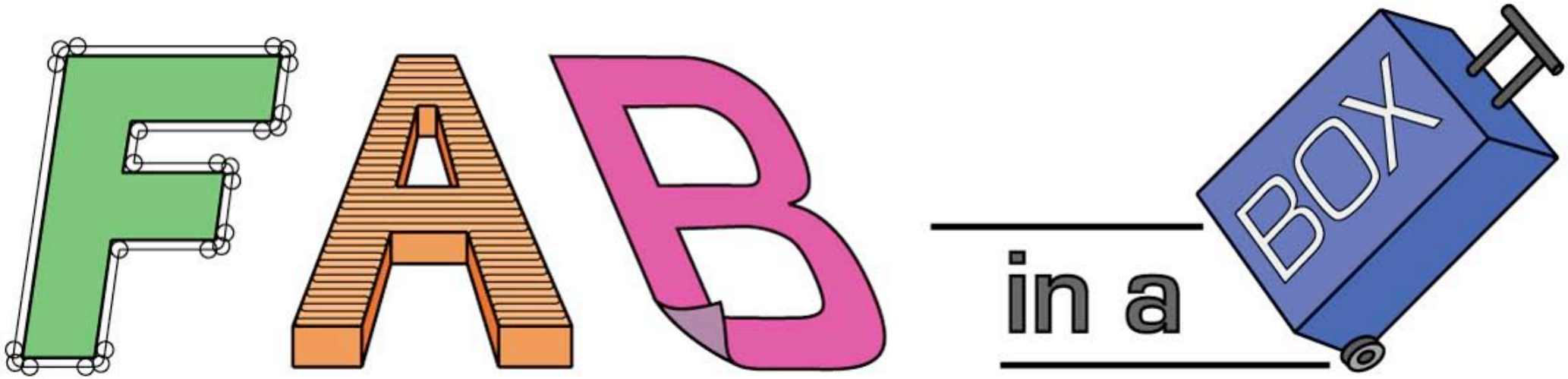


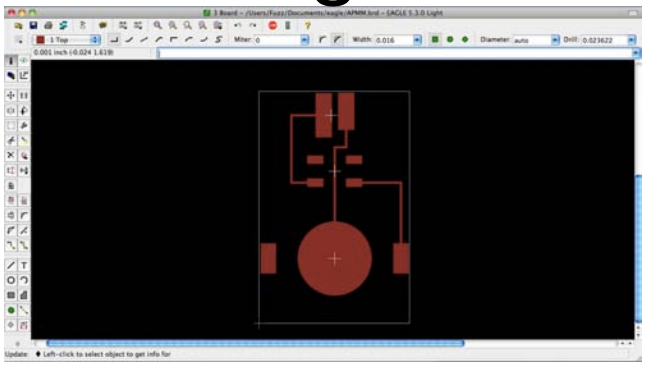
Fab Academy: Machine Design

Ilan Moyer

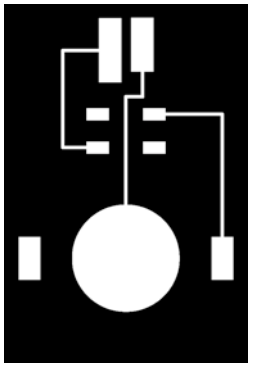
4/7/2010

FAB-IN-A-BOX: WORKFLOW

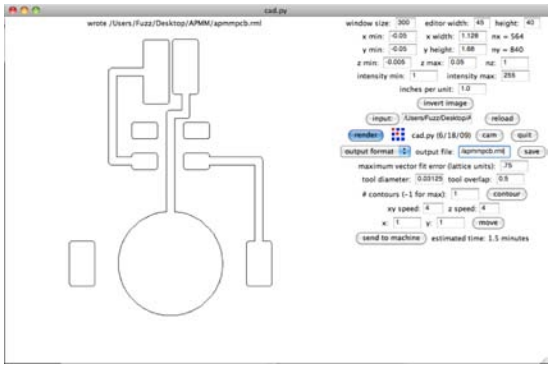
Eagle



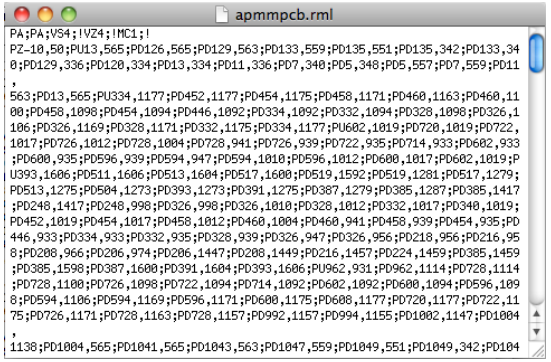
.PNG



CAD.PY



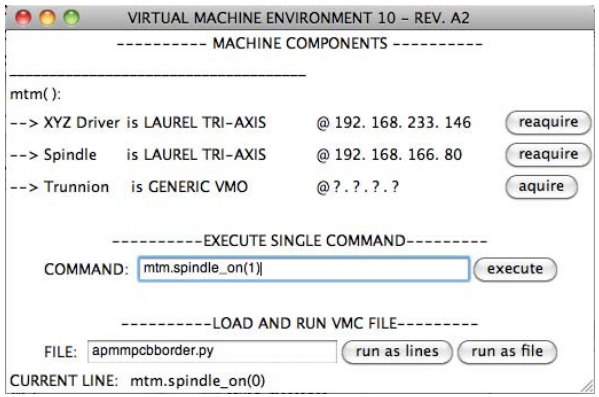
.RML



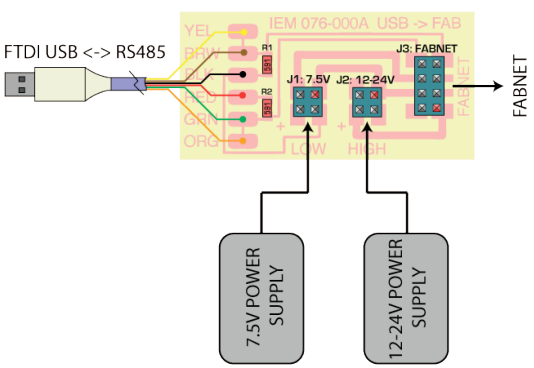
.PY

```
mtm.move( z = z_up, rate = retract_speed)
mtm.move(0.175,1.618, z_up, traverse_speed)
mtm.move( z = z_down, rate = plunge_speed)
mtm.open_group(0)
mtm.move(0.866,1.618, z_down, cutting_speed)
mtm.move(0.868,1.616, z_down, cutting_speed)
mtm.move(0.88,1.616, z_down, cutting_speed)
```

