Python: A "Toy" Language

David Beazley (@dabeaz)

Is this about toy problems?

```
def fibonacci(n):
    if n <= 2:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

No, let's talk about kids...

(oh no...)

An amazing science project

Advanced Biology

An amazing science project

Advanced Biology

... and Chemistry

An amazing science project

Advanced Biology

... and Chemistry

Machine Learning

An amazing science project

Advanced Biology

... and Chemistry

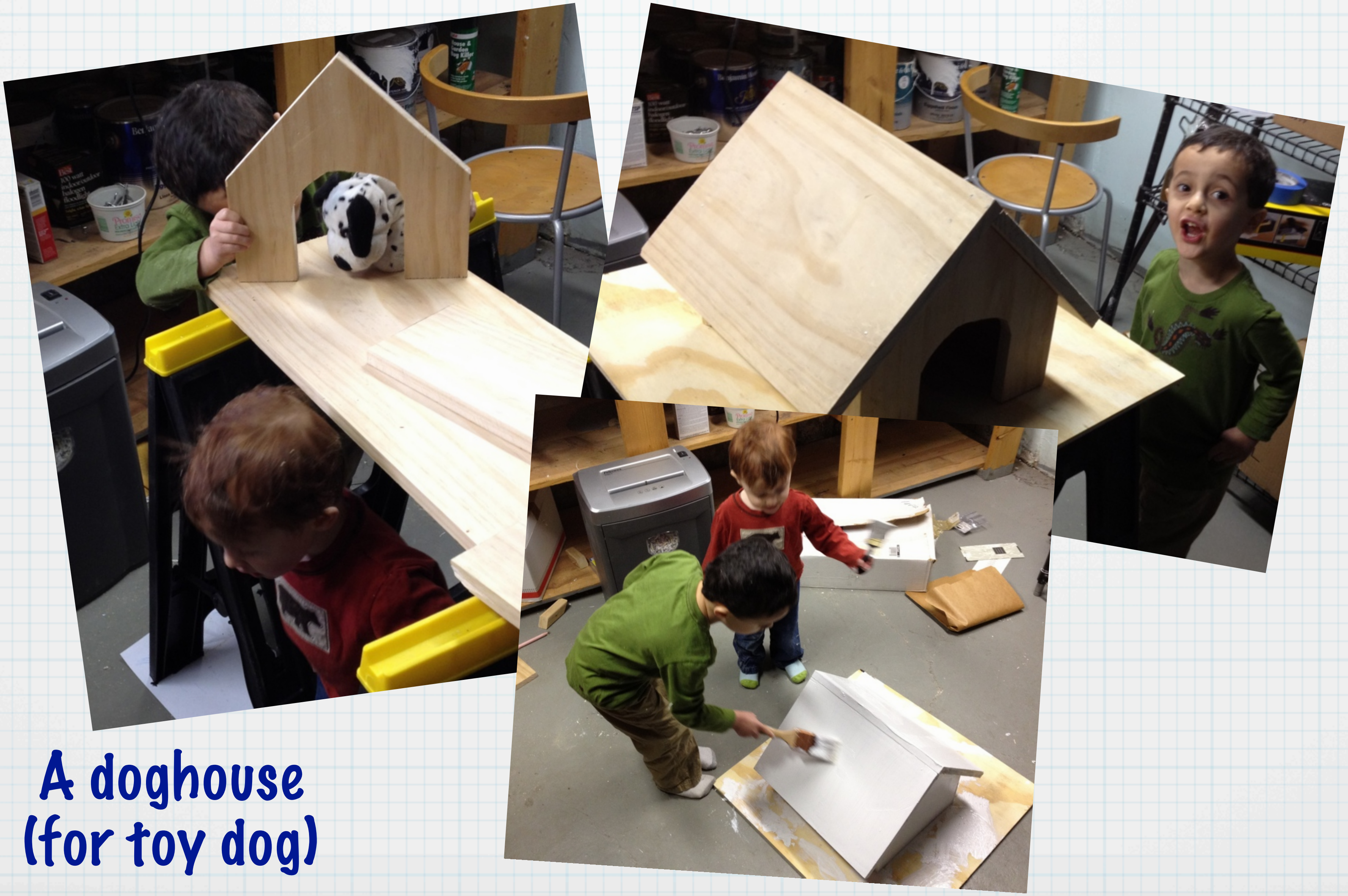
Machine Learning

A natural hacking instinct

You Get to Build Things

Diabolical block towers

You Get to Build Things



A doghouse
(for toy dog)

You Get to Build Things

Or maybe a birdhouse...
(for real birds)

Hacker Pro-Tip

Kids are the ultimate excuse for buying "tools"

- Drill press
- Telescope
- Magnifying glass
- Catapult
- Oscilloscope
- Soldering Iron
- Laser
- Welding torch

Hacker Pro-Tip

Kids are the ultimate excuse for buying "tools"

- Drill press
- Telescope
- Magnifying glass
- Catapult
- Oscilloscope
- Soldering Iron
- Laser
- Welding torch

You know, for kids. The big one. Yourself.

Question:

how do you get your kid to want to be a diabolical pirate scientist BDFL?

Question:

how do you get your kid to want to be a diabolical pirate scientist BDFL?

... in space.

A thought:

maybe I could make toys

A thought:

maybe I could make toys

with Python?

A thought:
maybe I could make toys
with Python?

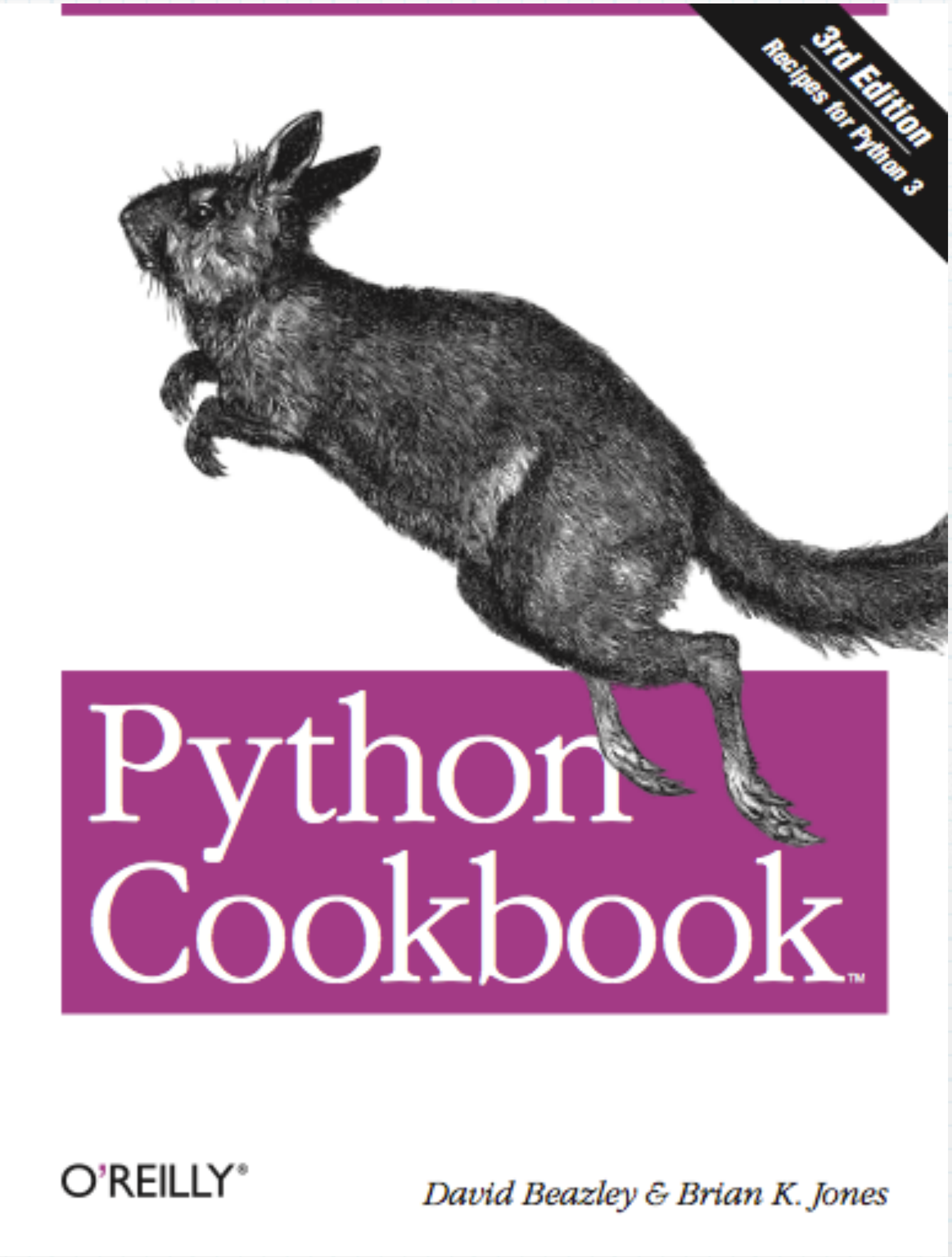
Python + Kids + Building Stuff == Fun

A thought:
maybe I could make toys
with Python?

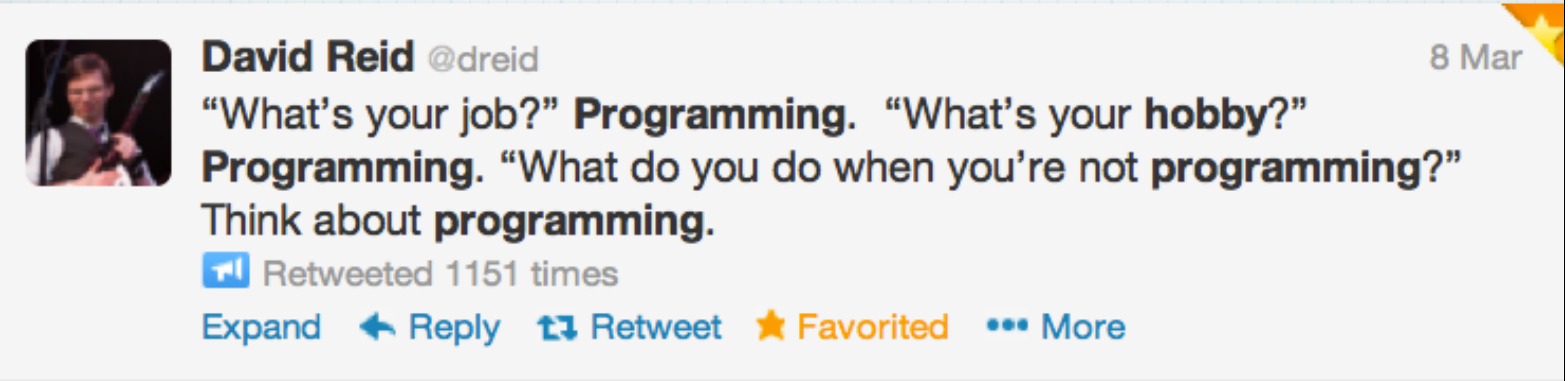
Python + Kids + Building Stuff == Fun

(also good way to avoid working on book)

Yes, this book



Seriously though...



David Reid @dreid 8 Mar

"What's your job?" **Programming.** "What's your **hobby**?" **Programming.** "What do you do when you're not **programming?**" Think about **programming.**

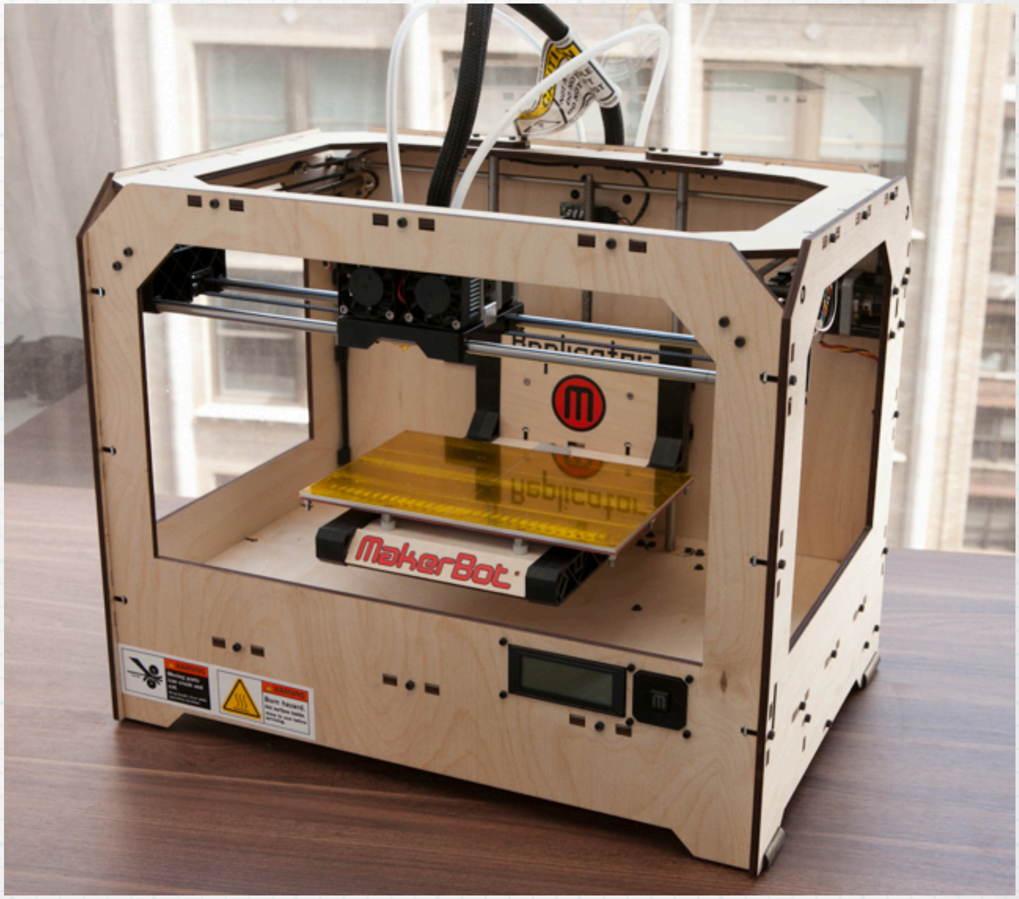
Retweeted 1151 times

Writing Python programs to make toys...
well, yeah. Duh!

Making Things

There's a bit of "maker" movement going on

- 3D Printing
- Hacker spaces
- Arduino, Raspberry Pi
- DIY



It's been on my mind

- "Oh, that might be cool"

However, I've never been that actively involved

Issue

Drawing and clicking "print" is a big "meh."

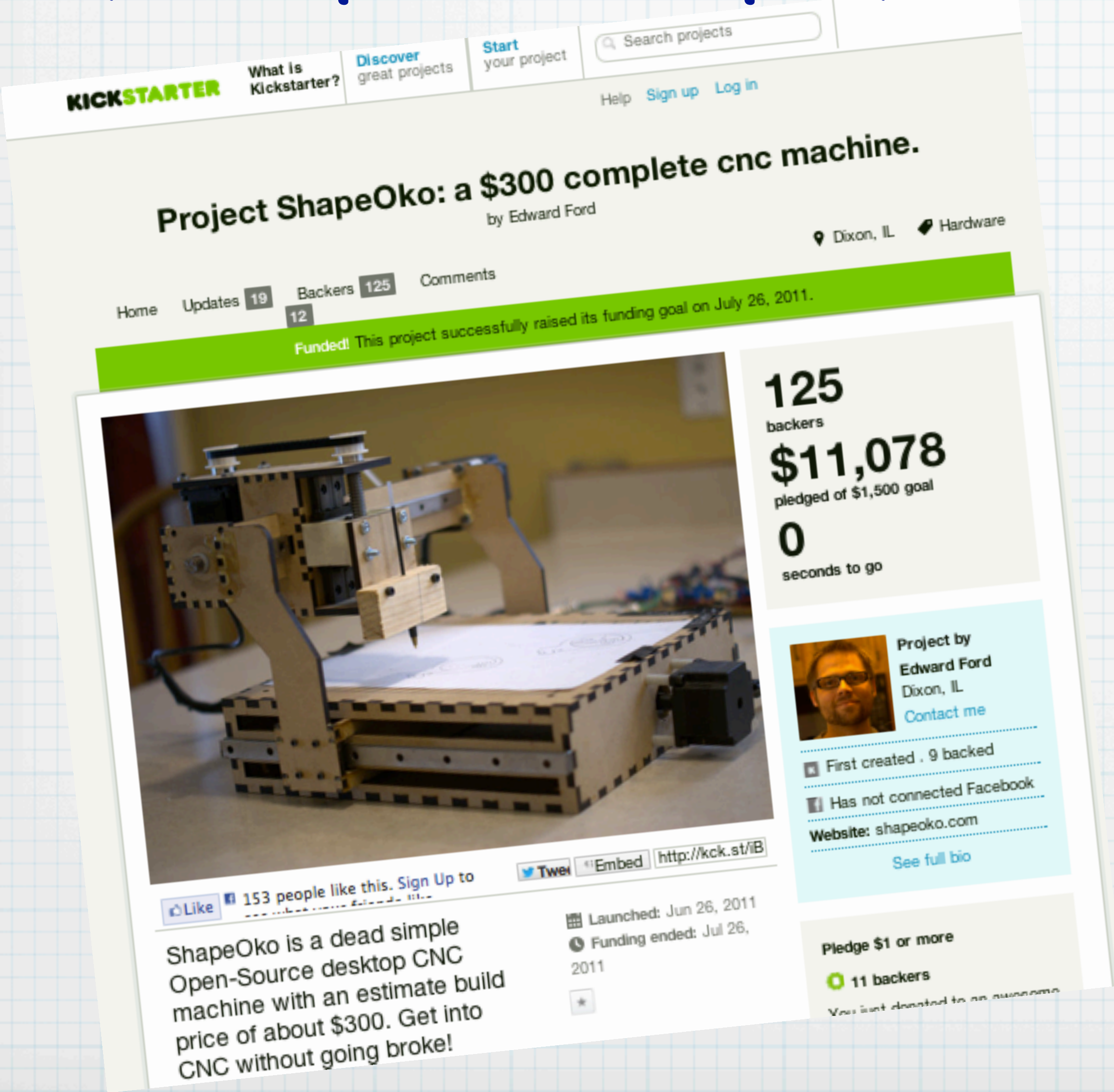
3D printing seems just a bit too magical

If machine breaks, could I figure out how to fix it?

More enjoyment from figuring out how to do something than actually doing it

DIY CNC Milling

So, shortly after last PyCon, this caught my eye...



Project ShapeOko: a $300 complete cnc machine.
by Edward Ford

Dixon, IL Hardware

Funded! This project successfully raised its funding goal on July 26, 2011.

125
backers

$11,078
pledged of $1,500 goal

0
seconds to go

Project by
Edward Ford
Dixon, IL
Contact me

First created . 9 backed
Has not connected Facebook
Website: shapeoko.com
See full bio

ShapeOko is a dead simple Open-Source desktop CNC machine with an estimate build price of about $300. Get into CNC without going broke!

Launched: Jun 26, 2011
Funding ended: Jul 26, 2011

Pledge $1 or more
11 backers

- ShapeOko Project
- 1-man project (Edward Ford)
- Chicago area
- Super cheap $

CNC Milling

In a nutshell: Computer controlled whirling knives

- Scary speed
- Lot's of noise
- Flying chips
- Thrilling Danger!

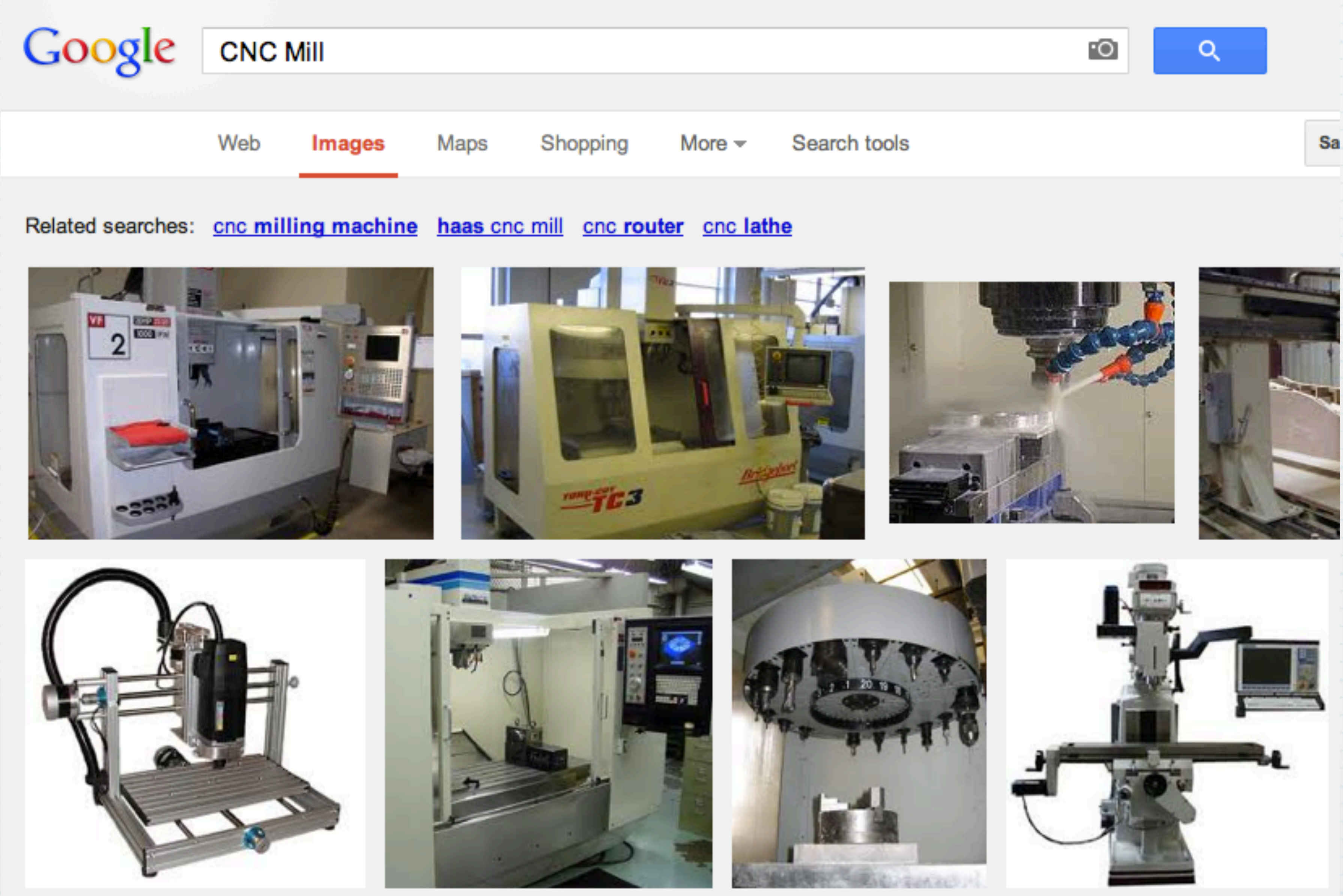
CNC Milling

In a nutshell: Computer controlled whirling knives

- Scary speed
- Lot's of noise
- Flying chips
- Thrilling Danger!

It sounded perfect! You know, for kids...

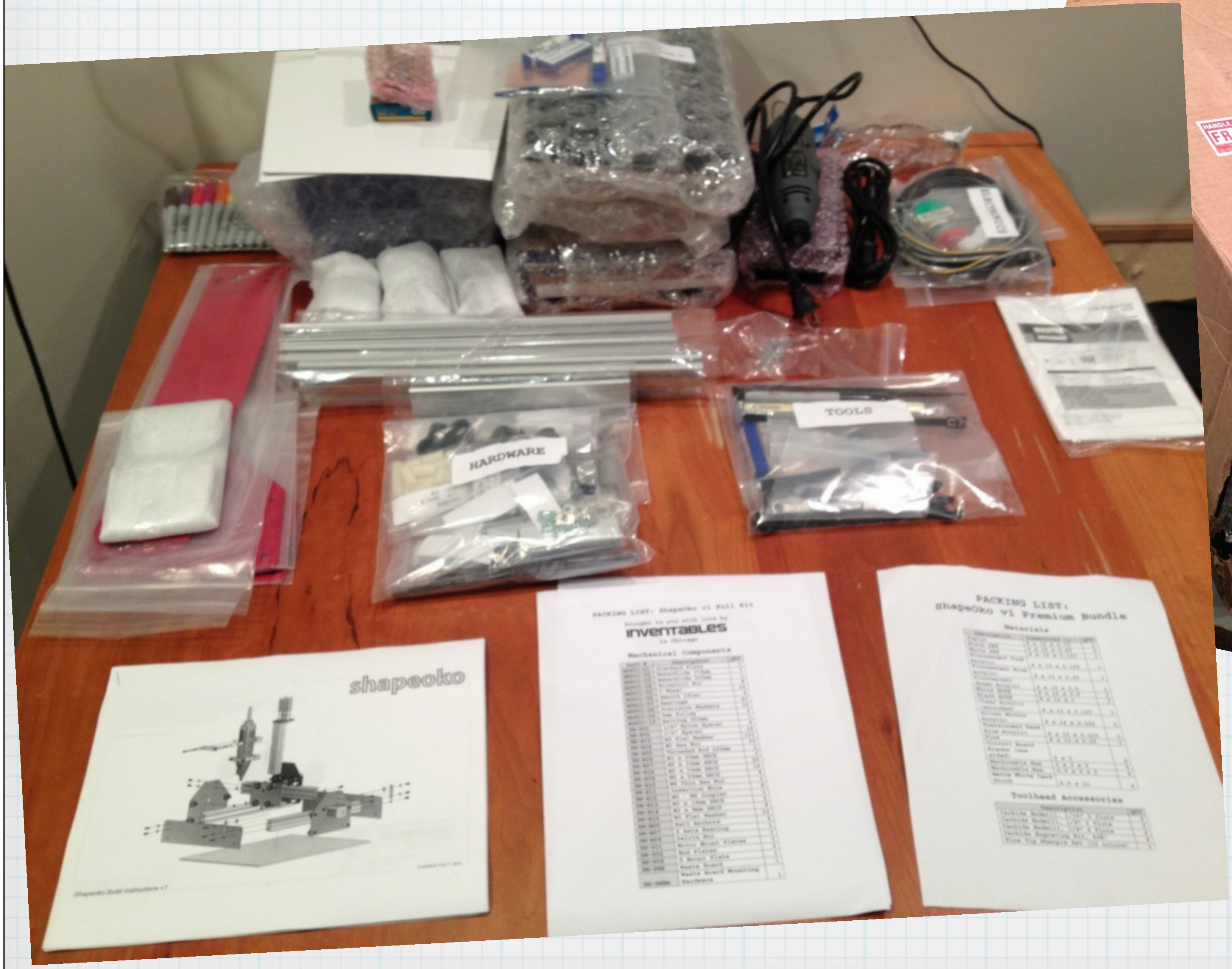
DIY != Industrial



CNC Mill

Related searches: cnc **milling machine** **haas** cnc mill cnc **router** cnc **lathe**

Kit Arrival


HARDWARE
TOOLS
shapeoko
PACKING LIST: ShapeOko v1 Full Kit
Inventables
Mechanical Components
PACKING LIST: ShapeOko v1 Premium Bundle
Materials
Toolhead Accessories

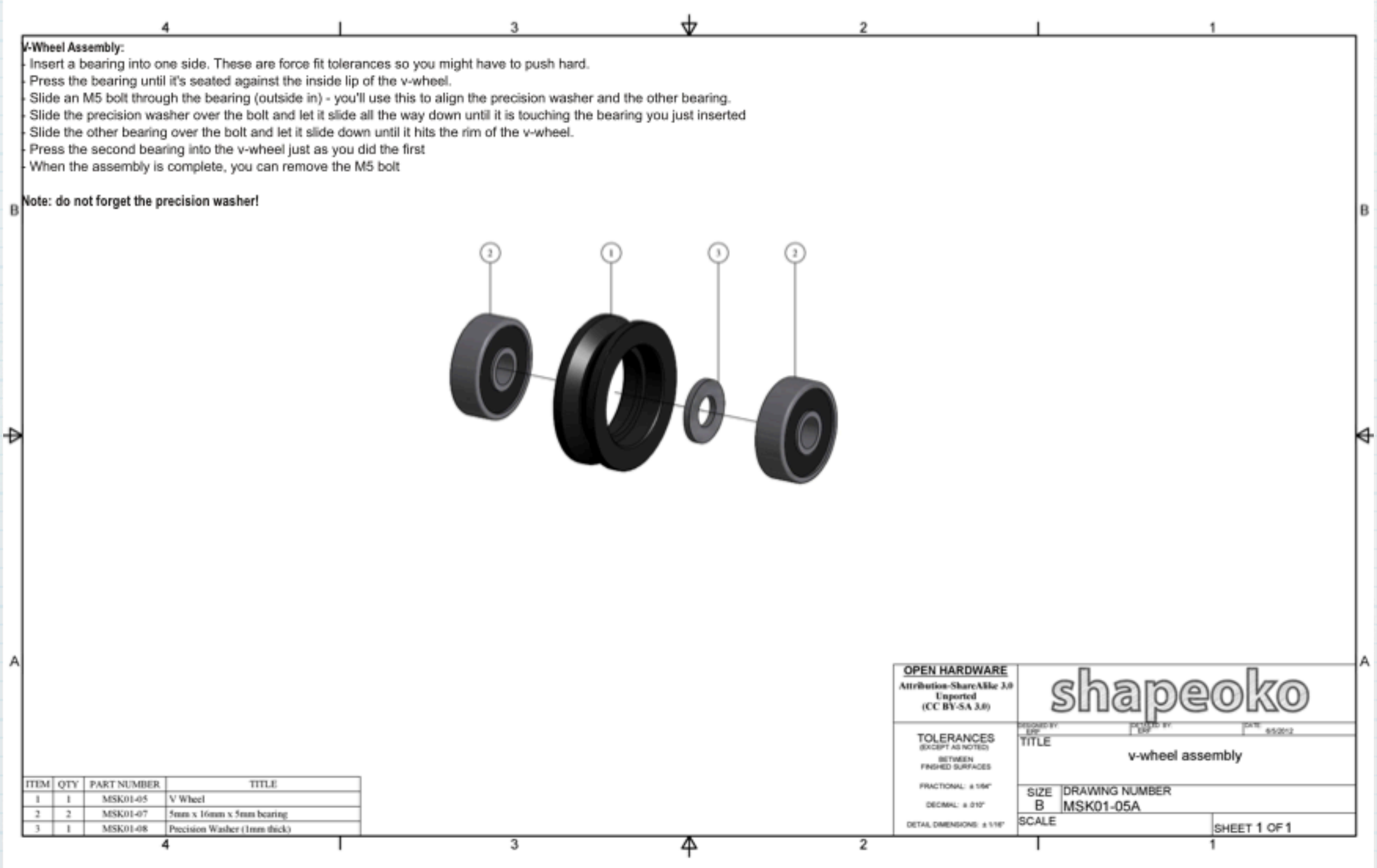
HANDLE WITH CARE
FRAGILE
THANK YOU
M60

Just to be clear...

... I have never done anything with CNC ever.

(I'm a software geek, what could possibly go wrong?)

Assembly



V-Wheel Assembly:

- Insert a bearing into one side. These are force fit tolerances so you might have to push hard.
- Press the bearing until it's seated against the inside lip of the v-wheel.
- Slide an M5 bolt through the bearing (outside in) - you'll use this to align the precision washer and the other bearing.
- Slide the precision washer over the bolt and let it slide all the way down until it is touching the bearing you just inserted
- Slide the other bearing over the bolt and let it slide down until it hits the rim of the v-wheel.
- Press the second bearing into the v-wheel just as you did the first
- When the assembly is complete, you can remove the M5 bolt

Note: do not forget the precision washer!

ITEM	QTY	PART NUMBER	TITLE
1	1	MSK01-05	V Wheel
2	2	MSK01-07	5mm x 16mm x 5mm bearing
3	1	MSK01-08	Precision Washer (1mm thick)

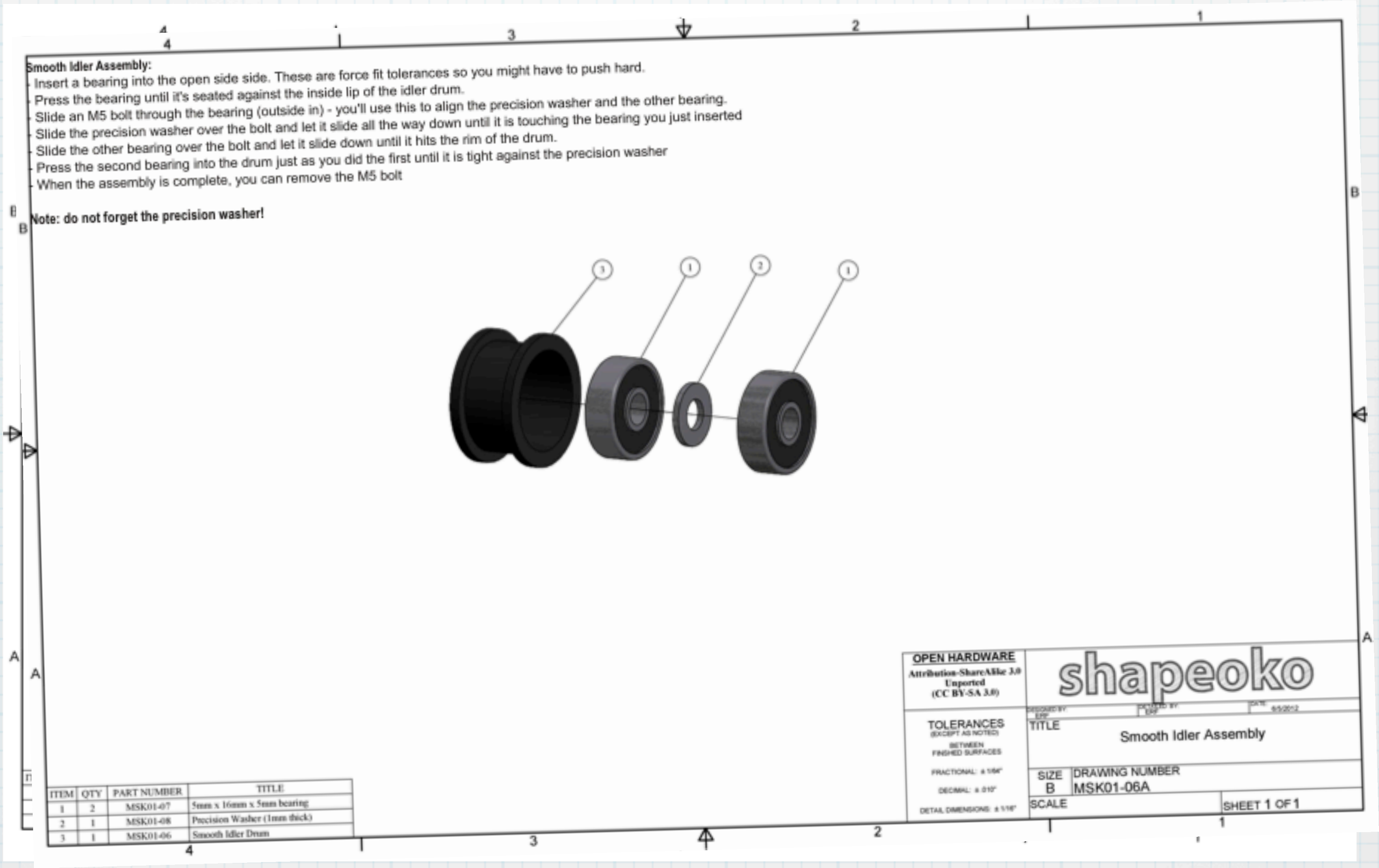
OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

shapeoko

DESIGNED BY ERF	DETAILED BY ERF	DATE 6/5/2012
TITLE v-wheel assembly		
SIZE B	DRAWING NUMBER MSK01-05A	
SCALE		SHEET 1 OF 1

Assembly



Smooth Idler Assembly:

- Insert a bearing into the open side side. These are force fit tolerances so you might have to push hard.
- Press the bearing until it's seated against the inside lip of the idler drum.
- Slide an M5 bolt through the bearing (outside in) - you'll use this to align the precision washer and the other bearing.
- Slide the precision washer over the bolt and let it slide all the way down until it is touching the bearing you just inserted
- Slide the other bearing over the bolt and let it slide down until it hits the rim of the drum.
- Press the second bearing into the drum just as you did the first until it is tight against the precision washer
