INTM 497 - Special Project

Construction of a bottle engraver

Author:
Mats Jonsson

Supervisor:
Prof. Blake Davis

May 2015

ILLINOIS INSTITUTE OF TECHNOLOGY

ILLINOIS INSTITUTE OF TECHNOLOGY

Abstract

Industrial Technology and Management
Illinois Institute of Technology

Bachelor of Mechatronics

Construction of a bottle engraver

by Mats JONSSON

There was no cheap solution available for automated engraving onto wine and beer bottles. The options were either expensive multi-purpose commercial engravers or CNC mills with special equipment. This project managed to solve this problem by inventing a small low-cost bottle engraver, that is now made Open Source over the Internet. It engraves onto the glass using a diamond engraving tool, that is being dragged across the surface using three computer contorlled stepper motors. It is able to engrave almost any shape onto both straight and non-uniform bottles, using a custom made application. This project is also a demonstration of the possibilities with Open Source software and hardware, since the all the results are produced strictly using only free tools.

Acknowledgements

I spent the last year of my Bachelor of Mechatronics from Chalmers University of Technology, Sweden, abroad as an exchange student at Illinois Institute of Technology, Chicago, USA. In order to pursue my degree I needed to fulfill a 6 credit (Swedish: 15 hp) degree project, which was made possible by registering for an INTM 497 Special Project course that could be granted as a replacement course. This project is examined as INTM 497 at IIT, but carried out with the degree project at Chalmers in consideration.

I want to thank the INTM department at IIT, in particular Prof. Blake Davis who made helped me making this possible and supported me thoughout the project as my advisor. Special gratitude goes to John Welin, manager at the Idea Shop in IIT, for providing me with great advice, ideas and material.

Contents

List of Figures

List of Tables

Abbreviations

CAD	**C**omputer **A**ided **D**esign
CAM	**C**omputer **A**ided **M**anufacturing
CLI	**C**ommand **L**ine **I**nterface
CNC	**C**omputer **N**umerical **C**ontrol
FOSS	**F**ree **O**pen **S**ource **S**oftware
OpenSCAD	**O**pen **S**ource **CAD**
OpenSCAM	**O**pen **S**ource **CAM**

Chapter 1

Introduction

1.1 Background

Automated machining with CNC machines is not a new invention, but last decade's progress in technogy, dropping prices of materials and especially the Internet has made CNC systems accessible to the public. With plans and instructions now available for free, you can make your own CNC router for home use. These can perform milling, routing, drilling, cutting and engaving, but most of them seem to work only on planar surfaces. For machining on cylindrical objects, e.g. wine bottles, a new solution is needed.

Custom engraving of bottles is achieved in industry using CNC machines with rotary attachments. Most of them engrave with lasers, but some use rotary grinding tools. These systems are very expensive, and not an option for the happy hobbyists who just want to engrave their home-brewn beer- or mulled wine bottles for small series production. For home use, the solution is instead a low-cost desktop-sized CNC bottle engraver, that can engrave any shape or image after being processed in a computer.

1.2 Purpose and goals

This project will present a complete low-cost bottle engraving solution. Everything, inlcuding this report, will be made using open source software and hardware only. All documentation and source files needed for replication of the machine will be published on established websites for open hardware.

1.3 Functional requirements

For this task, the final product should meet the following requirements:

- Low-cost; <$300
- Desktop-size; small enought to fit on a desk
- Can engrave on standard 750 ml (wine) and 350 ml (beer) glass bottles.
- High-quality engraving; 0.2 mm positioning accuracy
- Uses only open source software & hardware

1.4 Delimitations

The machine is built in the Idea shop in IIT, a prototyping lab that provides most tools needed for prototype making. There are machines that can do 3D printing, laser cutting, milling, routing and lathing, but machining with non-metal materials only. The bottle engraver therefore has to be designed taking these manufacturing methods into account. Since the aim of this project is to provide an easy-to-build solution for a bottle engraver, it is also important that it can be manufactured and assembled using common tools and accessible parts. Use of standardized components is preferable, since they are easy to obtain and replace.

Furthermore, this project will finish my last semester of my exchange year in the US. For being able to transport the machine back to Sweden it is an advantage if the design is able to disassemble in order to fit in a travel bag. Permanent assembly methods, eg gluing, should be avoided if it results less portability.

Chapter 2

Preparatory research

2.1 Functional analysis

To identify the actual problem, an analysis of the requested function is required. A great tool for this is a function tree. Each function is divided up into subfunctions, disected piece by piece until the point is reached where a solution is required to continue. This way, all the subfunctions can be identified and further analyzed.

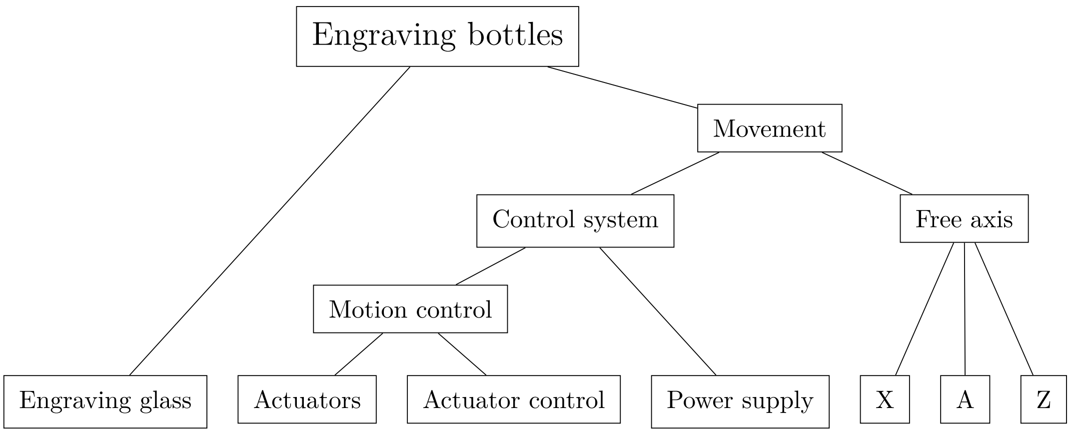


FIGURE 2.1: Function tree for bottle engaving

2.2 Coordinate systems

Most CNC mills use three linear axis: x, y and z. They make up a cartesian coordinate system, enabling motion in 3D space. Sometimes a fifth axis is used for enabling rotation of the object by using a rotary attachment. The linear machine axis can easily be calibrated into distance units (meters or inches), but the rotational device will work with angular units (degrees or radians). Therefore this rotational axis is being labelled a (as in angle).

In the case of engraving bottles, the x axis is not needed and the coordinate system will constitute of a, y and z, introducing a cylindrical coordinate system. Picturing the bottle surface simply as the projection of the XY plane onto a cylinder, the same control software can be used as for linear CNC mills.

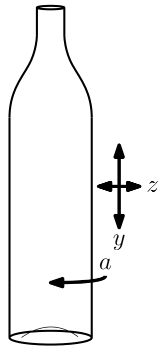


FIGURE 2.2: Coordinate system

2.3 Engraving glass

There are many ways to engrave glass. All forms of engraving aim to change the structure of the surface of the material, usually by damaging it in a controlled procedure. Chemical engraving is a completely different topic, also known as etching, used to roughen the surface of the material to achieve the effect of frosted glass. Sand blasting is another abrasive method for scratching the surface, but these methods are not very clean and therefore suitable for indoor home use. For machine engraving, the most common methods are [1]:

2.3.1 Laser engraving

This method uses a highly concentrated laser beam to heat up the glass. The rapid heatin causes microcracks on the surface of the glass, leaving a visible mark. This method is very accurate and can produce great results, but glass is a hard material which requiers a powerful laser. Unfortunately, they are too expensive for this project's budget.

A first glance at laser cutting/engraving makes one think that lasers have a big advantage – no Z axis is needed since it is just a beam of light. But this is not true – the laser beam has a limited focal range (depth of field) and a Z axis is needed to keep the focus constant to not get different results on different parts of the bottle.

2.3.2 Diamond-drag engraving

This is the easiest of the methods, and is simply just scratching the surface using a tool with a diamond in one end [2]. CNC routers can easily perform engraving with just the use of a diamond-drag toolbit mounted to the spindle. Since it simply scratches the surface, no rotating spindle is needed.

Source: `widgetworksunlimited.com`
FIGURE 2.3: Diamond-drag toolbit

The toolbit needs to hold a constant pressure against the surface in order to achieve good results. When engraving along the surface of the bottle, the Z axis must follow to compensate for the variable diameter. For this reason, diamond-drag toolbits for CNC machines often come with a built-in spring-loaded holder (see figure 2.3) that allows the toolbit to maintain a constant force.

2.3.3 Spindle cutter

By attaching a toolbit similar to the one used for diamond-drag to a motorized rotating spindle, cutting can be performed. The toolbit will, just like a mill or router, cut into the surface of the material and leave a path of the same shape as the tooltip. The path can be deeper than of diamond-drag and usually results in smoother edges. The extra trouble of attaching a rotary spindle to the device will pay off in quality, and it does not require as high lateral and normal forces to be able to engrave.

2.4 Bottle dimensions

It is essential to gather data about how bottle dimensions differ in order to determine the dimensions of the machine. Not only diameter and length is important, but even the variations in diameter along the bottle. Bottles are not perfect cylinders, and many times they come in many different shapes. Most of them are just a straight cylinder with a bottle neck, but some might have convex shapes with different diameters throughout the top and bottom of the bottle. It is necessary to know how much these dimensions differ in order to design the mechanisms, especially for the z axis.

2.4.1 Diameter

Trader Joe's grocery store at Roosevelt in Chicago has a decent supply of wine bottles which they let me measure in-store. A total of 14 bottles were measured with 4 dimensions per bottle: the diameter at the bottom and at the top just below the neck begins, then turned 90 ° and measured again. Different-shaped bottles were picked from various categories, and were measured using a digital caliper with a resolution of 0.01 mm. The data is presented in table 2.1.

#	Btm Ø1	Btm Ø2	Top Ø1	Top Ø2	Btm avg	Top avg	Btm diff	Top diff	Btm-top diff
1	81.21	81.16	80.60	81.36	81.19	80.98	0.05	-0.76	0.21
2	77.16	77.05	76.95	77.27	77.11	77.11	0.11	-0.32	-0.01
3	75.68	76.15	82.60	83.55	75.92	83.08	-0.47	-0.95	**-7.16**
4	72.10	73.12	76.91	77.52	72.61	77.22	**-1.02**	-0.61	-4.61
5	74.79	74.99	82.11	81.78	74.89	81.95	-0.20	0.33	-7.06
6	81.60	81.46	81.70	81.28	81.53	81.49	0.14	0.42	0.04
7	73.63	74.06	79.84	79.77	73.85	79.81	-0.43	0.07	-5.96
8	72.73	73.21	79.32	79.46	72.97	79.39	-0.48	-0.14	-6.42
9	104.30	104.33	102.82	103.90	**104.32**	**103.36**	-0.03	**-1.08**	0.95
10	84.66	84.07	88.71	88.84	84.37	88.78	0.59	-0.13	-4.41
11	81.99	81.76	81.76	81.03	81.88	81.40	0.23	0.73	0.48
12	76.10	76.78	76.20	76.73	76.44	76.47	-0.68	-0.53	-0.03
13	79.92	79.72	79.94	79.64	79.82	79.79	0.20	0.30	0.03
						St.dev	0.44	0.56	3.30

Table 2.1: Dimensions of measured bottles.