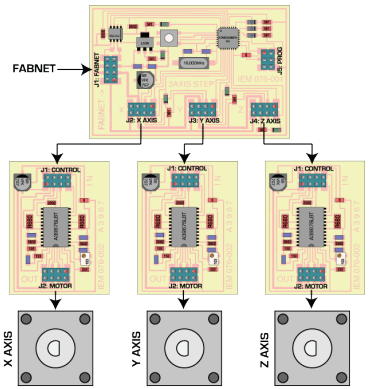
Virtual Machine Environment



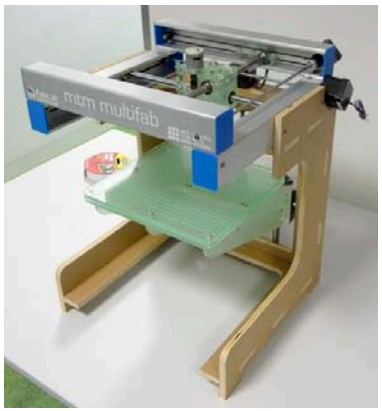
FABNET



Stepper Control

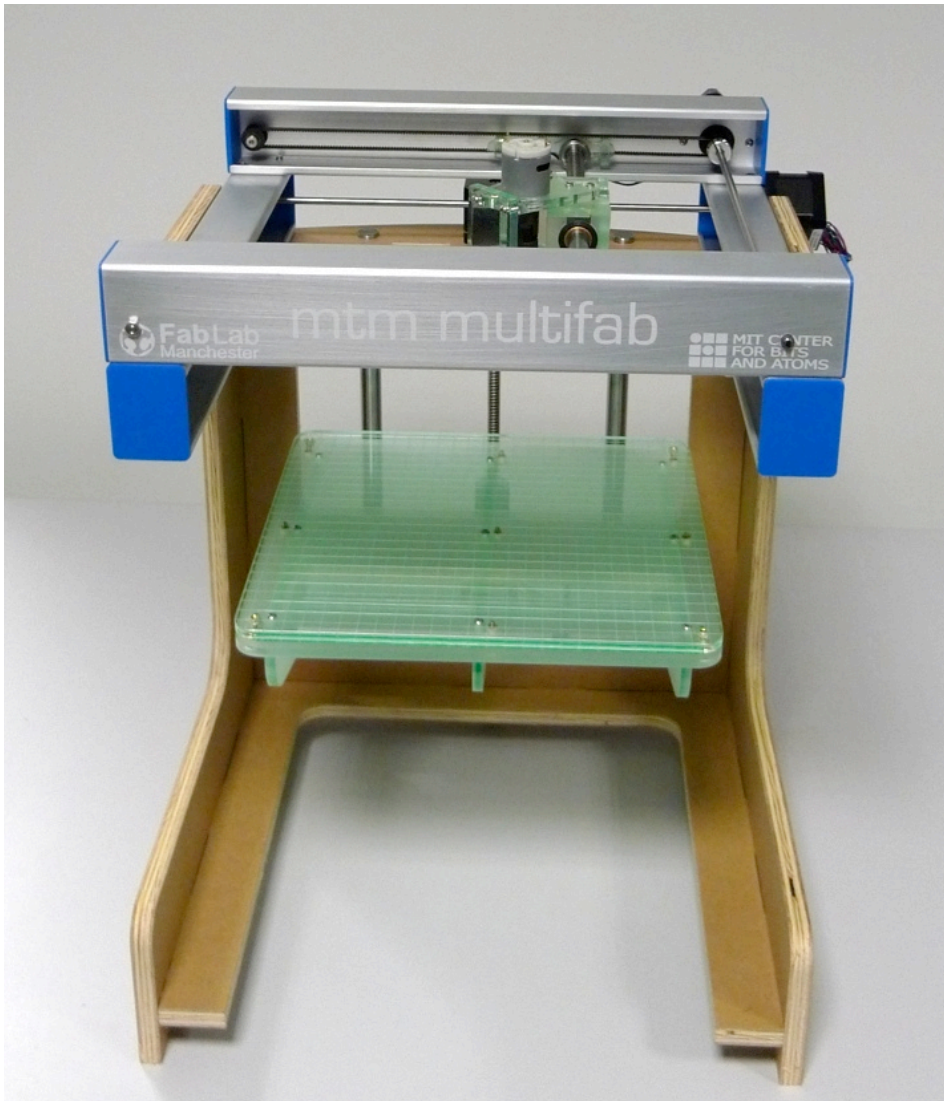


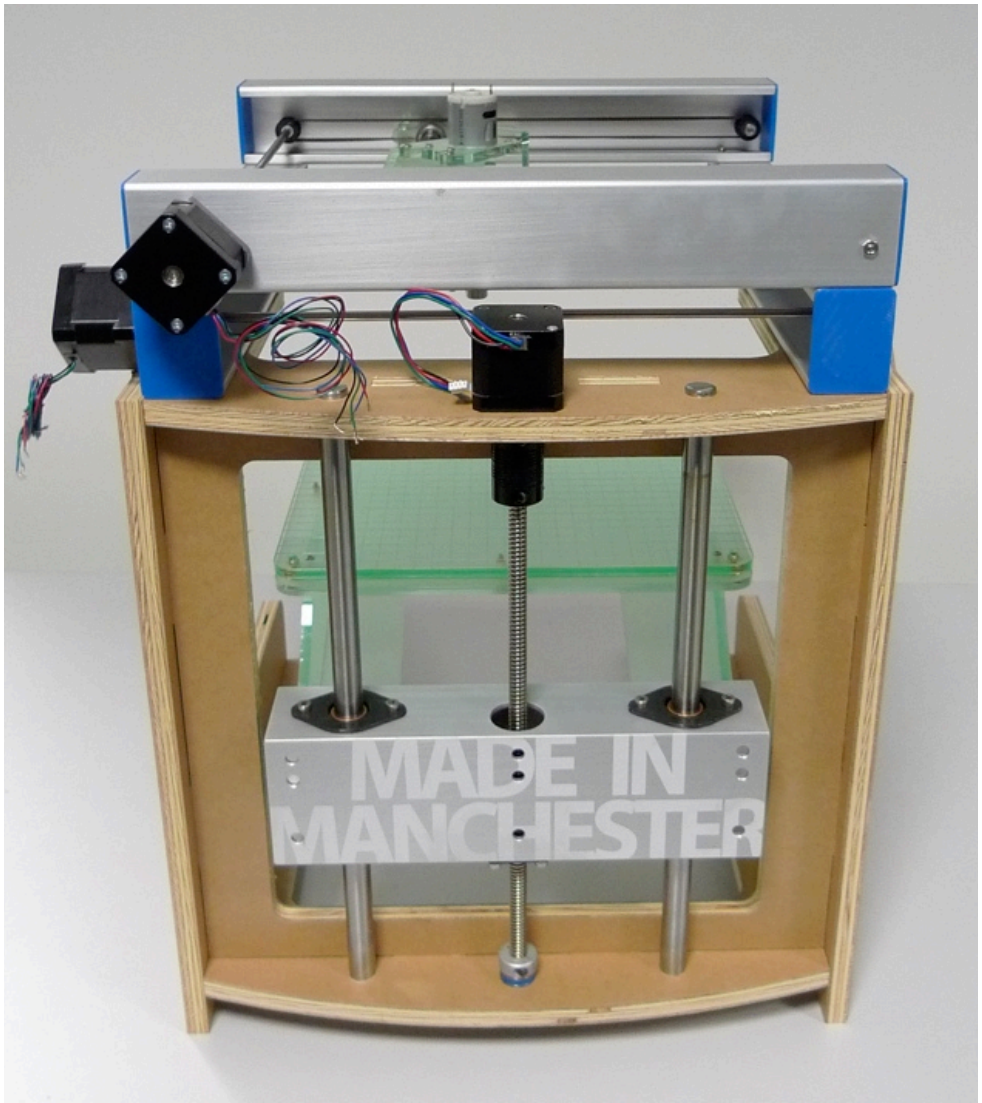
multifab