- When the assembly is complete, you can remove the M5 bolt

Note: do not forget the precision washer!

ITEM	QTY	PART NUMBER	TITLE
1	2	MSK01-07	5mm x 16mm x 5mm bearing
2	1	MSK01-08	Precision Washer (1mm thick)
3	1	MSK01-06	Smooth Idler Drum

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: ERF
DETAILED BY: ERF
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
Smooth Idler Assembly

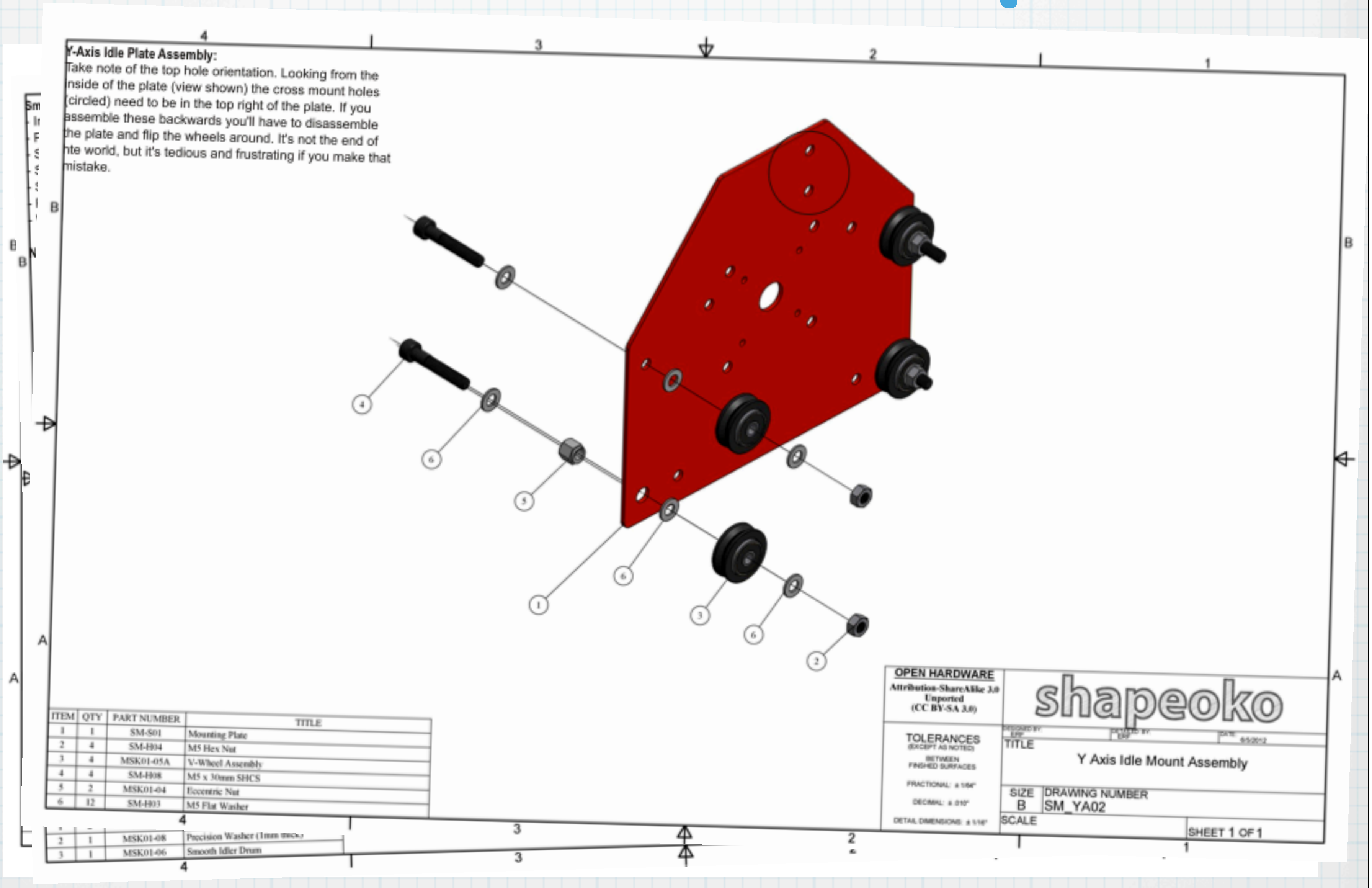
SIZE
B

DRAWING NUMBER
MSK01-06A

SCALE

SHEET 1 OF 1

Assembly



Y-Axis Idle Plate Assembly:
Take note of the top hole orientation. Looking from the inside of the plate (view shown) the cross mount holes (circled) need to be in the top right of the plate. If you assemble these backwards you'll have to disassemble the plate and flip the wheels around. It's not the end of the world, but it's tedious and frustrating if you make that mistake.

ITEM	QTY	PART NUMBER	TITLE
1	1	SM-S01	Mounting Plate
2	4	SM-H04	M5 Hex Nut
3	4	MSK01-05A	V-Wheel Assembly
4	4	SM-H08	M5 x 30mm SHCS
5	2	MSK01-04	Eccentric Nut
6	12	SM-H03	M5 Flat Washer

2	1	MSK01-08	Precision Washer (1mm thick)
3	1	MSK01-06	Smooth Idler Drum

OPEN HARDWARE
Attribution-ShareAlike 3.0 Unported
(CC BY-SA 3.0)

shapeoko

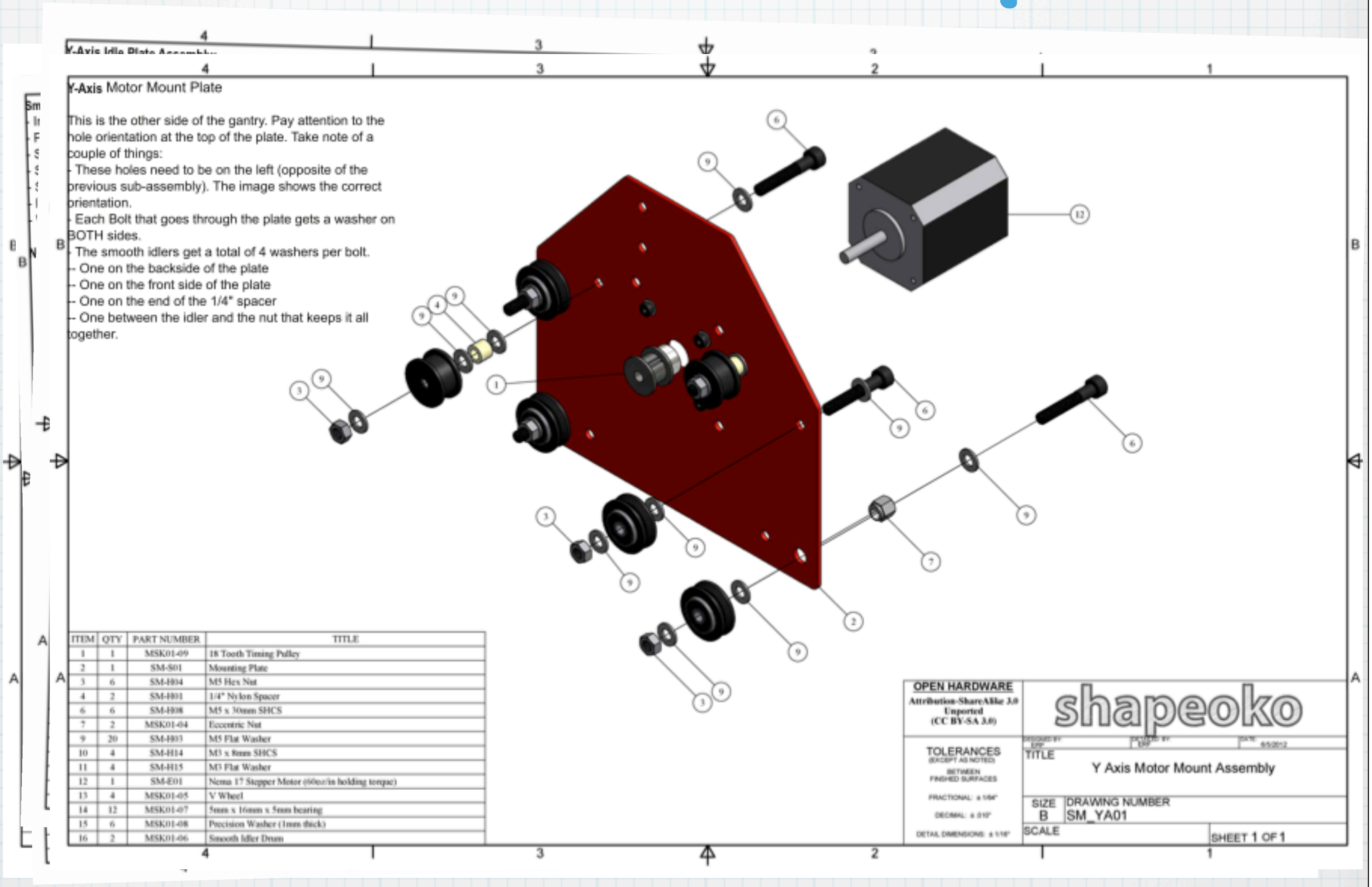
TOLERANCES
(EXCEPT AS NOTED)
BETWEEN FINSHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TITLE: Y Axis Idle Mount Assembly

SIZE: B
DRAWING NUMBER: SM_YA02
SCALE
SHEET 1 OF 1

Assembly



Y-Axis Idle Plate Assembly

Y-Axis Motor Mount Plate

This is the other side of the gantry. Pay attention to the hole orientation at the top of the plate. Take note of a couple of things:

- These holes need to be on the left (opposite of the previous sub-assembly). The image shows the correct orientation.
- Each Bolt that goes through the plate gets a washer on BOTH sides.
- The smooth idlers get a total of 4 washers per bolt.
  - One on the backside of the plate
  - One on the front side of the plate
  - One on the end of the 1/4" spacer
  - One between the idler and the nut that keeps it all together.

ITEM	QTY	PART NUMBER	TITLE
1	1	MSK01-09	18 Tooth Timing Pulley
2	1	SM-S01	Mounting Plate
3	6	SM-H04	M5 Hex Nut
4	2	SM-H01	1/4" Nylon Spacer
6	6	SM-H08	M5 x 30mm SHCS
7	2	MSK01-04	Eccentric Nut
9	20	SM-H03	M5 Flat Washer
10	4	SM-H14	M3 x 8mm SHCS
11	4	SM-H15	M3 Flat Washer
12	1	SM-E01	Nema 17 Stepper Motor (60oz/in holding torque)
13	4	MSK01-05	V Wheel
14	12	MSK01-07	5mm x 16mm x 5mm bearing
15	6	MSK01-08	Precision Washer (1mm thick)
16	2	MSK01-06	Smooth Idler Drum

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: ERP
CHECKED BY: ERP
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINSHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
Y Axis Motor Mount Assembly

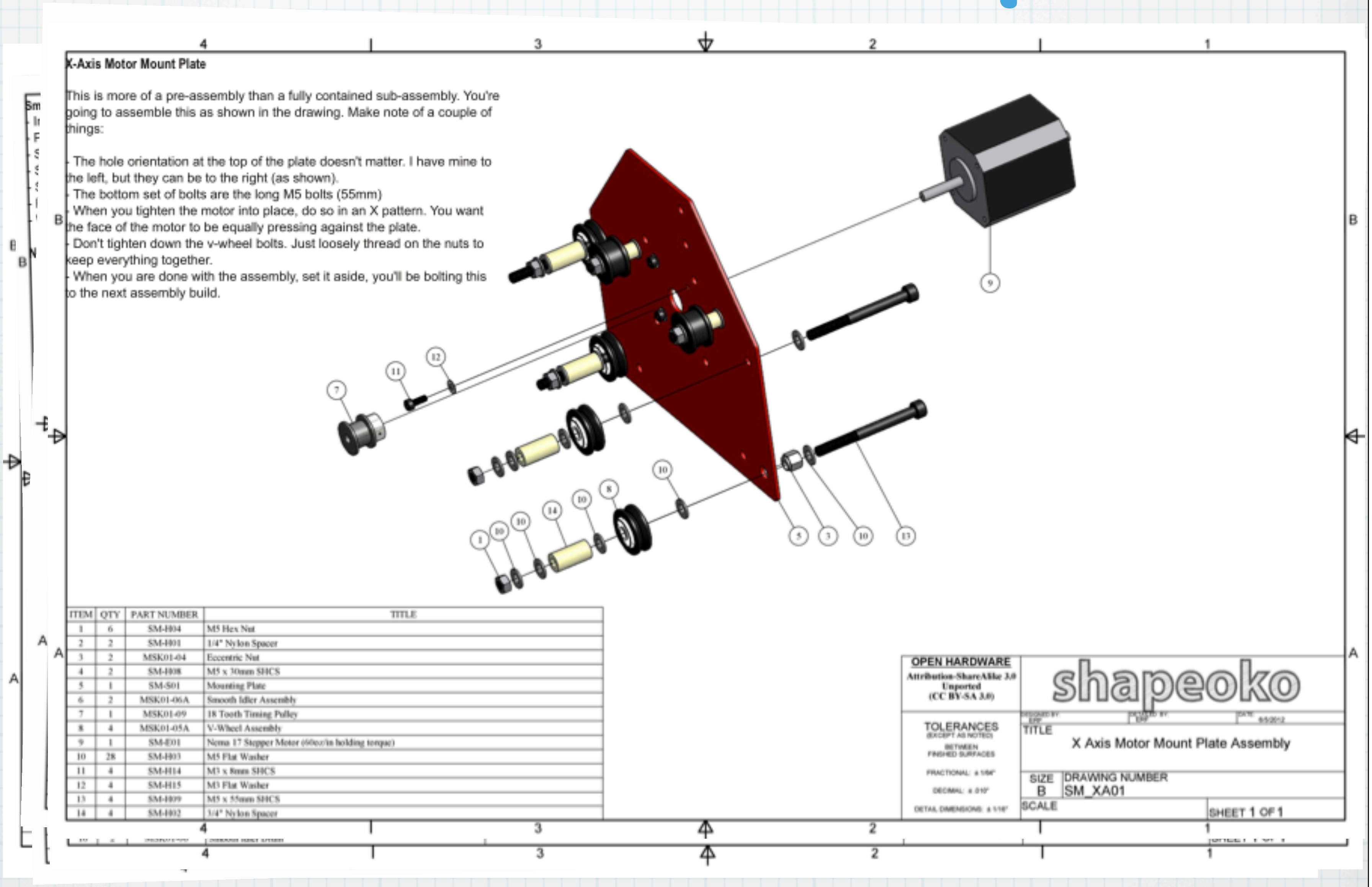
SIZE
B

DRAWING NUMBER
SM_YA01

SCALE

SHEET 1 OF 1

Assembly



X-Axis Motor Mount Plate

This is more of a pre-assembly than a fully contained sub-assembly. You're going to assemble this as shown in the drawing. Make note of a couple of things:

- The hole orientation at the top of the plate doesn't matter. I have mine to the left, but they can be to the right (as shown).
- The bottom set of bolts are the long M5 bolts (55mm)
- When you tighten the motor into place, do so in an X pattern. You want the face of the motor to be equally pressing against the plate.
- Don't tighten down the v-wheel bolts. Just loosely thread on the nuts to keep everything together.
- When you are done with the assembly, set it aside, you'll be bolting this to the next assembly build.

ITEM	QTY	PART NUMBER	TITLE
1	6	SM-H04	M5 Hex Nut
2	2	SM-H01	1/4" Nylon Spacer
3	2	MSK01-04	Eccentric Nut
4	2	SM-H08	M5 x 30mm SHCS
5	1	SM-S01	Mounting Plate
6	2	MSK01-06A	Smooth Idler Assembly
7	1	MSK01-09	18 Tooth Timing Pulley
8	4	MSK01-05A	V-Wheel Assembly
9	1	SM-E01	Nema 17 Stepper Motor (60oz/in holding torque)
10	28	SM-H03	M5 Flat Washer
11	4	SM-H14	M3 x 8mm SHCS
12	4	SM-H15	M3 Flat Washer
13	4	SM-H09	M5 x 55mm SHCS
14	4	SM-H02	3/4" Nylon Spacer

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: EPF
DETAILED BY: EPF
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
X Axis Motor Mount Plate Assembly

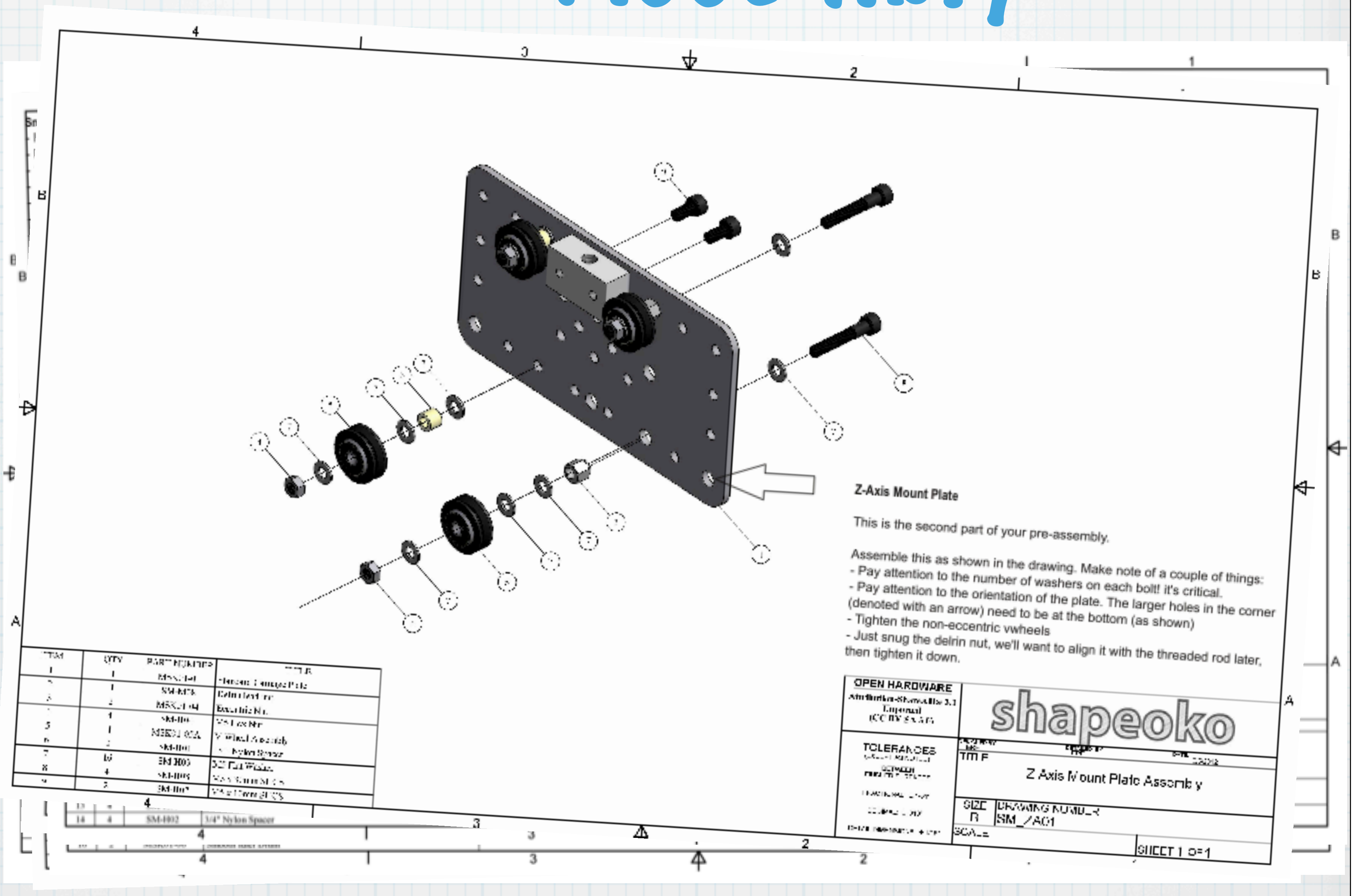
SIZE
B

DRAWING NUMBER
SM_XA01

SCALE

SHEET 1 OF 1

Assembly



Z-Axis Mount Plate

This is the second part of your pre-assembly.

Assemble this as shown in the drawing. Make note of a couple of things:
- Pay attention to the number of washers on each bolt! it's critical.
- Pay attention to the orientation of the plate. The larger holes in the corner (denoted with an arrow) need to be at the bottom (as shown)
- Tighten the non-eccentric vwheels
- Just snug the delrin nut, we'll want to align it with the threaded rod later, then tighten it down.

ITEM	QTY	PART NUMBER	TITLE
1	1	[illegible]	[illegible]
2	1	SM-M[illegible]	Delrin Lead[illegible]
3	2	MSS[illegible]	Eccentric Nut
4	4	SM-H0[illegible]	M5 Lock Nut
5	1	MBK01-02A	V Wheel Assembly
6	2	SM-H01	[illegible] Nylon Spacer
7	16	SM-H03	M5 Flat Washer
8	4	SM-H0[illegible]	M5 x [illegible] SHCS
9	2	SM-H0[illegible]	M5 x [illegible]mm SHCS

14	4	SM-H02	3/4" Nylon Spacer

OPEN HARDWARE
Attribution-ShareAlike 3.1
Unported
(CC BY SA 3.0)

shapeoko

TOLERANCES

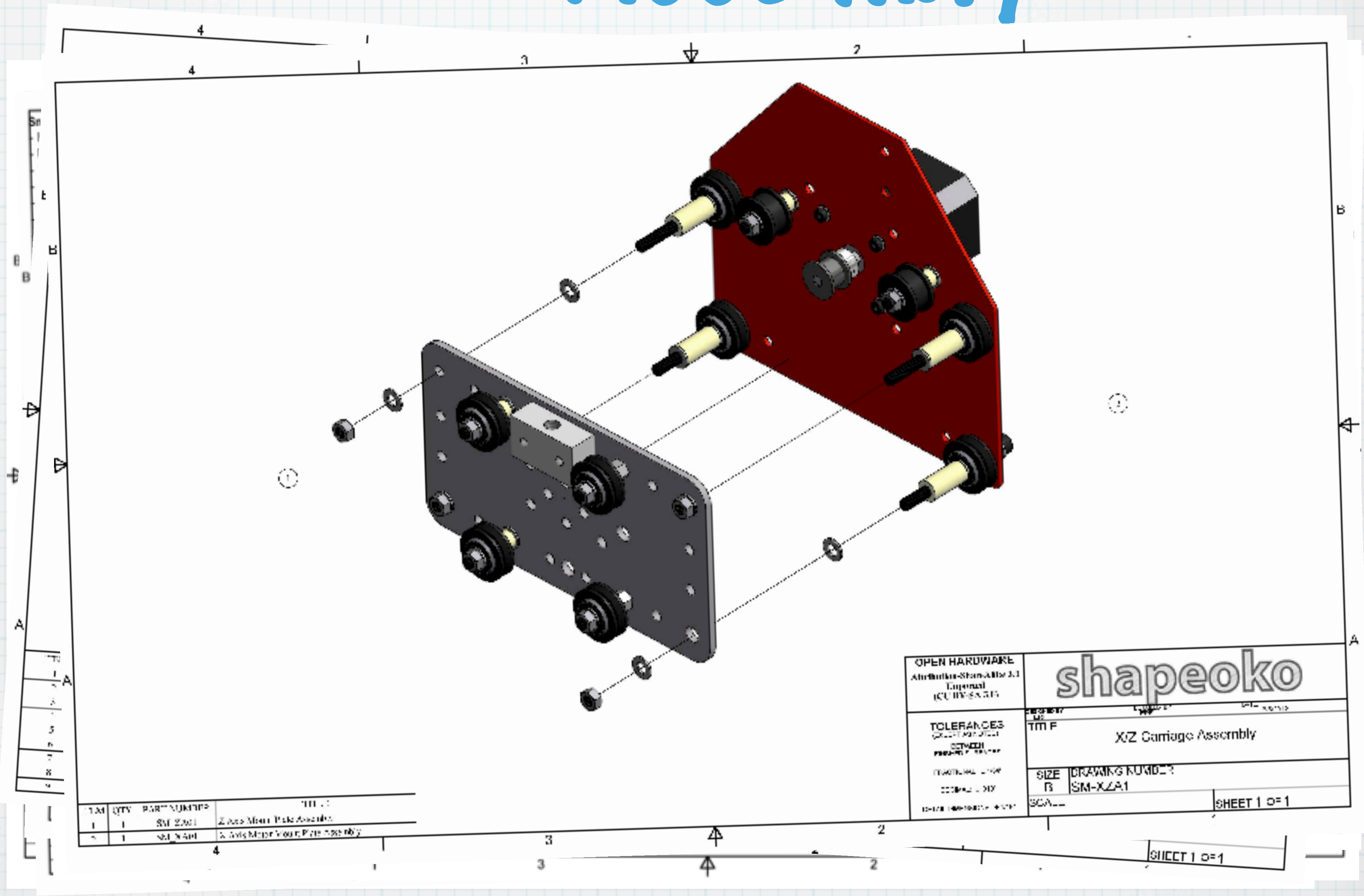
TITLE
Z Axis Mount Plate Assembly

SIZE B
DRAWING NUMBER SM_ZA01

SCALE

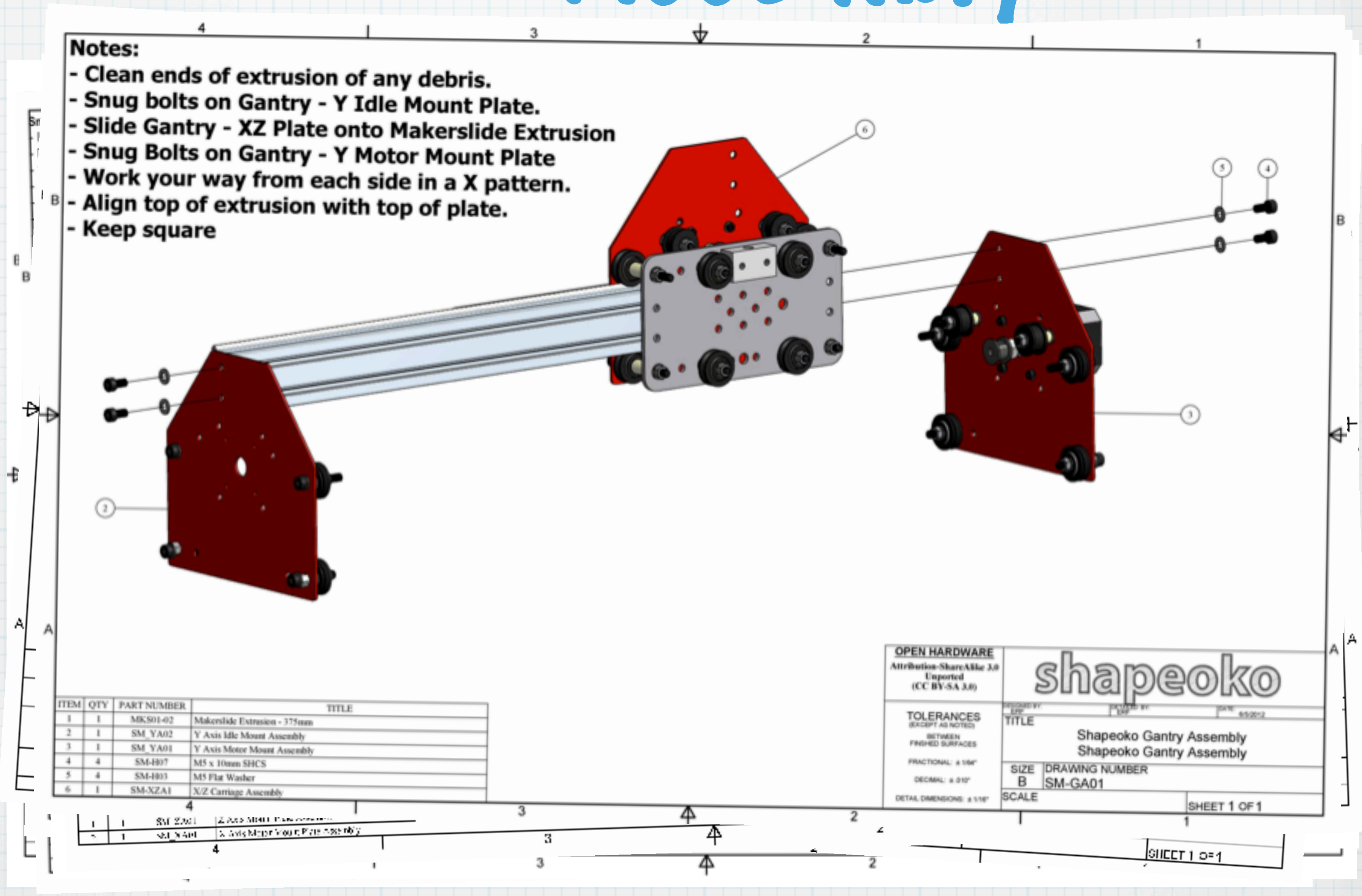
SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM_ZA01	Z Axis Mount Plate Assembly
2	1	SM_XA01	X Axis Motor Mount Plate Assembly

Assembly



Notes:
- Clean ends of extrusion of any debris.
- Snug bolts on Gantry - Y Idle Mount Plate.
- Slide Gantry - XZ Plate onto Makerslide Extrusion
- Snug Bolts on Gantry - Y Motor Mount Plate
- Work your way from each side in a X pattern.
- Align top of extrusion with top of plate.
- Keep square

ITEM	QTY	PART NUMBER	TITLE
1	1	MKS01-02	Makerslide Extrusion - 375mm
2	1	SM_YA02	Y Axis Idle Mount Assembly
3	1	SM_YA01	Y Axis Motor Mount Assembly