The highlighted entries are the maximum values in each column. These will be taken into account when deciding the maximum bottle size that the machine will be able to engrave. However, #9 has the greatest diameter because its volume is greater than 750 ml, therefore not covered by the functional requirements. For this sample, the greatest diameter of a 750 ml bottle is instead #10, with a diameter of 84.37 mm at the top. Adding some extra margin, a maximum diameter of 100 mm should be sufficient for the machine.

Excluding #9, the diameter difference is stretching from 0.05 mm (#1) and 1,02 mm (#4). Adding some extra margin, the machine should be able to compensate for change in diameter of 1.5 mm. #3 has the maximum difference of 7.16 mm between the top and bottom diameters because of its convex shape, which concludes that the machine must be able to support (with extra margin) a diameter difference of 10 mm throughout the axial direction (y axis).

The difference in diameter that is not related to the design is caused in the manufacturing process. All the bottles encountered had the same flaw as seen in figure 2.4: two seams on each side, which emerges from the blow mould where the red-hot glass is blown to the right dimension. [3]

Figure 2.4: Bottle seam.

Measurements on the single bottle above shown that this little ”bump” in the surface can be as high as 0.2 mm. The engraver must be able to handle this in order to engrave a full revolution.

2.4.2 Height

The heights bottles determine the maximum span that the y axis should be able to operate on, and the design of the bottle attachment. This data is not as important, and the method was fairly straight-forward: just visually scan the bottle heights on the shelves as in figure 2.5, and pick the tallest one.

The tallest bottle found was 325 mm high. Using this as a guidline, a maximum bottle height of 350 mm is an appropriate functional requirement.

2.4.3 Beer bottles

Assuming that beer bottles are manufactured the same way as wine bottles, no detailed measurements has been made. By designing for wine bottles, it should support smaller

FIGURE 2.5: Average bottle heights. Taller bottles can be seen in the back and to the right.

bottles too if a minimum diameter and height is added to the functional requirements. Most beer bottles in the store had a diameter of 50 mm, and the shortest found was 150 mm tall. Designing for a minimum diameter of 40 mm and heght of 130 mm should therefore include most bottles on the market.

2.4.4 Functional requirement summary

To summarize the results, the bottle engraver should be designed for bottles with:

- Heights: 130 - 325 mm
- Diameters: 40 - 100 mm
- Diameter difference along y axis: 10 mm
- Diameter difference along a axis: 1.5 mm

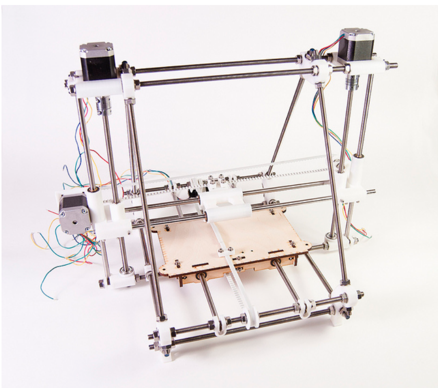
2.5 Existing solutions

Machine engraving on bottles has been done many times before, but most existing solutions seem to be multi-purpose CNC engraving machines capable of engraving on many different kinds of object and on many materials. Because of this, their price is very high – the cheapest found was $2,000 and the most expensive $20,000. The cheapest ones are laser engravers from China, while the more expensive use rotary cutters and are made in the US.

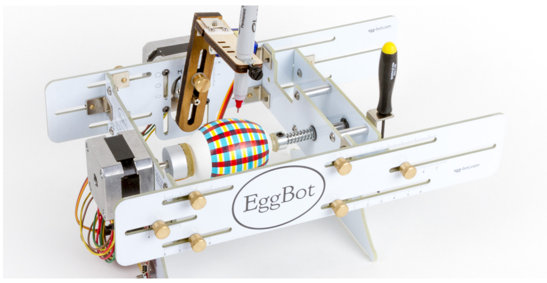
Another option is to use a CNC mill or laser with a rotary attachment. This is probably the most common and cheapest method since a CNC mill also can be used for many other tasks besides engraving.

It is not necessarily difficult to build your own CNC machine, and there are many Open Source CNC projects on the Internet. A very common machine is the RepRap (Figure 2.6) – a free desktop 3D printer whith the aim of self-replication [4]. It has a wide user base and is well documented – with everything being Open Source this is a very good source of inspiration.

Another project that is more similar in to a bottle engraver is the Eggbot (Figure 2.7). It is an Open Source egg painter, that is able to print art on eggs with a pen. [5]. It is very similar in function, but at the same time widely different in design – eggs should be handled with care, while bottle engraving should be more violent. The inspiration here is instead in the software bit, on how to convert art to machine movements.



Source: `reprap.org`
FIGURE 2.6: RepRap



Source: `egg-bot.com`
FIGURE 2.7: Eggbot

Chapter 3

Methodology

3.1 Brainstorming

Figure 3.1 shows sketches from the very early stage of the project. They were designed with mainly two different approaches, and combinations of them.

3.1.1 Laser-cut walls – Sketch (A), (J), (K) and (L)

This would be the cheapest method. Laser-cut walls made out of plywood, particle board or plastic build up the main frame. This brings rigidity to the design, but comes with the downside that it is not very flexible when it comes to adjusting it for different purposes or bottle dimensions.

3.1.2 Threaded rod frame – Sketch (B), (C) and (D)

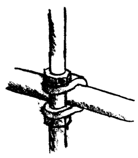


Figure 3.2: Rod clamp

Inspired by the RepRap, this frame uses threaded rods as base and a 3D printed rod clamp together with nuts for attaching them together. The rods are easy adjustable just using a wrench, which simplifies the adjusting/testing procedure. Threaded rods are cheap and easy accessible, and with the right design not many other custom made parts should be needed. But joints like these are not very rigid, and the frame has to compensate for this weakness, perhaps by adding crossing rods. This adds more complexity to the design.

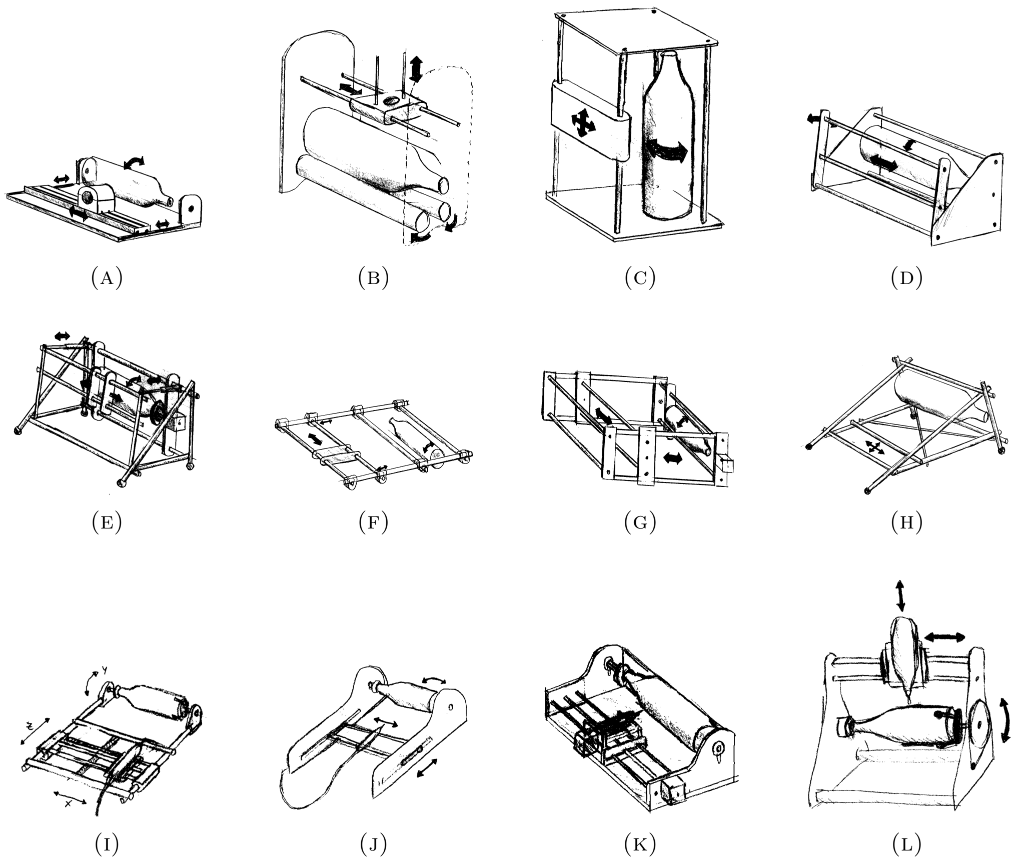


FIGURE 3.1: Different frame designs

3.1.3 Hybrid solutions – Sketch (J through (L)

By combining rods and laser-cut boards, the frame can be rigid but still adjustable to some extent. Bottles differ most in height, and so these designs take that in account by using laser-cut walls hold together with threaded rods.

3.2 Concept evaluation

Without detailed analysis, some of these 12 designs are personally more favorable than others. (A), (D), (E), (H) and (K) are all designs I value more, and good candidates for the final build. In order to get an objective revision of the possible solutions, a decision-matrix is a useful tool. Also called Pugh matrix [6], this is a method to evaluate a large set of options in order to point out the stronger and weaker candidates. By applying this method to the different frame designs, the best designs can be sorted out for further brainstorming.

Design (A) has been chosen as a reference. The other designs are then evaluated as better (+), worse (-) or equal to this design on a few criterion:

- Few parts – How many parts does it have compared to the reference design?
- Easy manufacturing – Are these parts easy to manufacture?
- Easy to build – Is the design easy to assemble?
- Cheap – Are the needed parts cheaper or more expensive?
- Accurate – Will it be able to produce accurate results?
- Flexible – Is it easy to adjust in order to fit different bottle sizes and tweak dimensions during the build?
- Robust – Is the design robust and not wobbly?

Many of these criterion depend on each other, and it is sometimes hard to to a fair comparison of the designs. A brief estimation will still give a hint on which designs are bad when the total score is calculated. For each design, all pluses are counted and subtracted by the number of minuses. The sum will then tell if the design is better (¿0), worse (¡0) or about the same as the reference design (A).