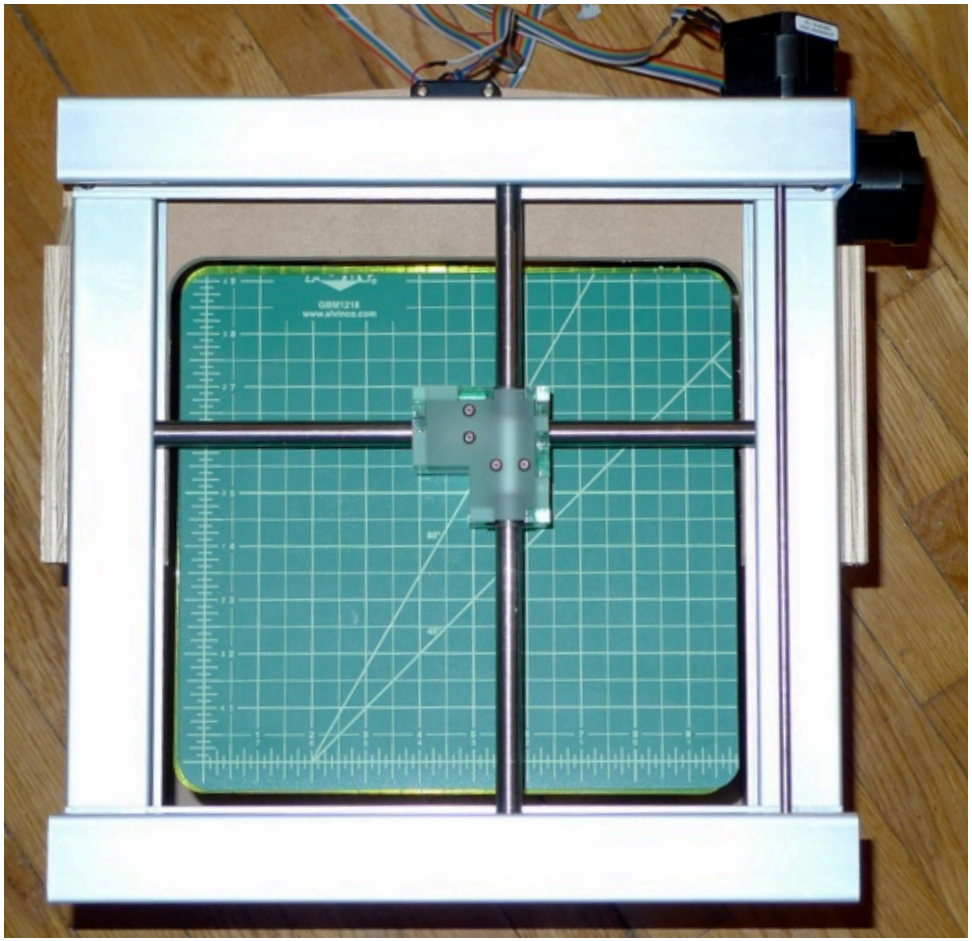
MTM MULTIFAB

Ilan Moyer and Maxim Lobovsky

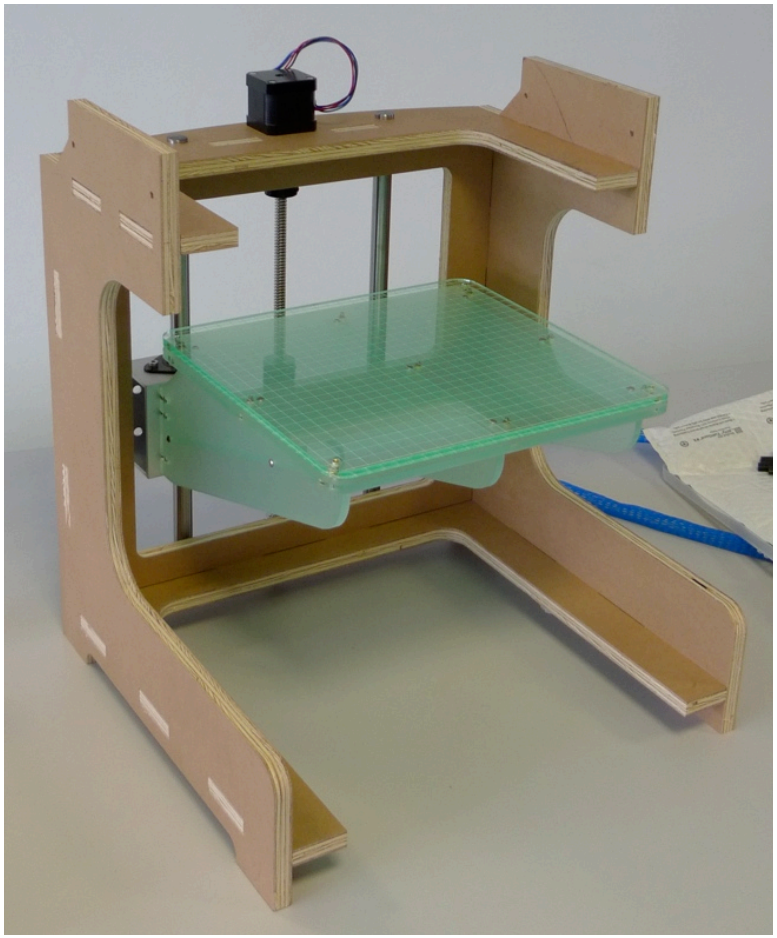




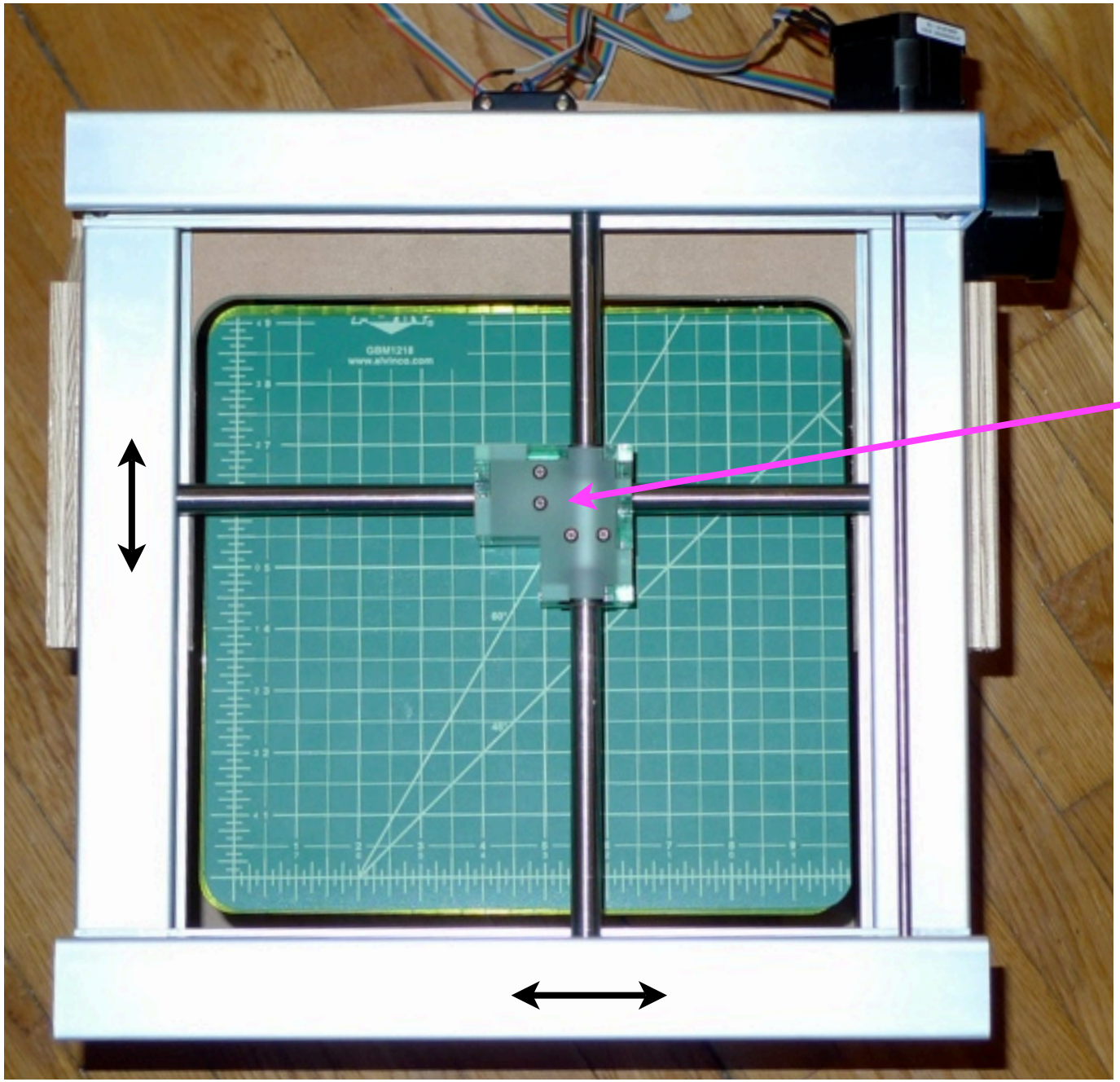
XY STAGE



Z FRAME

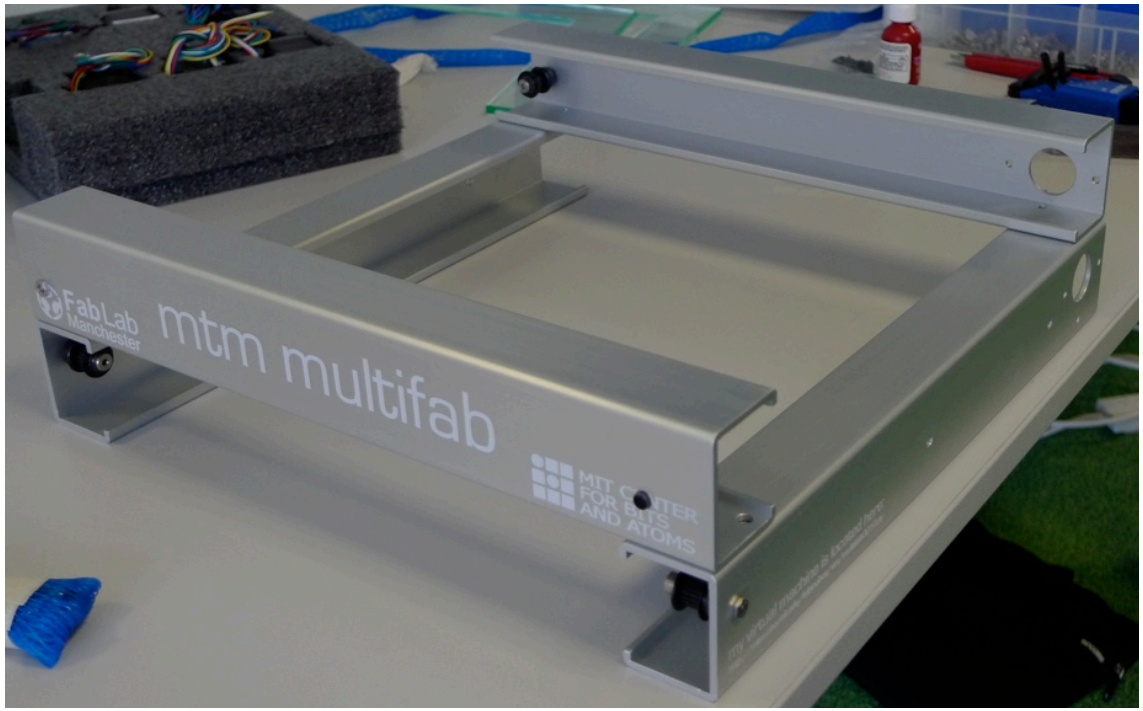


