measurement is reset by clicking + SHIFT key off to the side

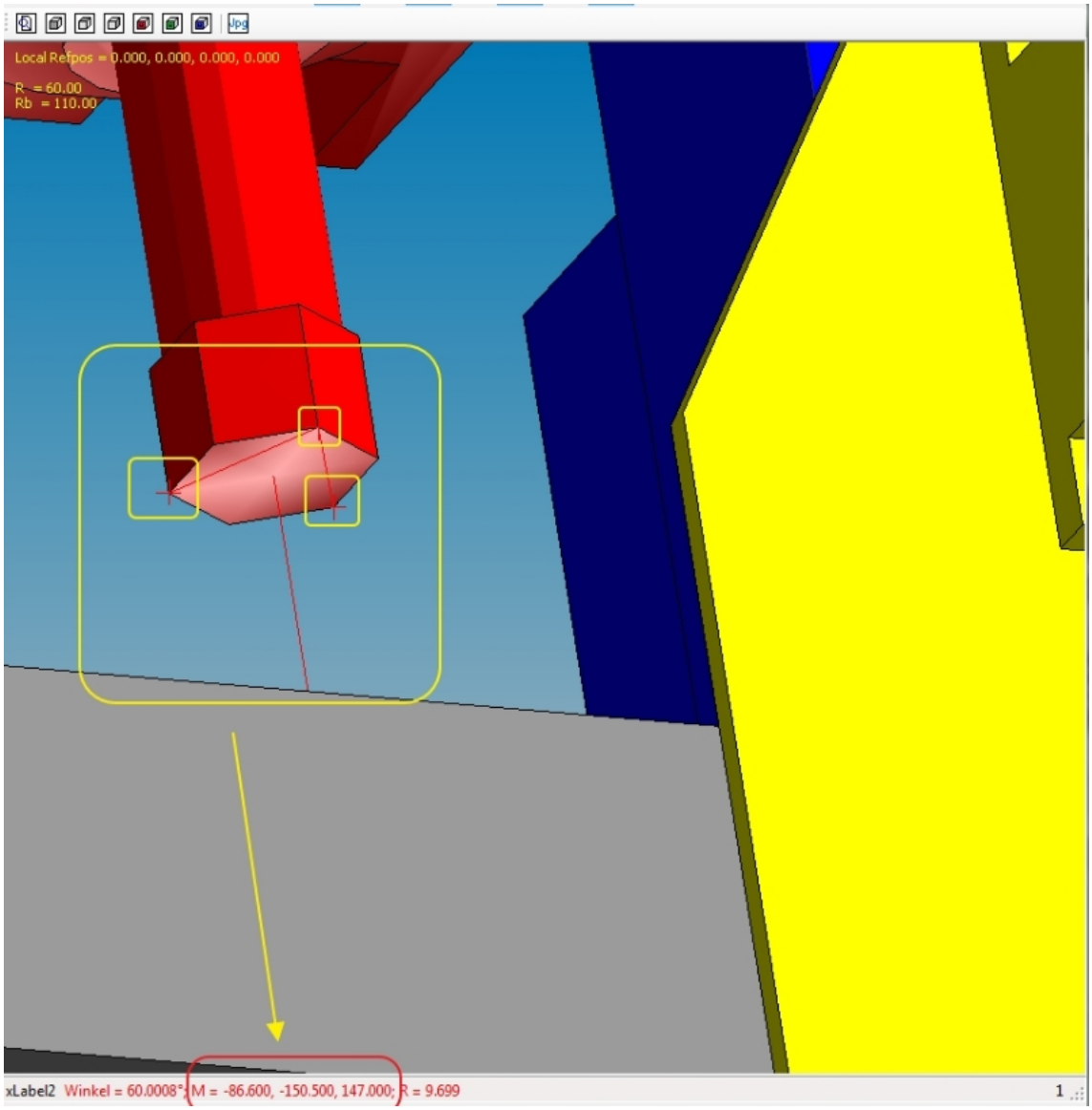


Figure 99: Measurement of tool origins

The X, Y and Z components of the coordinate are then entered in the dialog of the Machine Management under Number 6. Tool.

> Watch out ! Remember to press the button 'Accept data' after the entry or any changes are lost.