4	4	SM-H07	M5 x 10mm SHCS
5	4	SM-H03	M5 Flat Washer
6	1	SM-XZA1	X/Z Carriage Assembly

1	1	SM_ZA01	Z Axis Motor Plate Assembly
[illegible]	1	SM_XA01	X Axis Motor Mount Plate Assembly

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINSHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TITLE
Shapeoko Gantry Assembly
Shapeoko Gantry Assembly

SIZE
B

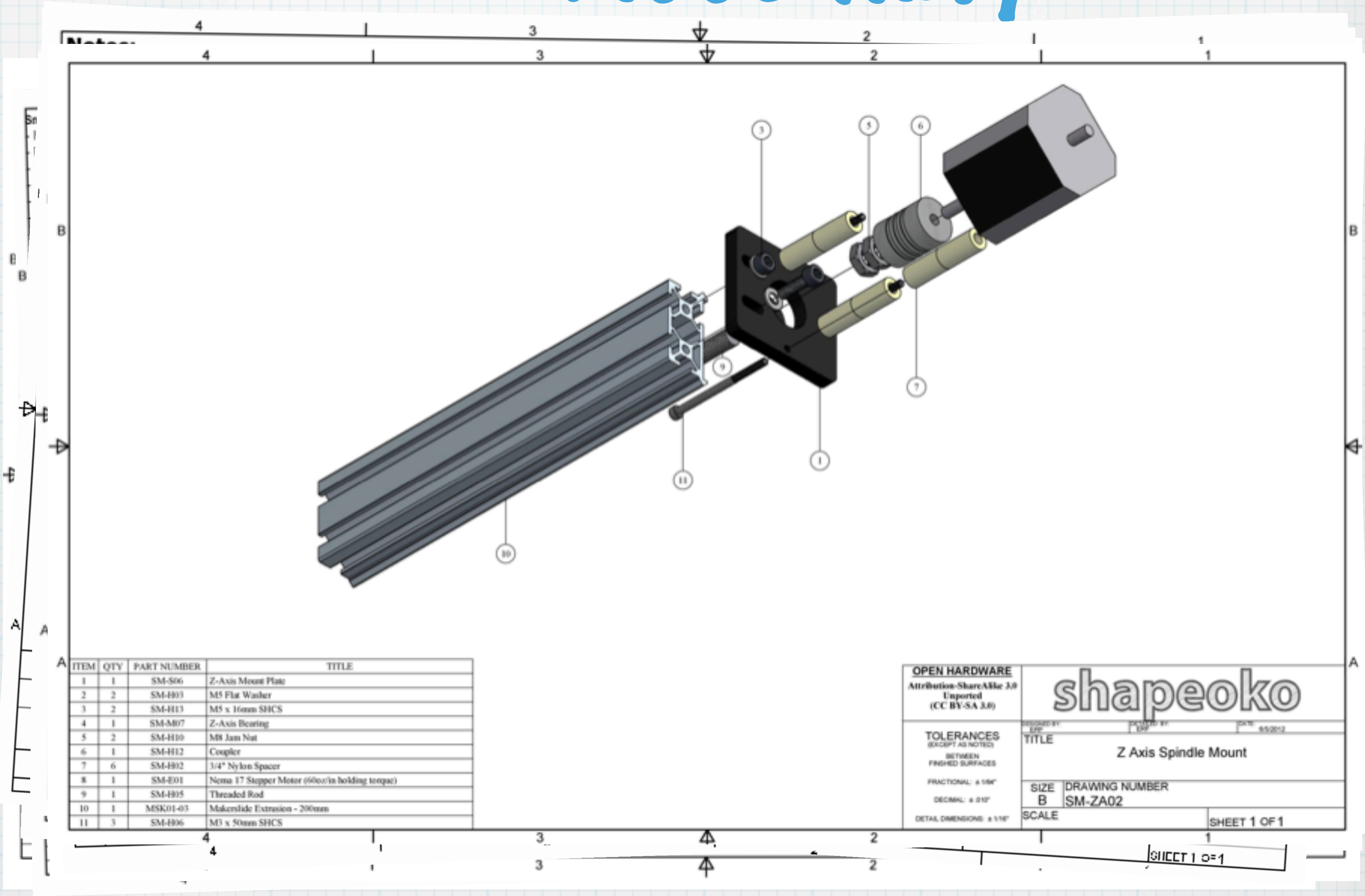
DRAWING NUMBER
SM-GA01

SCALE

SHEET 1 OF 1

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM-S06	Z-Axis Mount Plate
2	2	SM-H03	M5 Flat Washer
3	2	SM-H13	M5 x 16mm SHCS
4	1	SM-M07	Z-Axis Bearing
5	2	SM-H10	M8 Jam Nut
6	1	SM-H12	Coupler
7	6	SM-H02	3/4" Nylon Spacer
8	1	SM-E01	Nema 17 Stepper Motor (60oz/in holding torque)
9	1	SM-H05	Threaded Rod
10	1	MSK01-03	Makerslide Extrusion - 200mm
11	3	SM-H06	M3 x 50mm SHCS

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TITLE
Z Axis Spindle Mount

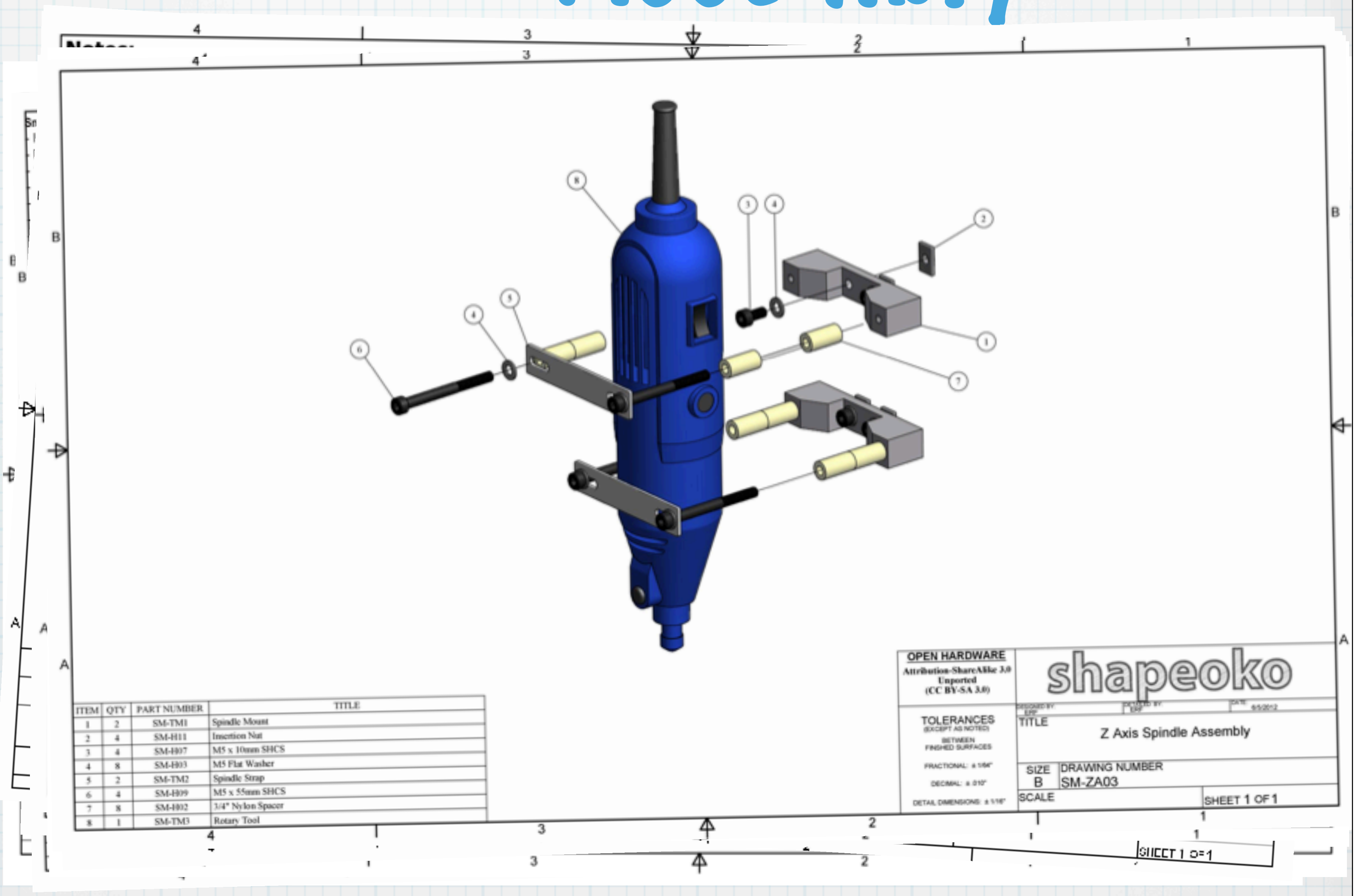
SIZE
B

DRAWING NUMBER
SM-ZA02

SCALE

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	2	SM-TM1	Spindle Mount
2	4	SM-H11	Insertion Nut
3	4	SM-H07	M5 x 10mm SHCS
4	8	SM-H03	M5 Flat Washer
5	2	SM-TM2	Spindle Strap
6	4	SM-H09	M5 x 55mm SHCS
7	8	SM-H02	3/4" Nylon Spacer
8	1	SM-TM3	Rotary Tool

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
Z Axis Spindle Assembly

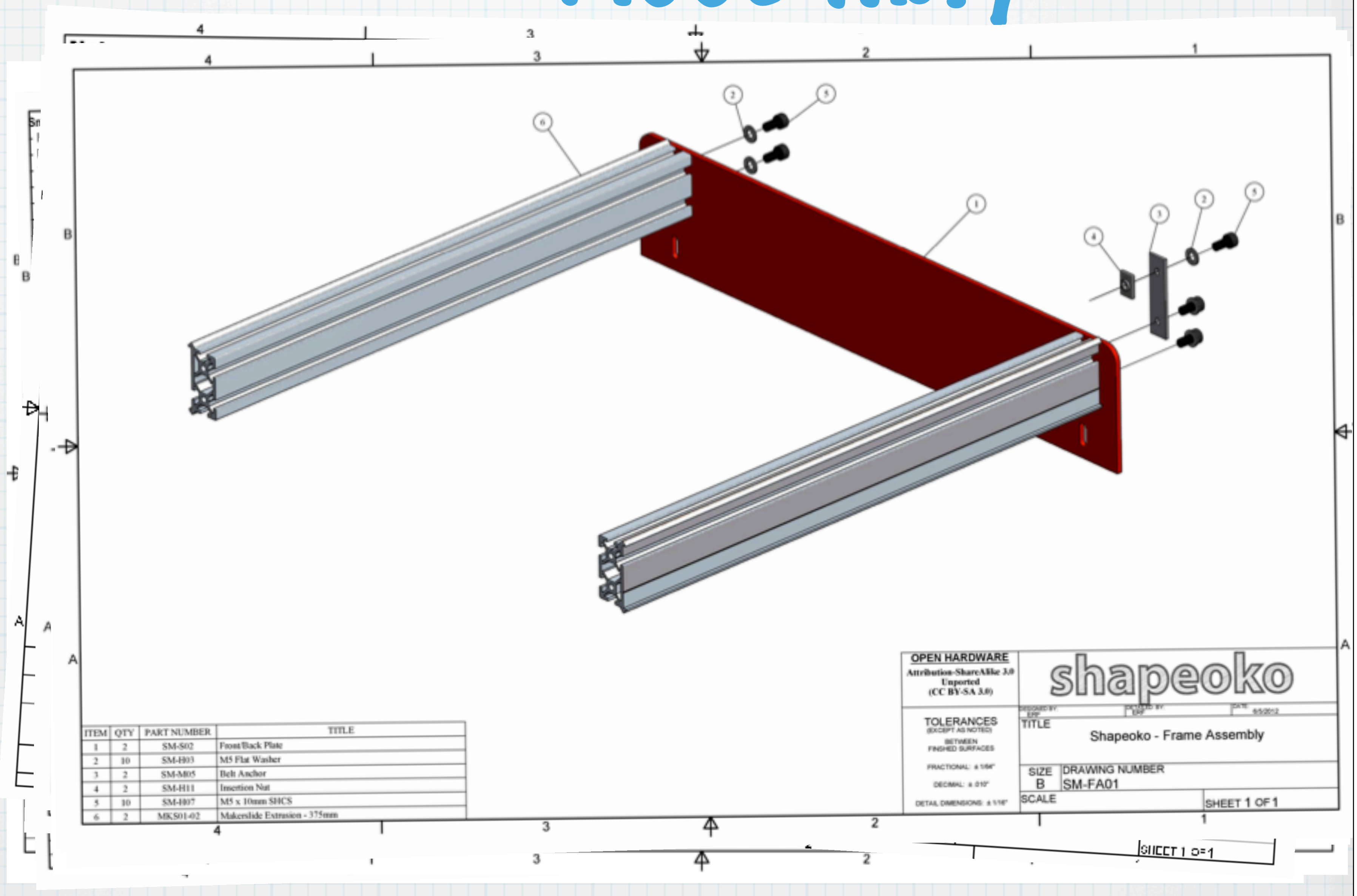
SIZE
B

DRAWING NUMBER
SM-ZA03

SCALE

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	2	SM-S02	Front/Back Plate
2	10	SM-H03	M5 Flat Washer
3	2	SM-M05	Belt Anchor
4	2	SM-H11	Insertion Nut
5	10	SM-H07	M5 x 10mm SHCS
6	2	MKS01-02	Makerslide Extrusion - 375mm

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
Shapeoko - Frame Assembly

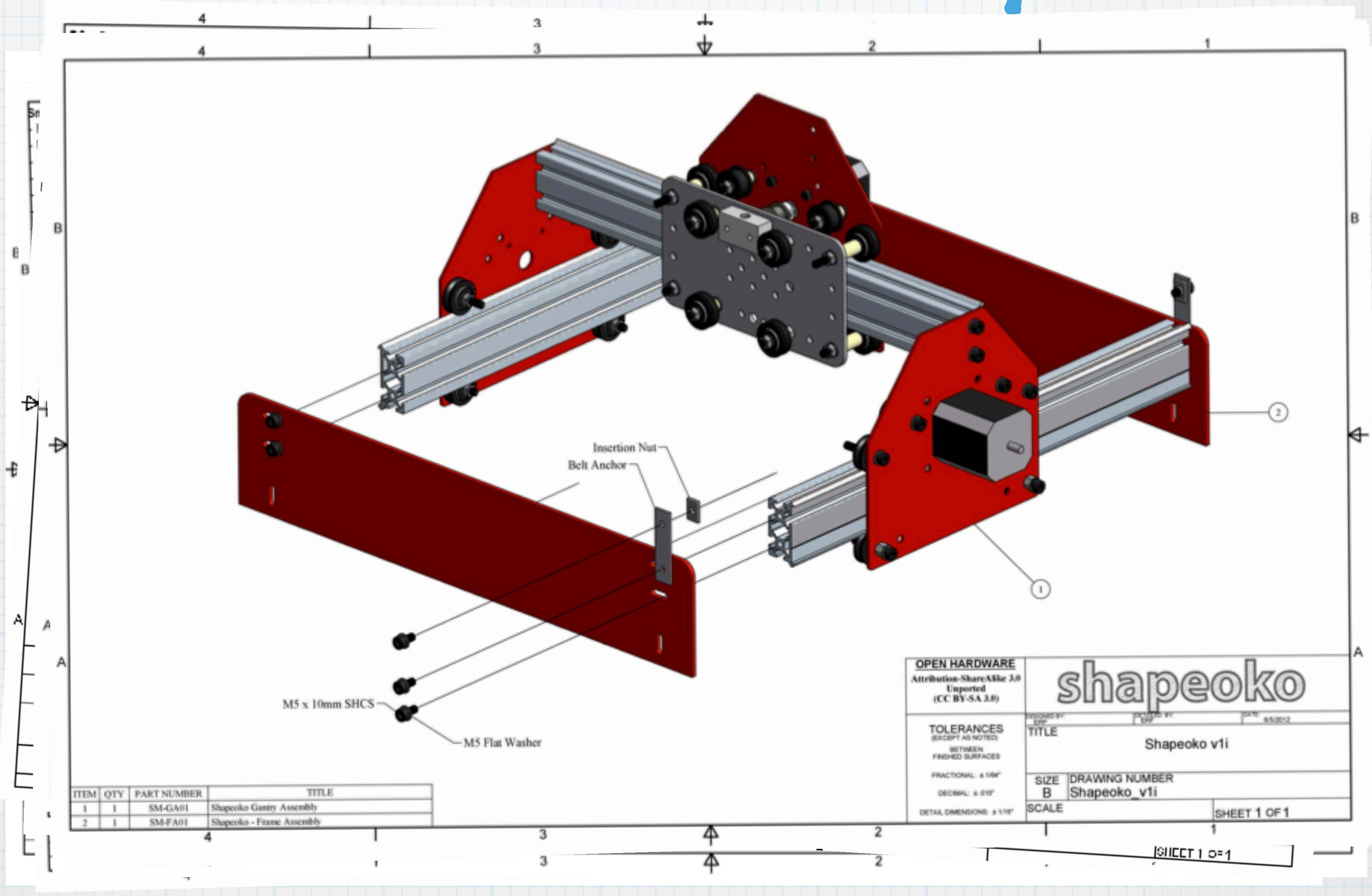
SIZE
B

DRAWING NUMBER
SM-FA01

SCALE

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM-GA01	Shapeoko Gantry Assembly
2	1	SM-FA01	Shapeoko - Frame Assembly

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

DESIGNED BY: EPP
DETAILED BY: EPP
DATE: 6/5/2012

TITLE
Shapeoko v1i

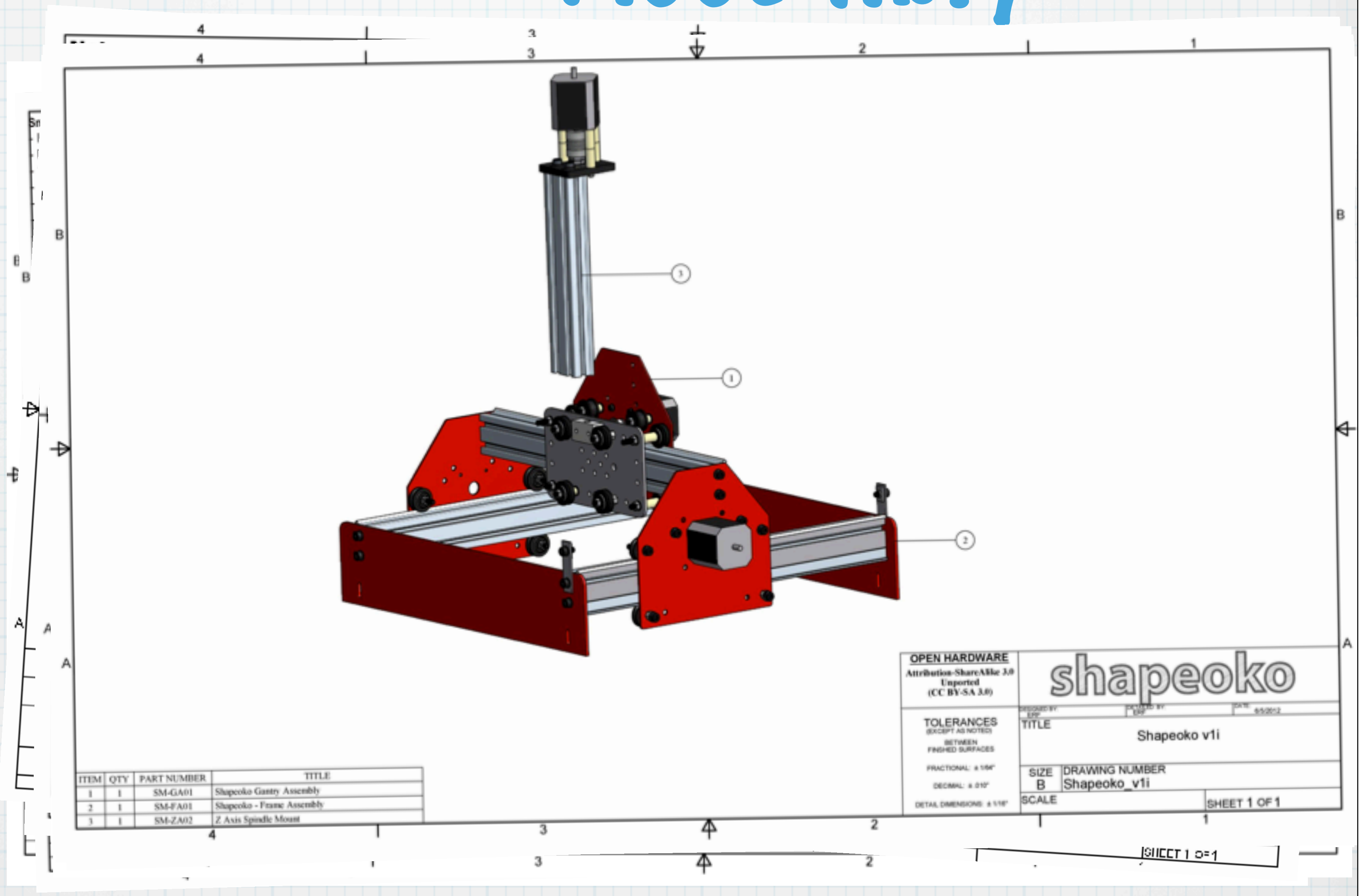
SIZE
B

DRAWING NUMBER
Shapeoko_v1i

SCALE

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM-GA01	Shapeoko Gantry Assembly
2	1	SM-FA01	Shapeoko - Frame Assembly
3	1	SM-ZA02	Z Axis Spindle Mount

OPEN HARDWARE
Attribution-ShareAlike 3.0
Unported
(CC BY-SA 3.0)

shapeoko

DESIGNED BY: ERF
DETAILED BY: ERF
DATE: 6/5/2012

TOLERANCES
(EXCEPT AS NOTED)
BETWEEN
FINISHED SURFACES
FRACTIONAL: ± 1/64"
DECIMAL: ± .010"
DETAIL DIMENSIONS: ± 1/16"

TITLE
Shapeoko v1i

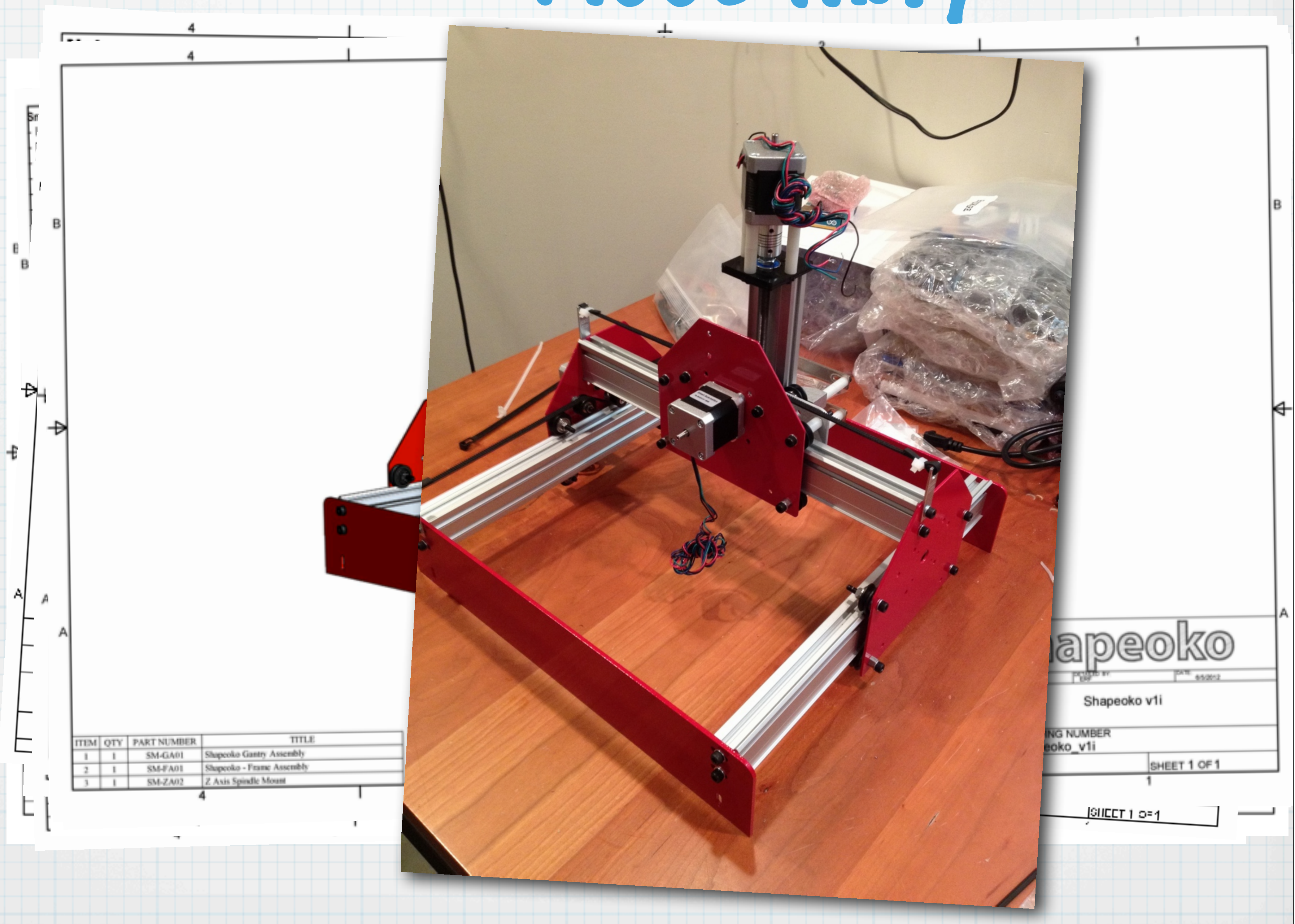
SIZE
B

DRAWING NUMBER
Shapeoko_v1i

SCALE

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM-GA01	Shapeoko Gantry Assembly
2	1	SM-FA01	Shapeoko - Frame Assembly
3	1	SM-ZA02	Z Axis Spindle Mount

apeoko

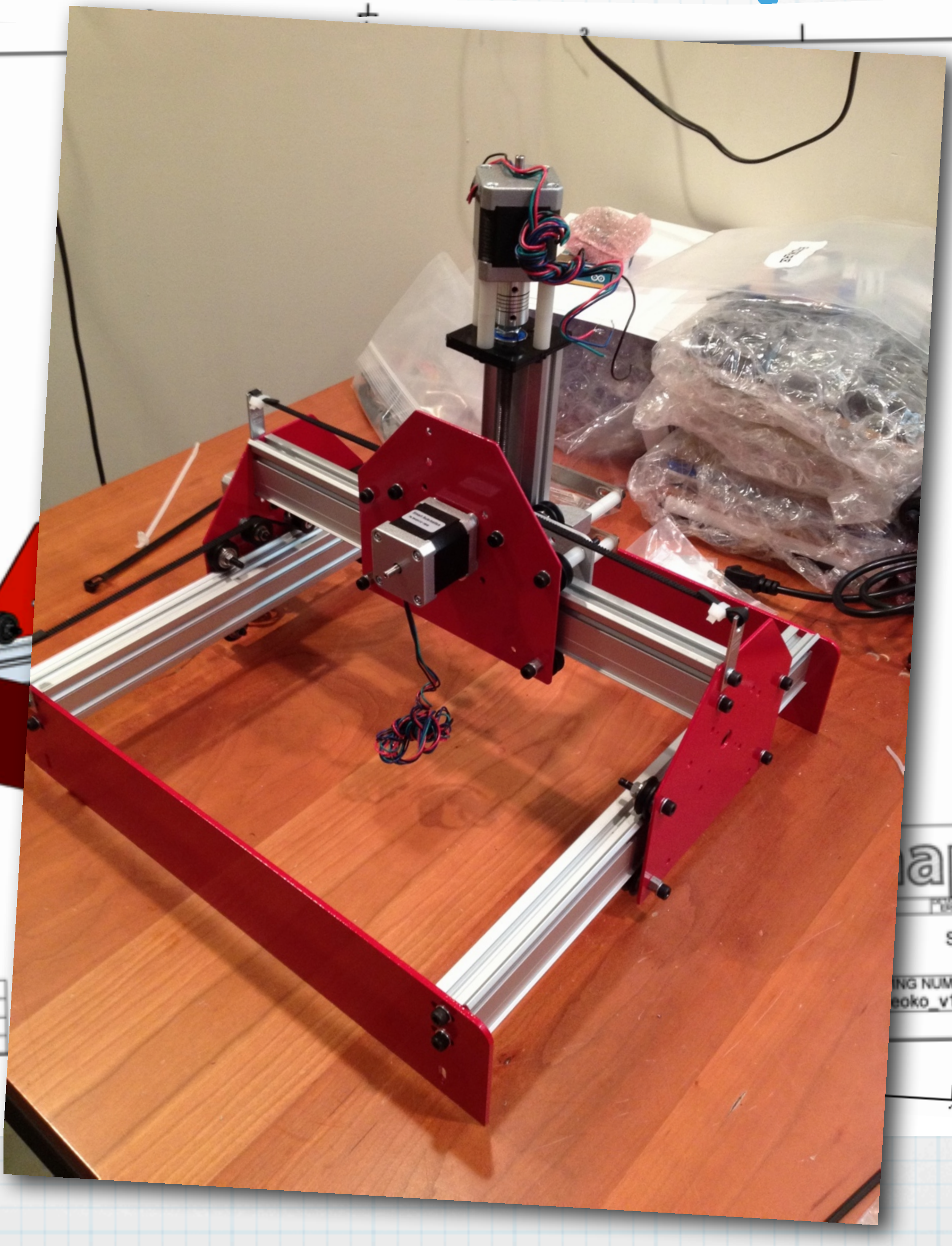
DATE 6/5/2012

Shapeoko v1i

ING NUMBER
eoko_v1i

SHEET 1 OF 1

Assembly



ITEM	QTY	PART NUMBER	TITLE
1	1	SM-GA01	Shapeoko Gantry Assembly
2	1	SM-FA01	Shapeoko - Frame Assembly
3	1	SM-ZA02	Z Axis Spindle Mount

apeoko

DATE 6/5/2012

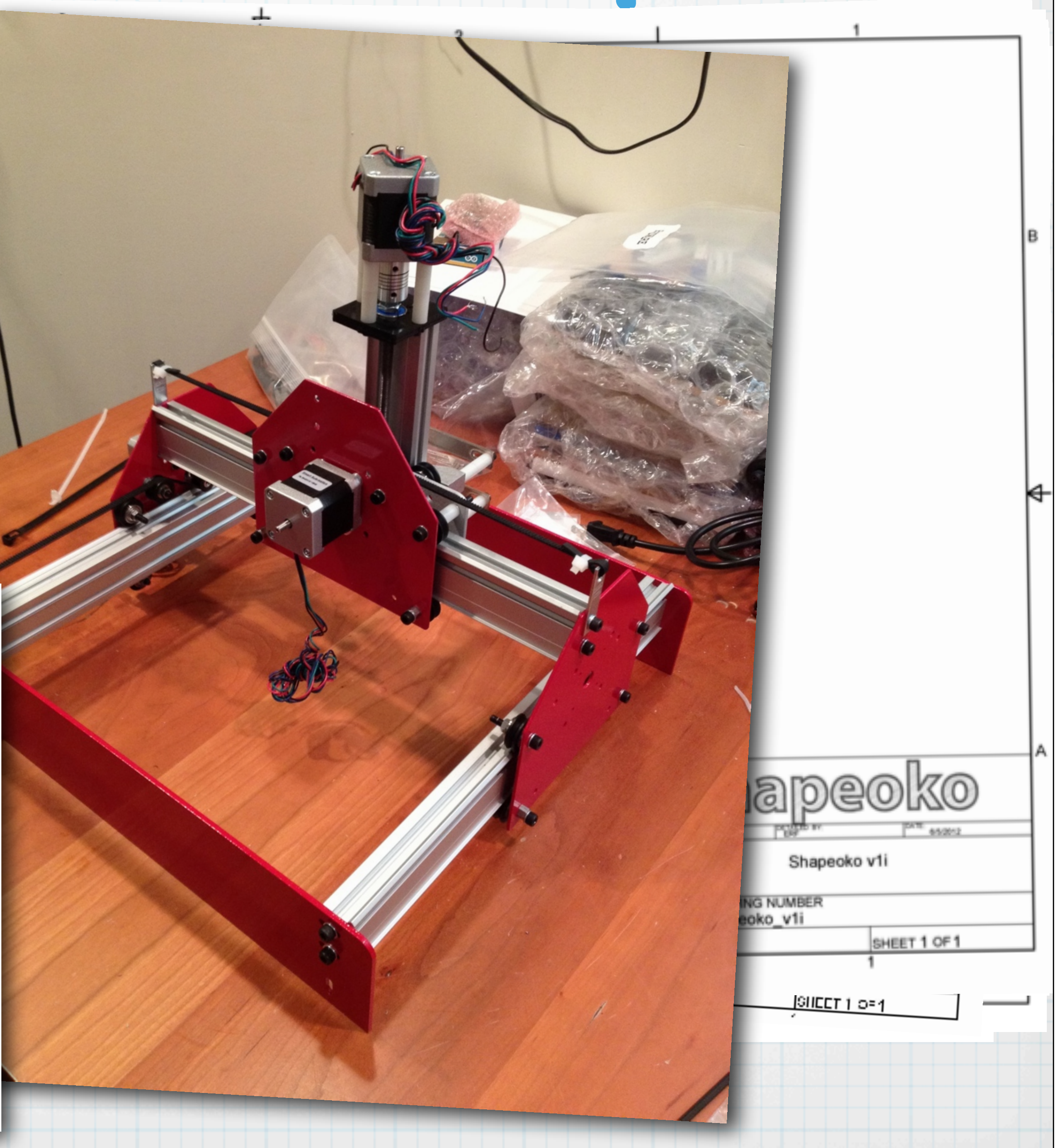
Shapeoko v1i

NG NUMBER
eoko_v1i

SHEET 1 OF 1

SHEET 1 OF 1

Assembly

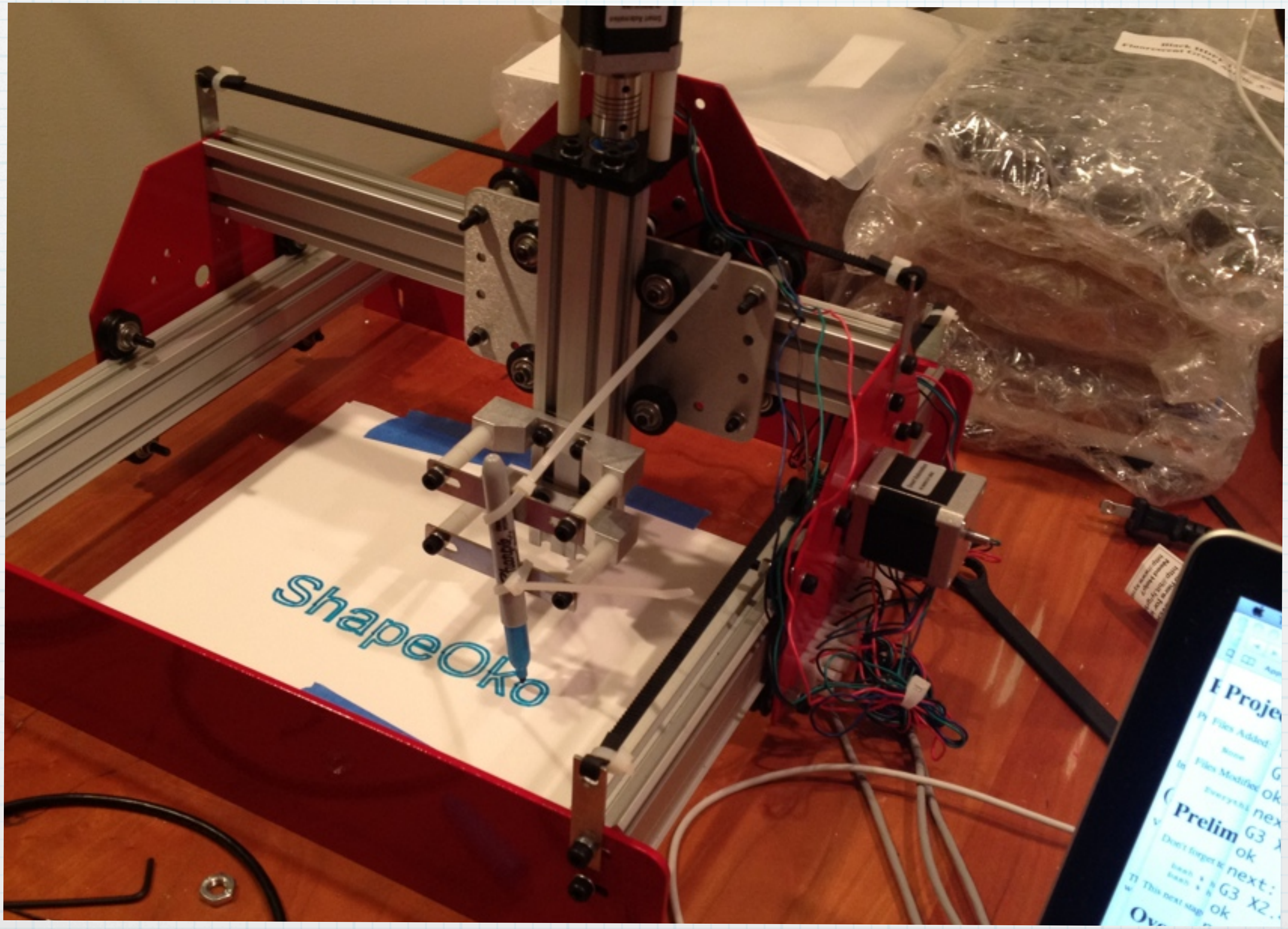


Curse Words Uttered:

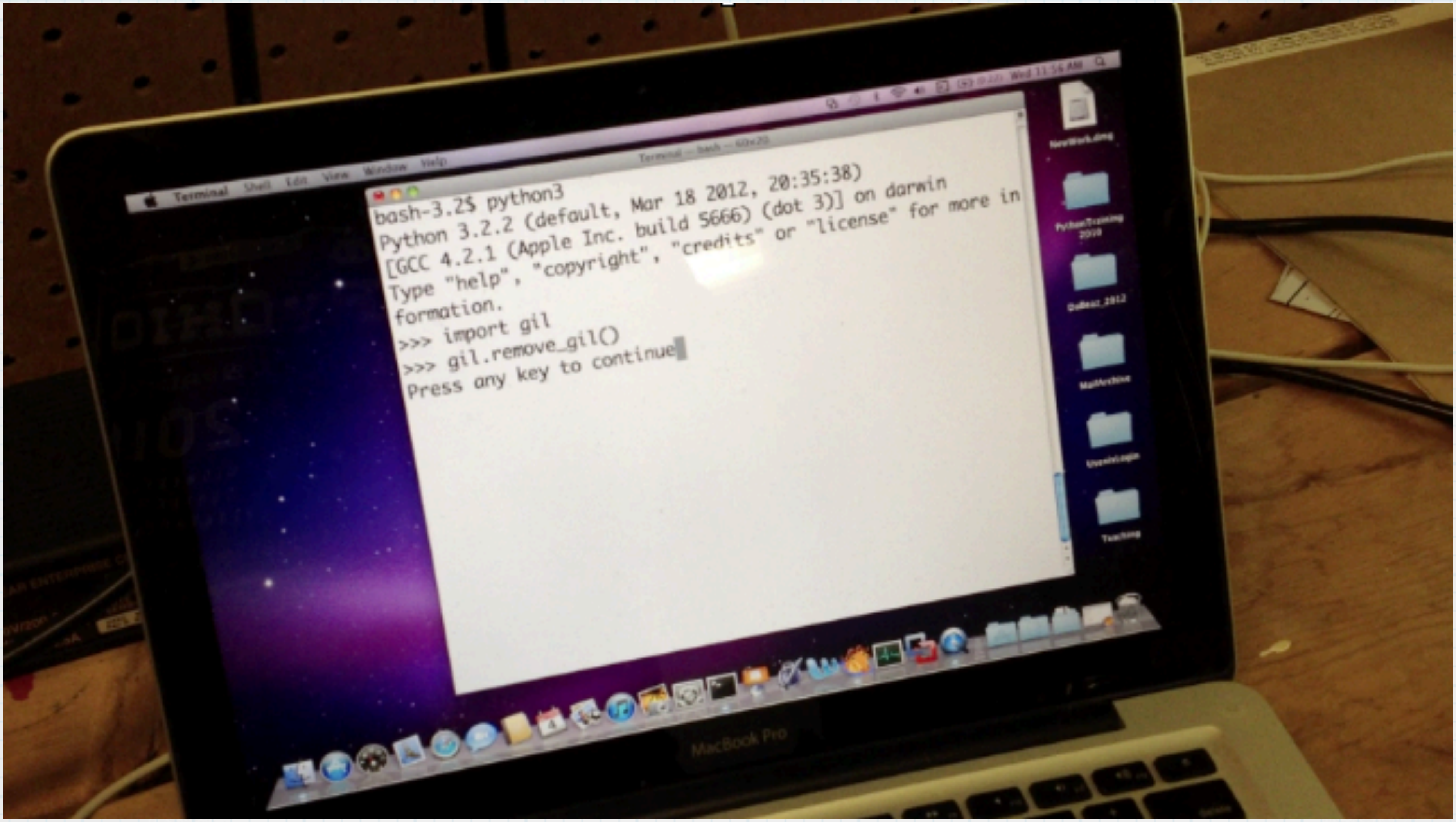
CNC Mill - 0
Stroller - 137

(Stroller cost more)

Hello World



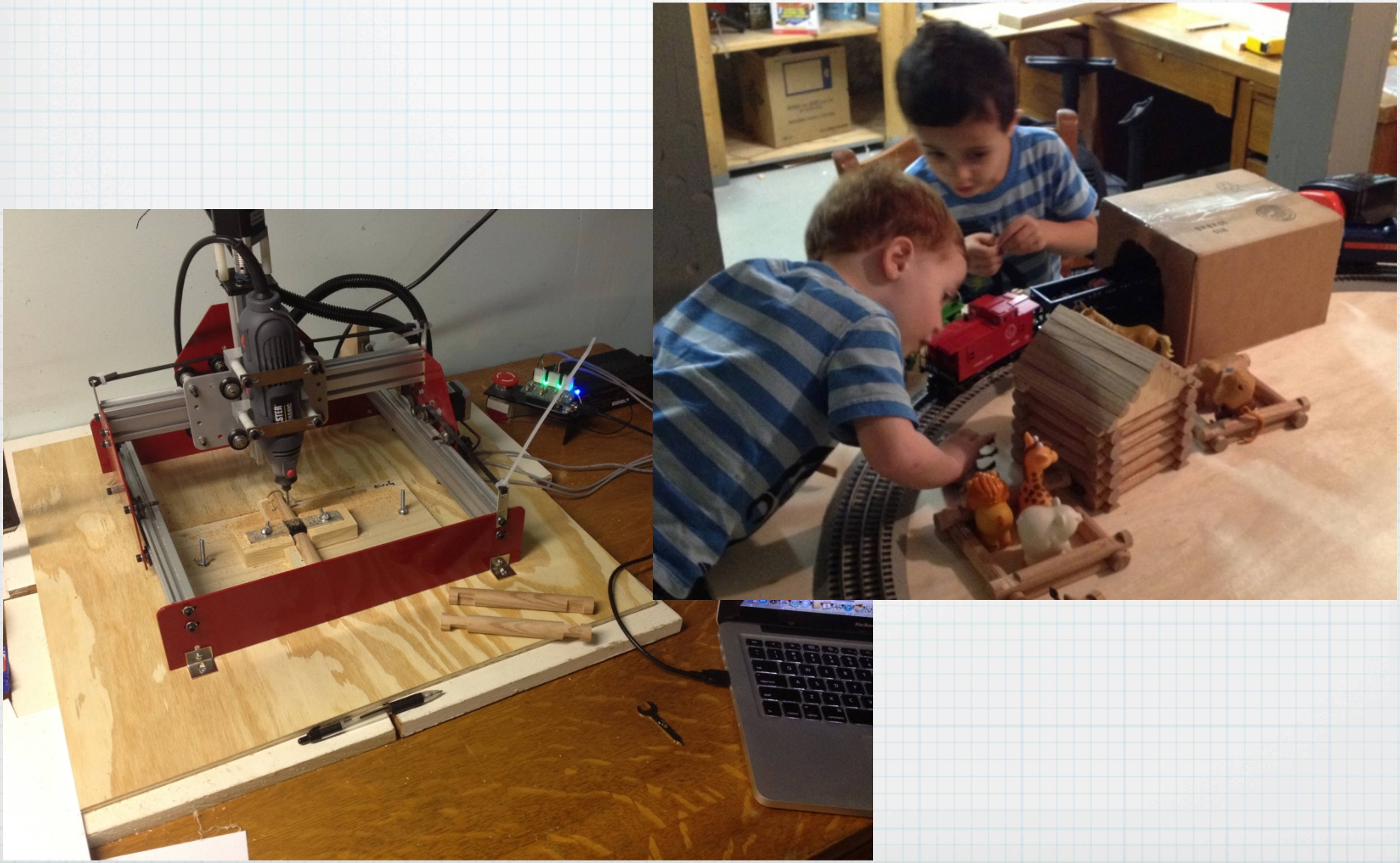
Practical First Job



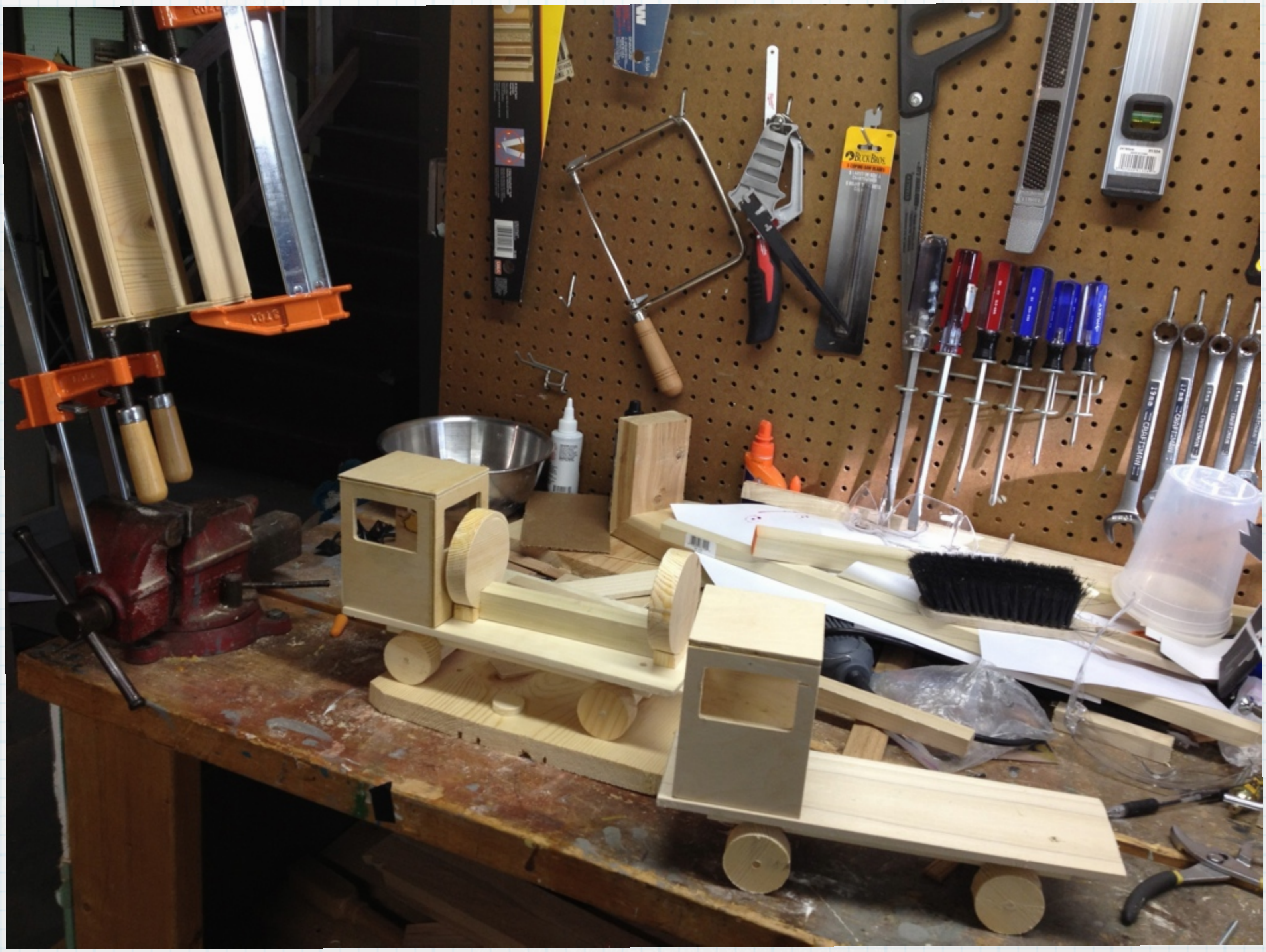
```
bash-3.2$ python3
Python 3.2.2 (default, Mar 18 2012, 20:35:38)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more in
formation.
>>> import gil
>>> gil.remove_gil()
Press any key to continue
```

GIL

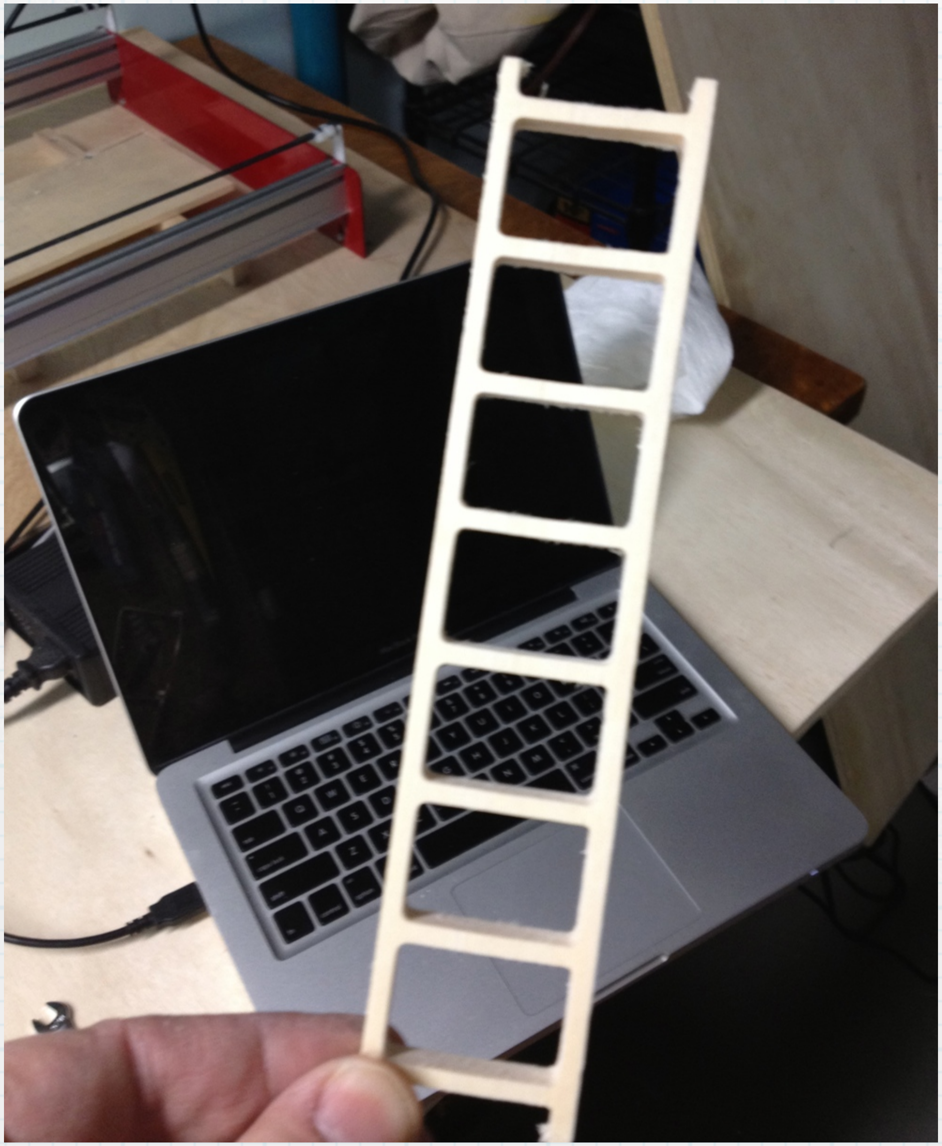
Lincoln Logs



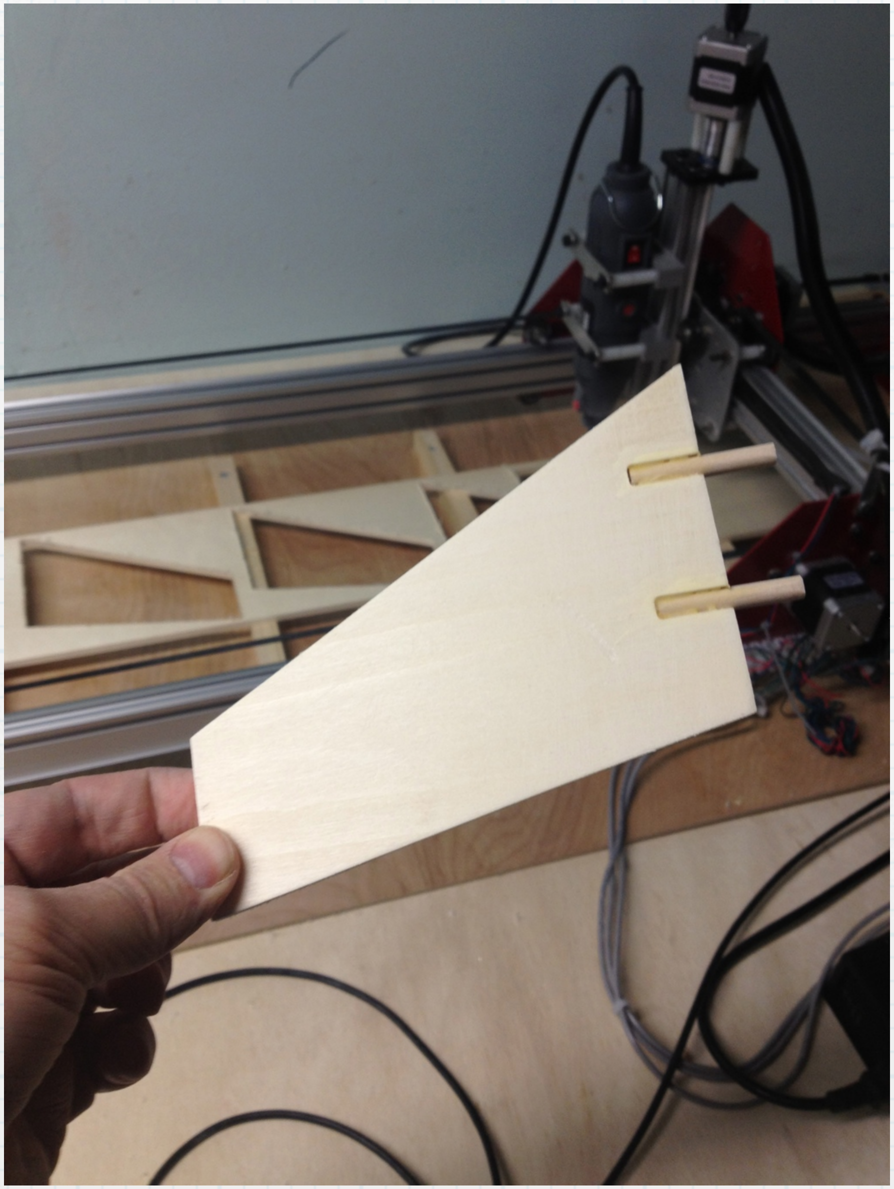
Wood Train & Fire Truck



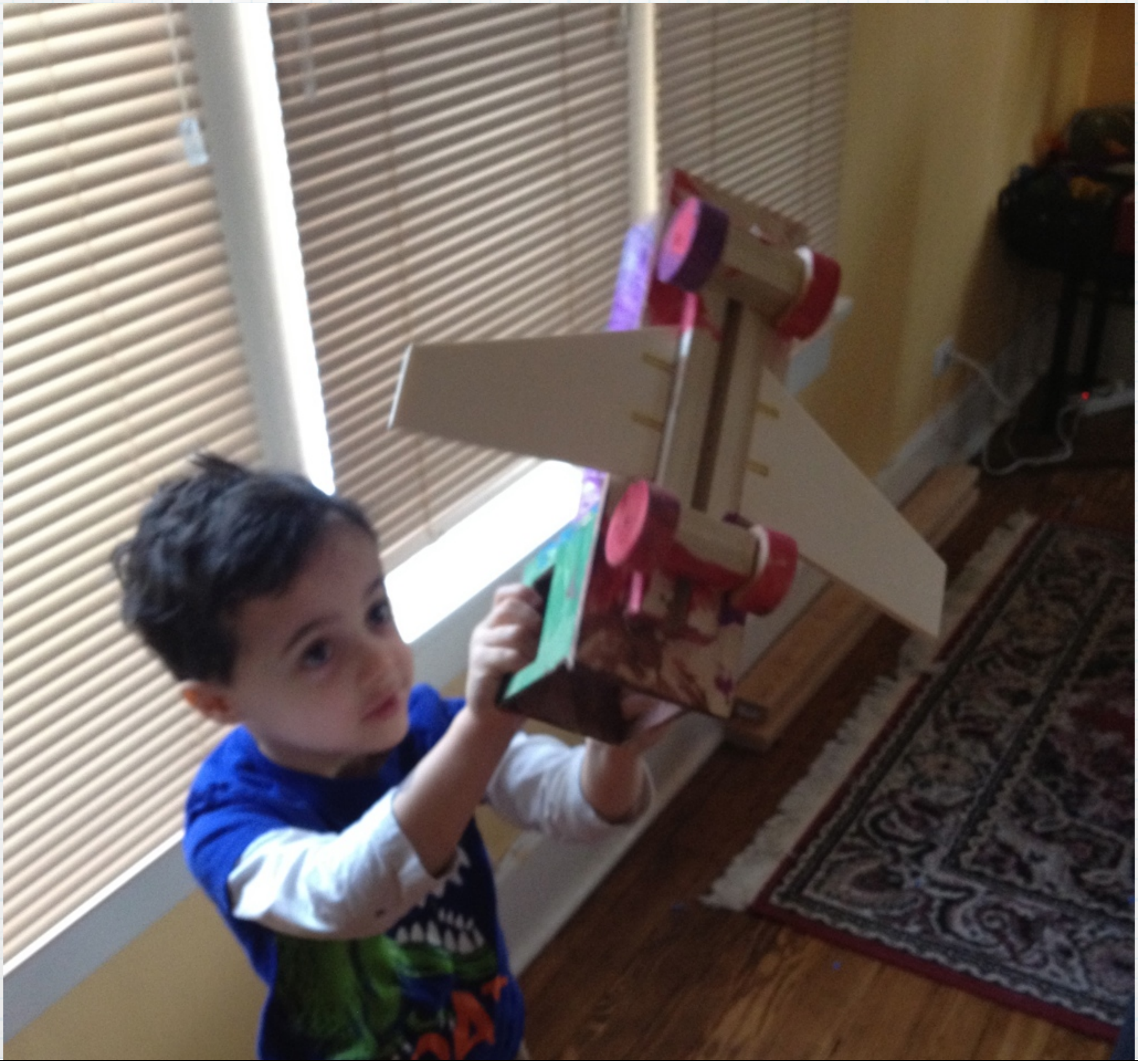
Ladder: For Fire Truck



Detachable Wing

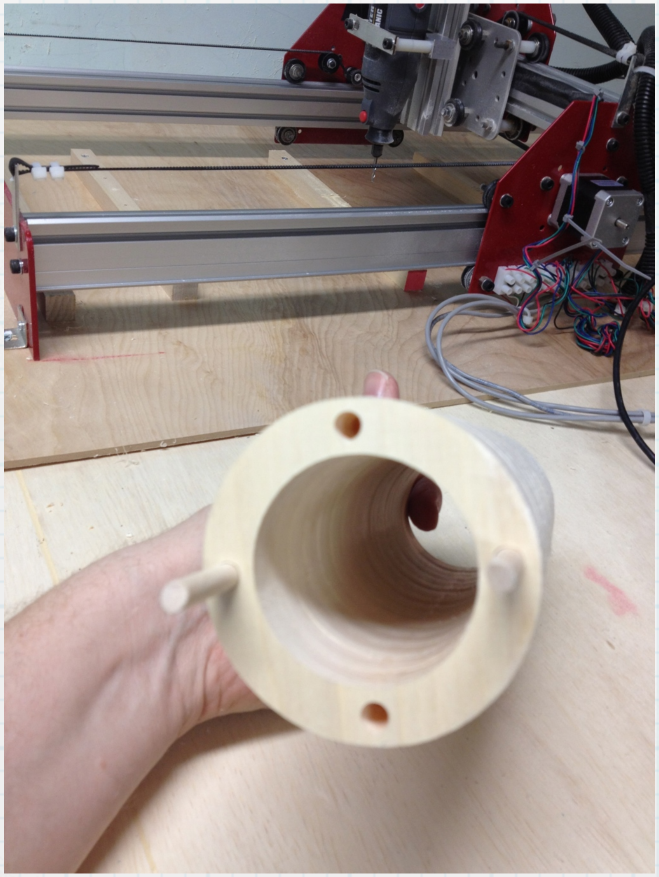


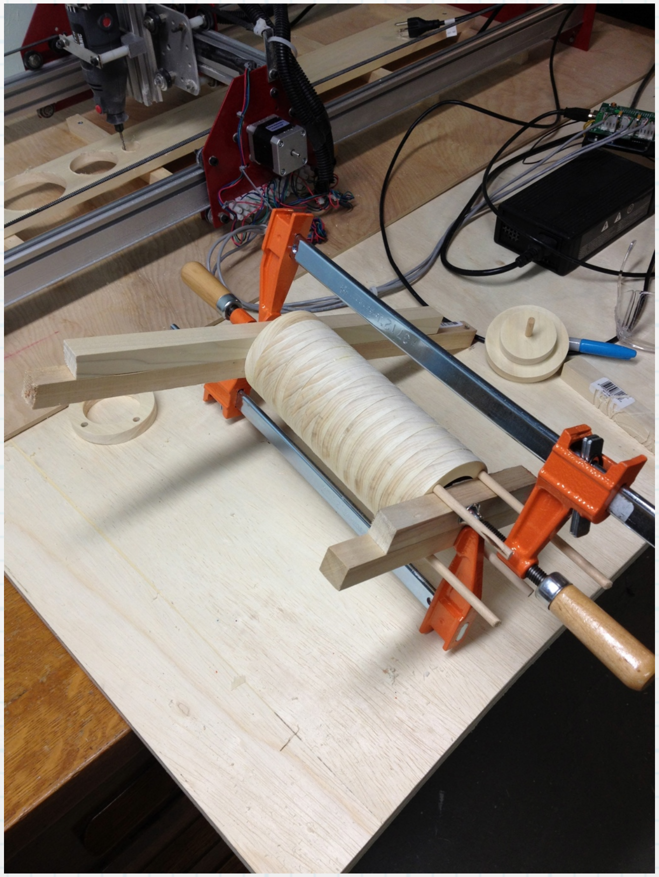
Flying Wooden Train



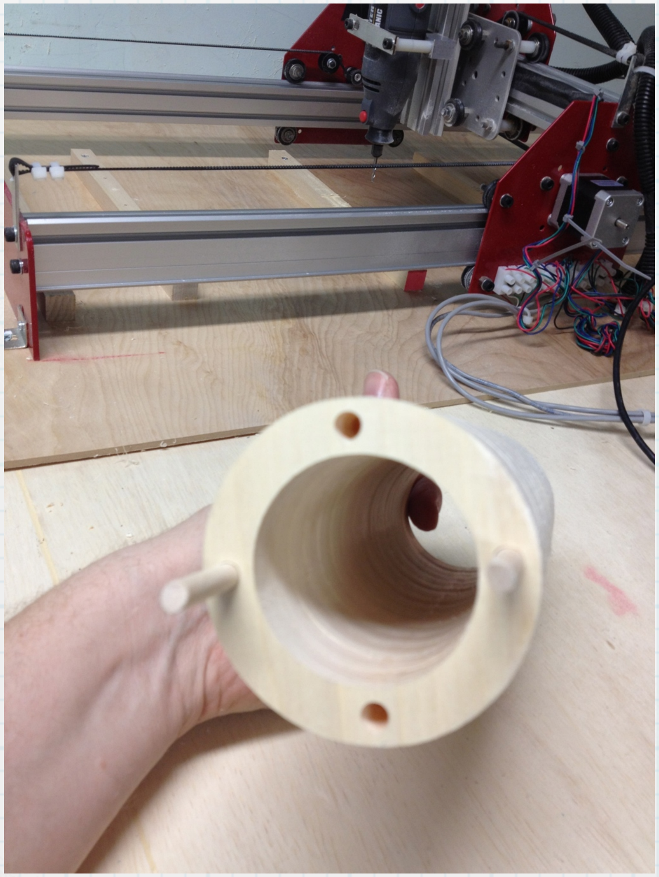
Flying Wooden Train with Hidden Pirate Treasure Chest

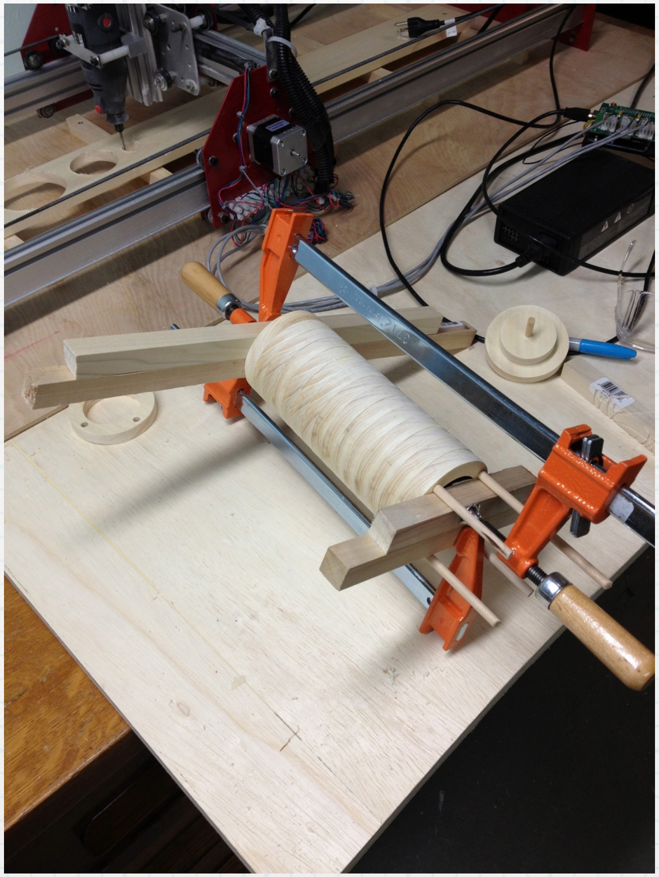
Flying Wooden Train with Hidden Pirate Treasure Chest





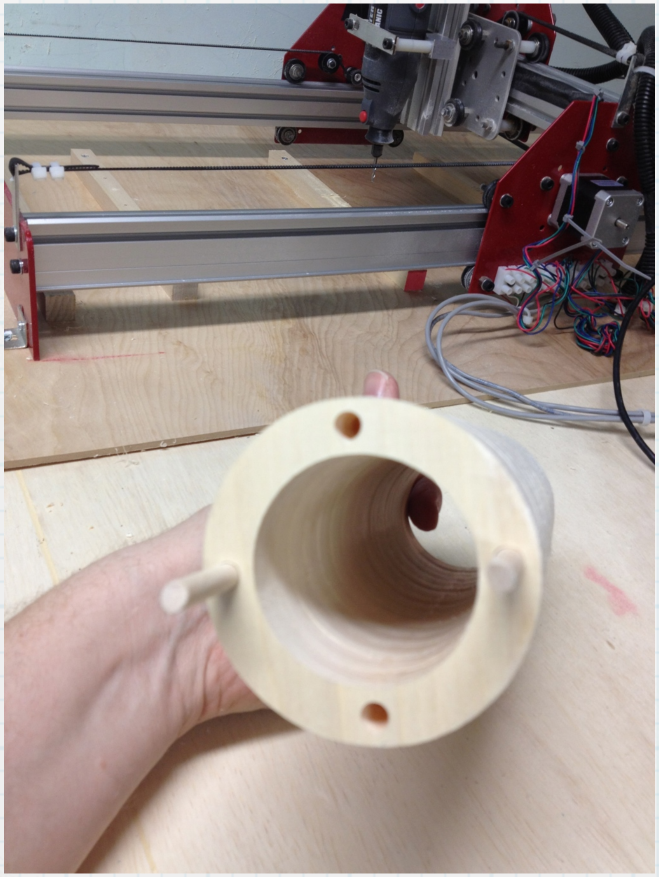
Flying Wooden Train with Hidden Pirate Treasure Chest

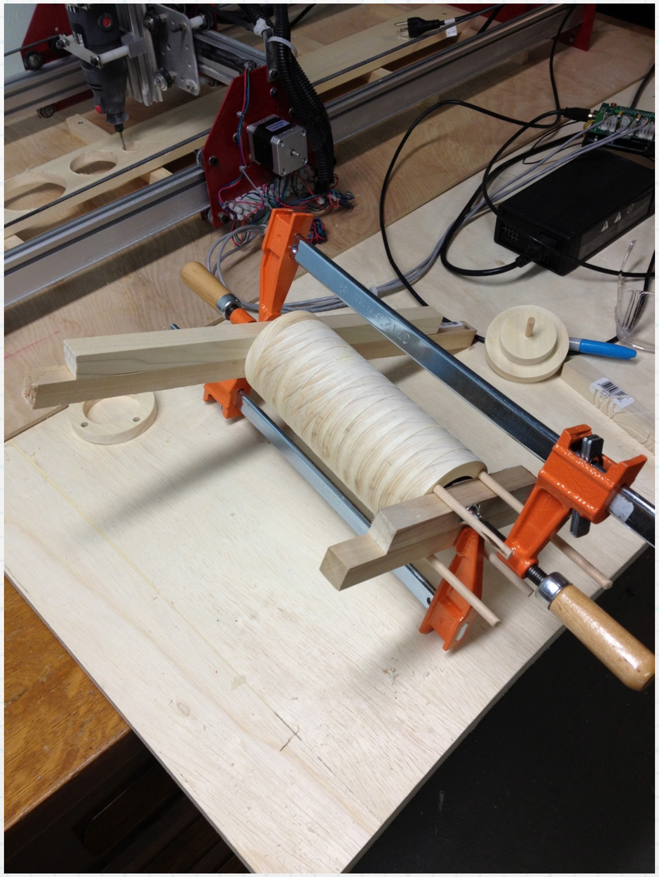




"Kid, if this doesn't stop, I'm going to start making you write a requirements doc."

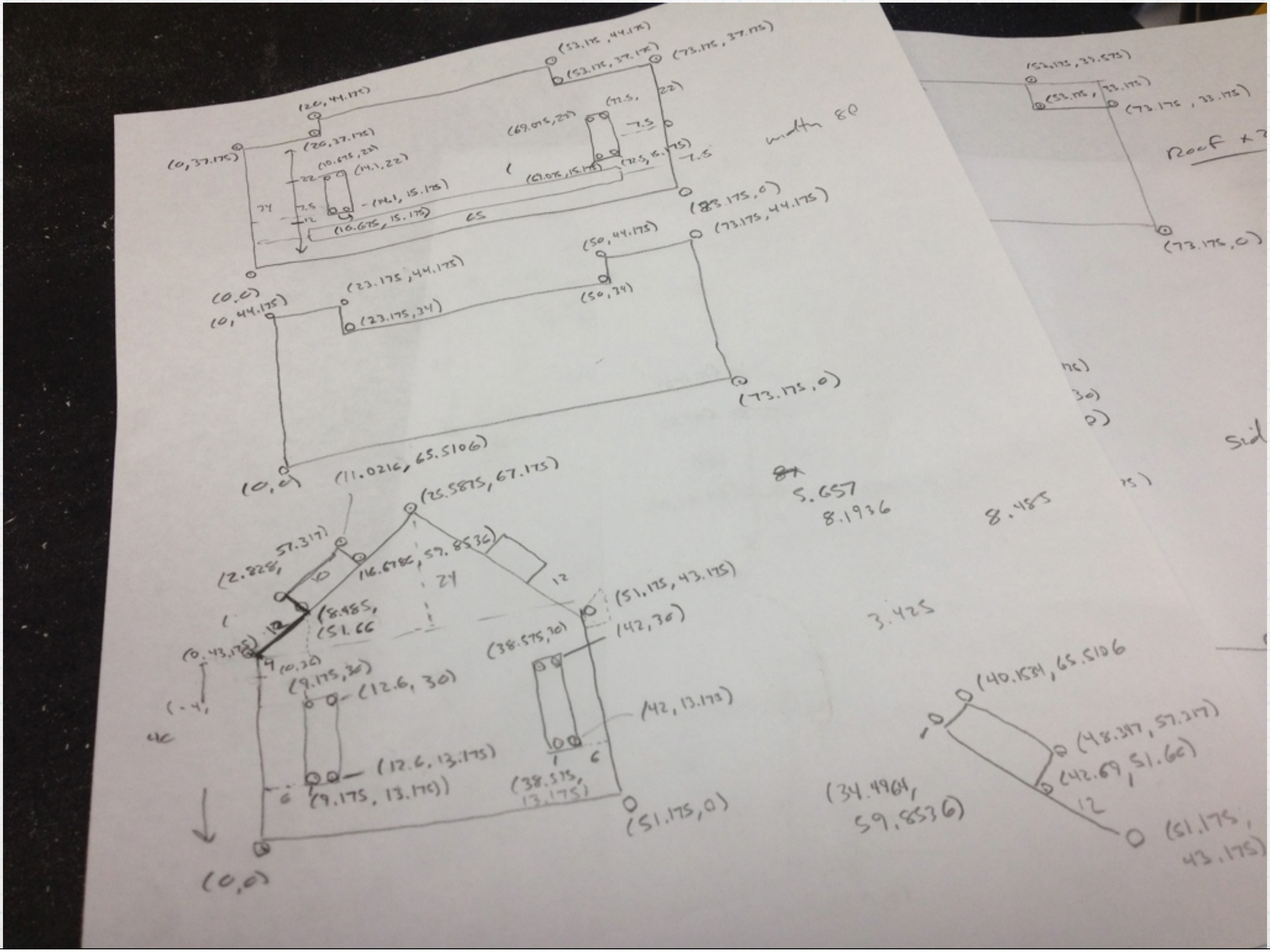
Flying Wooden Train with Hidden Pirate Treasure Chest





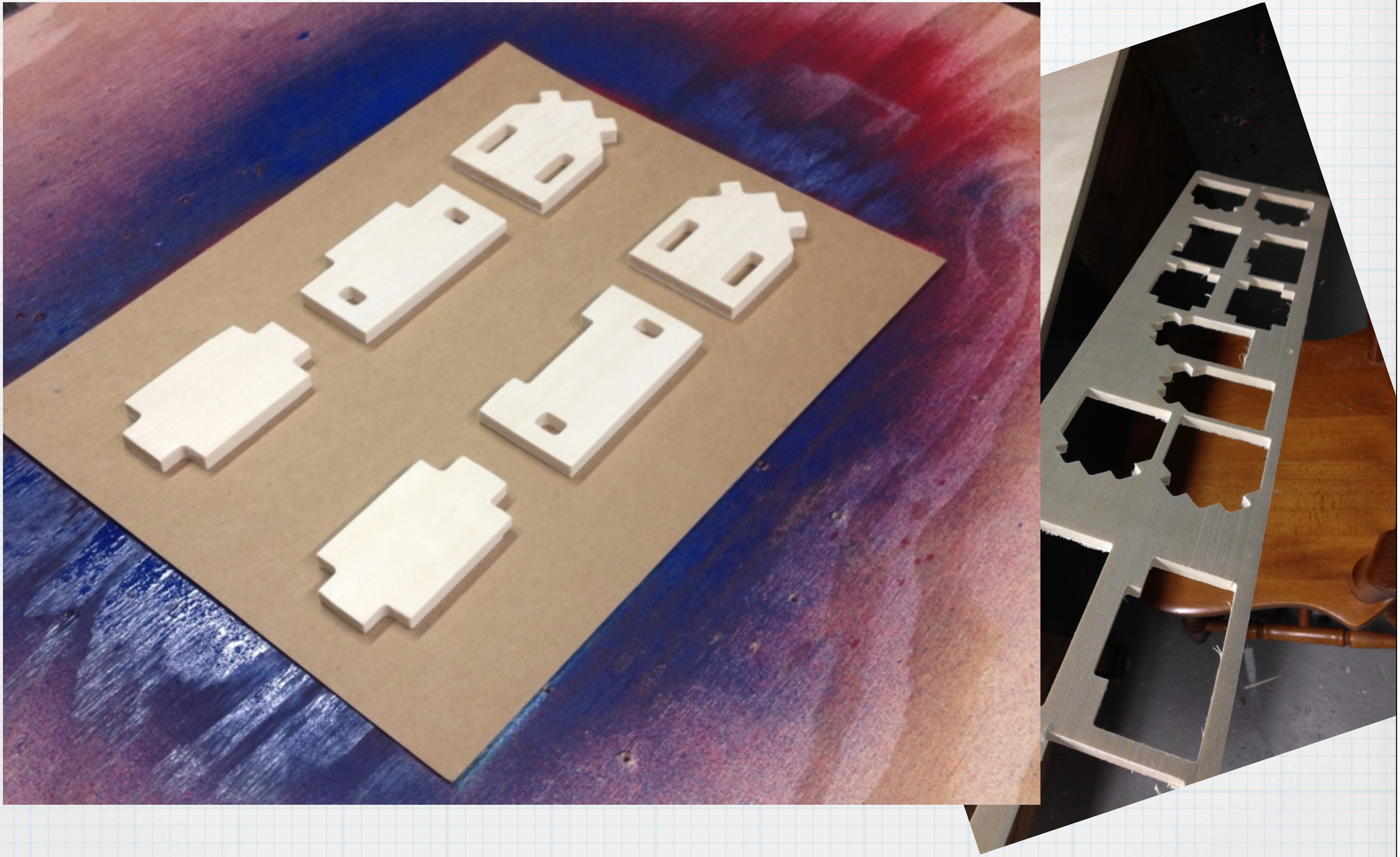
"Kid, if this doesn't stop, I'm going to start making you write a requirements doc--in Docbook XML"

Or maybe...



(53.175, 44.175)
(53.175, 37.175)
(73.175, 37.175)
(20, 44.175)
(20, 37.175)
(0, 37.175)
(10.675, 22)
(14.1, 22)
(69.075, 22)
(72.5, 22)
7.5
(14.1, 15.175)
(10.675, 15.175)
(69.075, 15.175)
(72.5, 15.175)
24
width 80
(83.175, 0)
(73.175, 44.175)
(50, 44.175)
(0, 0)
(0, 44.175)
(23.175, 44.175)
(23.175, 34)
(50, 34)
(73.175, 0)
(0, 0)
(11.0216, 65.5106)
(25.5875, 67.175)
5.657
8.1936
8.485
(2.828, 57.317)
(16.6786, 57.8536)
24
12
(8.485, 51.66)
(51.175, 43.175)
(42, 30)
(0, 43.175)
(0, 30)
(9.175, 30)
(12.6, 30)
(38.575, 30)
3.425
(42, 13.175)
(12.6, 13.175)
(9.175, 13.175)
(38.575, 13.175)
(51.175, 0)
(0, 0)
(40.1534, 65.5106)
(48.397, 57.317)
(42.69, 51.66)
12
(34.4964, 59.8536)
(51.175, 43.175)
(53.175, 33.575)
(53.175, 33.175)
(73.175, 33.175)
Roof x 2
(73.175, 0)

... wait for it



A miniature bike shed!

Random Consequence...

"Daddy, can you make something?"

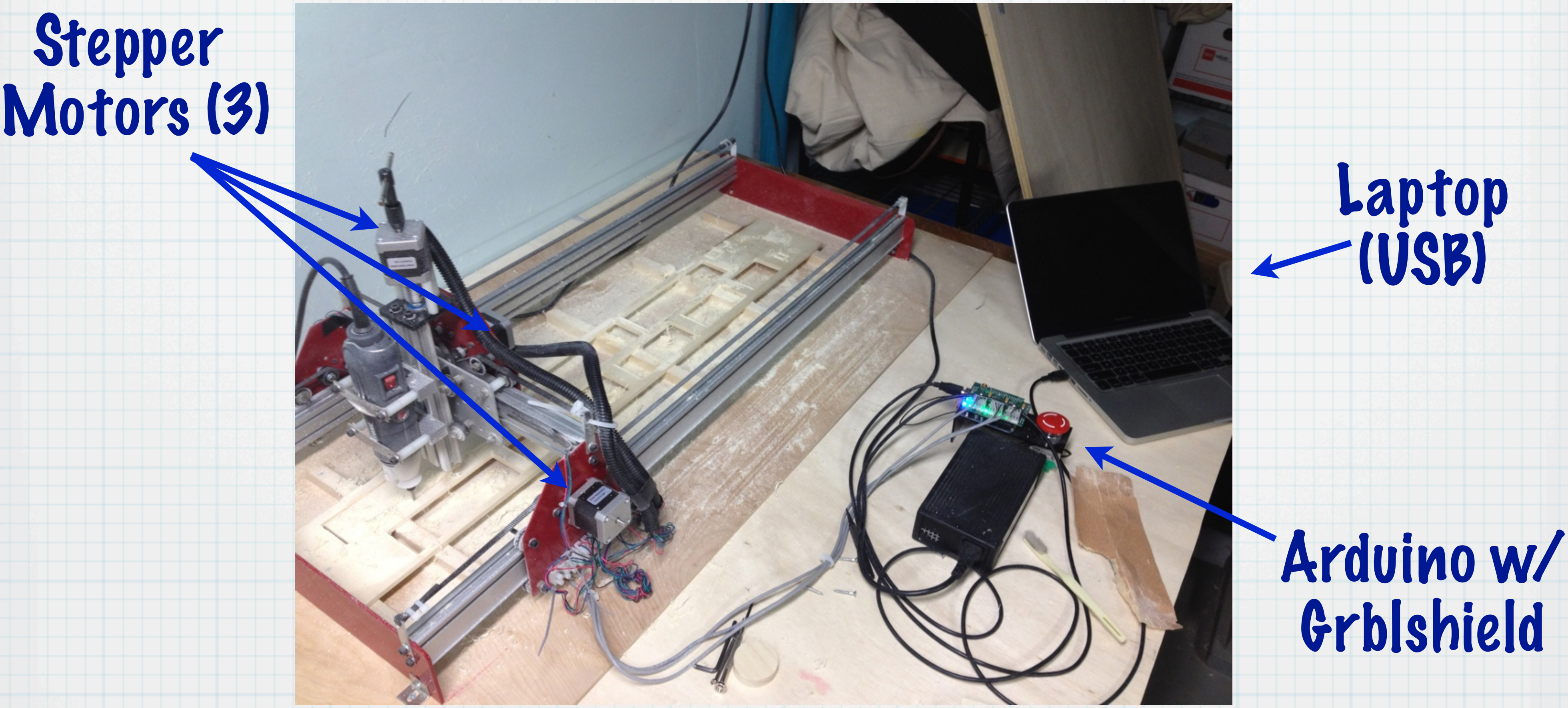
Random Consequence...

"Daddy, can you make something?"

Actual meaning: Install a new iPad app.

(I digress)

Electronics/Tech



Stepper Motors (3)

Laptop (USB)

Arduino w/ Grblshield

Software

It's just serial ports... use pyserial

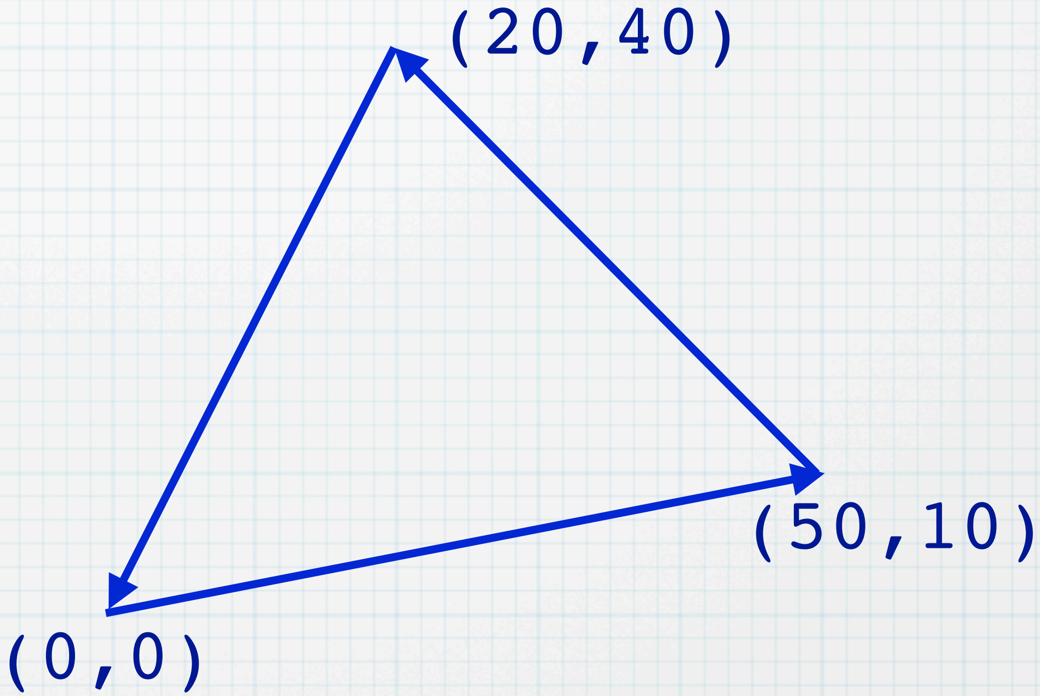
```
import serial
ser = serial.Serial(
     '/dev/tty.usbmodem641', 9600)