THE XY STAGE: KINEMATICS

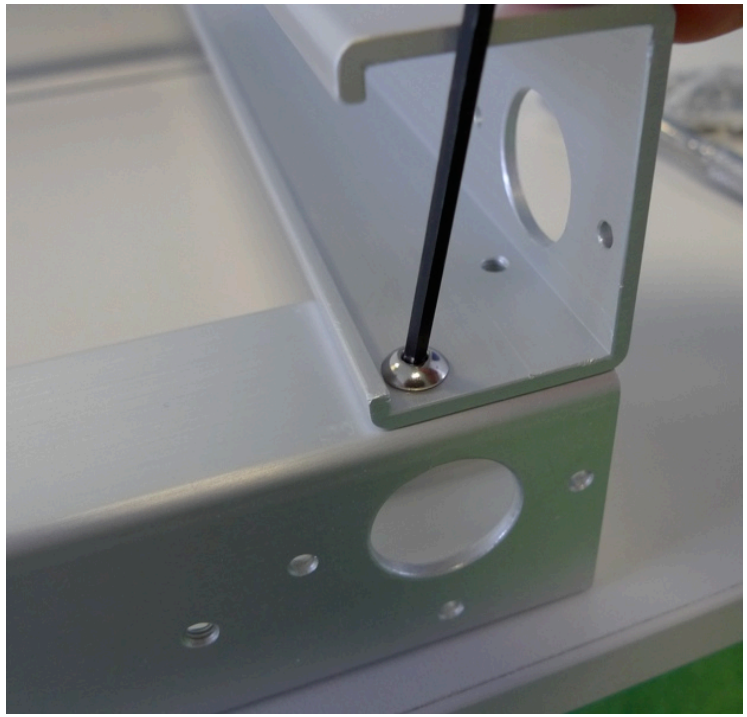


THE XY STAGE: STRUCTURE

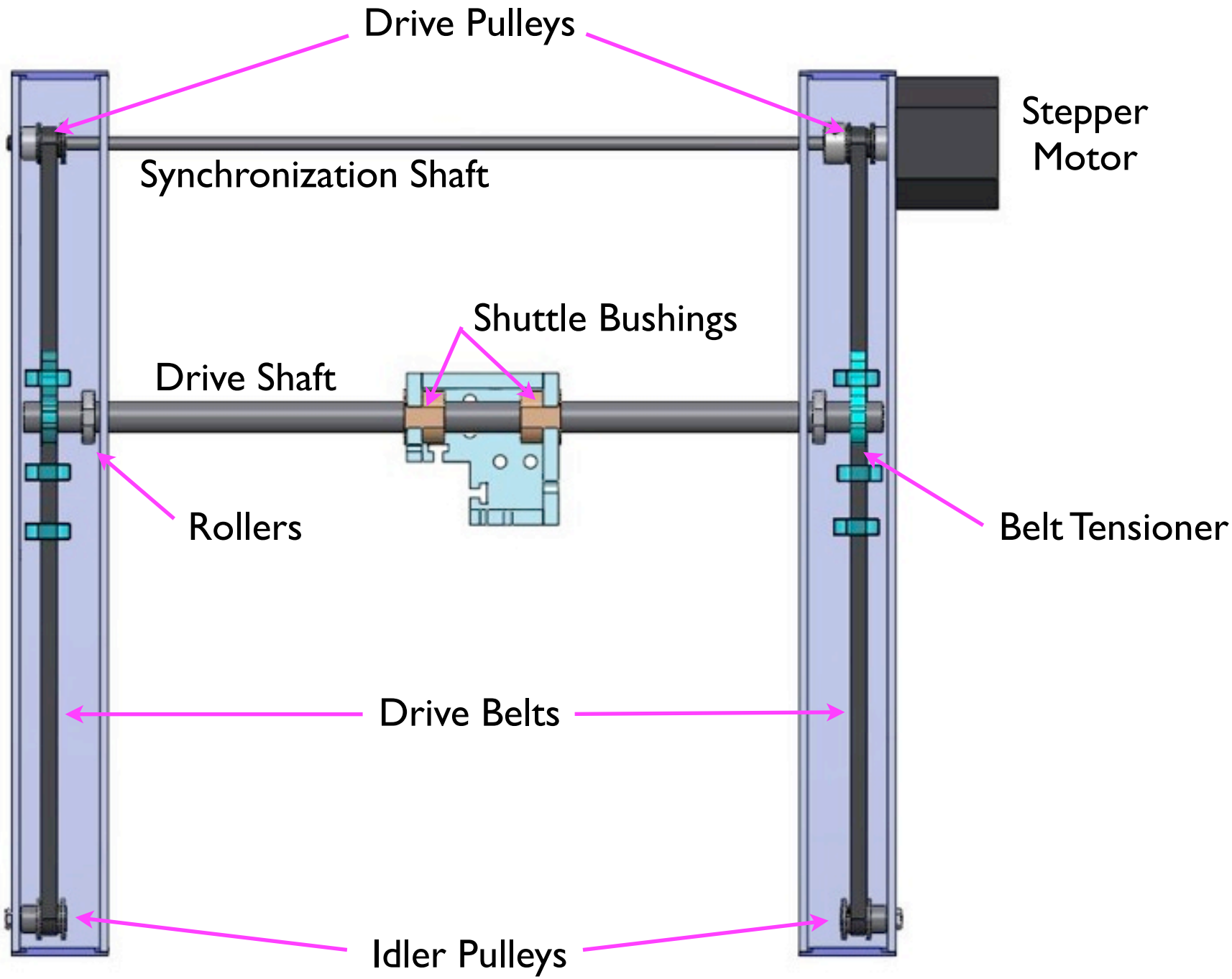
Four C-Chanel Extrusions



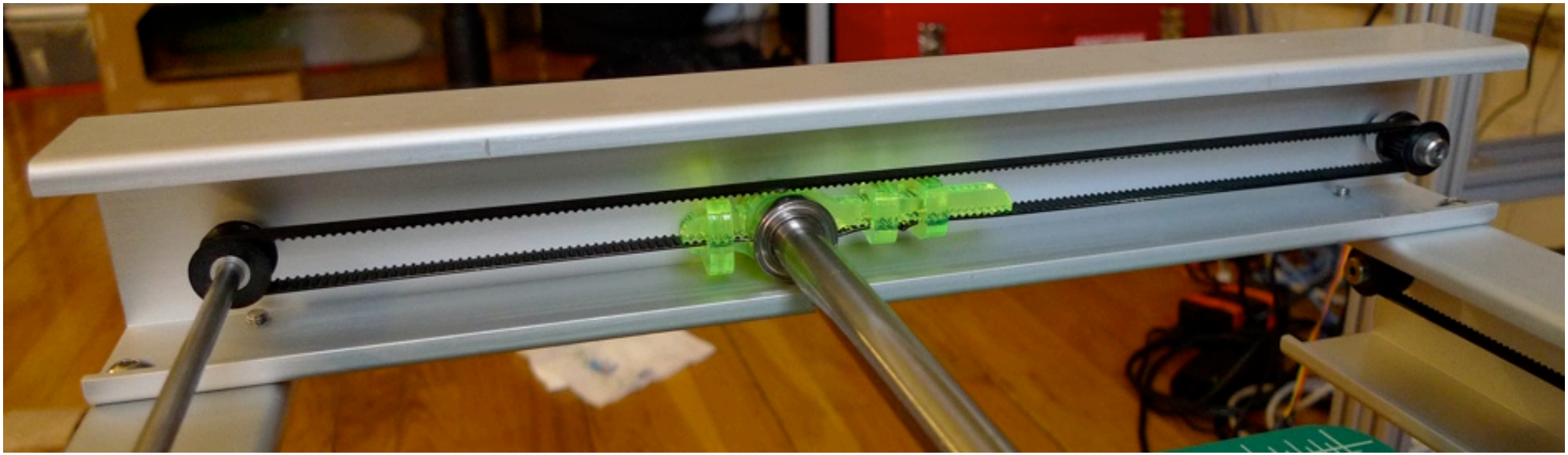
Symmetric Bolt Pattern

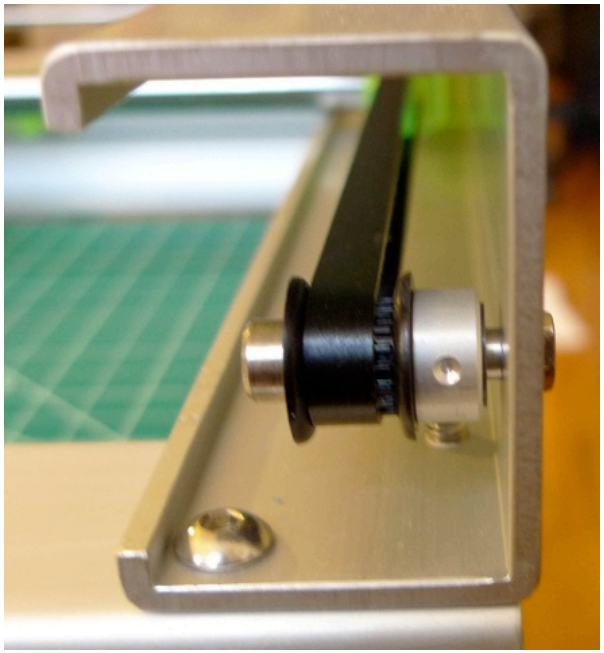