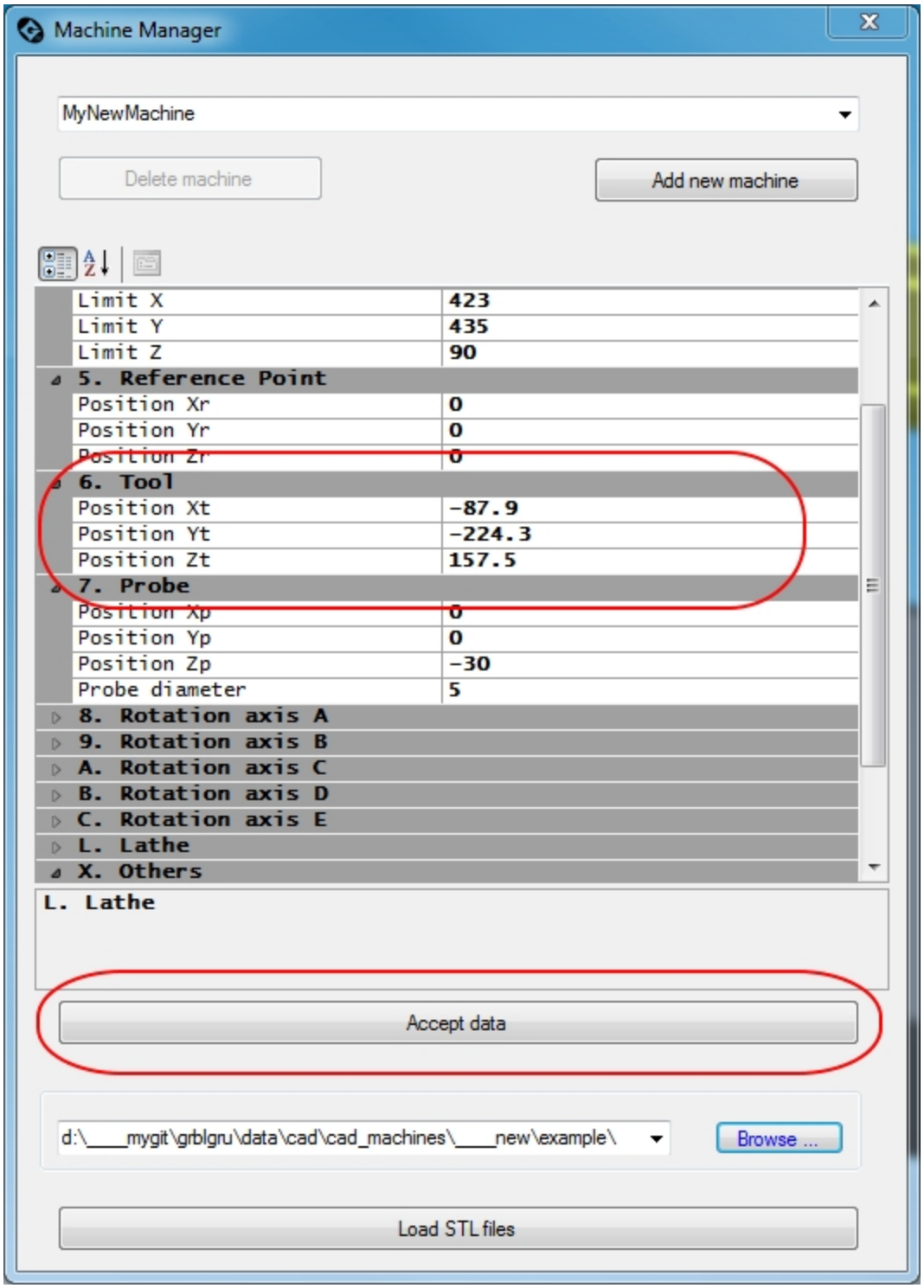


Figure 100: Entering the Origin tools

8.8 Rotary Axes position and direction

> The designations of the two axes of rotation A and B do not correspond to the basic rules for axis designations for purely pragmatic reasons. This is the reason why, in the following example, the rotation axis that rotates around the Z-axis does not bear the name C, but can be called A or B.

In order for ***GrblGru*** to handle the axes of rotation correctly, a point on the axis of rotation and the direction of the axis are required. To measure a point on the axis of rotation, proceed analogously to the tool origin. The direction can usually be determined without measurement, because it usually extends in the X, Y or Z direction. In the example 'lathe chuck' it points upwards, ie in the Z direction. That's why we enter 0, 0, 1 in this case. The input of the direction always refers to the basic position of the model. So as it was constructed and drawn.