def command(cmd):
    ser.send(cmd.encode('ascii')+b'\n')
    resp = ser.readline()
    if resp != b'Ok\n':
        raise RuntimeError(resp)
```

Simple command/response protocol

GCode

Movement controlled by simple commands

```
G1 Z10
G1 X0 Y0
G1 Z-2
G1 X50 Y10
G1 X20 Y40
G1 X0 Y0
G1 Z0
```

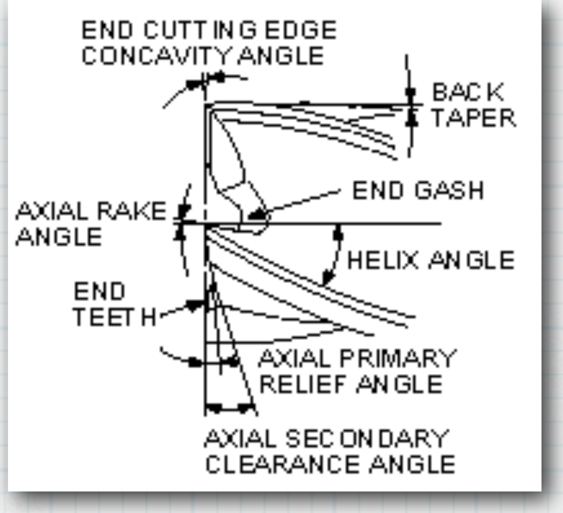


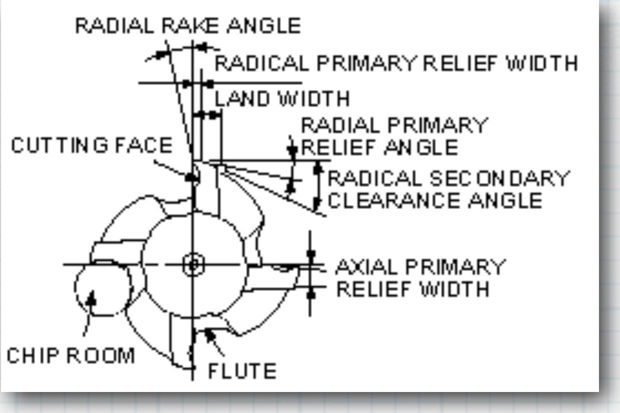
It's a lot like plotting/turtle graphics

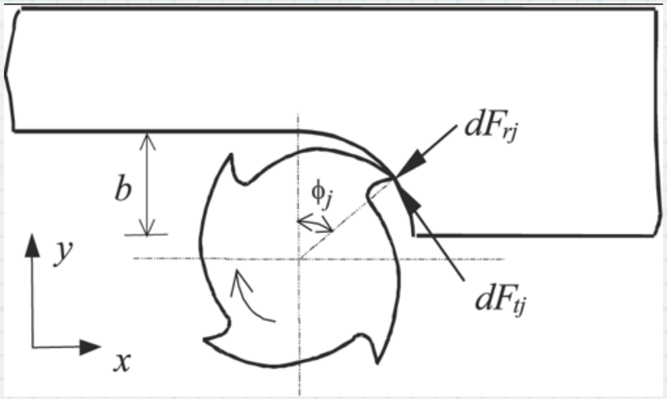
Whirling Knives

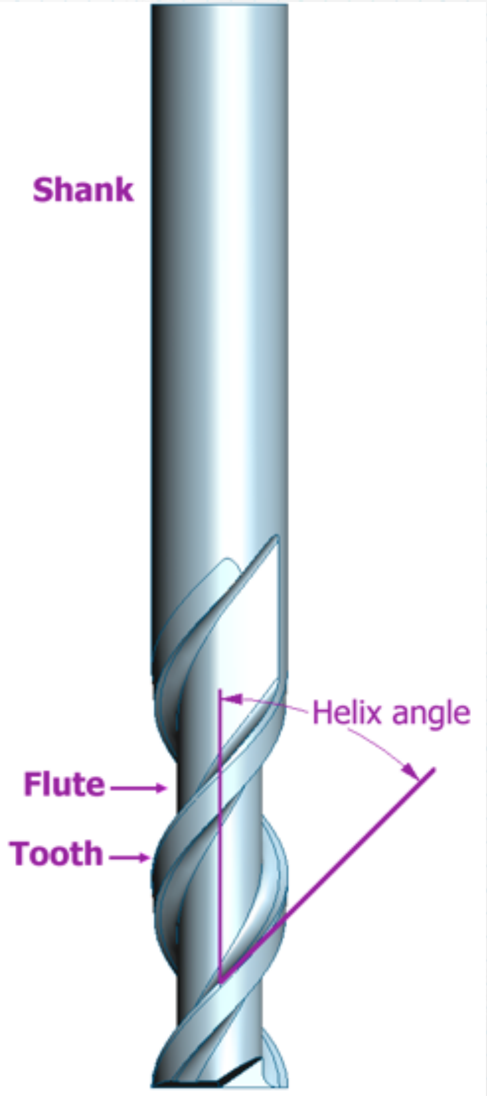
You're in the physical world

Plotting with 25000 RPM end mill


END CUTTING EDGE CONCAVITY ANGLE
BACK TAPER
END GASH
AXIAL RAKE ANGLE
HELIX ANGLE
END TEETH
AXIAL PRIMARY RELIEF ANGLE
AXIAL SECONDARY CLEARANCE ANGLE

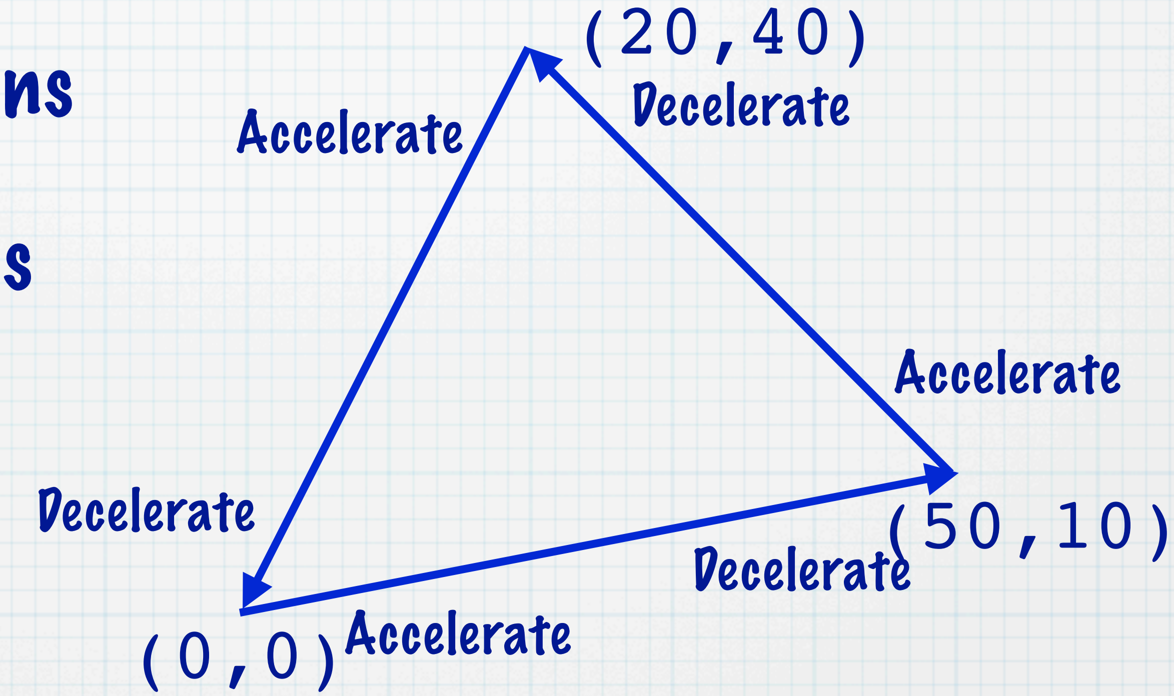

RADIAL RAKE ANGLE
RADICAL PRIMARY RELIEF WIDTH
LAND WIDTH
RADIAL PRIMARY RELIEF ANGLE
CUTTING FACE
RADICAL SECONDARY CLEARANCE ANGLE
AXIAL PRIMARY RELIEF WIDTH
CHIP ROOM
FLUTE



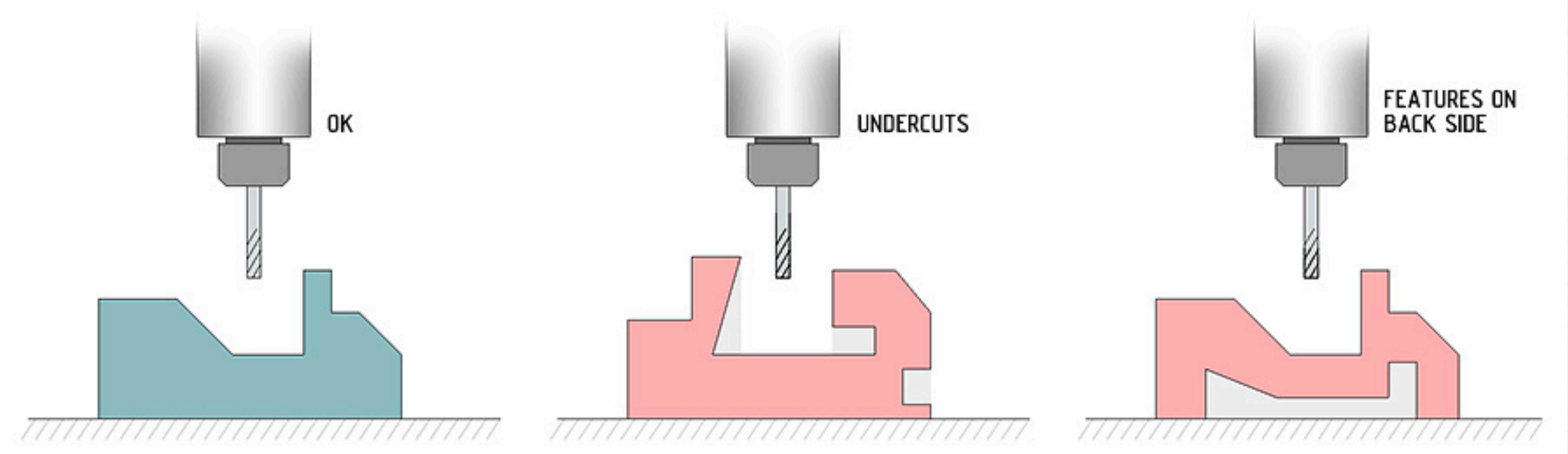


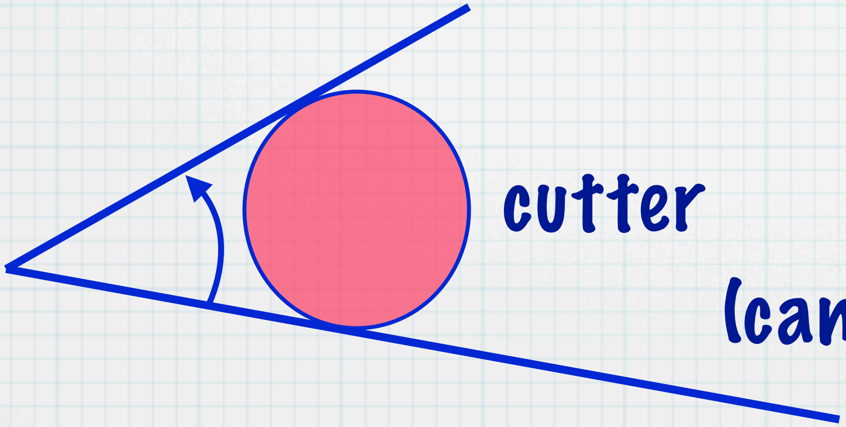
Physics

- No instantaneous motion
- Hardware limitations
- Material properties



Geometry Restrictions





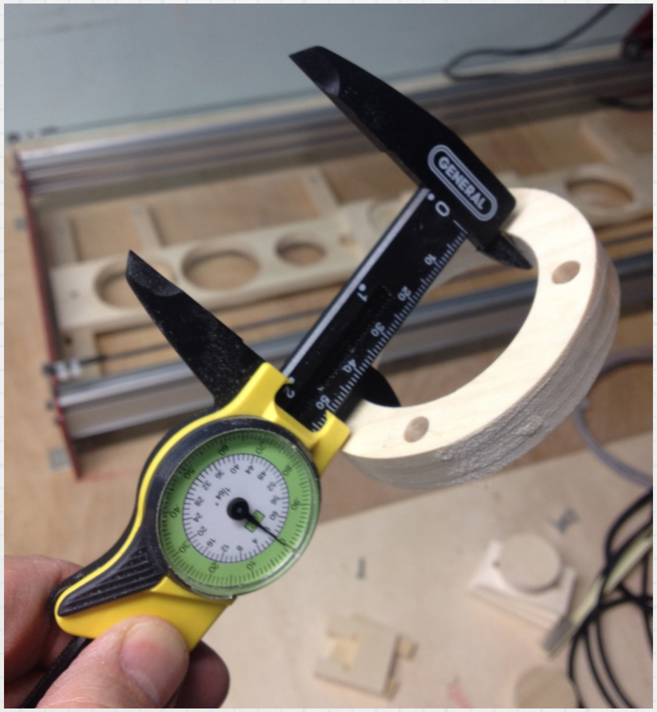
cutter

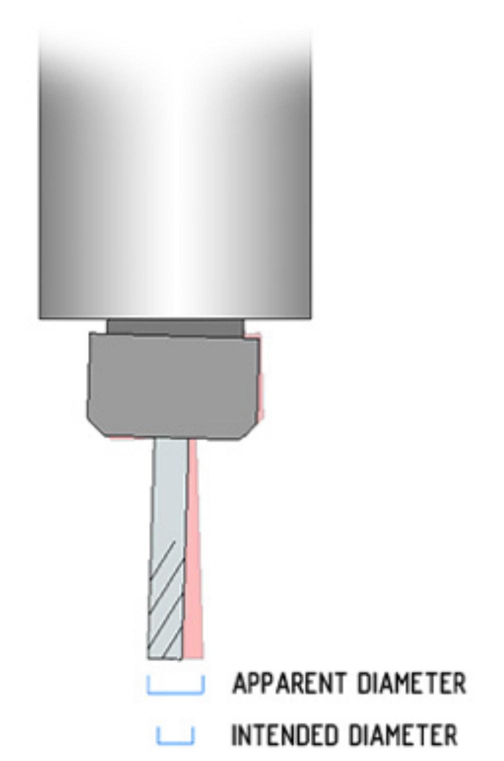
(can't cut exactly into vertices)

It's Inexact

There are real hardware "errors"

- Wobble in rotary tools (runout)
- Misalignment/centering issues



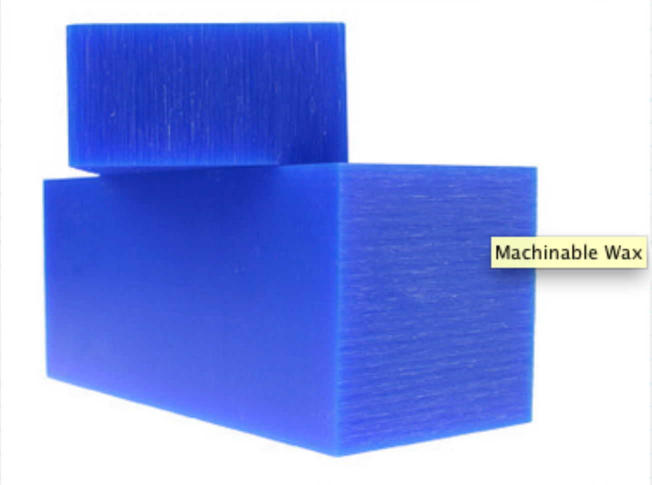


Materials Science

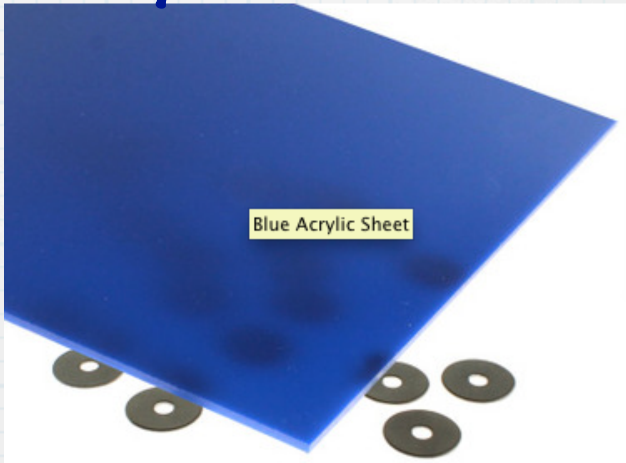
Wood

HPDE Plastic

Machinable Wax

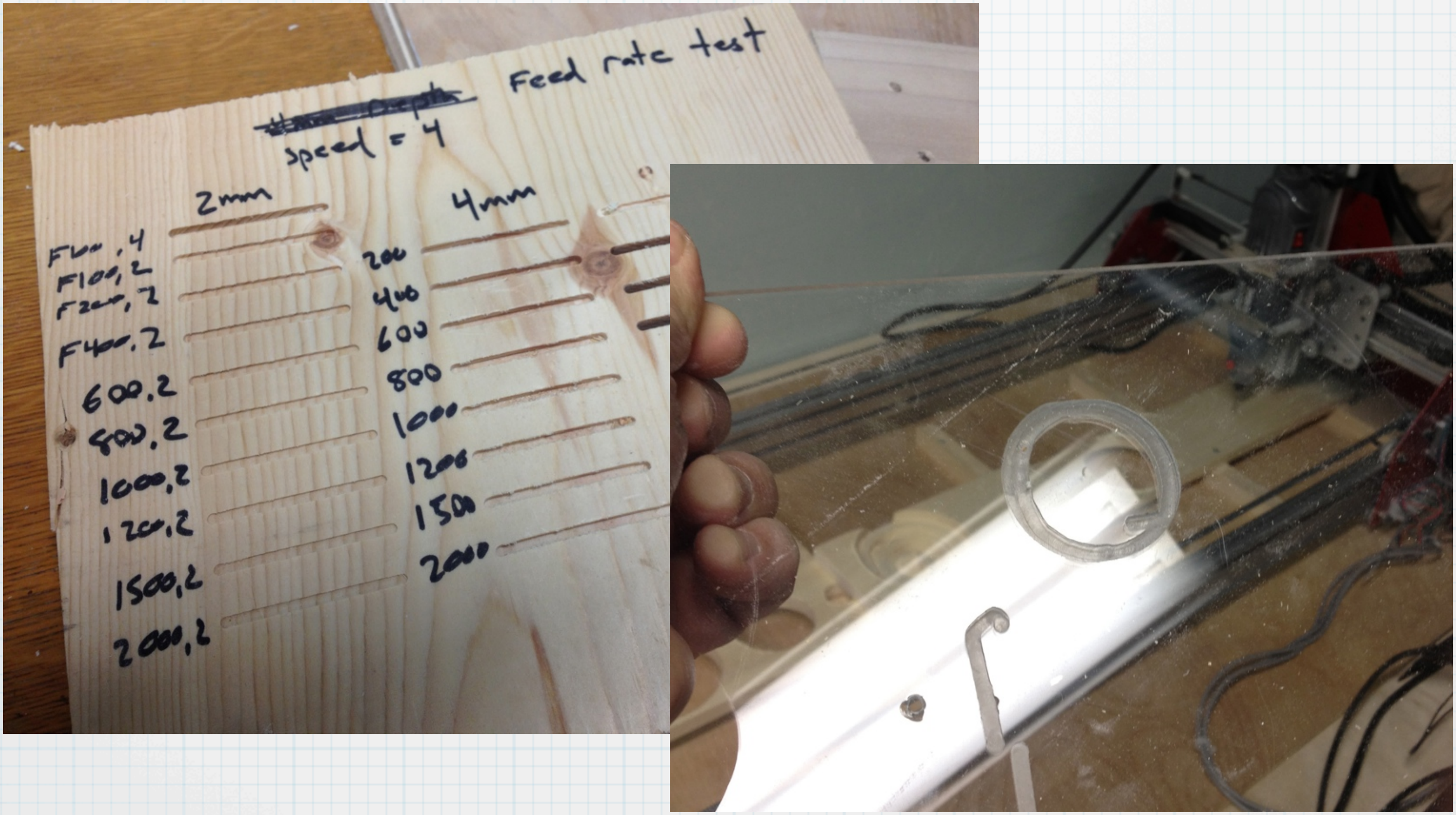


Acrylic



A Science Experiment?

Feed rates, rotational speed, cut depth, etc.

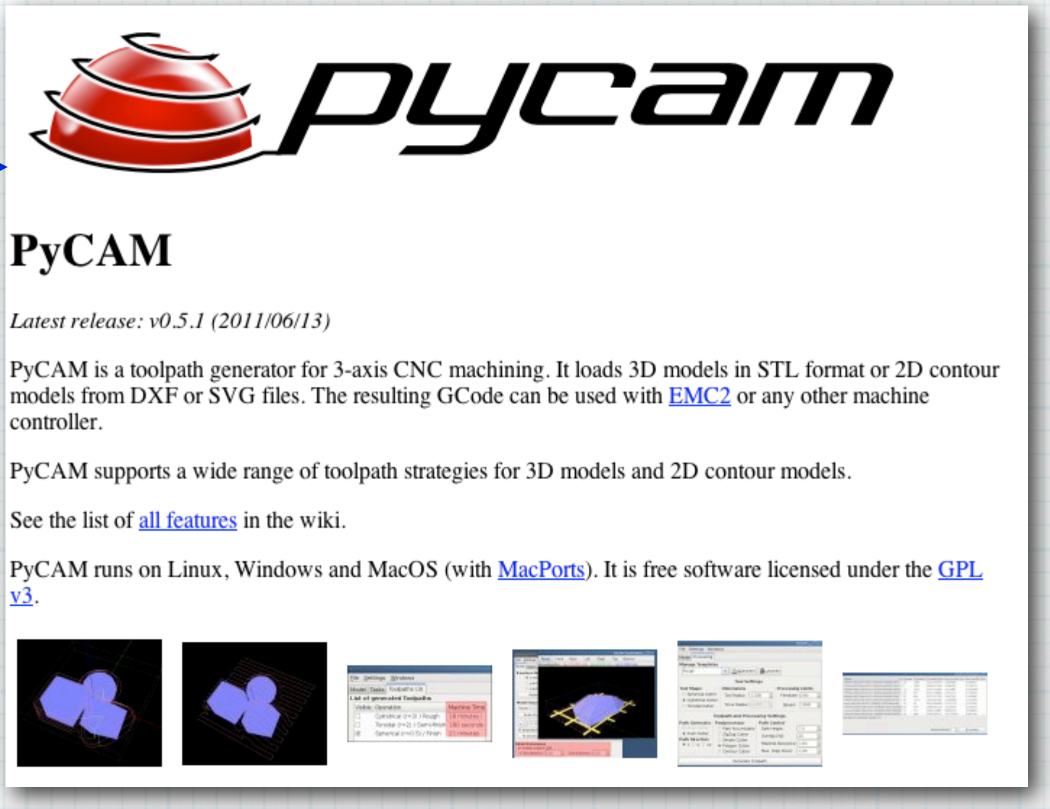


CAM Software

Maybe there's a reason why there's a whole industry of expensive "Computer Aided Manufacturing" Software

Of course, there's a free Python one too...

(I have not used it)



PyCAM

Latest release: v0.5.1 (2011/06/13)

PyCAM is a toolpath generator for 3-axis CNC machining. It loads 3D models in STL format or 2D contour models from DXF or SVG files. The resulting GCode can be used with EMC2 or any other machine controller.

PyCAM supports a wide range of toolpath strategies for 3D models and 2D contour models.

See the list of all features in the wiki.

PyCAM runs on Linux, Windows and MacOS (with MacPorts). It is free software licensed under the GPL v3.

DIY CAM

CNC is "simple" enough to write scripts