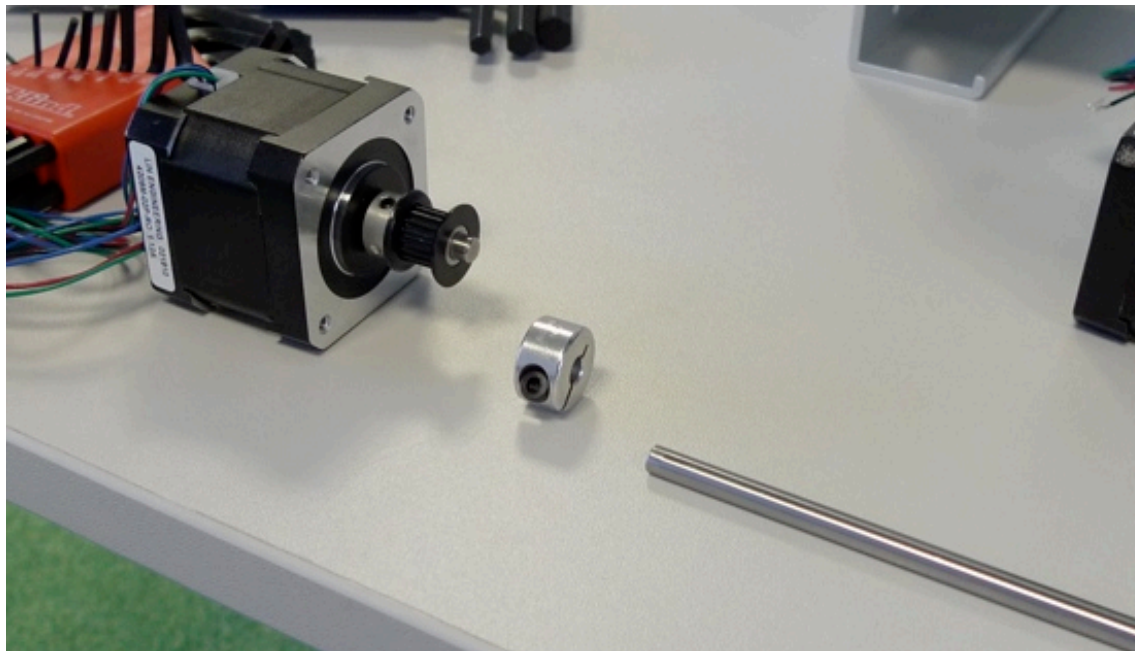
THE XY STAGE: MECHANISM



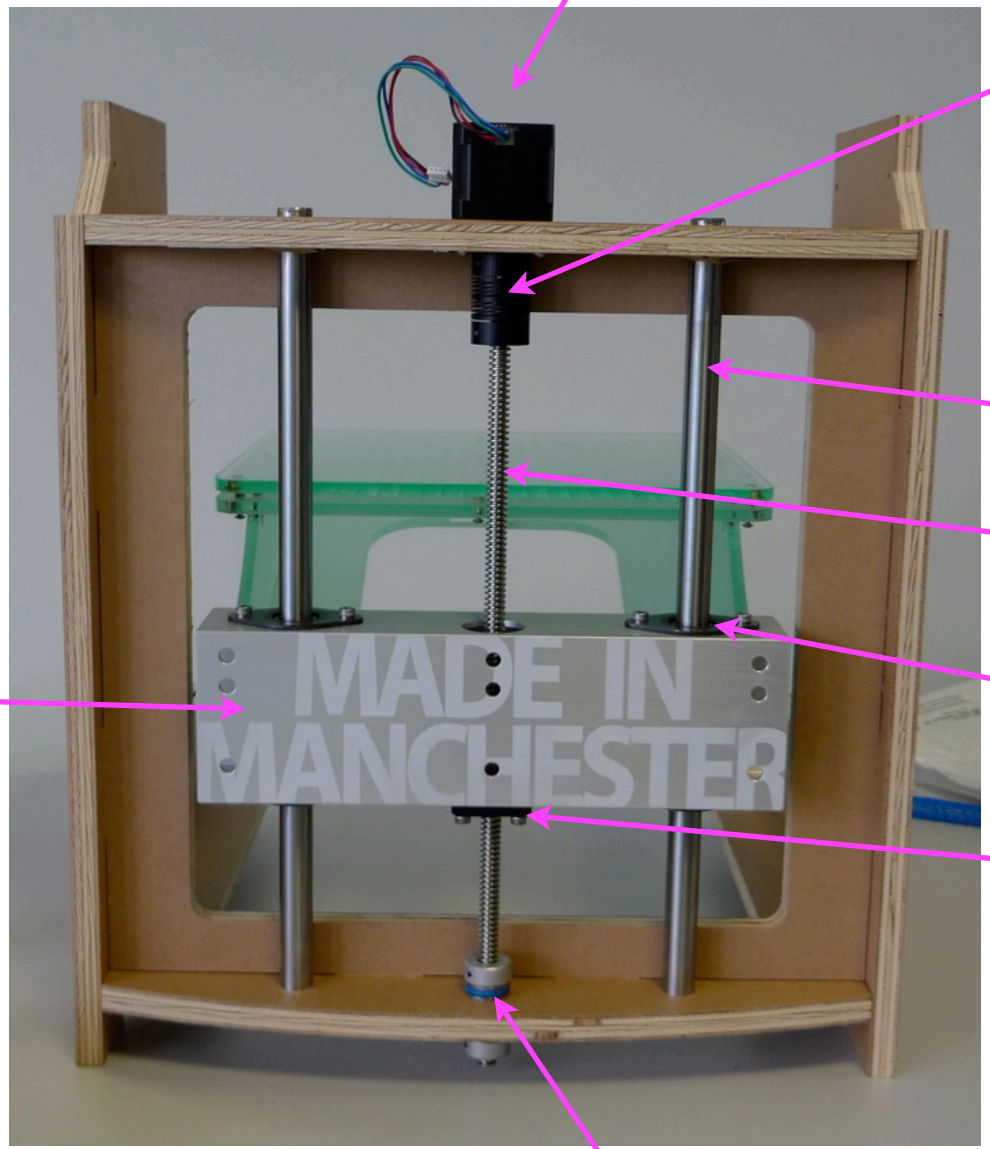
THE XY STAGE: MECHANISM







THE Z AXIS: DRIVE MECHANISM



Stepper Motor

Helical Beam Coupler

Guide Rail (x2)