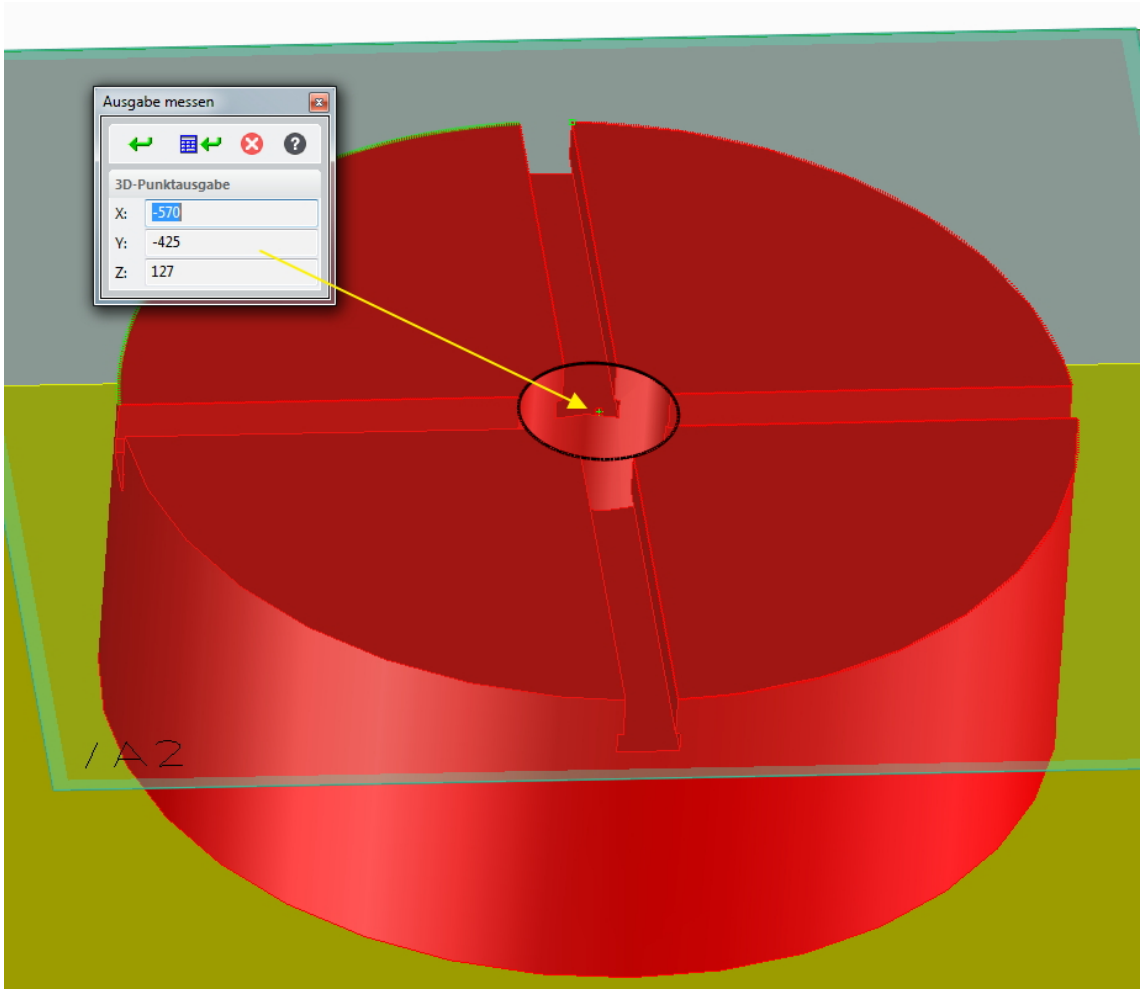


Figure 101: Measurement of a point on the A-axis in CAD (but can also be measured with GrblGru)

Position and direction of the A-axis are entered in the dialog of the Machine Manager in Chapter 8.