```
part =  [
    (0, 0),
    (0, 44.175),
    (23.175, 44.175),
    (23.175, 37.175),
    (60, 37.175),
    (60, 44.175),
    (83.175, 44.175),
    (83.175, 0),
    (0,0)
]
```

DIY CAM

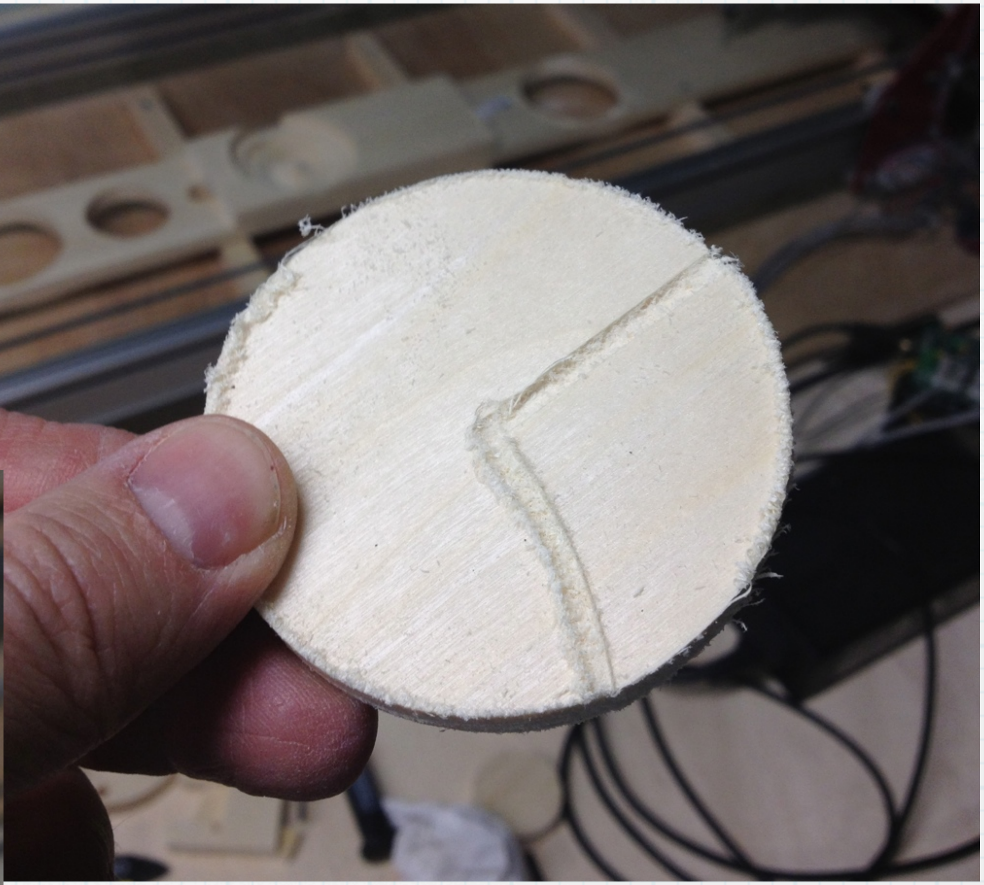
Direct streaming of GCode

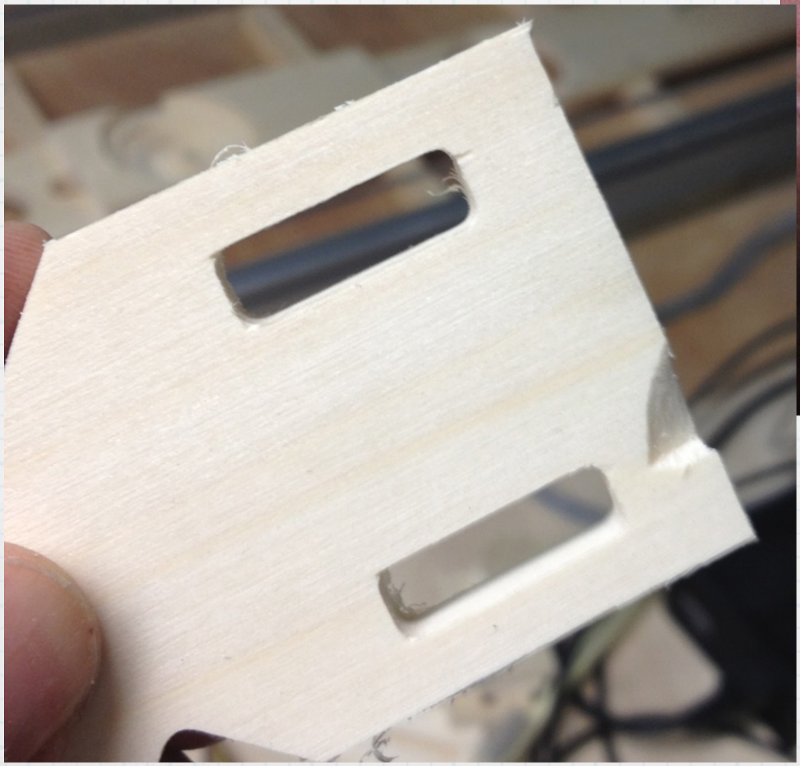
```
command('F1000')
for z in range(1,10):
    command('G1 Z-%s' % z)
    for x, y in part:
        command('G1 X%s Y%s' % (x,y))
command('G1 Z0')
```

You'll see those whirling knives moving around!

Programming Errors

They take physical form!





Real Dangers

Rotating Knives!

- Drill through table
- Jamming
- Shattered End Mill
- Harmonic vibration
- Destruction

A simple sign error can get interesting

Real Dangers

Rotating Knives!

- Drill through table
- Jamming
- Shattered End Mill
- Harmonic vibration
- Destruction

A simple sign error can get interesting

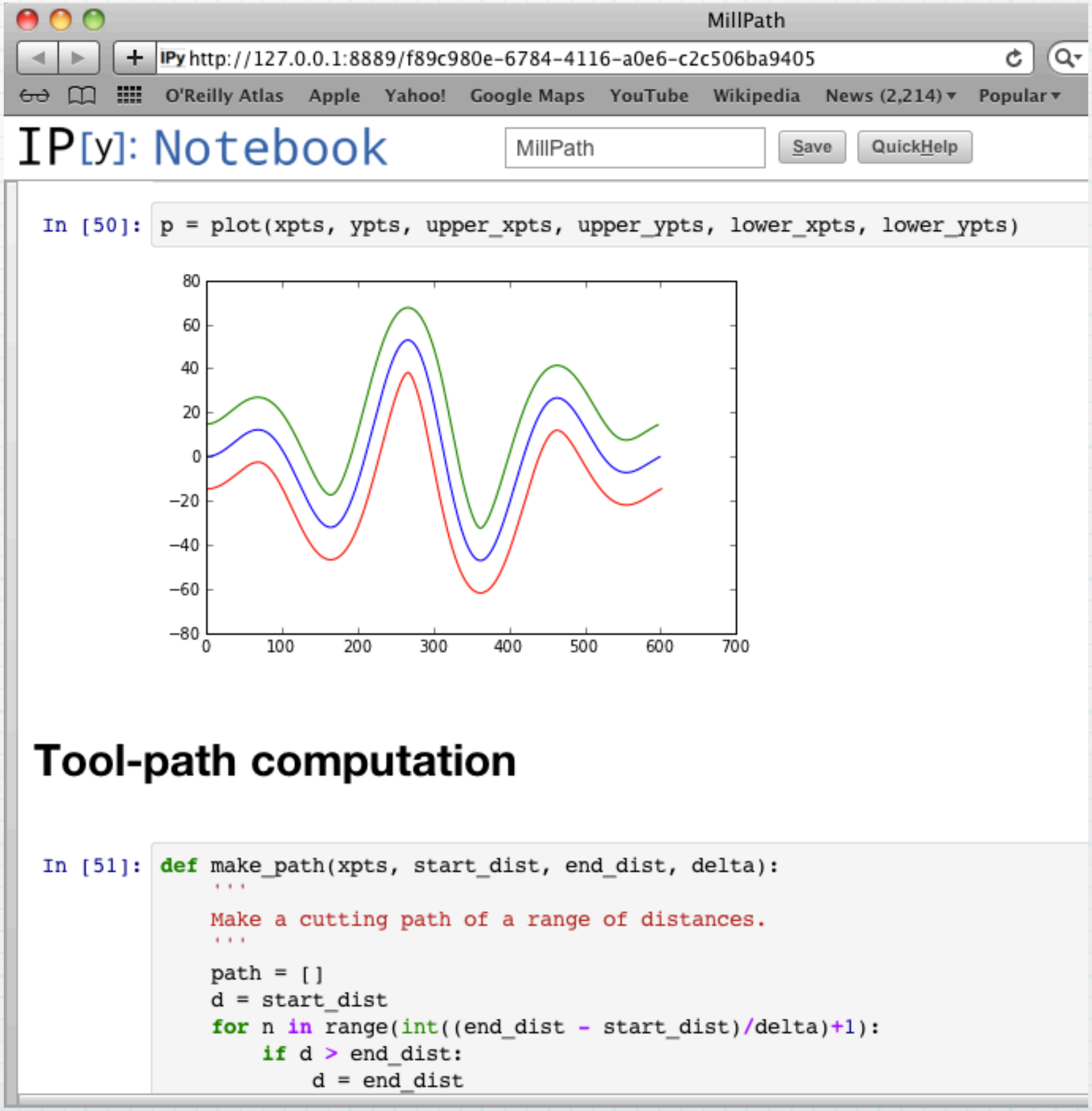
TDD?

Solution?

IPython Notebook?

Is there anything it can't do?

Why not CAM?