Leadscrew

Bushings

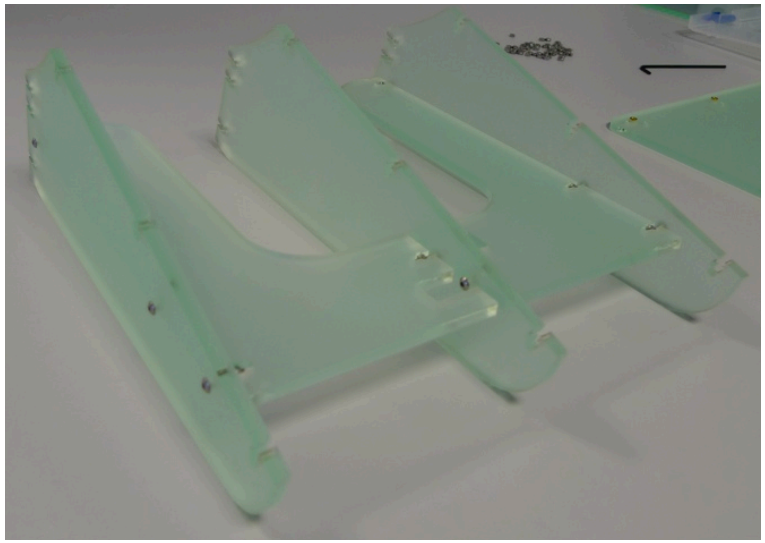
Tube

Leadnut

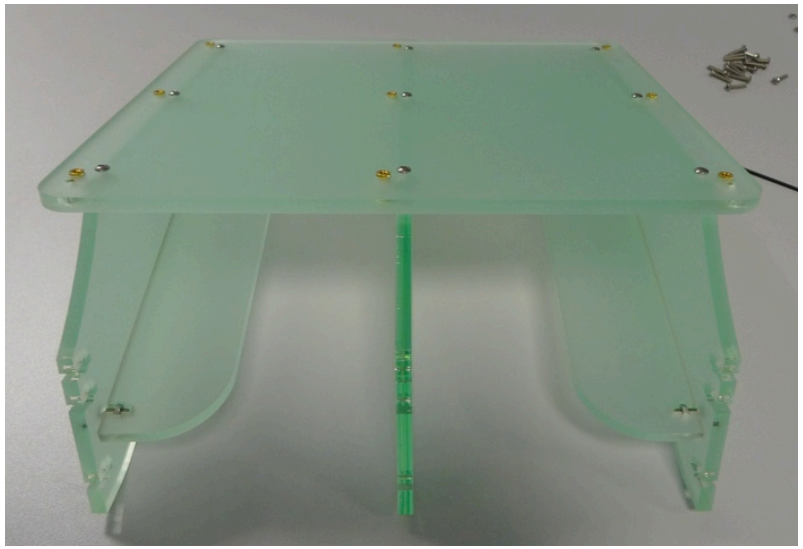
Thrust Bearing Assy

THE Z AXIS: TABLE

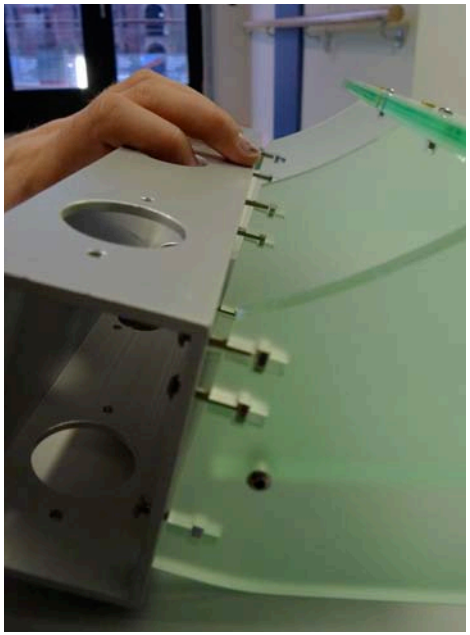
Sub-Frame



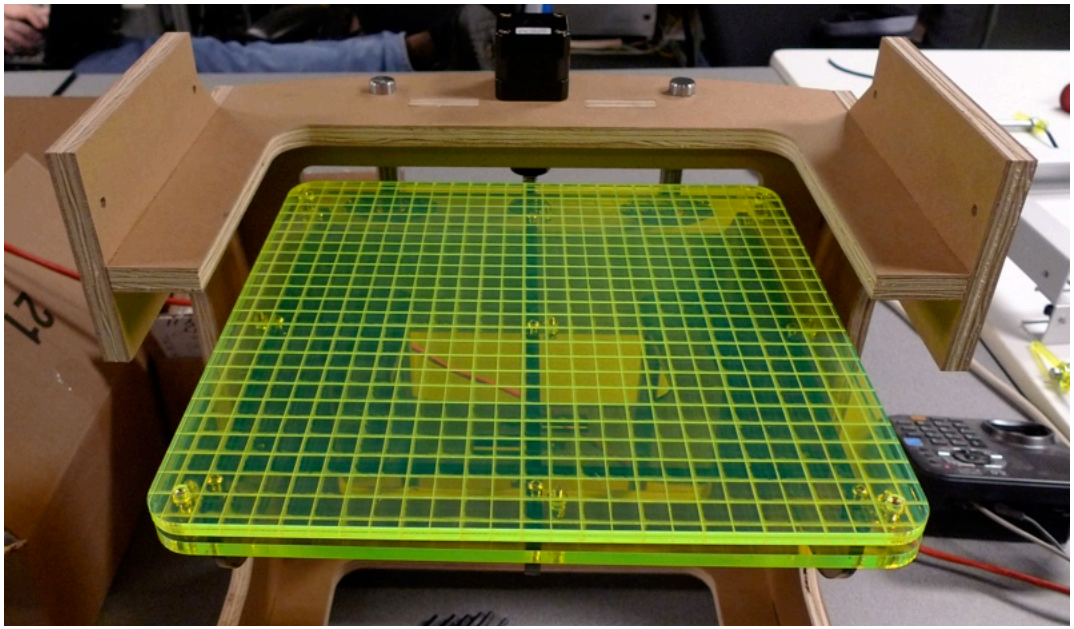
Sub-Table



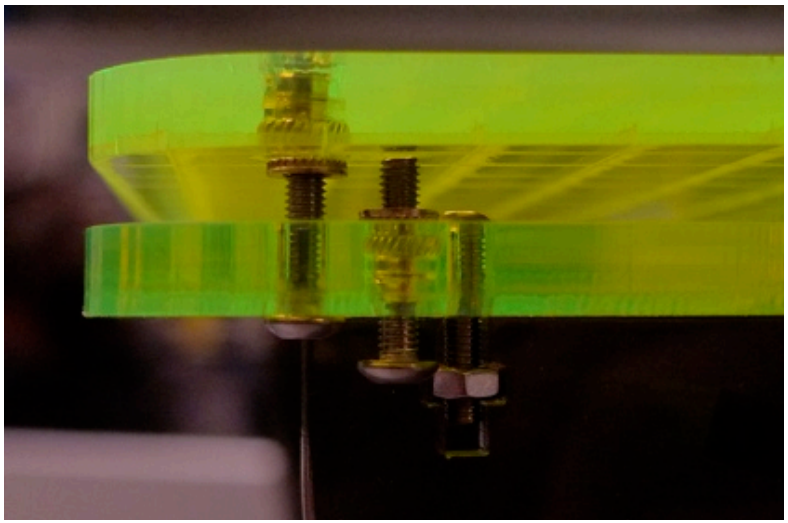
Mounting



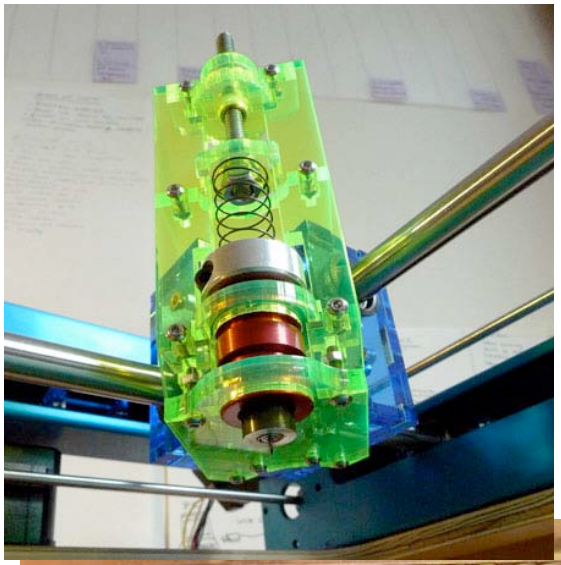
Working Surface

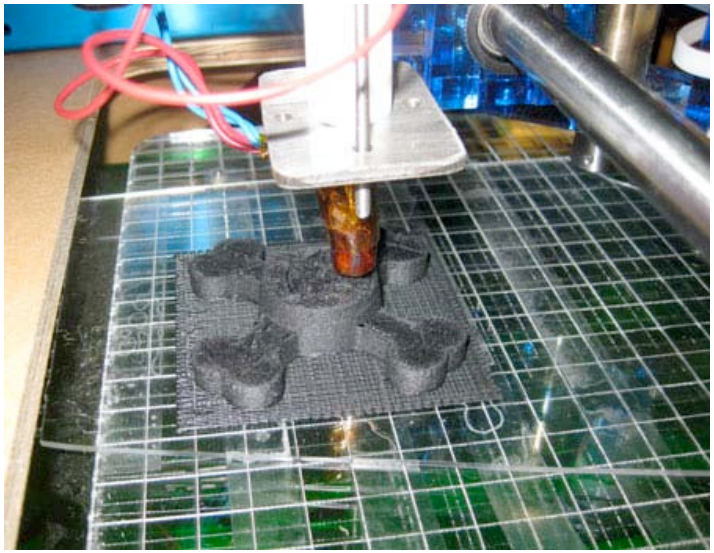


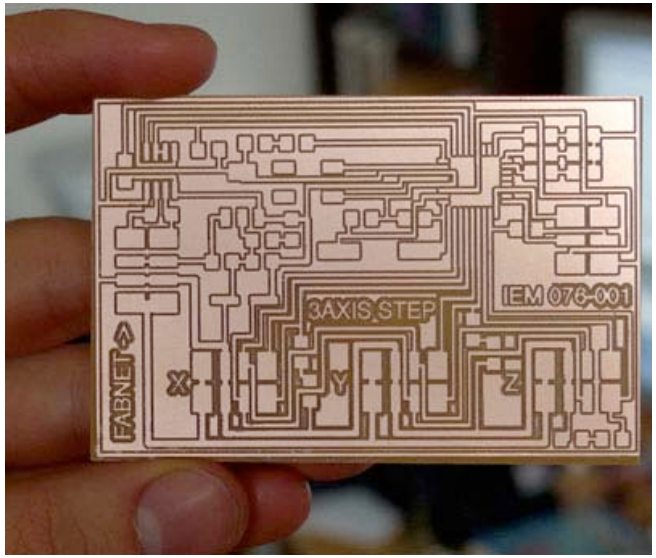
Leveling