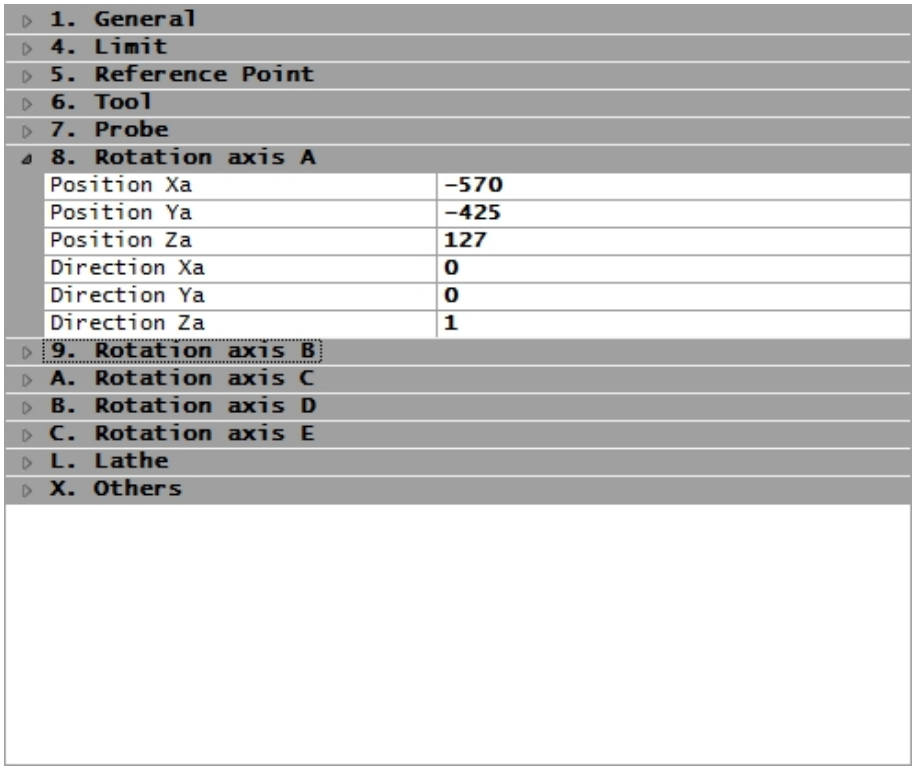


Figure 102: Entering the position and direction of the A-axis

For the sake of completeness, here is the same game with the B axis. The direction here is the X direction, that is 1,0,0

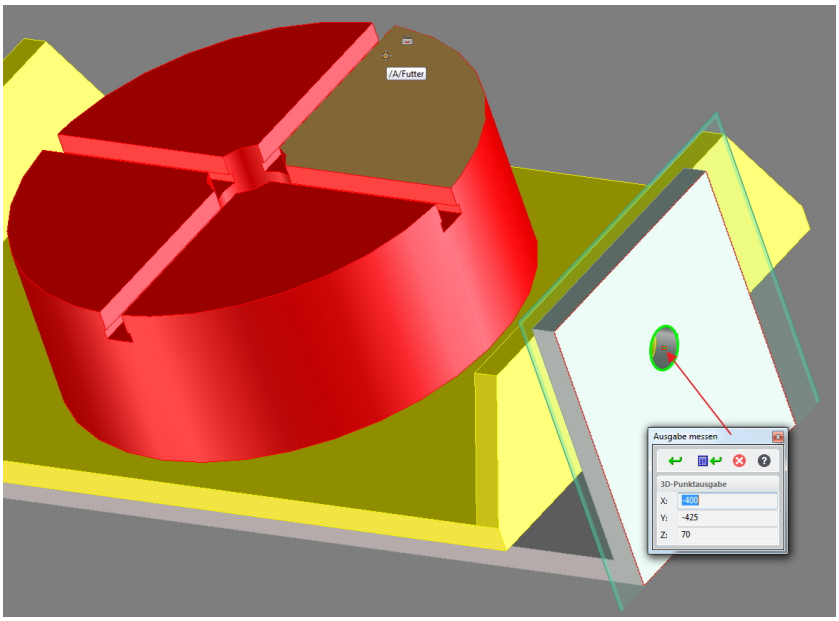


Figure 103: Measurement of a point on the B axis

1. General	
4. Limit	
5. Reference Point	
6. Tool	
7. Probe	
8. Rotation axis A	
Position Xa	**-570**
Position Ya	**-425**
Position Za	**127**
Direction Xa	**0**
Direction Ya	**0**
Direction Za	**1**
9. Rotation axis B	
Position Xb	**-400**
Position Yb	**-425**
Position Zb	**70**
Direction Xb	**1**
Direction Yb	**0**
Direction Zb	**0**
A. Rotation axis C	
B. Rotation axis D	
C. Rotation axis E	
L. Lathe	
X. Others	

Figure 104: Entering the position and direction of the B axis

8.9 Input of the Reference Position

In Chapter 3 you can specify the position of a reference point on which the model is positioned when you press the Toolbar button  . In conjunction with rotary axes, it has proved helpful to place the reference point in the middle of the A-axis.