	Frame design										
Criteria	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)
Few parts	-			-	-	-	-				
Easy manufacturing	-		+	+		-	+		-		-
Easy to build	-		+	+	+	-			-		
Cheap	-	+	+	+	+						
Accurate			-	-							
Flexible			+	+	+	+	+	+			
Robust	-	-	-	-	-	-	-	-		+	-
Total	-5	0	**2**	**1**	**1**	-3	0	0	-2	**1**	-2

Table 3.1: Pugh matrix

(B) was never a favorite in the first place and got a very bad score, worse than (A) in every aspect. (G), (J), and (L) also have low rank, exkluding them from further evaluating. (D) is the best candidate, but with only one plus ahead of (E), (F) and (K) it is too early to choose it as the winning design. They need to be evaluated against each other.

Chapter 4

Evaluation

4.1 Overview

Ideally the evaluation process begins with experiments and follows by designing, modelling, building and testing. This project has been far from that simple – it has been rebuilt several times with numerous modifications. In a project like this, problems are expected, but more time should have been spent on designing the details before the building started. Due to a limited timeframe, the building started before all the details were completely thought through. This section is attempted to give the process some structure, when it actually was an intense alternation between designing, building and testing.

Since there was no time for further analysis of the sketches, (D) in Figure 3.1 was chosen for continued development. It is easy to manufacture with a laser cutter, rigid, and angular motion in the Z axis is very easy to achieve. It has been one of the favourable designs since the sketching started.

4.2 Engraving experiments

To motivate choice of motors and actuators, it is essential to know the force required for engraving. The normal force N is perpendicular to the surface, hence operating in the z direction to push down the engraving bit. F is the force required for moving the bit along the surface in the y and a directions.

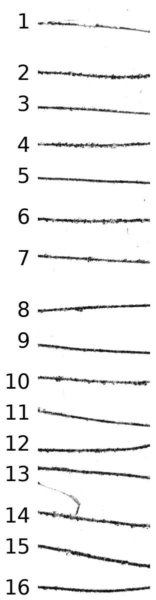


FIGURE 4.1: Result of engraving while measuring normal force

#	Normal force (N)
1	7.36
2	7.85
3	9.17
4	9.71
5	10.50
6	11.67
7	12.21
8	13.44
9	13.73
10	14.72
11	15.70
12	16.68
13	17.66
14	18.64
15	19.62
16	20.60

TABLE 4.1: Normal force data

Two different sets of experiments were made using a CNC mill in order to be able to estimate these forces. A 3 mm thick glass pane was used for diamond-drag engraving, first to estimate N and then F. With this CNC mill, the z axis can be manually adjusted with an accuracy of 1 µm.

4.2.1 Normal force

The glass was first placed on a digital scale with a resolution of 5 g. The scale was reset to zero and the toolbit was lowered until dragging of the glass would result in a scratch of the surface. The "weight" was recorded and the glass was dragged by hand to engrave a line into the glass. The data for 16 lines are presented below, together with a scanned and enhanced image of the glass.

Since the scale measures weight in grams, the normal force N is obtained using $N = mg$ where m is the measured weight and $g = 9.81\,\mathrm{m/s^2}$. A tiny increase of line thickness is seen in the image, but after #1 the quality of the lines are almost identical and varies most likely due to the quality of the glass. For achieving an engraving result of line #2 and following, a minimum normal force of 7.85 N is required.

4.2.2 Lateral force

The force F that is required in the tangent of the surface was acquired in a separate experiment by putting the glass on a custom made rig with metal slides that the glass could slide upon. First the frictional force was measured with no engraving taking place. The normal force is just the gravitational force for the glass, $N = mg = 2.84\,\mathrm{N}$, where the glass weight $m = 290\,\mathrm{g}$. The glass was dragged by hand with an analog force gage, mutiple times, and the average force was estimated to $0.25\,\mathrm{N}$. The coefficient μ was then calculated with $\mu = N/F = 0.09$. This value is low enough for the frictional force to be neglected, such that $F_\mu = 0$. The measured force F can then be used as an estimation for the engraving force F_e.

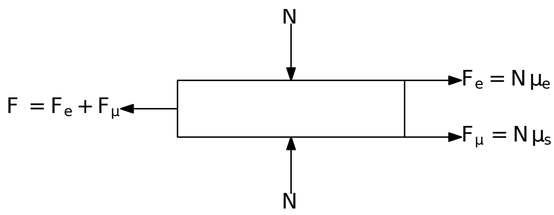


FIGURE 4.2: Forces acting on the glass

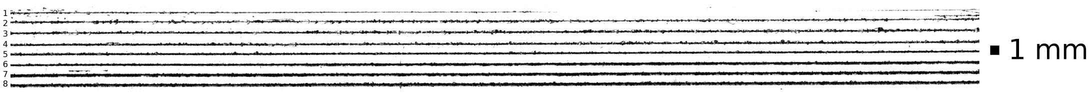


FIGURE 4.3: Result from lateral force measurements

The results above were acquired in a way similar to the previous experiment. The toolbit was lowered until it was in contact with the glass ($z = 0$), which was dragged with a force gage. It was dragged two times, and the average $\bar{F_e}$ is shown in the table above. The spindle was then lowered with 0.02 mm, and the procedure was repeated 8 times until the force became too big in order to achieve good results. The were scanned and are shown above, ordered the same way as the table.

As in the previous experiment, the lines are getting slightly thicker the more the toolbit is lowered. From line #3 and below, the engraving looks good. For achieving #3, a force of 4.6 N was required. By visually comparing these two experiments, this would correspond to line #2 for the normal force experiments. We can conclude that the normal forces are similar, and these two together is a good estimation of what forces should be expected for performing engraving.

#	z	$\bar{F_e}$
1	0	2.00
2	-0.02	3.75
3	-0.04	4.60
4	-0.06	5.50
5	-0.08	8.00
6	-0.1	10.70
7	-0.12	13.70
8	-0.14	15.50

TABLE 4.2: Lateral force measurements

4.2.3 Conclusions

To achieve good engraving results, the z axis must be able to push against the bottle with a minimum force of 8 N, while the y and a axes must manage a tangential force of 5 N.

4.3 Design evolution

The design was modelled in OpenSCAD and went through several iterations. Common for all of the following is that they all have two-layered walls which allows for making extruded holes that bearings will fit in. They all have a bottle that rotates around an axle as the a axis and Ø8 mm smooth rods with bushings for the linear motion on the y axis. The walls are held together by M8 threaded rods. Since the engraver should be cheap, the goal is to have all custom made parts laser cut.

4.3.1 Version 1 and 2

The first CAD model can be seen in Figure 4.4. The tilting yz table is powered by servos mounted in the rectangular holes in the walls. The y axis is controlled by a stepper motor and a leadscrew, and the carriage (not modelled) is held in place by the leadscrew and an additional smooth rod. The a axis is also driven by a stepper motor that is geared through 3D printed pulleys and a belt.

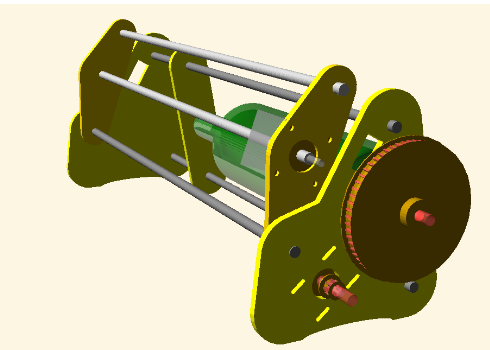


FIGURE 4.4: Version 1

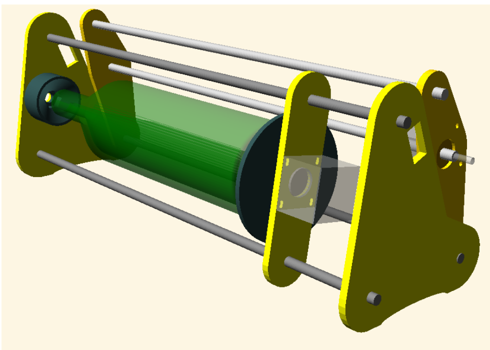


FIGURE 4.5: Version 2, viewed from behind

The second version in Figure 4.5 is not much different from version 1, but uses a stepper motor with a 19:1 reduced planetary gear box (Figure 4.6) that is directly mounted to the bottle grip.

4.3.2 Issues with version 1 and 2

Using a gearbox simplifies the design, but later on a closer look at the data sheet revealed a backlash as high as 1 °. With a maximum bottle diameter of 100 mm, this would lead to an error of

$$\frac{100\,\text{mm} \cdot \pi}{360^\circ} = 0.87\,\text{mm}$$

Source: `amazon.com`
FIGURE 4.6: NEMA 17 with 19:1 planetary gearbox

which is unacceptable. The planetary gearbox idea was therefore discarded.

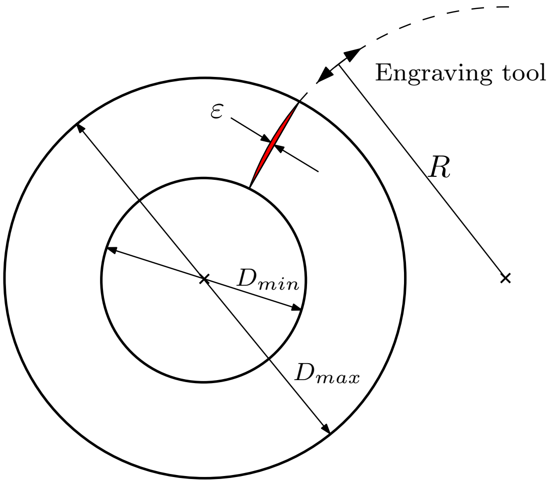


FIGURE 4.7: Flaw with design 1 and 2

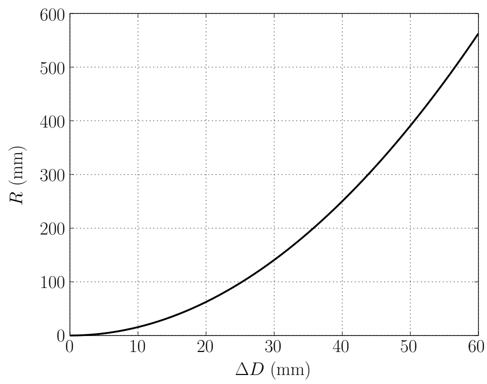


FIGURE 4.8: Required arm length R for desired ΔD

The idea with the tilting yz table is easy and cheap to implement. But it also comes with a drawback since the frame is not adjustable depending on the size of the bottle. The toolbit is lowered and raised in an arc trajecory which will cause a distortion of the engraved figure as the diameter of the bottle varies. Curved bottles will get a position displacement as seen in the red-filled area in Figure 4.7. The maximum error ε that can be tolerated is specified in the functional requirements as the precision accuracy of 0.2 mm. Assuming that the center of the engraving tool trajecory is positioned in such way that the error is minimized, the required arm radius R is a function of the ε

and the difference in diameter ΔD:

$$R = \frac{(\Delta D)^2 + 16\varepsilon^2}{32\varepsilon}$$

Since the machine should be capable of bottles with diameters between $D_{min} = 40$ and $D_{max} = 100\,\text{mm}$, we get $\Delta D = 60\,\text{mm}$. But the machine will probably never operate in this interval, but at most with $\Delta D = 10\,\text{mm}$ for the variations in diameter of one bottle. As seen in Figure 4.8, this requires an arm length of only around 10 mm, but as soon as ΔD increases the arm length grows exponentially. The machine would only be optimized for one preset bottle diameter and would produce different results when operating outside of this range. To have it optimized between the full range, $\Delta D = 60\,\text{mm}$ would require an arm length of 550 mm which is obviously not feasible.