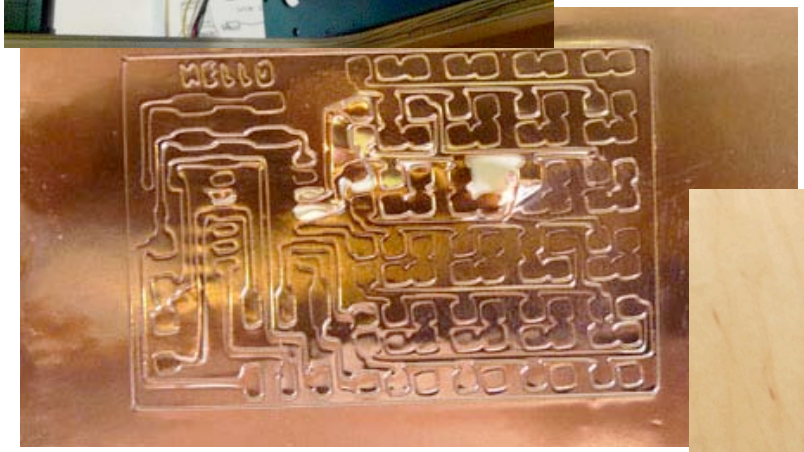


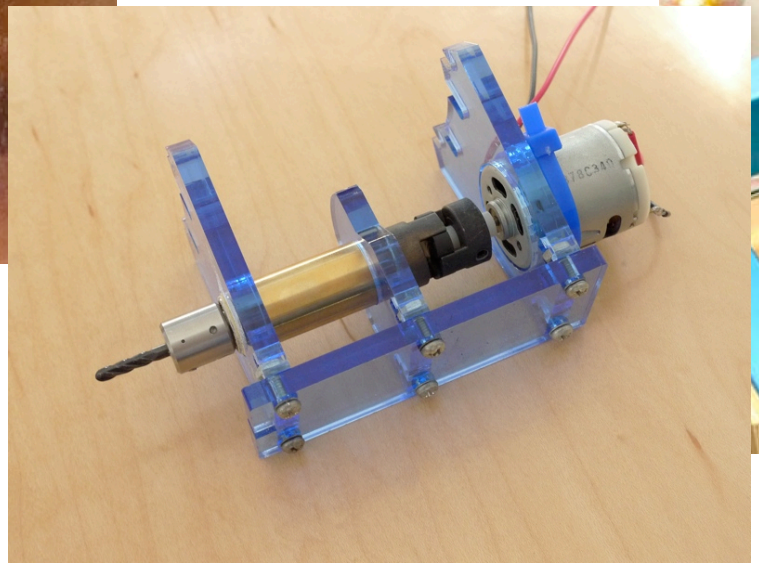
TOOL-HEADS AND CAPABILITIES

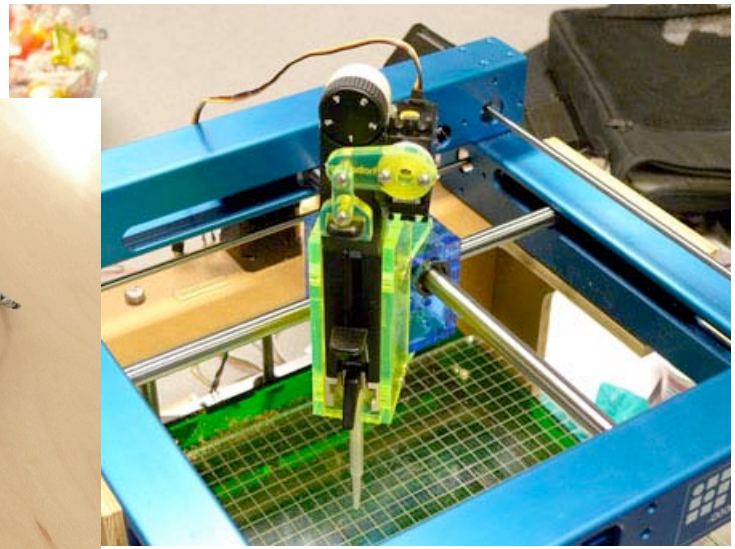




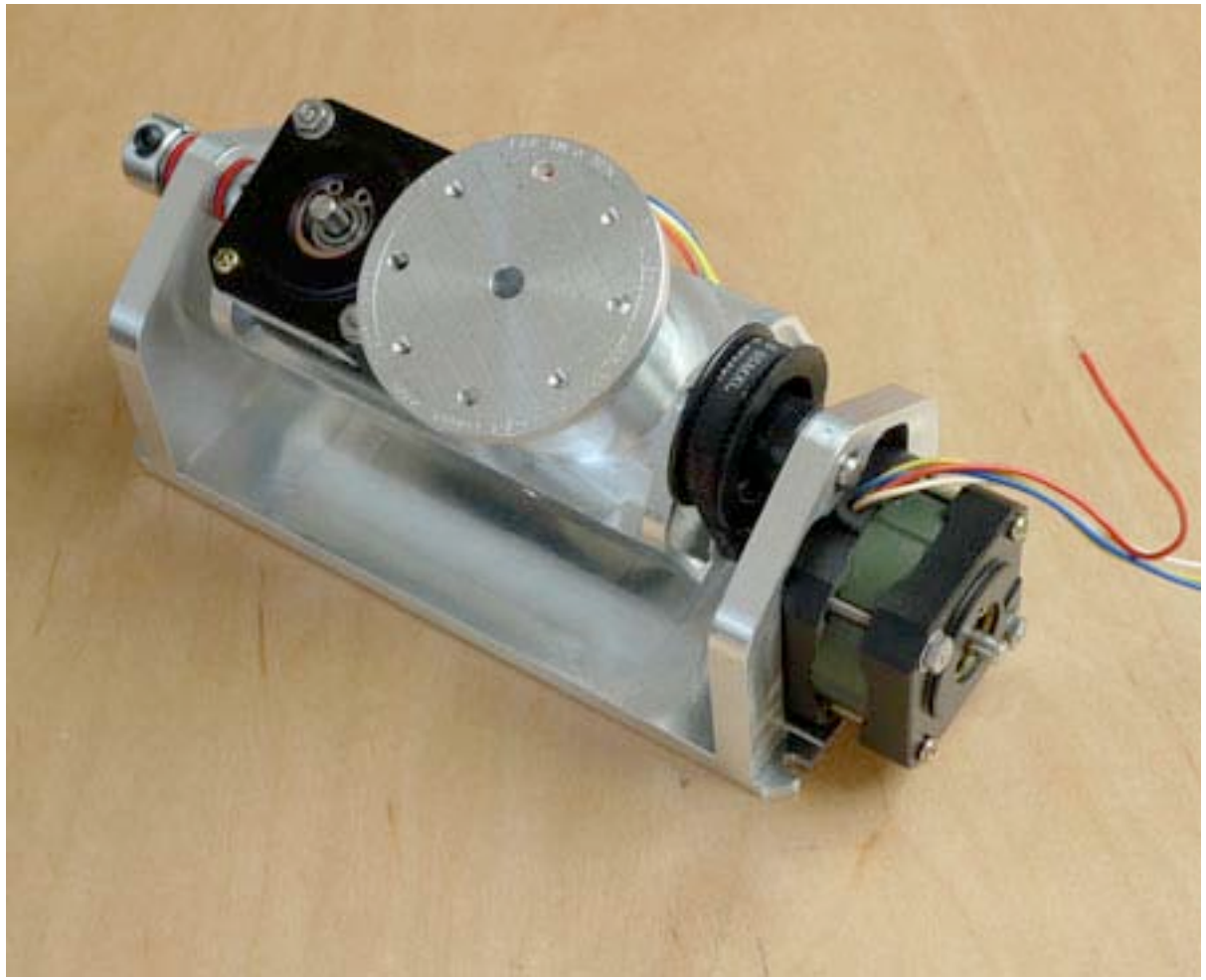








5 AXIS TRUNNION



STRENGTHS

- Low Inertia -> High Acceleration
- Direct Drive -> High Speed
- 200mm^3 Work Volume
- Fixed Table -> Tall 3D Prints

WEAKNESSES

- Low Stiffness -> No Heavy Milling
- Low Basic Resolution: 0.0035” (~0.0015 with half stepping.)