The position of the X and Y components can be calculated simply by subtracting the A-axis position and the tool position. Z stays at 0.

$$X_r = X_a - X_t \tag{1}$$
$$Y_r = Y_a - Y_t \tag{2}$$
$$Z_r = 0 \tag{3}$$
$$\tag{4}$$
$$X_r = -570 - (-45) = -525 \tag{5}$$
$$Y_r = -425 - (-175) = -250 \tag{6}$$
$$Z_r = 0 \tag{7}$$

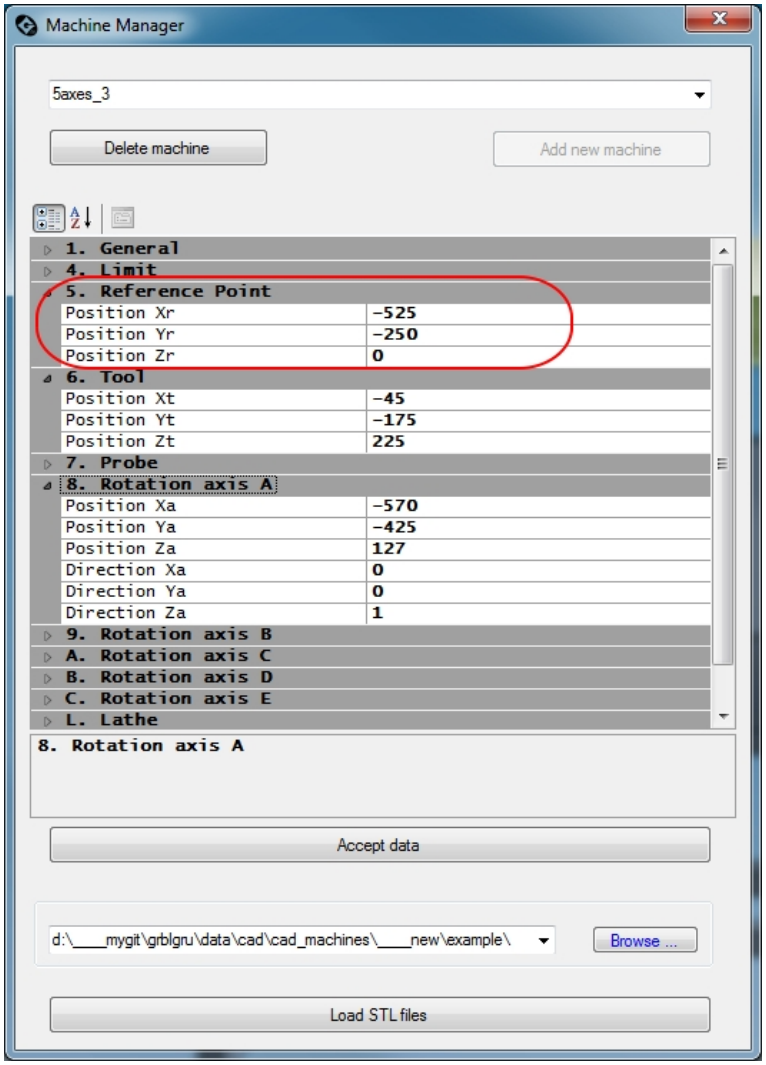


Figure 105: Entering the position of the reference point

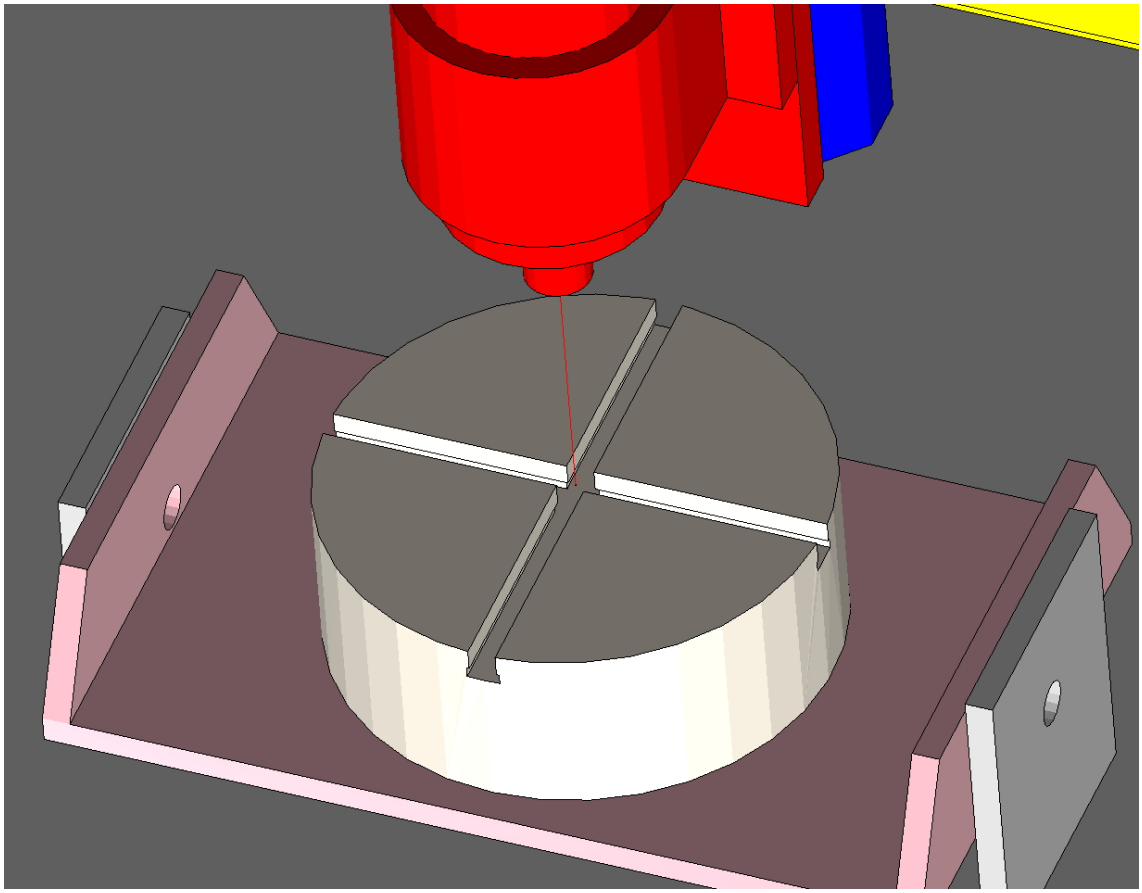


Figure 106: Reference point = center A-axis

8.10 Limits (limit switches)

If you want, you can enter the axis limits in Chapter 4. When the V in the settings is on monitoring, the axes remain in manual mode upon having reached the end position.