4.3.3 Version 3 and 4

The z axis mechanism was redesigned by having it push the engraving tool in a straight line towards the center of the bottle. Linear motion is a little bit more complex to achieve, but one way is by using a solenoid – an electromagnet that pushes/pulls a metal rod. The construction is very simple and they are very cheap. It is assumed to have a long lifetime since there is no mechanical wear other than in the bushing. Version 3 in Figure 4.9 has the yz table fixed to the frame and the engraving tool attached to another pair of rods that is being actuated by

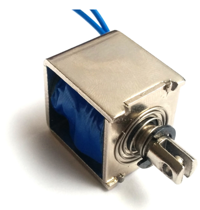


FIGURE 4.11: JF-0826 Solenoid

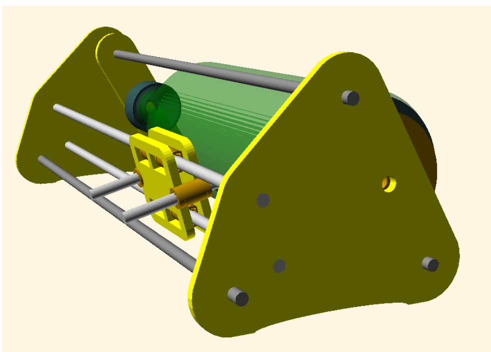


FIGURE 4.9: Version 3

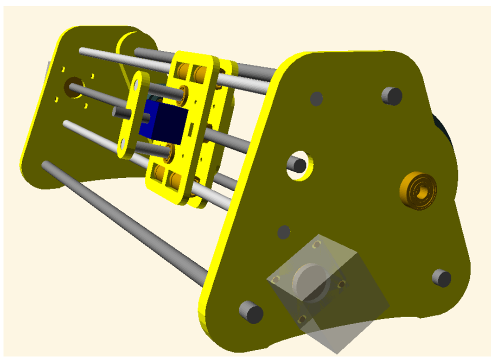


FIGURE 4.10: Version 4

a solenoid. A solenoid that would suit this purpose is the JF-0826 of unknown brand, seen in Figure 4.11, with a pulling force of 20 N and a stroke length of 10 mm. The z rods are horizontally positioned in version 3 and the engraving tool is mounted between them. This might interfere and collide with the bottle if there is a great variation in diameter. Therefore, version 4 (Figure 4.10) has vertical z rods instead. Its dimensions has been adjusted to optimize the volume of the machine.

The pulley has been placed on the inside of the wall. The y axis is still controlled with a stepper motor and a leadscrew.

4.3.4 Issues with version 3 and 4

Version 3 has not yet any solution for controlling the y axis. A leadscrew is preferred to be located on the same plane as the engraving tool, but the z rods are taking up all space. Version 4 is able to fit a leadscrew in between the rods, but the question is instead how to easily mount the bushings for the z rods. There is just a hole for them through the carriage plates in the model, without any attachment features. One idea was that screwing the two plates together will bend theplates so that the deformed plates will grip the bearings, but this is highly uncertain. Another problem is how to mount the solenoid, which only has threaded holes on its side and not on the bottom.

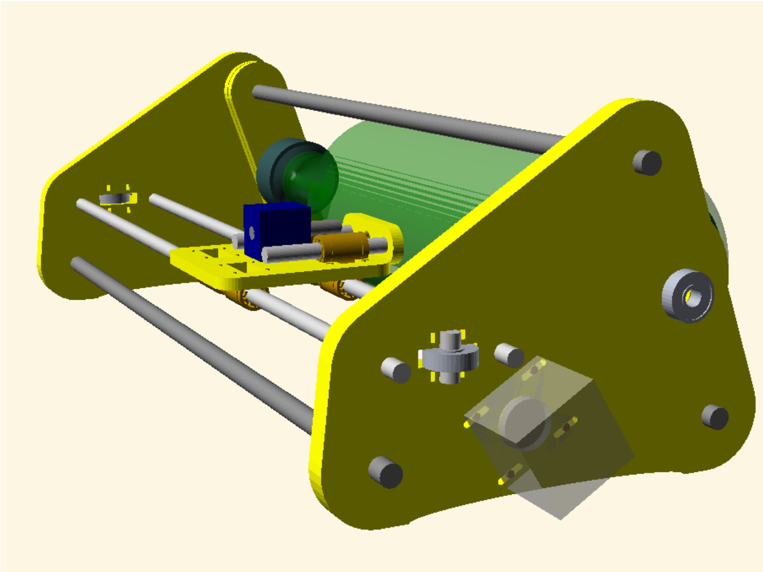


FIGURE 4.12: Version 5

4.3.5 Version 5

The y rods are positioned horizontally this time, which makes mounting of solenoid and bushings easy. The solenoid is screwed onto the plate, while the bushings are strapped in place with cable ties. The y axis is controlled with a stepper and a belt (though not shown in model), which is wrapped around two bearing guides that is integrated into the walls. The stepper motor for this belt is not yet designed, but should be mounted on the rods.

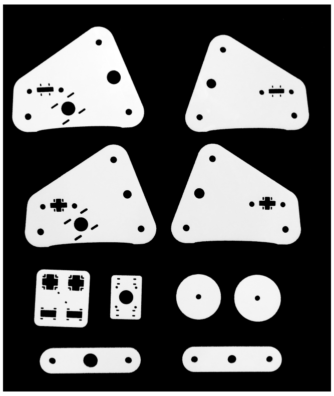


FIGURE 4.13: Laser cut pieces

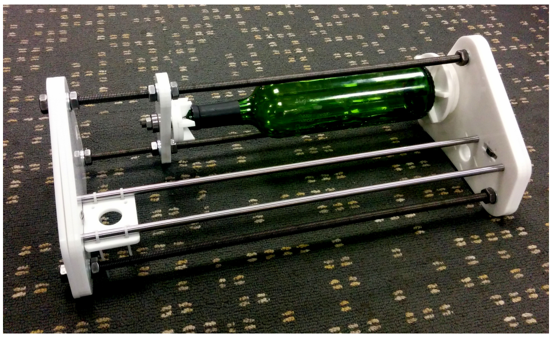


FIGURE 4.14: Version 5 built, with modifications

4.4 Building process

Even though version 5 was not completely finished down to every detail, the frame base could be built. All pieces was laser cut out of 5.5 mm thick white Acrylic and can be seen in Figure 4.13. The first modification was to add a third wall layer that could act as an end stop for the smooth rods and increase stability. After a mount for the y stepper motor and the bottle grips was designed, the frame looked as in Figure 4.14. This prototype immediately revealed a big problem with this choice of design.

4.4.1 Deflection in rods

The y rods seemed to be too flexible – bending them by hand was too easy. This design would probably not meet the requirement of 0.2 mm positioning accuracy since the engraving tool would displace during engraving. This can be reduced either by using thicker rods, or by choosing another slide system such as MakerSlide in Figure 4.17

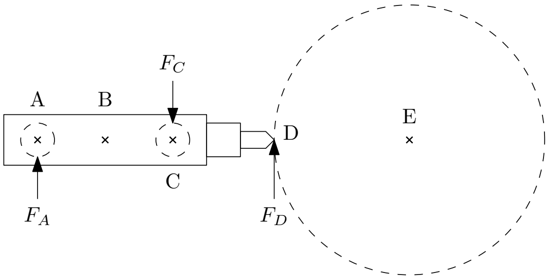


FIGURE 4.15: Forces acting on the rods

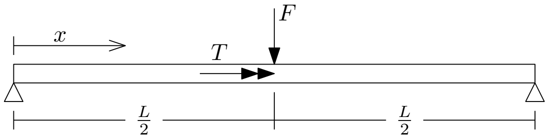
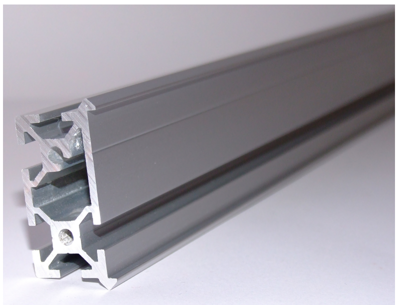


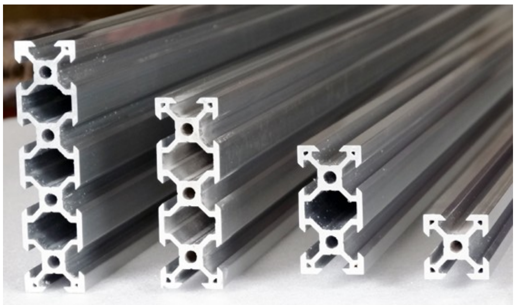
FIGURE 4.16: Beam with simple supports

or V-Slot in Figure 4.18. The theoretical deflections were calculated in order to pick the best option, by simplifying the design as in Figure 4.15. All calculation steps are shown in Appendix A.1, leading to Equation A.5:

$$\delta_D = -\frac{L_{BD} L^3 F_D}{48 L_{AC} E I}$$



Source: `inventables.com`
FIGURE 4.17: MakerSlide



Source: `openbuilds.com`
FIGURE 4.18: V-Slot

The formula for the deflection in an aluminium extrusion is different since it is only one single beam. Following the calculations in Appendix A.2 gives us Equation A.12:

$$\delta_D = \frac{FL^3}{48EI} + \frac{L_{BD}^2 F L}{GJ}$$

These ecuations was used for comparison between different rod diameters, V-Slot of different dimensions and MakerSlide. The result can be seen in Table 4.3, using the rod length $L = 470\,\text{mm}$ with a lateral force $F = 10\,\text{N}$.