```
In [50]: p = plot(xpts, ypts, upper_xpts, upper_ypts, lower_xpts, lower_ypts)
```

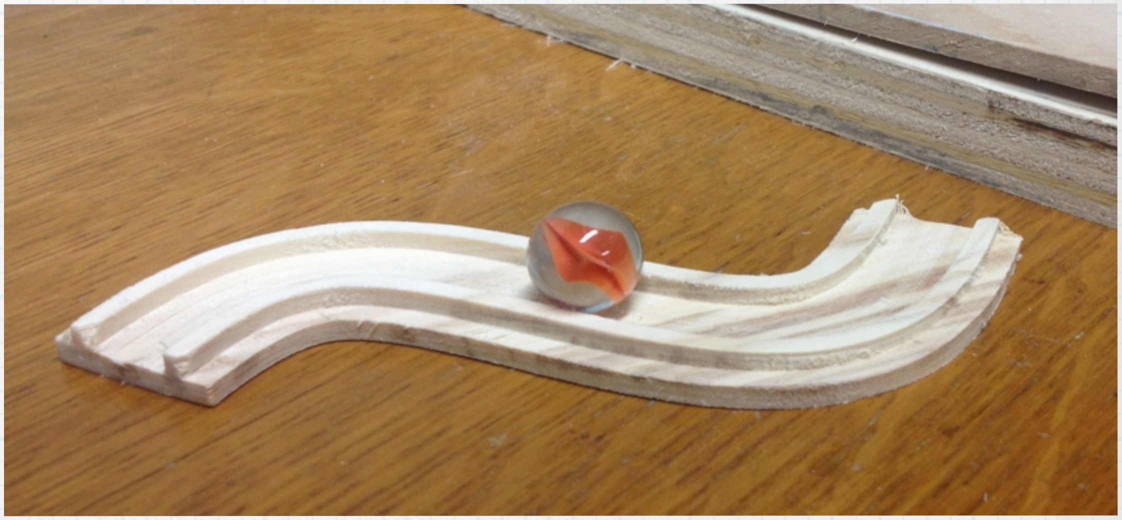
Tool-path computation