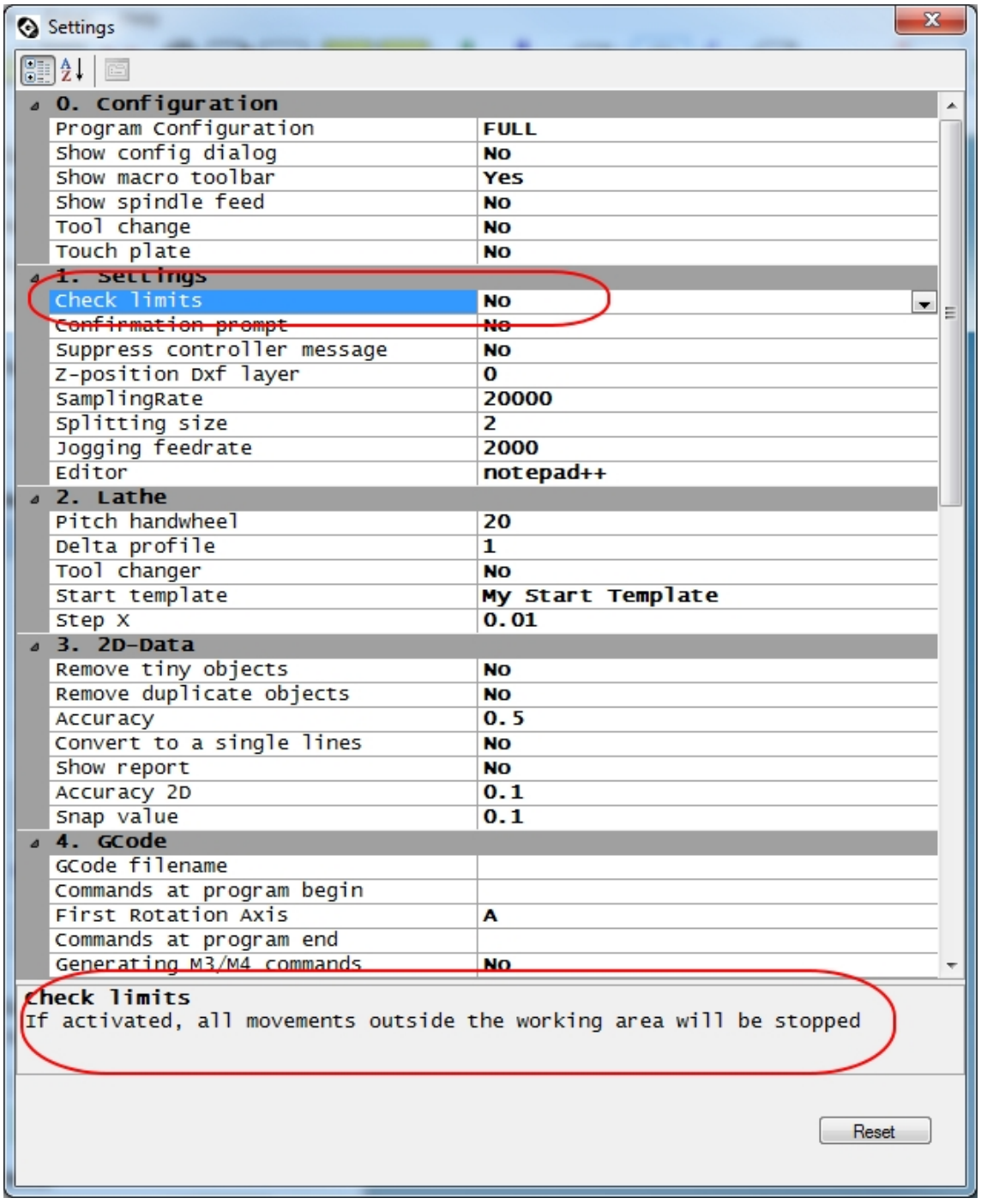


Figure 107: Dialog Settings

8.11 Remaining settings

Please make sure that you have not accidentally set one of the option buttons. These are only intended for special machines and can considerably influence the machine function.