The results show that the rods have to be at least 12 mm thick to deflect less than 0.2 mm. All of the V-Slots and the Makerslide would not have this problem. Since

Type	**Dim.**	$\mathbf{I}_x$ **(mm^4)**	$\mathbf{I}_y$ **(mm^4)**	**E (Mpa)**	**G (Mpa)**	**Width (mm)**	**d (mm)**
Rods	Ø 8	4.02E+02	6.36E+04	2.14E+05	7.93E+04	50.00	0.729
Rods	Ø 10	9.82E+02	1.00E+05	2.14E+05	7.93E+04	50.00	0.299
Rods	Ø 12	2.04E+03	1.45E+05	2.14E+05	7.93E+04	50.00	0.144
Rods	Ø 8	4.02E+02	1.62E+05	2.14E+05	7.93E+04	80.00	0.550
Makerslide	40 × 20	1.61E+04	6.10E+04	7.17E+04	2.69E+04	40.00	0.029
V-slot	20 × 40	1.36E+04	5.13E+04	7.17E+04	2.69E+04	40.00	0.034
V-slot	20 × 60	1.97E+04	1.60E+05	7.17E+04	2.69E+04	60.00	0.021
V-slot	20 × 80	2.56E+04	3.60E+05	7.17E+04	2.69E+04	80.00	0.015
V-slot	40 × 20	1.61E+04	6.10E+04	7.17E+04	2.69E+04	20.00	0.012
V-slot	60 × 20	1.97E+04	1.60E+05	7.17E+04	2.69E+04	20.00	0.005
V-slot	60 × 20	2.56E+04	3.60E+05	7.17E+04	2.69E+04	20.00	0.002

TABLE 4.3: Deflection of different rail systems

the MakerSlide is more expensive, the V-Slot 20 × 40 was chosen for a redesign of the frame.

4.4.2 Bad solenoid

The implementation of the solenoid was the most difficult task. It requires a custom-made driving circuit to be powered, and modifications of the controlling software. When having all this working, further testing revealed a big issue with this approach – the solenoid gets very hot.

Having it connected for 5 minutes and measuring its temperature with an infrared thermometer revealed its temperature to be 105 ° C on the metal frame, and 180 ° C on the coil. After the experiment, the inner plastic cylinder seem to have been deformed and the piston got stuck when it was pulled all the way in. The solenoid was proved to be useless. Better solenoids exist but they are not particularly cheap. Another option is to use another stepper motor even for the z axis, as was the idea from the beginning.

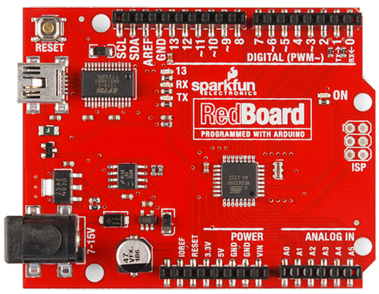
In addition to the temperature issue, the solenoid that is specified to have a stroke length of 10 mm. This was in reality 7 mm, and the pull force appeared to be highly non-linear – it would only pull 20 N at its maximum stroke. This might cause bad results due to variations of the bottle diameter, and need for an extra feature to adjust the solenoid manually for each bottle size. The solenoid idea was eventually discarded in favor the stepper motor solution.

Chapter 5

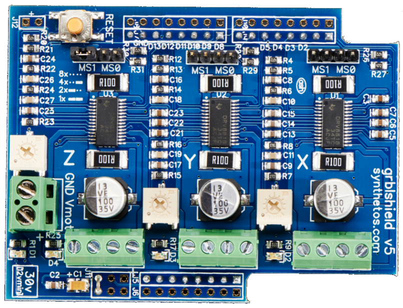
Solution

5.1 Detailed description of electronics

The heart of the machine is a Redboard (Figure 5.1), which is a cheaper clone of an Arduino Uno. It runs Grbl, an Open Source G-Code interpreting software that is specifically designed for the Arduino platform. Grbl translates the G-Code into pulse signals to control the stepper motors. These are driven by another module called gShield (Figure 5.2), that is designed to fit right onto the Arduino's standard pins. It has three stepper motor drivers and not only provides power for the motors, it also has safety functions for overheating and overcurrent. It also provides up to 8x microstepping, that can increase the motors' resolution by a factor of 8.



Source: `sparkfun.com`
FIGURE 5.1: Redboard



Source: `raspberry.com.ua`
FIGURE 5.2: gShield

5.1.1 Choice of stepper motors

The stepper motors has been chosen such that they will meet the requirements of precision accuracy, engraving forces and price. They all need different calculation methods that are presented in Table 5.1.

	y	a	z
Torque T	$\frac{Ftp}{\pi}$	$\frac{FD_{bottle}}{2\mathrm{r}}$	$\frac{FD_s}{2} \cdot \frac{\pi\mu D_s + l}{\pi D_s - \mu l}$
Resolution ρ	$\frac{N}{tp} \cdot n$	$\frac{N}{\pi D_{bottle}} \cdot n \cdot \mathrm{r}$	$\frac{N}{l} \cdot n$

F = engraving force, t = teeth on pulley, p = belt pitch, n = microsteps, r = gear ratio, N = steps / revolution, D_{bottle} = bottle diameter, D_s = leadscrew diameter, μ = leadscrew friction coefficient, l = leadscrew pitch

TABLE 5.1: Formulas used to calculate stepper motor requirements

y axis

A good starting point for choice of motors are the NEMA 17. They are the standard for RepRaps and other CNC homebuilt machines, hence cheap and available from many sources. But the NEMA standard only refers to its shape, and it exists in many variations. The cheapest, most common one has a resolution of 1.8 ° [7]. Using this with the formulas in Table 5.1 allows for calculating the resulting resolution ρ of the axis. To be able to move the engraving tool and operate with high speed, a minimum holding torque T must be overcome by several magnitudes.

$$\left.\begin{aligned} F &= 5\,\mathrm{N} \\ t &= 16\,\mathrm{teeth} \\ p &= 2\,\mathrm{mm} \\ N &= 200\,\mathrm{steps/rev} \\ n &= 8\,\mathrm{x} \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} T_y &= 81.5\,\mathrm{N\,mm} \\ \rho_y &= 50\,\mathrm{steps/mm} \end{aligned}\right. \tag{5.1}$$

The minimum resolution is obtain from the functional requirements for precision accoracy as $1/\epsilon = 5\,\mathrm{steps/mm}$. The bipolar NEMA 17 sized ROB-09238 from Sparkfun has a holding torque of 226 N mm and is well above these requirements. But the price of stepper motors is not necessarily corresponding to its size – smaller NEMA motors actually costs more. Extra power is usually not a bad thing, so this motor was chosen for the y axis.

a axis

Following the same procedure as for the y axis with the additional gear ratio r = 7.1 (see Section 5.2.2) and maximum bottle diameter $D_{bottle} = 100\,\text{mm}$, the torque and resolution is:

$$T_a = 56.3\,\text{N mm} \qquad \rho_a = 36.2\,\text{steps/mm}$$

The ROB-09238 will work even here. However, a higher resolution NEMA 17 (ROB-10846) was instead obtained due to error in calculations and not taking microstepping into acount. This motor has $N = 400\,\text{steps/rev}$, hence twice the resolution, and a holding torque of 480 N mm. It is even more overqualified, but was kept in the design. Higher resolution and torque will only produce better results.

z axis

The calculations for the z axis differs since it uses a leadscrew. With $N = 200\,\text{steps/rev}$ (which seems to be most common for NEMA motors of these sizes), $F = 8\,\text{N}$, $D_s = 5\,\text{mm}$, $\mu = 0.2$ [8] and $l = 0.8$ (standard M5 thread), the torque and resolution becomes:

$$T_z = 5.0\,\text{N mm} \qquad \rho_z = 250.0\,\text{steps/mm}$$

The motor should be as small as possible in order to minimize the weight of the yz carriage. But smaller motors increase dramatically in price. The NEMA 11 with 60 N mm holding torque is also much more powerful than needed, but was obtained because of its price.

5.1.2 Power supply

The gShield is recommended to be powered with 24 V according to the manufacturer [9], but should even work within 12 to 30 V. Since the Arduino and the previously used solenoid run on maximum 12 V, a 12 V, 6 A power supply was purchased for the earlier design. This was a bad choice since it did not provide the motors with enough power – they would skip steps when accelerating too fast. When not using the solenoid, the Arduino can be powered through the USB cord separately from the gShield that then can be operated on higher voltages. This will also have the motors draw less current which will make them run cooler. An 18 V, 4 A laptop charger proved work very well and is still being used in the final version.

5.2 Detailed description of design

The switch to V-Slot resulted in a new wall design, and the switch to stepper motor required a new sliding carriage. The new result can be seen in Figure 5.3. The laser cut parts are now white to better match the color of the Acrylic.

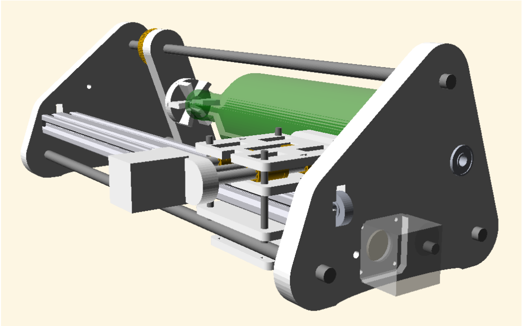


FIGURE 5.3: Final CAD version

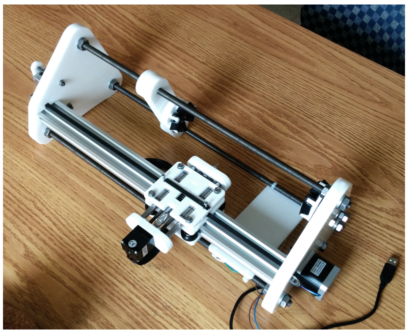


FIGURE 5.4: Final build

5.2.1 Bottle attachment

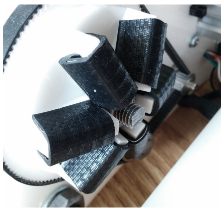


FIGURE 5.5: Bottom bottle grip

A bottle attachment (Figure 5.5) was designed with inspiration from existing bottle grips for rotary CNC attachments. The idea is that one cone-shaped base fits into the punt (the dimple in the bottom) of the bottles, and a funnel-shaped smaller base grips the opening when when they are tightened together. The grips are laser cut and hold together by tight fit. Many attempts were made before the fitting was right – a too tight fit would make the Acrylic crack and a too lose fit would not make it stay in place.

5.2.2 Gear reduction for the a axis

When using a rotary axis, the positioning accuracy will depend on the diameter of the bottle. An increase in diameter will result in a decrease of accuracy, since the angular step remains constant but the circumference will change. To meet the functional requirement of a positional accuracy of 0.2 mm, the stepper motor needs to have a gear reduction. This will also provide a greater moment to the bottle, which probably will be required in order to overcome the forces involved in engraving.