```
In [51]: def make_path(xpts, start_dist, end_dist, delta):
             '''
             Make a cutting path of a range of distances.
             '''
             path = []
             d = start_dist
             for n in range(int((end_dist - start_dist)/delta)+1):
                 if d > end_dist:
                     d = end_dist
```

Project: Marble Track

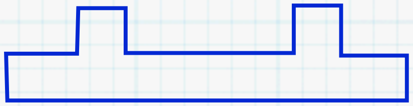
Plot mathematical functions into curvy marble track

f(x) = 30*sin(0.15*x) →

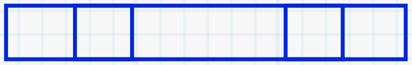


Project: Marble Track

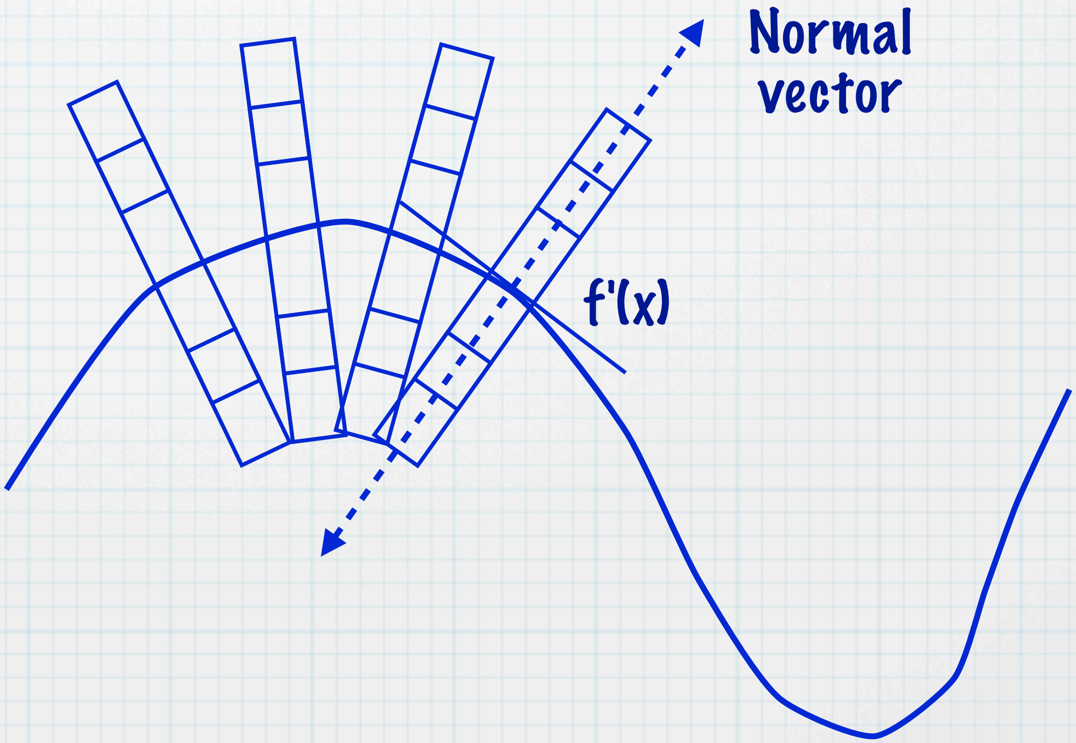
Track Cross-section



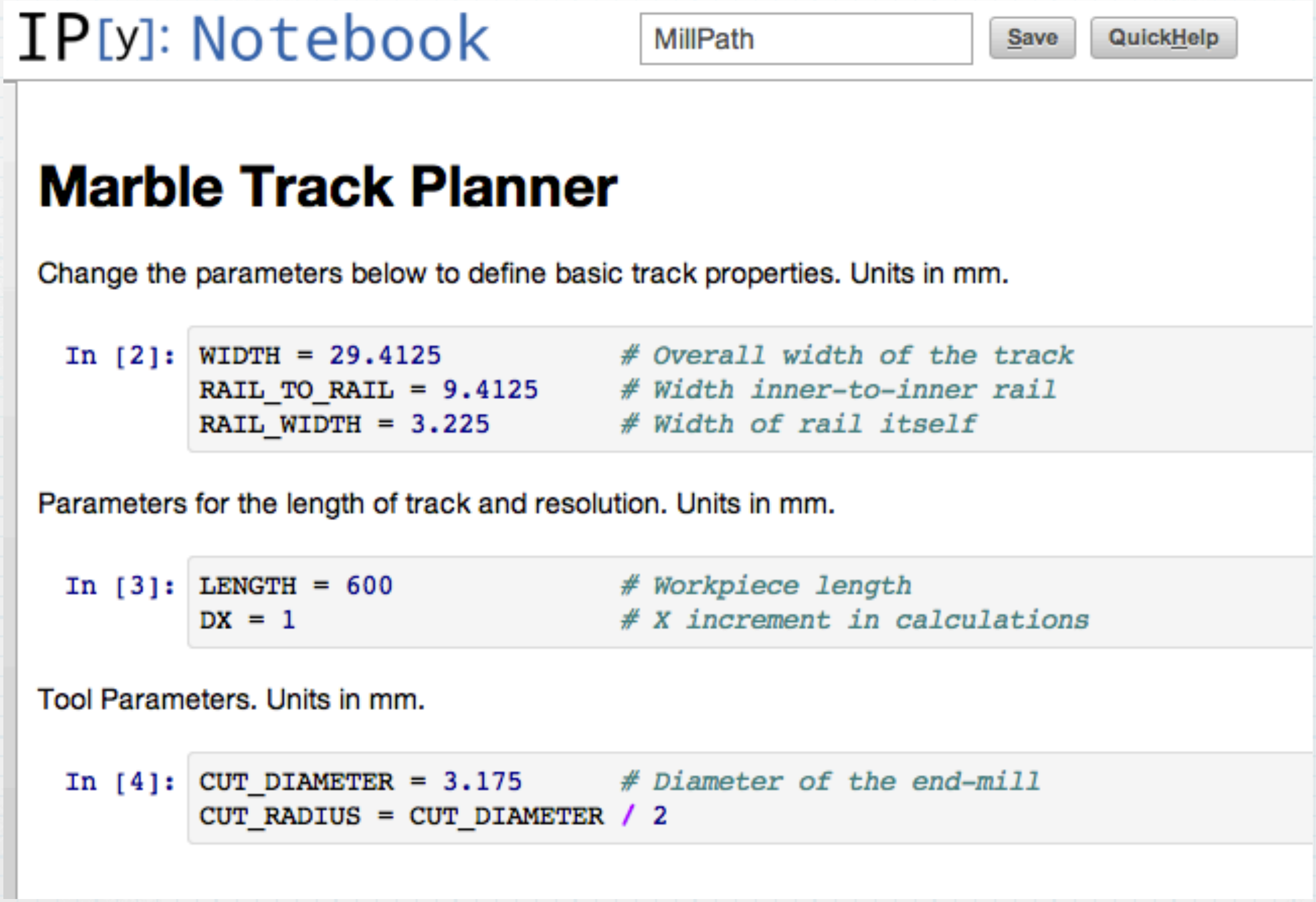
Top View



Track Path



Demo : IPython Notebook



Marble Track Planner

Change the parameters below to define basic track properties. Units in mm.

```
In [2]: WIDTH = 29.4125            # Overall width of the track
        RAIL_TO_RAIL = 9.4125      # Width inner-to-inner rail
        RAIL_WIDTH = 3.225         # Width of rail itself
```

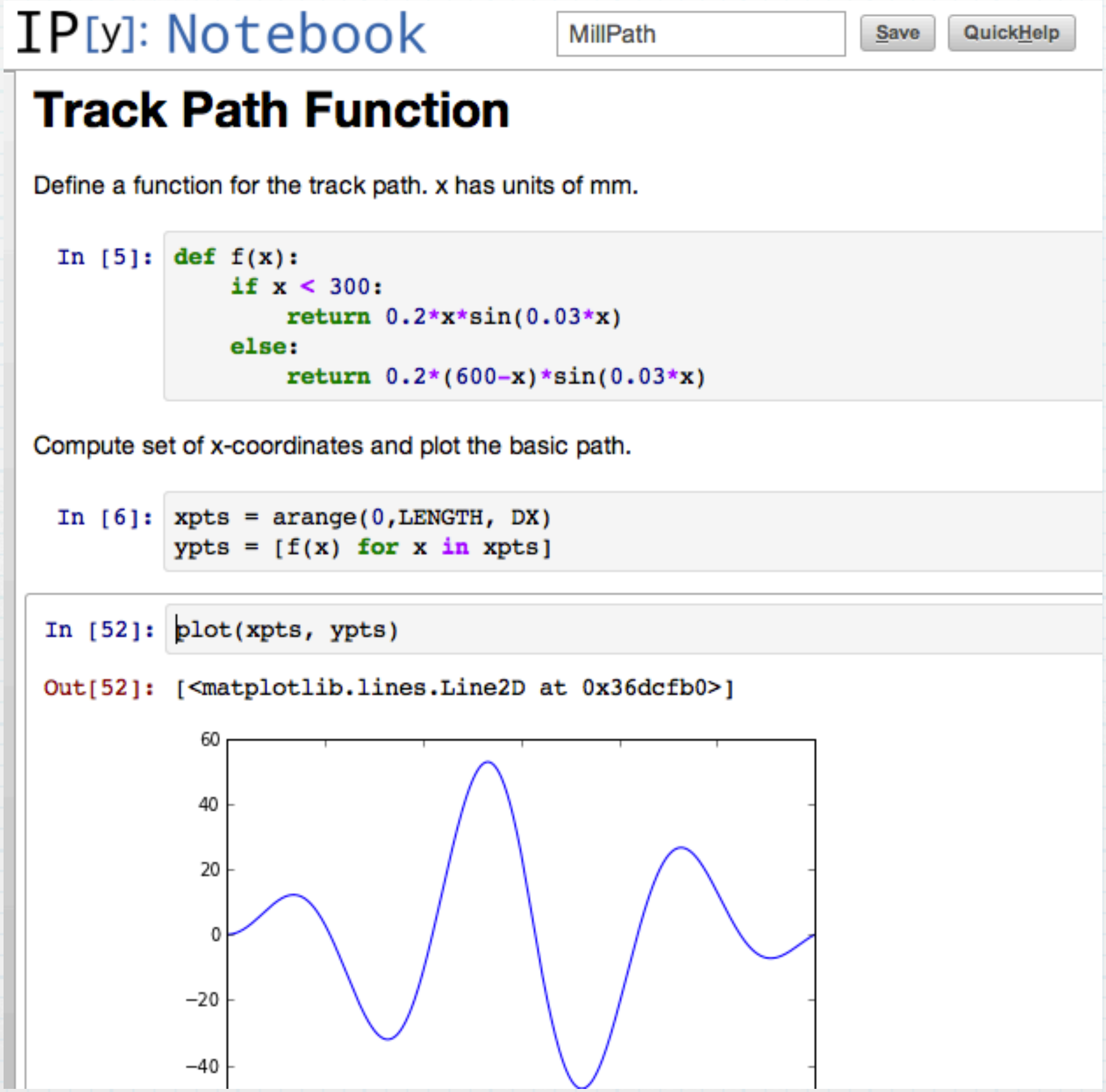
Parameters for the length of track and resolution. Units in mm.

```
In [3]: LENGTH = 600               # Workpiece length
        DX = 1                     # X increment in calculations
```

Tool Parameters. Units in mm.

```
In [4]: CUT_DIAMETER = 3.175       # Diameter of the end-mill
        CUT_RADIUS = CUT_DIAMETER / 2
```

Demo : IPython Notebook



Track Path Function

Define a function for the track path. x has units of mm.

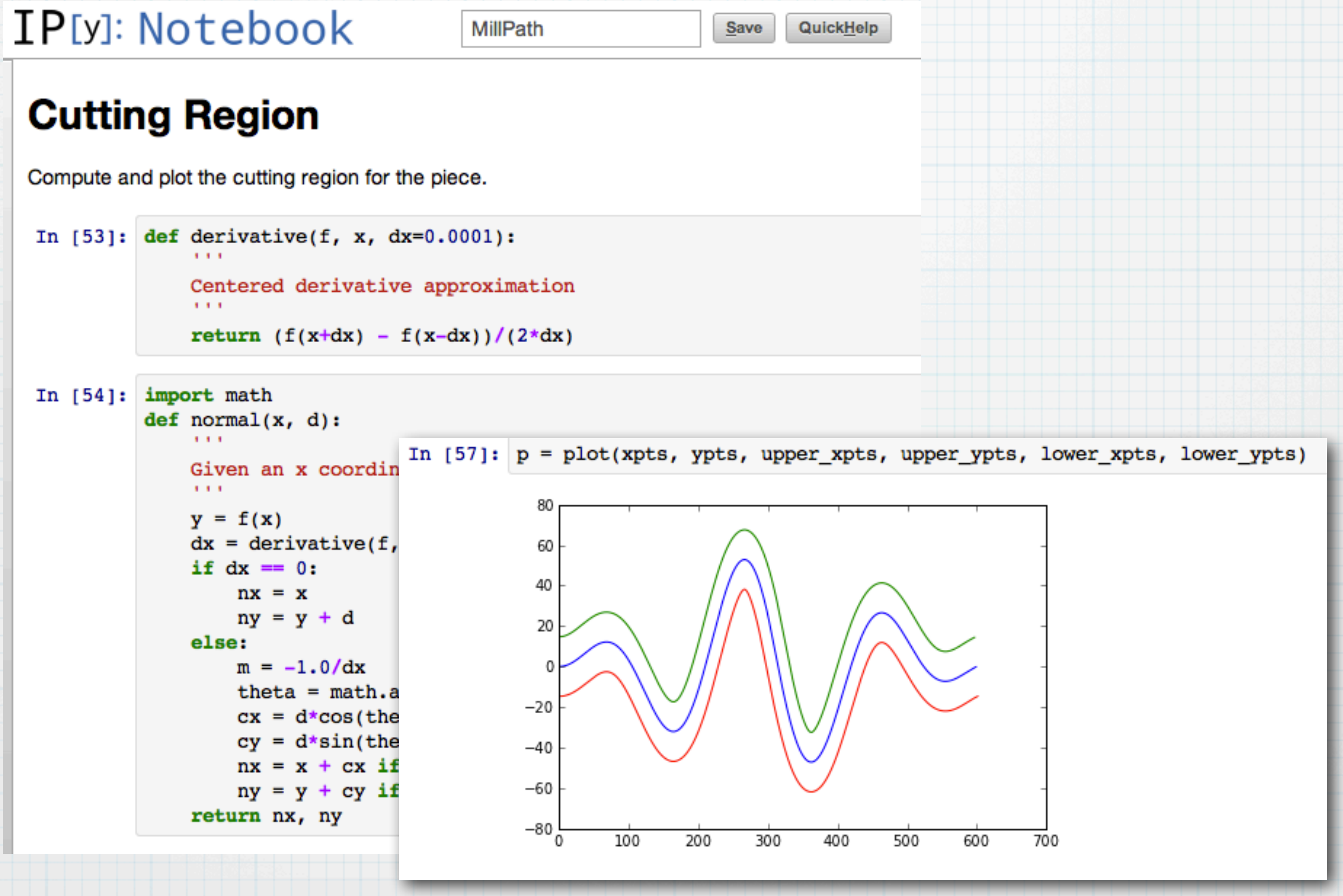
```
In [5]: def f(x):
            if x < 300:
                return 0.2*x*sin(0.03*x)
            else:
                return 0.2*(600-x)*sin(0.03*x)
```

Compute set of x-coordinates and plot the basic path.

```
In [6]: xpts = arange(0,LENGTH, DX)
        ypts = [f(x) for x in xpts]
```

```
In [52]: plot(xpts, ypts)

Out[52]: [<matplotlib.lines.Line2D at 0x36dcfb0>]
```

Demo : IPython Notebook



Cutting Region

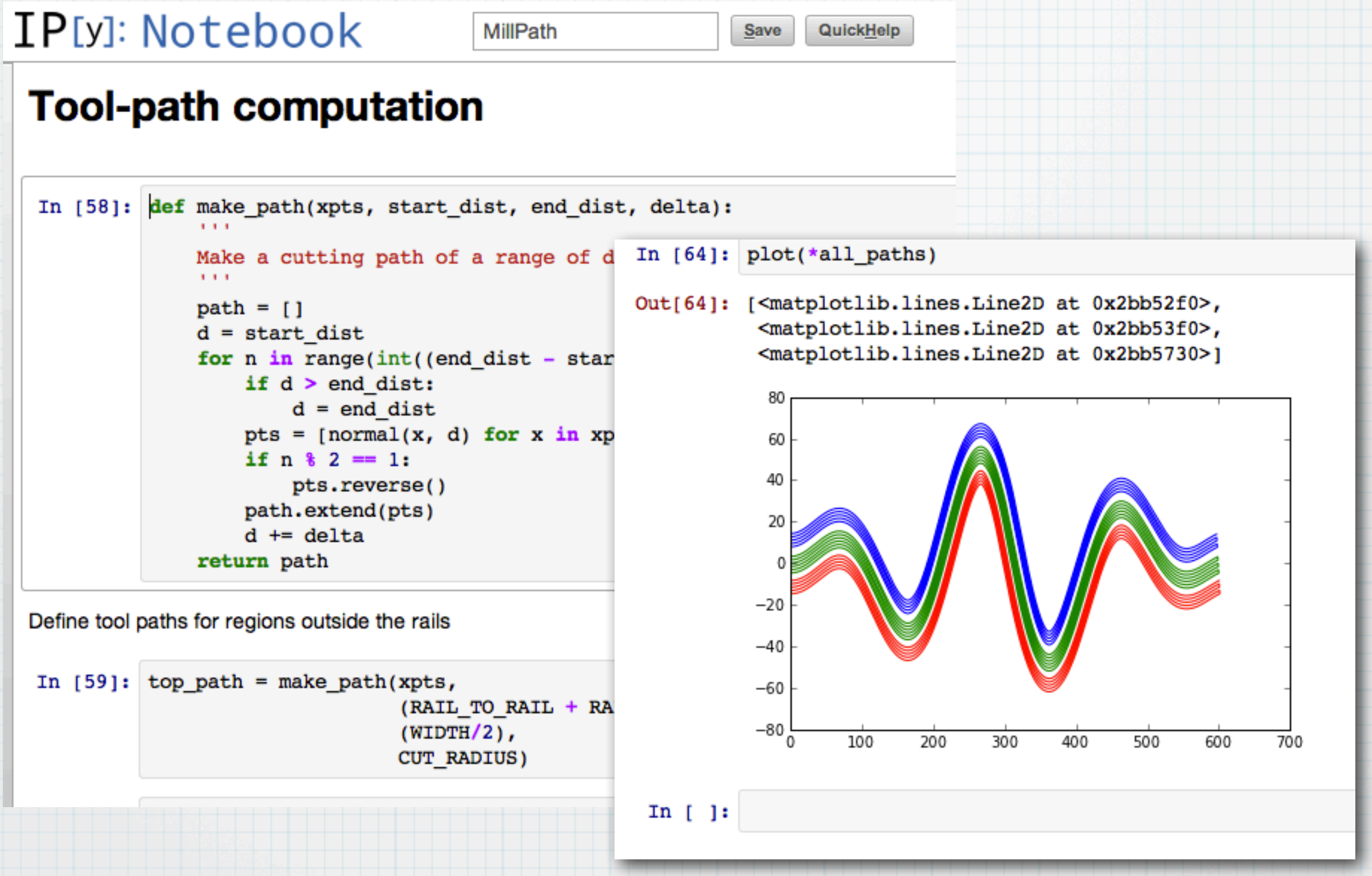
Compute and plot the cutting region for the piece.