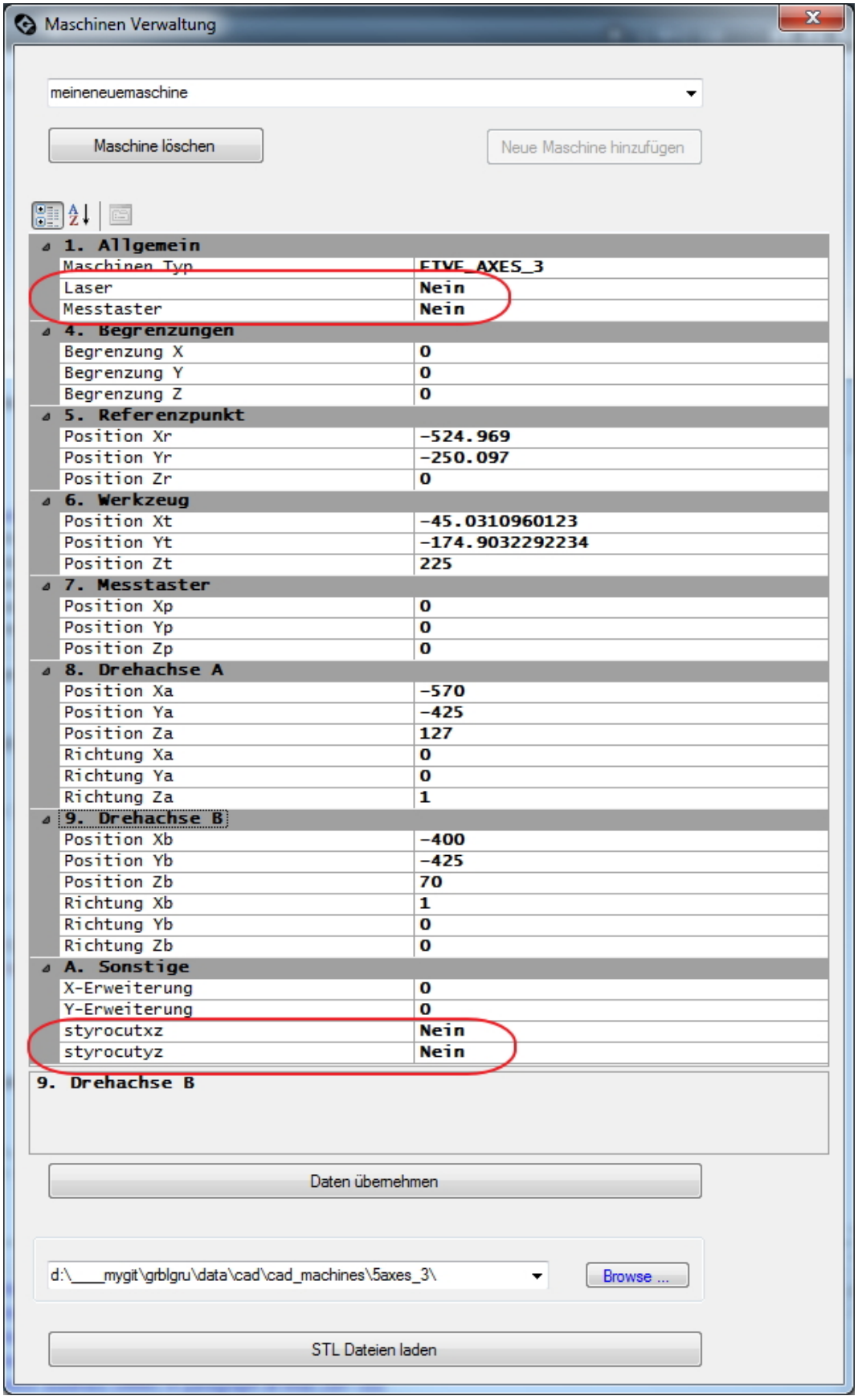


Figure 108: Normal position of option switches = NO

8.12 Special models (lathes)

In principle, the creation of Metal Lathe 3D models works just like the milling. However, there are some supplements that I would like to talk about here.