The necessary gear ratio depends on the accuracy of the stepper motor and the desired maximum diameter of the bottle. See Appendix A.3 on how it can be calculated with Equation A.17:

$$\mathrm{r} = \frac{D\pi}{nd_{\mathrm{des}}}$$

Refer to Appendix A.3 for complete calculations. For using a Nema17 stepper motor with $n = 200$ steps per revolution together with a bottle diameter of $D = 90$ mm, with $d_{\mathrm{des}} = 0.2\,\mathrm{mm}$ will require a gear ratio of:

$$\mathrm{r} = \frac{90 \cdot \pi}{200 \cdot 0.2} = 7.1$$

A GT2 belt system was chosen. It is a metric belt with 2 mm teeth width and 6 mm belt width, the same as is being used for most RepRaps, which indicates low price. A 16 teeth timing pulley was obtained as the lower gear, while the bigger gear was laser cut using a parametric OpenSCAD design found on Thingiverse. It was very hard to get the dimensions right, many gears has been cut and failed to get the right fit for the belt. The pulley generation code had a bug which generated incorrect gears for bigger dimensions, which was could be fixed after some troubleshooting. The number of teeth needed to achieve the required gear ratio is $16 \cdot 7.1 = 114$ teeth. The timing belt length can then be calculated from with [**?**]:

$$L = \sqrt{4C^2 - (D-d)^2} + \frac{1}{2}(D\theta_D + d\theta_d) \tag{5.2}$$

where

$$\theta_d = \pi - 2\sin^{-1}\frac{D-d}{2C} \qquad\qquad \theta_d = \pi + 2\sin^{-1}\frac{D-d}{2C}$$

D and d is the diameter of the bigger versus smaller pulley, and C is the center-to-center distance between them. Since we want to know the number of teeth and not the length, are we using the pitch diameter instead and the pitch center-to-center distance instead:

$$D = \frac{114}{\pi} \qquad\qquad d = \frac{16}{\pi} \qquad\qquad C = \frac{L}{2\,\mathrm{mm}}$$

The distance L between the pulleys is 96.92 mm. Plugging this into Equation 5.2 gives us a belt length of 163 teeth.

5.2.3 Engraving tool

The diamond-drag toolbit used was first the 1/8" diamond tool that was used for the engraving experiments, but later on replaced by a 120 ° drag engraver purchased from Diamond-Tools.com. It is mounted to to the z slider using a propeller mount for radiocontrolled airplanes (see Figure 5.7). It is has M5 threads in one end for the intended propeller. and a 1/8" motor shaft mount in the other. When mounting this on an Acrylic plate with a regular M5 nut as in Figure 5.10, tensioning the nut will also grip the engraving tool. This is a very cheap solution compared to generic CNC tool chucks.

The same propeller mount is also being used as the coupler that attaches the z stepper motor to the leadscrew (see Figure 5.11. By drilling the shaft hole to 5 mm and

FIGURE 5.6: Diamond-drag toolbit

FIGURE 5.7: Propeller holder

cutting the top off the spinner, it fits on to the stepper motor shaft and can join with the leadscrew.

5.2.4 Integrated cables

Using three-layered walls allows for completely integrating the cables into the walls by making cutouts in the middle layer as seen in Figure 5.8. Figure 5.9 shows how the four cables for the y stepper motor has been completely hidden by leading them through both walls and the V-Slot. All cables for the whole machine exits through one single whole on the right wall underneath the V-Slot.

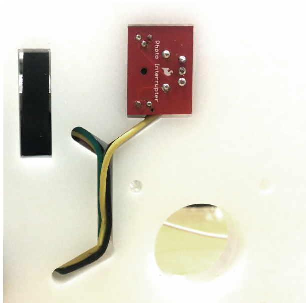


Figure 5.8: Cable channels from the optical limit switch

Figure 5.9: y stepper cables led through the V-Slot

5.2.5 XZ slider

The slider in Figure 5.10 is the most complex detail on the machine. The z axis is sliding through four linear bearings that was first 3D printed out of ABS plastic from a model found at Thingiverse.com. They are hold in place by tightening the plates together.

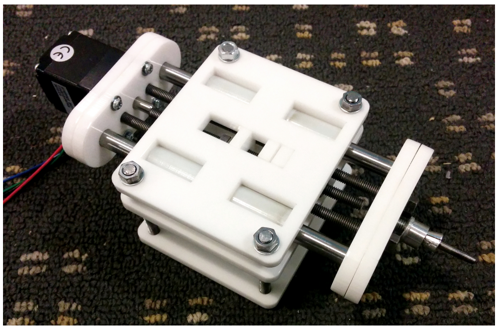


Figure 5.10: yz carriage

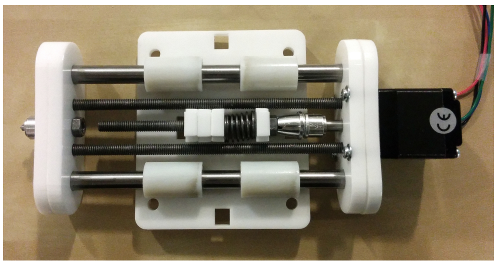
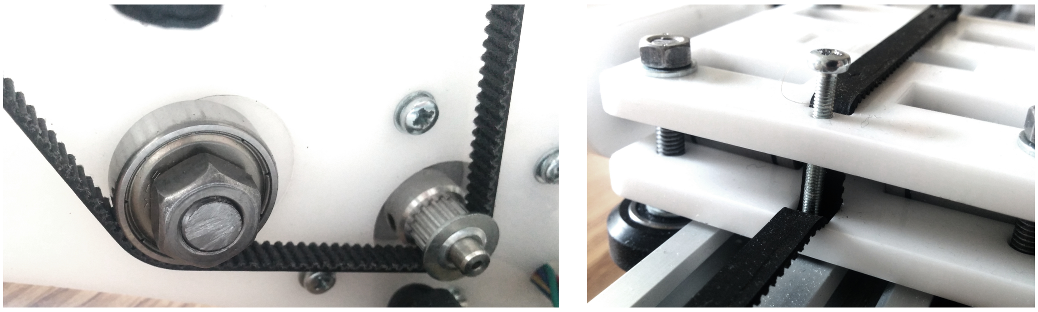


Figure 5.11: Mechanism of the carriage

The z stepper is not directly pushing the engraving bit, but instead compressing a spring that is pushing it against the surface of the bottle. This will allow for variations of the diameter as the bottle rotates, due to imperfect bottles or non-centered alignment. The mechanism is seen in Figure 5.11.

5.2.6 Belt tensioning

The downside of using belts instead of leadscrews is that they need to be tensioned. The y belt is open and attached to the carriage while the a belt is in a closed loop. These needed two different solutions that are shown in Figure 5.12. With the goal of using only leftover parts from the build, the a tensioner in (A) is simply just two M3 screws and a few laser cut features. These also attaches the belt to the carriage. For the a axis, a circular cutout in the wall with an eccentric hole makes base for an adjustable bearing as seen in (B), that is tensioning the belt by pushing it upwards.



(A) (B)

FIGURE 5.12: Belt tensioners for the a and y axis.

5.3 Software toolchain

The process of converting an image into G-Code is made in several steps. The source image can be practically anything – a photo, a sketch, a photo of a sketch, vector graphics etc. Most of the time, it is raster images from a sketch or from graphics found on the Internet. Different motives are obviously more or less suitable – engraving is about lines and contours, no colors or shading is possible. Following is an example on how to translate the hand-drawn sketch as seen in Figure 5.13 sketch into motions for the machine. It is first captured by simply taking a photo with a smartphone.

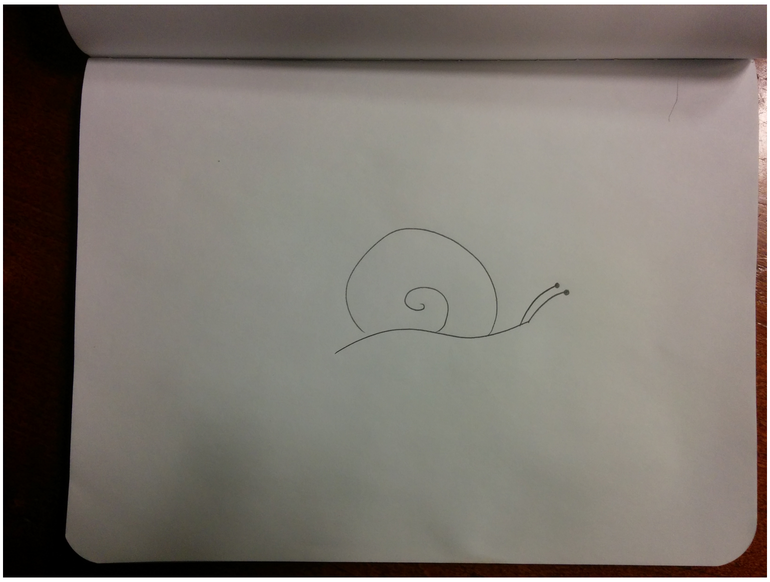


FIGURE 5.13: The original photo of the sketch that is going to be converted.

5.3.1 Enhancing

The raster image first needs to be pre-processed by enhancing its contours. This is easily done with GIMP, using the Levels and Threshold functions. The image is then cropped to contain only the sketch, and looks as Figure 5.14.

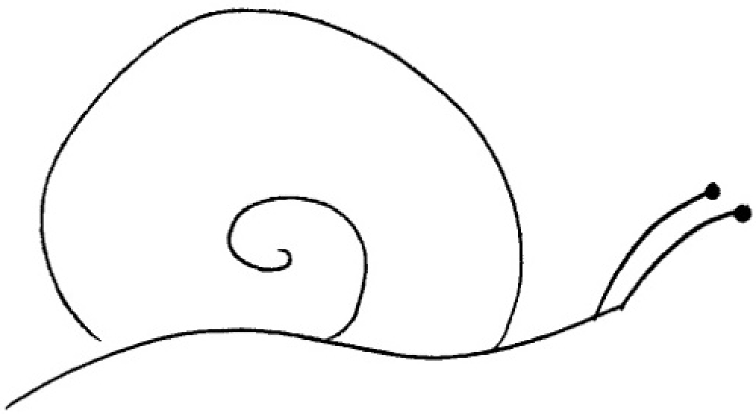


FIGURE 5.14: Enhanced using GIMP

5.3.2 Tracing

The image should now be converted into vectors. What method to use depends on what kind of graphics it is. There is no way to sketch actual lines on paper – in a magnifying glass they are all just thin filled regions. The outlines of these are easy for a computer program to trace. Figure 5.15 shows the image traced using the Trace bitmap feature in Inkscape. Every line will result in a double engaved line, following the outlines of the original sketched lines.

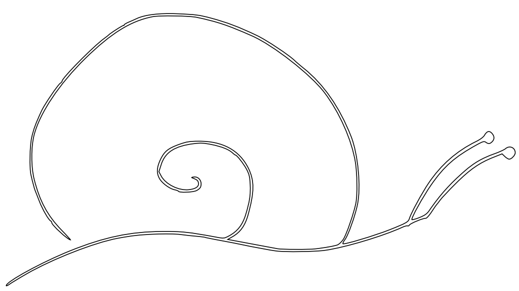


FIGURE 5.15: Traced outlines using Inkscape

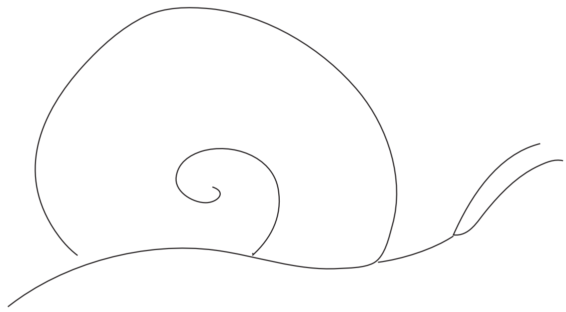


FIGURE 5.16: Traced centerlines using autotrace

A better solution in this case would instead be to use a centerline tracing algorithm. Inkscape does not provide this feature, but another CLI application named autotrace does. Figure 5.16 shows how the image looks after being traced with autotrace. This only consists of single vector lines, which results in that filled areas are not traced – the eyes of our snail is gone. There is no way to get a perfect conversion from raster to vectors, and manual editing is often needed.