```
In [53]: def derivative(f, x, dx=0.0001):
             '''
             Centered derivative approximation
             '''
             return (f(x+dx) - f(x-dx))/(2*dx)
```

```
In [54]: import math
         def normal(x, d):
             '''
             Given an x coordin
             '''
             y = f(x)
             dx = derivative(f,
             if dx == 0:
                 nx = x
                 ny = y + d
             else:
                 m = -1.0/dx
                 theta = math.a
                 cx = d*cos(the
                 cy = d*sin(the
                 nx = x + cx if
                 ny = y + cy if
             return nx, ny
```

```
In [57]: p = plot(xpts, ypts, upper_xpts, upper_ypts, lower_xpts, lower_ypts)
```

Demo : IPython Notebook



Tool-path computation

```
In [58]: def make_path(xpts, start_dist, end_dist, delta):
             '''
             Make a cutting path of a range of d
             '''
             path = []
             d = start_dist
             for n in range(int((end_dist - star
                 if d > end_dist:
                     d = end_dist
                 pts = [normal(x, d) for x in xp
                 if n % 2 == 1:
                     pts.reverse()
                 path.extend(pts)
                 d += delta
             return path
```

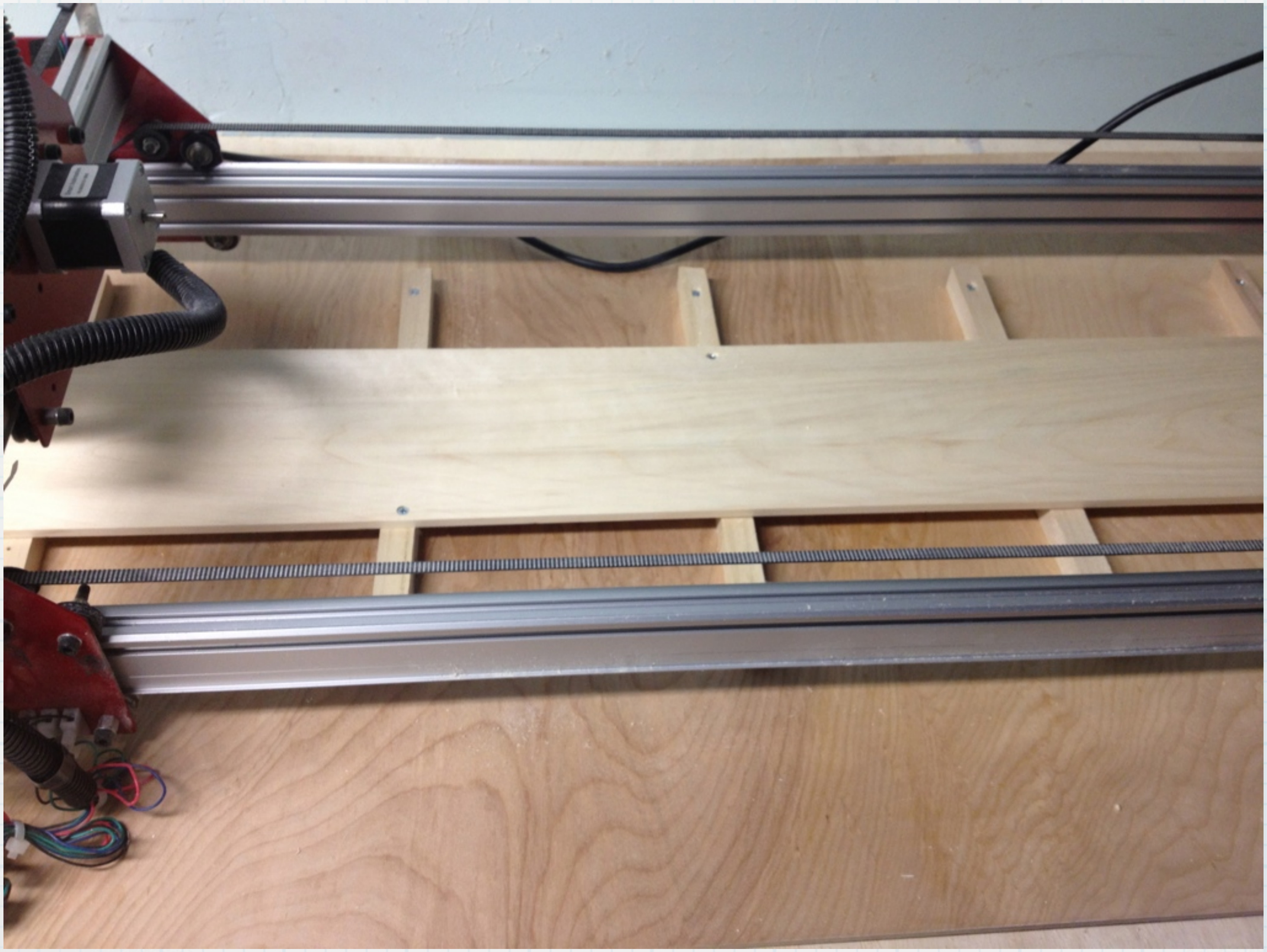
Define tool paths for regions outside the rails

```
In [59]: top_path = make_path(xpts,
                                  (RAIL_TO_RAIL + RA
                                  (WIDTH/2),
                                  CUT_RADIUS)
```

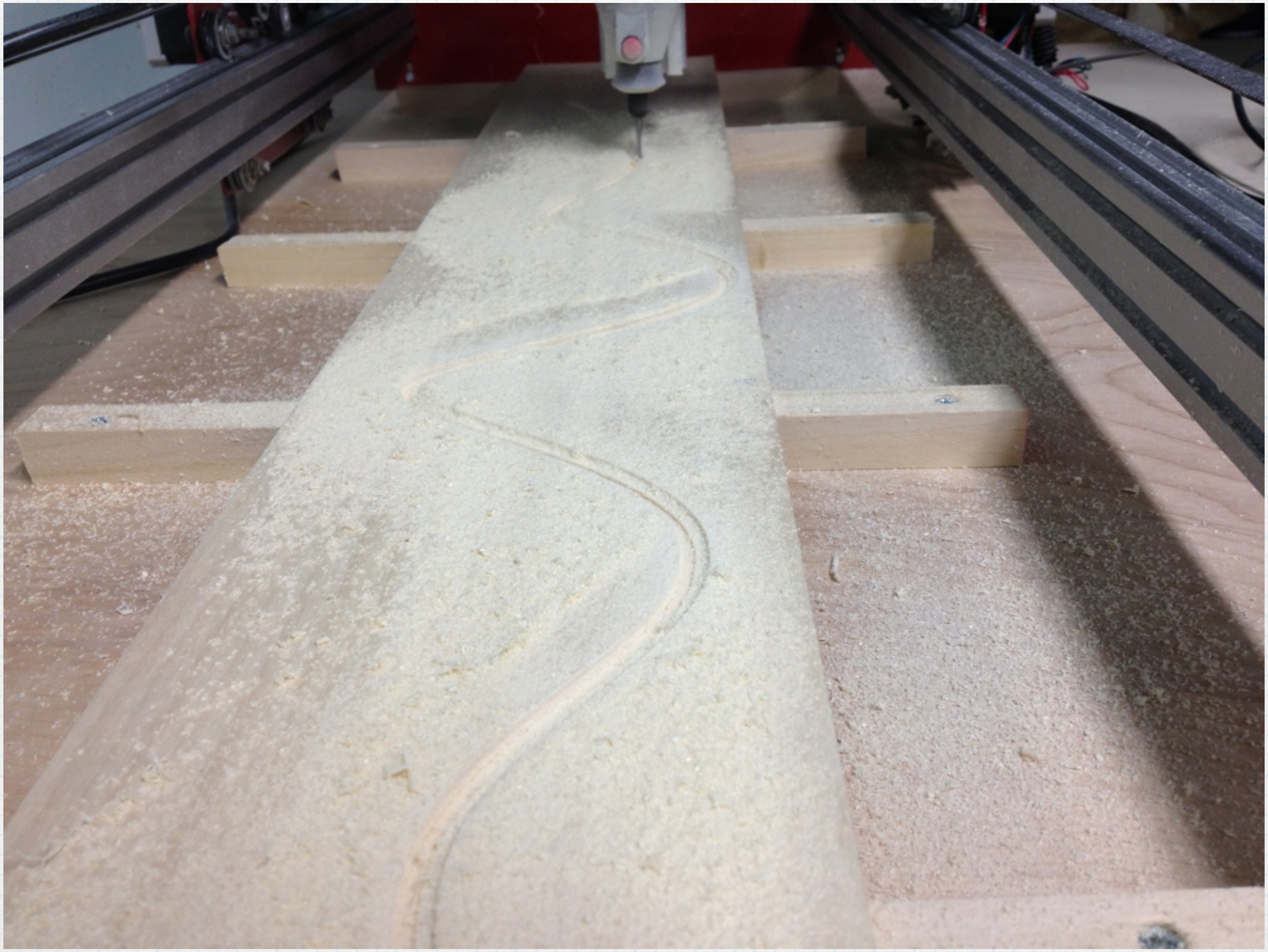
```
In [64]: plot(*all_paths)

Out[64]: [<matplotlib.lines.Line2D at 0x2bb52f0>,
          <matplotlib.lines.Line2D at 0x2bb53f0>,
          <matplotlib.lines.Line2D at 0x2bb5730>]
```

```
In [ ]:
```

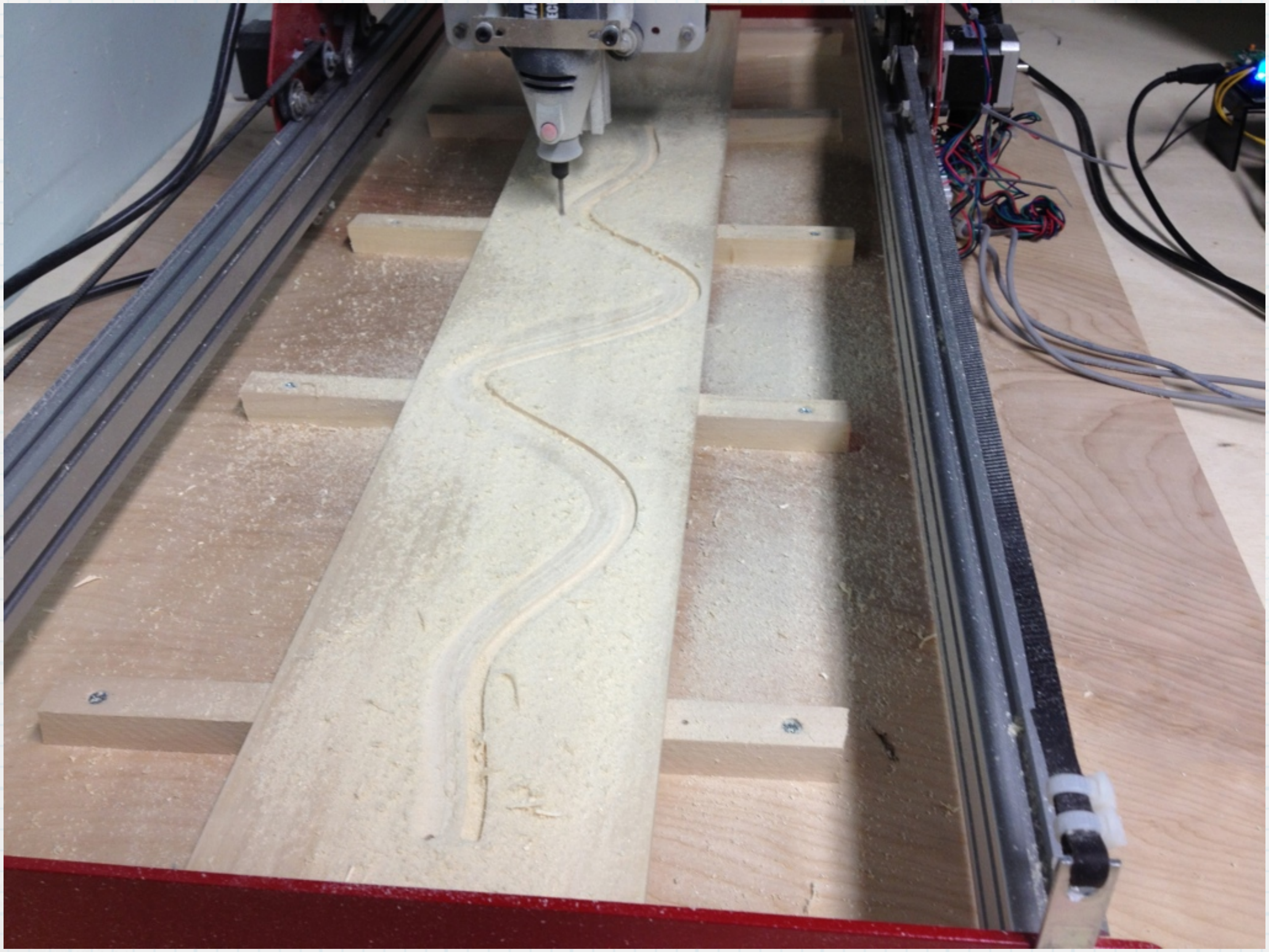
Milling



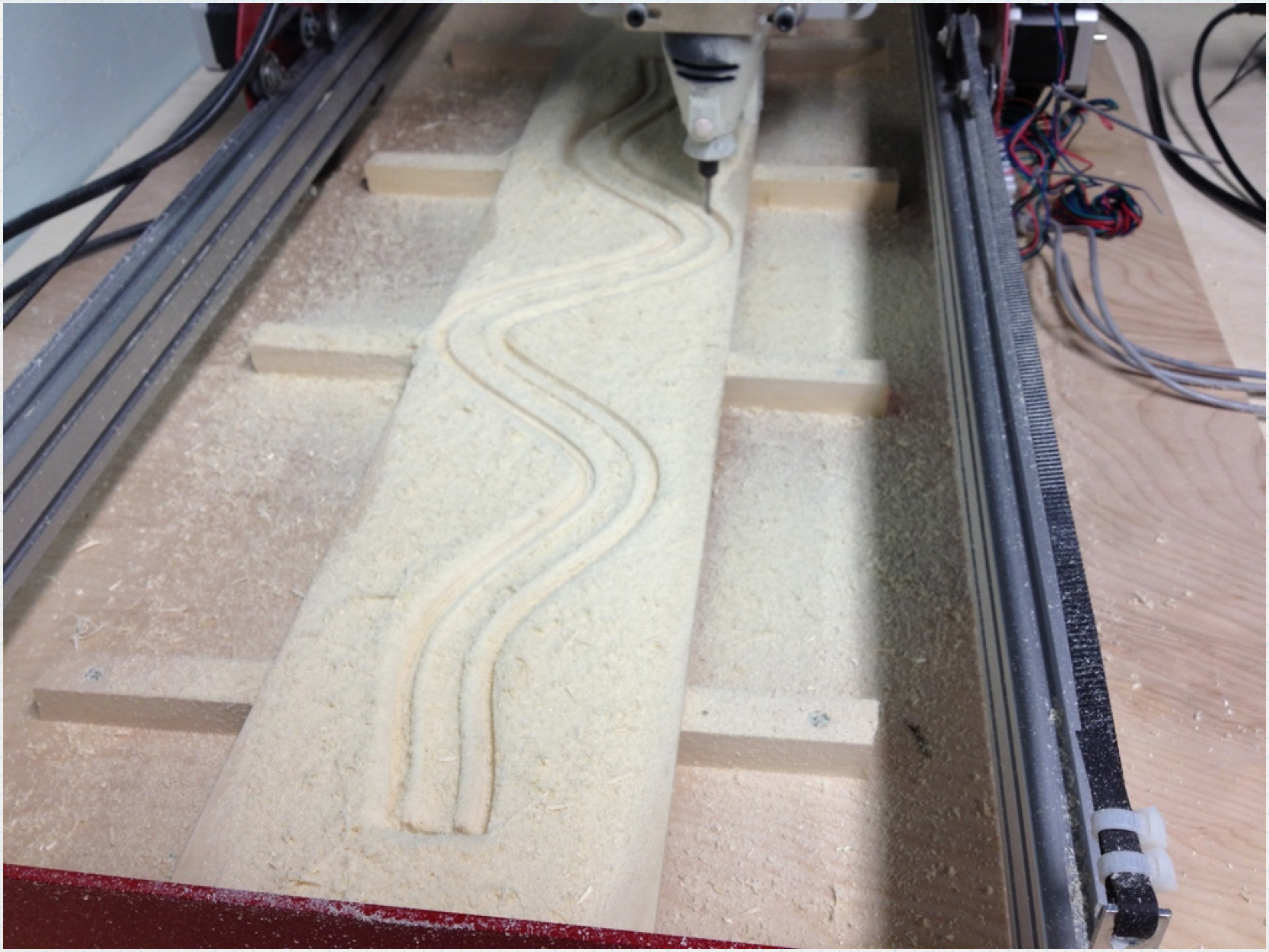
Milling



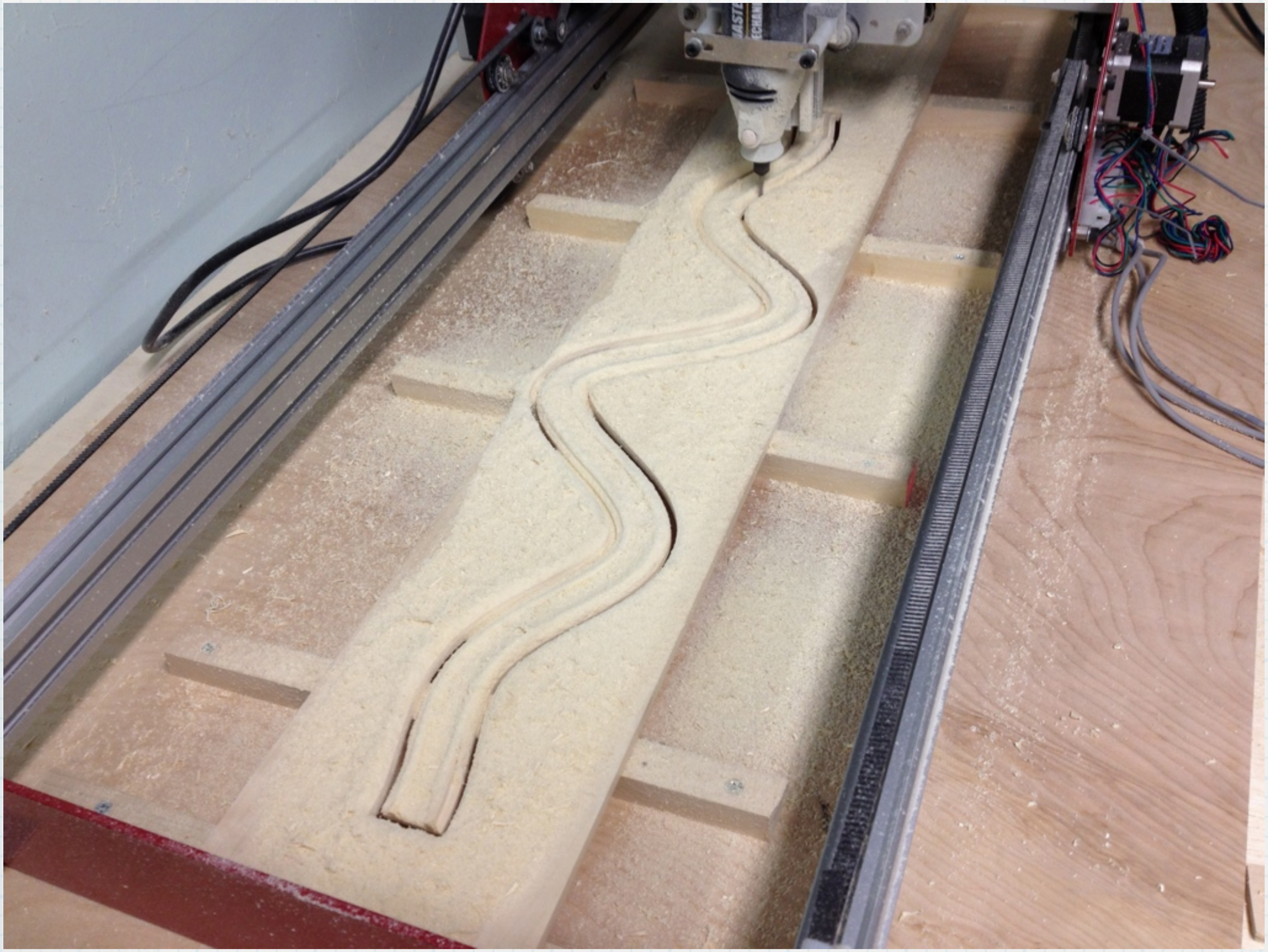
Milling



Milling



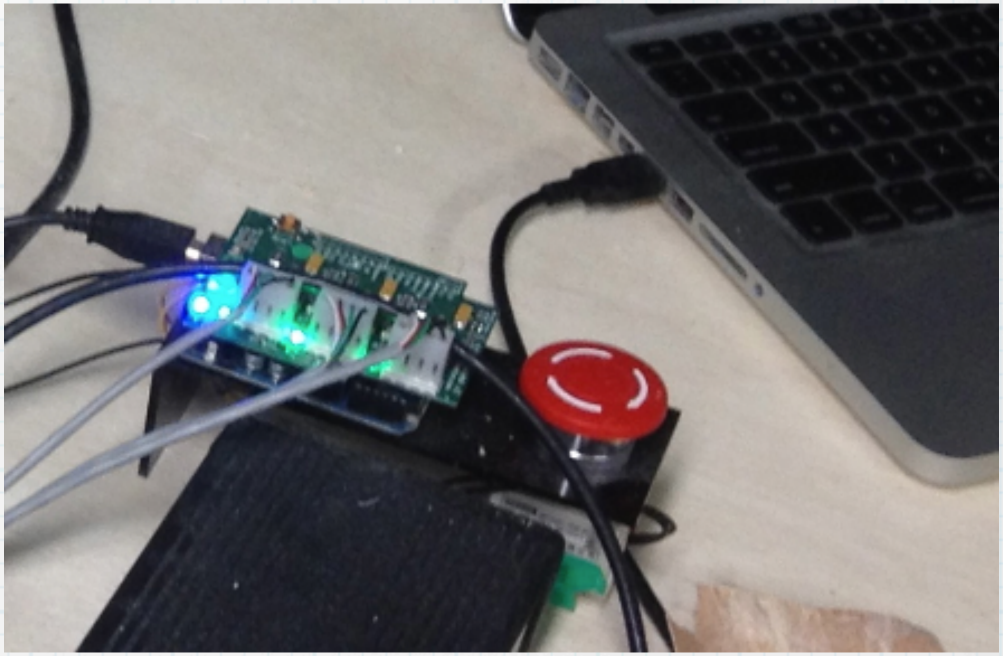
Milling



Rough Cut

Challenges

- Exceptions are common
- There's a "panic" button
- Often stop/resume
- Still working on software for it...



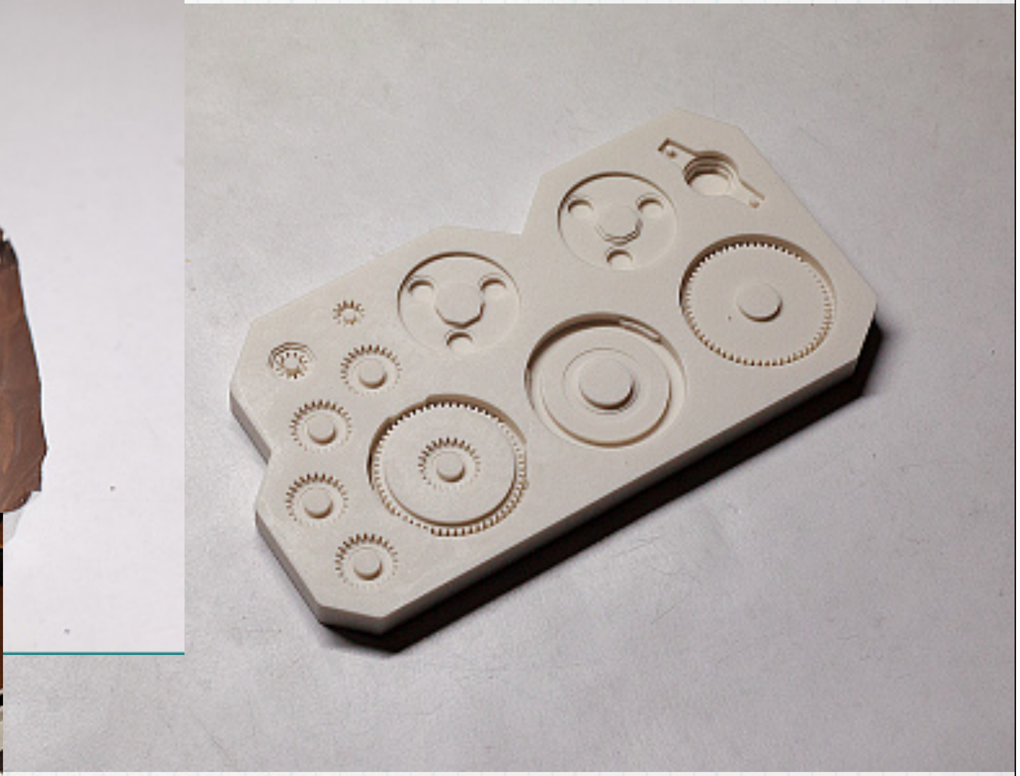
User Testing

Integration Testing

Variation

Ripple Track

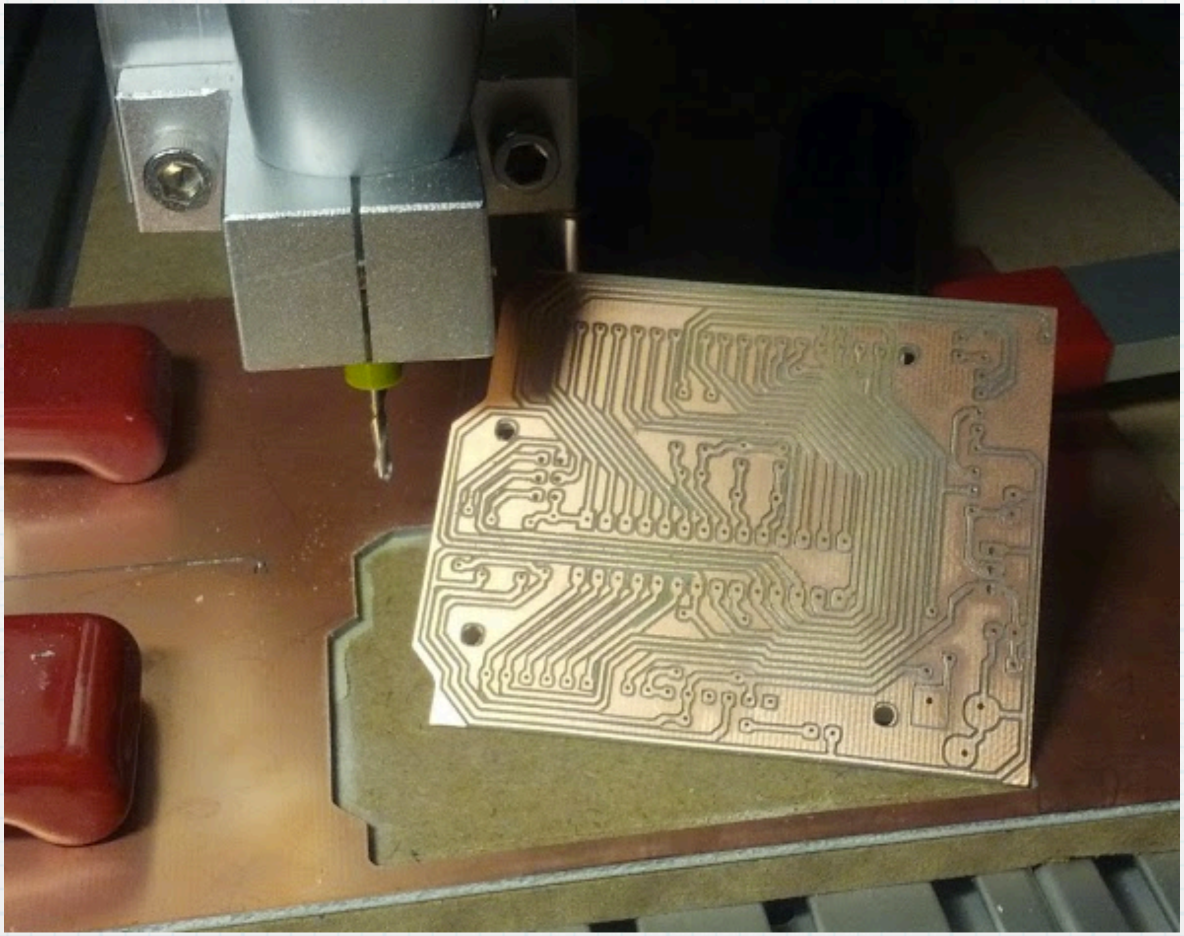
Future Direction?



Resin Casting

(Photos: Guerilla Guide to CNC)

Future Direction?



Printed Circuit Boards?

Is There a Point?

- CNC is a long-solved problem
- CAD/CAM is long-solved
- Why bother?

Is There a Point?

- CNC is a long-solved problem
- CAD/CAM is long-solved
- Why bother?

Answer: It's every bit as fun as my first computer...

Is There a Point?

- CNC is a long-solved problem
- CAD/CAM is long-solved
- Why bother?

Answer: It's every bit as fun as my first computer...

(and in 1978, programming was a solved problem).

Is There a Point?

- CNC is a long-solved problem
- CAD/CAM is long-solved
- Why bother?

Answer: It's every bit as fun as my first computer...

(and in 1978, programming was a solved problem).

And nothing shouts out "fun" like a JCL script.

It's Supposed To Be Fun

Why did you start using Python?

... it's okay to admit it. It was probably because it was fun.

(if it's not fun, you're not doing it right)

Bigger Point?

"Every child is an artist. The problem is how to remain an artist when we grow up."

- Pablo Picasso

s/artist/hacker/

Final Comments

- You can do this!
- ShapeOko (http://www.shapeoko.com)
- Guerrilla Guide to CNC (http://lcamtuf.coredump.cx/gcnc/)
- Thanks!
- Follow at @dabeaz