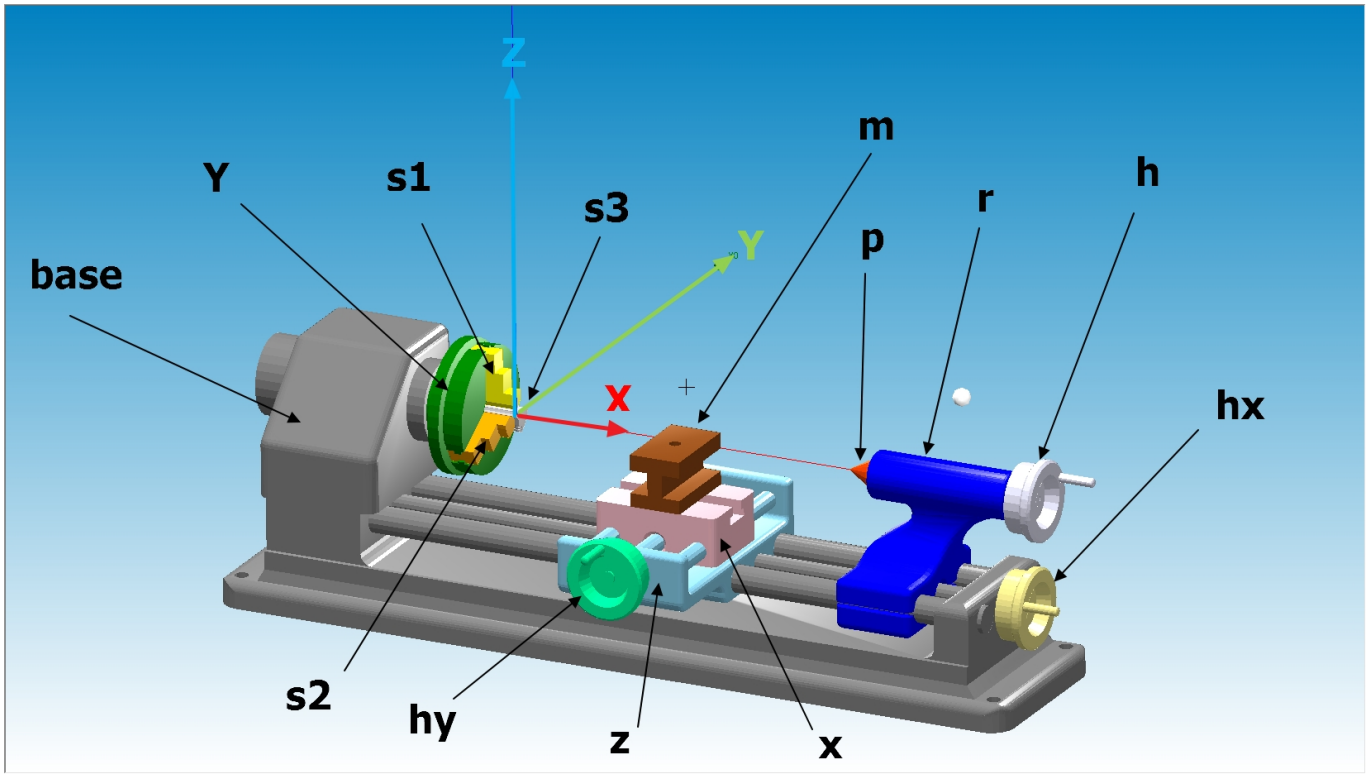


Figure 109: The lathe components

First of all, the 3D model **MUST** be drawn in the following coordinate system:

- The origin lies in the centre of the chuck. If you do not take that into account, the rotation of the food will not work.
- The chuck is saved as **y+.stl**.
- The blade holder m should be positioned so that the leading edge is at Y = 0. It is saved under **m+.stl**.
- The chuck must be drawn so that a clamping claw is perpendicular to the top.
- The vertical jaw must be saved as **s1+.stl**. Then the other two jaws follow with counterclockwise view of the chuck.
- The handrails are saved as**h+.stl**, **hx+.stl** and **hy+.stl**.
- The tailstock is saved as **r+.stl** and the quill as **p+.stl**.
- Tool changers are saved as **m+.stl** like a normal knife holder.