5.3.3 Convert to G-code

With the sketch in vector format there are numerous ways to convert it into G-Code. Inkscape supports exporting to DXF which is the default format for most G-Code generatorion software, but this is also capable from within Inkscape itself with the third-party extension GCodeTools.

The extension is being used by first creating orientation points for getting the right scale of the engraving, and then specify parameters for the tool that is being used. These orientation points are different for bottles with different diameters, but can be saved as templates for repeated use. It is enough by having two templates for diameters of 50 and 75 mm to cover most bottles – it does not have to be very precise. A miscalculated orientation point in the a direction will result in a stretched or compressed image.

When these properties are set, the paths can be selected and converted directly to a G-Code file.

5.3.4 rezet.py for non-straight bottles

If engraving on staight bottles as in Figure 5.17, this step can be skipped. Since GCodeTools does not provide any suitable tools for adjusting the z depth for non-straight bottles, a customized G-parser was programmed in Python to solve this problem. It resets z according to a provided diameter table, hence the name rezet.py.

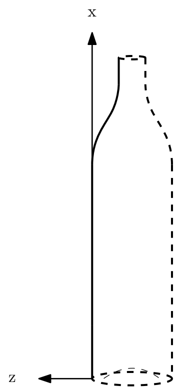


FIGURE 5.17: Straight bottle, $z = 0$

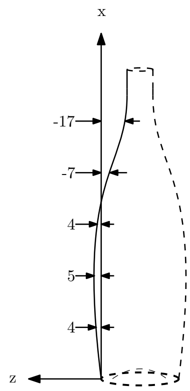


FIGURE 5.18: Curved bottle, z = variable

The required arguments that needs to be provided to the script are listed in Table 5.2. The target bottle first has to be measured on a number of points that will represent

Identifier	Default value	Description
-i	gcode.ngc	The file that contains the G-Code source
-o	gcode-rezet.ngc	The file to save the result to
-r	1	The desired path splitting resolution
-x	—	Sample points in the x direction
-y	—	Sample points in the y direction
-zx	—	Sample points for diameter difference in the x direction
-zy	—	Sample points for diameter difference in the y direction
-z1	999	Maximum allowed z value
-z0	-999	Minimum allowed z value

TABLE 5.2: Arguments for rezet.py

the shape of the bottle. All diameters are then subtracted by the diameter at $y = 0$ and divided by 2 in order to get the radius difference $x(y)$ function for the bottle as seen in Figure 5.18.

When executing rezet.py with this data as parameters, it will parse the G-Code file and trace the toolpath. Since G-Code is not just points in space but instead directions for motions in straight lines and arcs, this is more complicated than it seems.

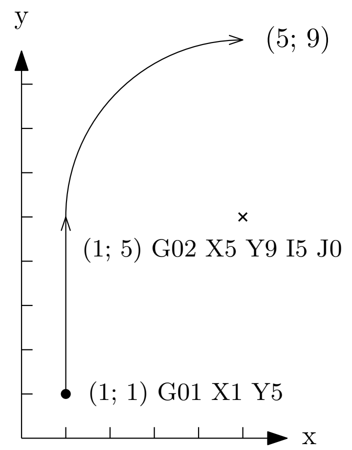


FIGURE 5.19: Example of a straight line and arc motion

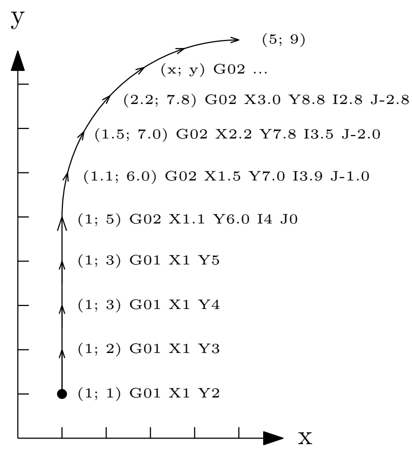


FIGURE 5.20: Illustration of the result of rezet.py

Linear motions are defined with the codes G00 and G01, and arc motions with G02 and G03. The linear motions are simply just coordinates coordinates for the new machine position, while arc motions provide the destination coordinates to which the machine will travel, but in a clockwise (G02) or counter-clockwise (G03) arc around the centerpoint specified by two offset coordinates I and J. Figure 5.19 shows an example of what the two codes can look like.

rezet.py will parse the G-Code and look for these codes. For every motion, it will calculate the travel distance and divide it up into a number of subpaths depending on the chosen resolution. The new subpaths will then have their z values adjusted by interpolating between the provided z sample points for the given coordinates, and adding it to the existing z value. The result is illustrated in 5.20.

5.3.5 Non-cylindrical objects

Since the script also takes parameters for z values in the x direction, almost any shape can be compensated for. The radius difference in a symmetric polygon with N sides can be calculated with

$$z = R\left(\frac{1}{\left|cos\left(\left(\theta \mod \frac{2\pi}{N-1}\right) - \frac{\pi}{N-1}\right)\right|} - 1\right) \tag{5.3}$$

where θ is the angle in radians and R is its minimum radius. This formula can easily be sampled and plotted as in Figure 5.22 using numerical computation software such as Octave. These sample points can then be fed into rezet.py.

In theory, all regions with $N \geq 5$ should be possible to engrave on. Squares and a triangles are impossible – the diamond tip with a 120 ° cone shape will not be able to reach the surface. In reality, a bad engraving result is to be expected on shapes that differ too much from a cylinder. The engraving tool will not be perpendicular to the surface which will cause different cuts depending on where the engraving is taking place.

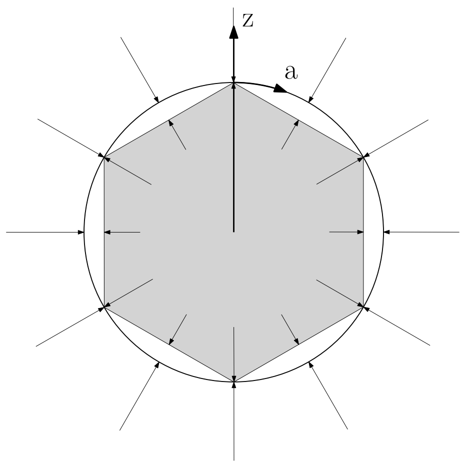


FIGURE 5.21: Hexagon, $N = 6$

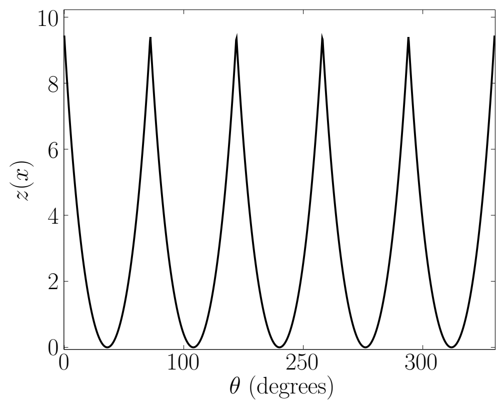


FIGURE 5.22: Radius difference of a hexagon with $R = 40$

5.3.6 Send to Grbl

The G-Code is now ready to be sent to the machine. There are several options for this, since the process is quite basic – just send the G-Code over to Grbl over the USB interface, one line at a time. When the command finished executing, Grbl responds with an 'ok' and the next command can be sent.

Universal G-Code Streamer is an Open Source cross-platform application that does this and also provides other useful features like visualizing the G-Code to check if it

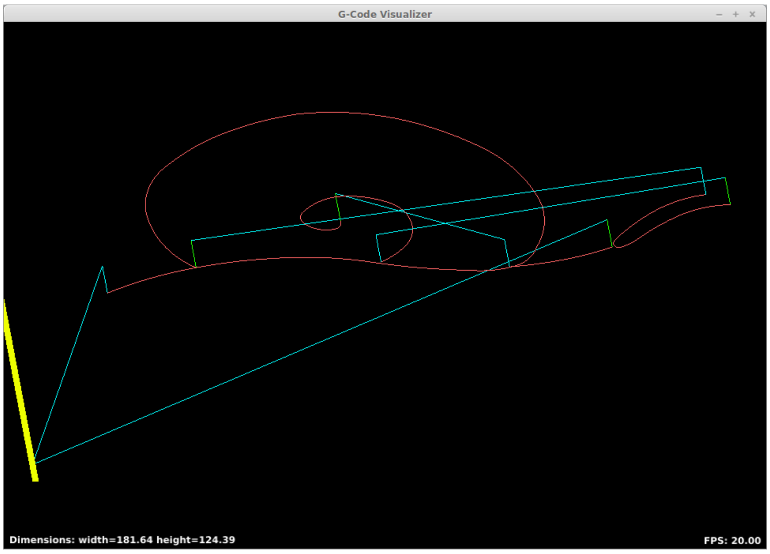


FIGURE 5.23: G-Code as previewed by Universal G-Code Streamer

is correct as in Figure 5.23. The G-Code file is streamed to Grbl with the click of a button. All that is left is to sit back and watch the engraving show.

5.4 Testing and results

The first engraving results with the first engraving toolbit was not satisfying enough – the cut was shallow and not very smooth. Tiny glass fragments would break loose along the engraved path. The change to the new tool drastically improved the results – it cuts deeper and produces a smooth, white line as in Figure 5.25. This figure required 16 minutes to complete.

FIGURE 5.24: Result from the first engraving tool

FIGURE 5.25: Result with the new engraving tool

Colored bottles are best suited for engraving. The result is not as distinct on transparent bottles, since the white strokes does not differ much in color from the glass.

Curved bottles are no problem to engrave on after using rezet.py, but non-cylindrical bottles have not yet been tested.

5.4.1 Source code

OpenSCAD is basically just coding, and no traditional drawings have ever been made in this project. Therefore, no drawings will be presented here. Neither will the source files, since code is useless on paper. All source files are instead available online at `github.com/farbo/bengraver`. This includes over 1000 lines of code plus scripts as rezet.py, DXF and STL file exporters.

5.4.2 Bill of materials

The complete bill of materials is show in Table 5.3. This only includes the material used in the final model, not the material that was purchased but ended up not being used. It does not include shipping expenses.

5.5 Assessments

The engraving results are well above what was expected when designing this machine. It fulfills most functional requirements, but there are some minor notes that need to mentioned.

5.5.1 Cost

The budget of $300 was a very good estimation of the total cost, but not if shipping costs are included. This could be reduced by purchasing tha fasteners in a lower quantity – they all shipped in packs of 100, so most of them remain unused.

Part	**Location**	**ID**	**Qty**	**Each**	**Total**
Nema 17	Sparkfun	ROB-09238	2	$14.95	$29.90
gShield	Inventables		1	$50.00	$50.00
Arduino	Sparkfun	DEV-12757	1	$25.29	$25.29
12V power supply	Amazon		1	$9.31	$9.31
Nema 11	Amazon		1	$17.99	$17.99
Photo interrupter	Sparkfun	SEN-09299	1	$1.95	$1.95
Photo interrupter breadboard	Sparkfun	BOB-09322	1	$1.50	$1.50
Spring, Ø8.5 x 20 mm					$0.00
608ZZ bearings	Amazon		1	$9.00	$9.00
Smooth rod, Ø8 mm	McMaster-Carr	88625K67	2	$5.70	$11.40
Threaded rod, M10 x 1.5	McMaster-Carr	99055A130	2	$6.47	$12.94
Threaded rod, M8 x 1.25	McMaster-Carr	98861A080	1	$2.28	$2.28
Nuts, M10 x 1.5	McMaster-Carr	90592A024	1	$8.52	$8.52
Nuts, M8 x 1.25	McMaster-Carr	90592A022	1	$4.51	$4.51
Washers, M3 x 3.2	McMaster-Carr	91455A330	1	$2.61	$2.61
Washers, M10	McMaster-Carr	91166A280	1	$4.36	$4.36
Washers, M8	McMaster-Carr	91166A270	1	$3.23	$3.23
M3 x 0.5 * 25 mm machine screws	McMaster-Carr	92005A130	1	$3.28	$3.28
GT2 11t pulley	Inventables	26054-09	2	$5.87	$11.74
GT2 closed belt 163t	Misumi	GBN3262GT-60	1	$5.50	$5.50
GT2 belt open	Inventables	26053-01	4	$1.99	$7.96
RC Prop mount 1/8 in	Amazon		1	$5.95	$5.95
Diamond-drag toolbit	diamondtool.com	DR1251	1	$10.45	$10.45
V-slot wheel	OpenBuilds	SKU 480	6	$3.95	$23.70
V-slot rail	OpenBuilds	SKU LP-155	1	$6.50	$6.50
Acrylic 1/8 in			2	$15.00	$30.00
				Total	**$299.07**

Table 5.3: Bill of materials

5.5.2 Bottle dimension capabilities

It is capable of engraving on diameters between 30 and 90 mm. This includes most beer and wine bottles on the market. The maximum diameter was in Section 2.4.4 specified to 100 mm, a requirement that could not be met. Making the machine able to engraver big bottles will also reduce its quality in small engravings, since the increased length of the z axis will increase the deflection of the construction.

5.5.3 Positioning accuracy

The theoretical resolution is as much as 10 times higher than the functional requirement, but this is hard to confirm with measurements. A visual study of the results show that artifacts smaller than about 0.2-0.5 mm will not be engraved with precision due to deflection in the carriage. This can probably be improved by having the z axis use a V-Slide system as well.

5.5.4 Open Source

The only proprietary software that has been used for this project is the software that is being used for preparing the laser cutting in the Idea Shop – mainly AutoCAD. The machines used for manufacturing also use proprietary sofware, but the whole design process made completely with FOSS.

5.5.5 Reliability

The success of engraving depends on how well prepared the G-Code is. The software that is being used contains bugs – especially GCodetools which often outputs faulty GCode when converting complex paths. The G-Code should always be previewed in a visualizer before engraving.

The engraving process sometimes fails when an axis gets stuck. If engraving too fast over a surface seam as seen in Figure 2.4, the stepper motors might skip a few steps. When this happens, the engraver gets completely lost and there is nothing else to do than discard the bottle and try again. There is no feedback for the position of the engraving tool, the machine has to be calibrated for the right acceleration, speed and engraving force to avoid that this happens.

Chapter 6

Conclusions

This project is a success. A very competent machine has been designed and built for a very small price. It is lightweight and practical compared to existing solutions. It is not very user friendly and it takes a bit of learning in order to operate it, but it is not intended for the ordinary user. It is a cheap and fun solution people with technical interests and crafting skills. Many Open Source CNC machines can be found on the Internet, and this is yet another one but with an different purpose.

Another intention with this project was to prove the potential of FOSS. Most of the problems that has been dealt with is already solved by the Open Source community, it is just a matter of finding these solutions. If they are not completely applicable onto the intended application, everything is provided for in order to make it work yourself. This machine would probably not exist without free software – the software is by far the most complex part of this machine. Using proprietary products would not be possible within this budget.

Chapter 7

Future development

There are many improvements that can be made to the machine to make it cheaper and simpler. It uses some special parts that are expensive and difficult to obtain. Most critical is the belt for the a axis. It is in a closed loop and has to be of a specific length. It is not necessarily expensive, but hard to get. If the belt could be open-ended, the same belt can be used as for the y axis. An even cheper solution would be to use laser-cut gears or another pulley system, but this would introduce another problem with backlash.

Furthermore, there are some minor an major changes features that can be added in order to increase the quality of the engravings and make it more user friendly.

- By simply adding a contact or distance sensor to the engraving toolbit, it is just a matter of software changes to have the engraver automatically adjust the Z depth to the bottle without using rezet.py. It would take measurements of the variation in the diameter by point by point lower the engraving bit until it get in touch with its surface. The depth would be recorded and used to adjust the depth when engraving. If using a force sensor to measure the pressure on the engraving tool, the depth could be automatically changed during the engraving instead. This would be a much more complicated implementation.

- Rotary engraving can be made possible by redesigning the carriage for implementation of a rotary spindle. This would not only produce smoother engraving results, but also open up a whole new world of other ways to use this machine.

By being able to attach standard routing tools to this machine, it would be able to perform lathing and milling on a wide range of different materials.

- Buttons for controlling the machine manually should be implemented in order to easily position the machine for different kinds of bottles. This can now be made from within Unviversal G-Code Sender, but is not very user friendly.

- Design other grips that makes it easier to attach other objects, e.g. glasses, to the machine.

- Make it wireless either with Bluetooth or by using a server running on RaspberryPI. There are already solutions out there to have the RaspberryPI stream G-Code and accessible through a web interface, which would get rid of the tedious work of always having to connect the USB cable all the time.

- An upgrade to 24 V would increase the engraving speed.

Many people have shown interest in this machine, and the development will continue even after completion of my degree. The project is already published on the Internet, but noone has found it yet. The next step will be to make sure that people do.

Appendix A

Calculations

A.1 Deflection in rods

The calculations are simplified by neglecting torsion in the rods and bending in z. The rods are only supported by the scores in the walls at each end, and they can be calculated as a beam with two simple supports (see Figure 4.16). Maximum deflection can then be obtained by [**?**]:

$$\delta = \frac{FL^3}{48EI} \tag{A.1}$$

Where F is the load in the middle of the rod. The deflection the two rods at A and C will result in a larger deflection in D:

$$\delta_D = L_{BD}\frac{\delta_A - \delta_C}{L_{AC}} \tag{A.2}$$

Since I and E are the same for both rods, the equations can be simplified to:

$$(A.1),(A.2) \Rightarrow \delta_D = -\frac{L_{BD}L^3(F_A - F_C)}{48L_{AC}EI} \tag{A.3}$$

Static equilibrium gives $F_A - F_C$:

$$\sum F_y = F_A - F_C + F_D = 0 \Rightarrow F_A - F_C = -F_D \tag{A.4}$$

The deflection in D will be:

$$(A.3), (A.4) \Rightarrow \delta_D = -\frac{L_{BD}L^3F_D}{48L_{AC}EI} \tag{A.5}$$

A.2 Deflection in aluminium extrusions

The deflection in the aluminium extrusions differ from the rods since they are only one single beam. It can be calculated using Castigliano's Theorem, which says that the deflection in the direction of FD is equal to the derivative of the strain energy of the beam with respect to the force. [**?**]

$$\delta = \frac{dU}{dF} \tag{A.6}$$

The tangential force at the engraving bit results in a bending force F and a torsion T on the beam. The bending moment M caused by F is:

$$M = \begin{cases} \frac{Fx}{2}, & \text{if } 0 < x < \frac{L}{2}, \\ \frac{F}{2}(L-x), & \text{if } \frac{L}{2} < x < L \end{cases} \tag{A.7}$$

The torsion is constant throughout the beam.

$$T = L_{BD}F \tag{A.8}$$

The torsion and the bending moment accounts for the total strain energy:

$$U = U_T + U_M \tag{A.9}$$

$$U_T = \frac{T^2L}{2GJ} \tag{A.10}$$

By symmetry, from (A.7):

$$AU_M = 2\int_0^{\frac{L}{2}} \frac{M^2}{2EI}dx = \frac{F^2L^3}{96EI} \tag{A.11}$$

Substituting into (A.8) and taking the derivative with respect to F finally gives the deflection:

$$\delta = \frac{FL^3}{48EI} + \frac{L_{BD}^2FL}{GJ} \tag{A.12}$$

A.3 Gear reduction

The number of degrees per step can be calculated with

$$\text{deg/step} = \frac{360°}{n} \tag{A.13}$$

where n is the number of steps per revolution. 1 º of rotation will correspond to a distance along the surface according to

$$\text{distance/deg} = \frac{D\pi}{360°} \tag{A.14}$$

where D is the diameter of the bottle. Multiplication of (A.13) and (A.14) gives the distance d acquired per step:

$$d = \frac{360°}{n} \cdot \frac{D\pi}{360°} = \frac{D\pi}{n} \tag{A.15}$$

For achieving the desired accuracy d_{des}, d needs to be reduced with a factor of $\frac{1}{\text{r}}$ where r is the gear ratio:

$$d \cdot \frac{1}{\text{r}} = d_{\text{des}} \tag{A.16}$$

Substituting (A.15) and solving for r finally gives us the required gear ratio

$$\text{r} = \frac{D\pi}{nd_{\text{des}}} \tag{A.17}$$

References

[1] Jackie Zack. Engraver's journal, 2015. URL https://www.engraversjournal.com/article.php/2597/index.html.

[2] Johnson Plastics. The engraver's bible, 2015. URL http://www.johnsonplastics.com/engraversbible/diamonddrag.html.

[3] Productions MAJ. How it's made, season 8, episode 2. 2007.

[4] RepRap, 2015. URL http://reprap.org.

[5] Evil Mad Scientist Laboratories, 2015. URL http://egg-bot.com.

[6] Göran Gustafsson. Konstruktionsmetodik. 2012.

[7] RepRap.org. Stepper motor, 2015. URL http://reprap.org/wiki/Stepper_motor.

[8] Inc Daycounter, 2015. URL http://www.daycounter.com/Calculators/Lead-Screw-Force-Torque-Calculator.phtml.

[9] Synthetos. Using gshield, 2015. URL https://github.com/synthetos/grblShield/wiki/Using-grblShield.