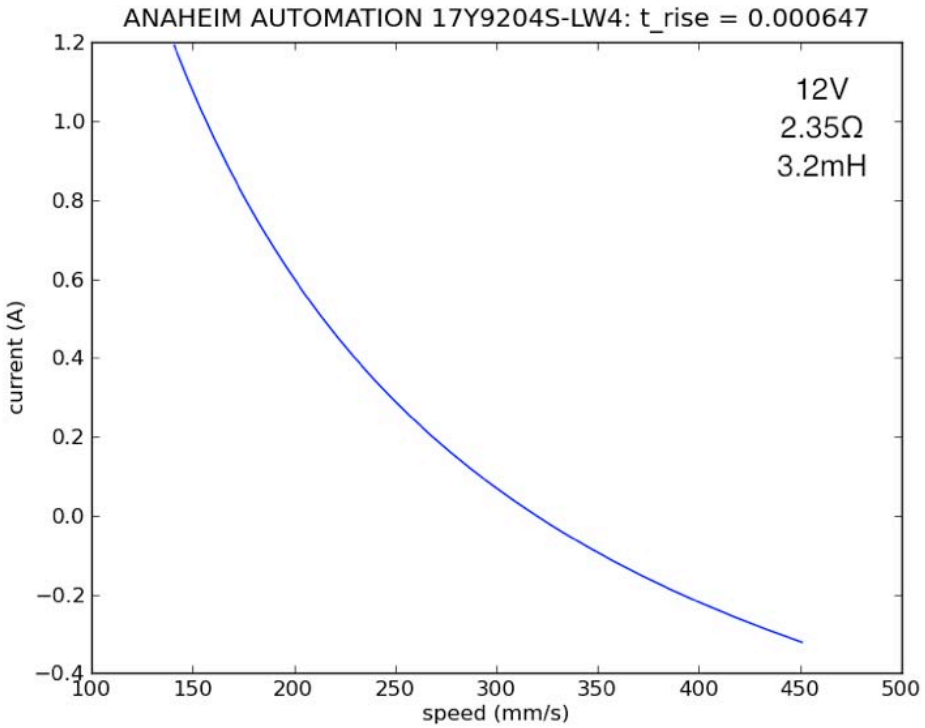
STEPPER MOTOR SELECTION

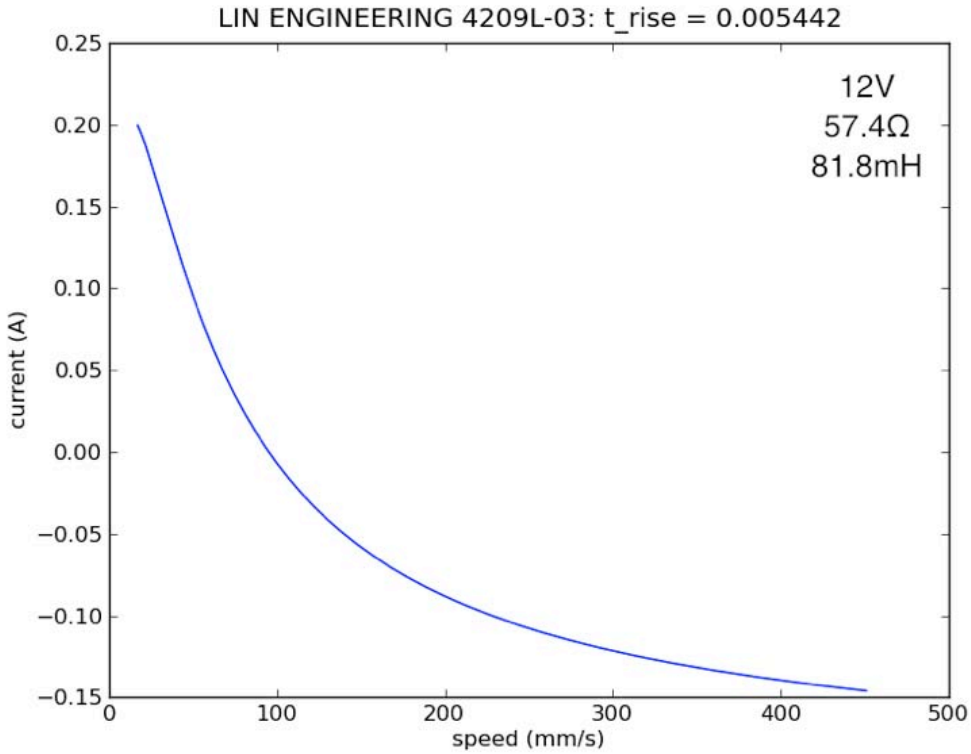
Lin Engineering 4209M-02P

Resistance = 2.35 Ohms
Inductance = 3.2 mHenries

Lin Engineering 4209L-03P

Resistance: 58 Ohms
Inductance: 82 mHenries

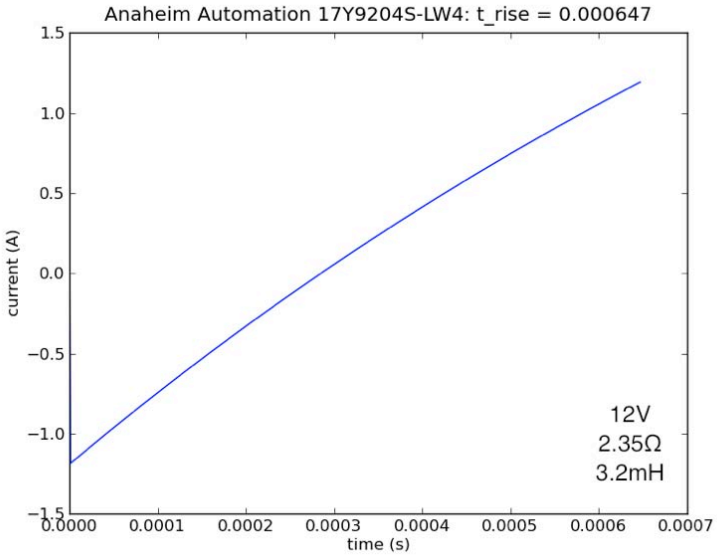




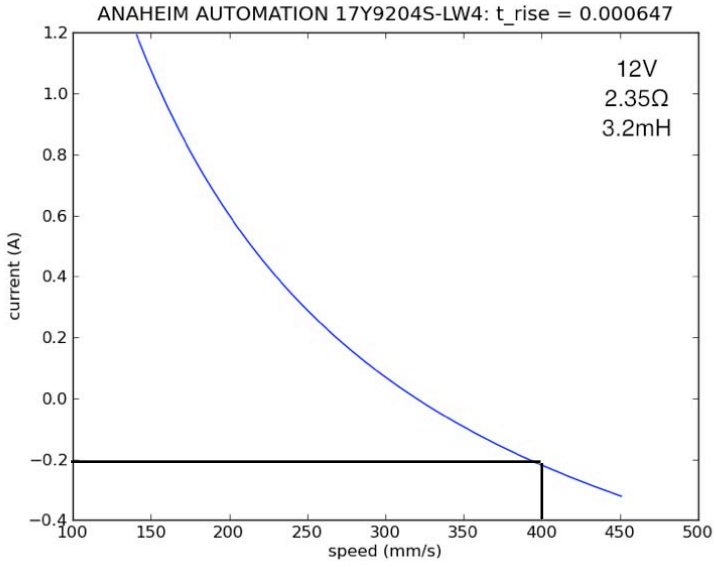
NOTE: t_rise is the time for motor current to reach 100%.

DRIVE VOLTAGE SELECTION

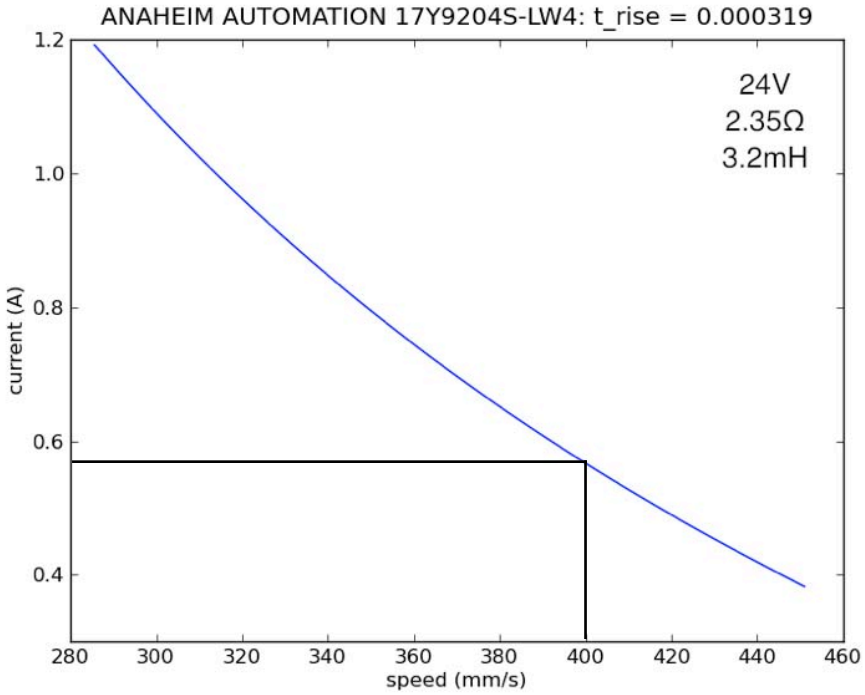
Current Rise Time: 12V



Max Speed: 12V
41% Power @ 400mm/s

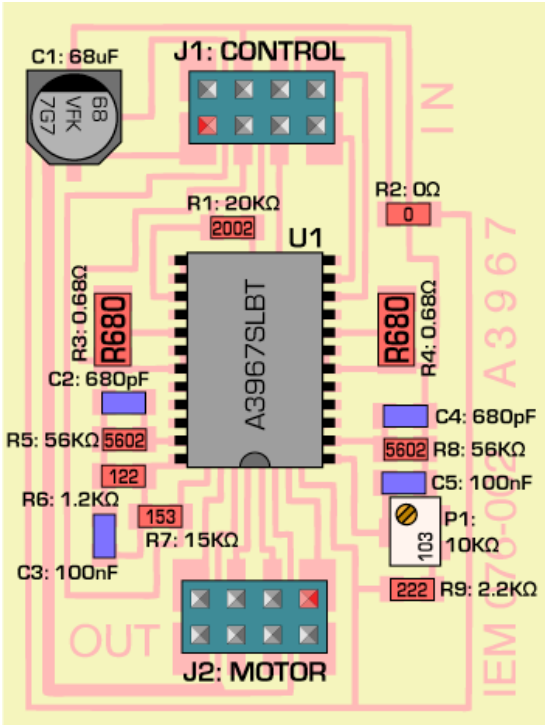


Max Speed: 24V
75% Power @ 400mm/s



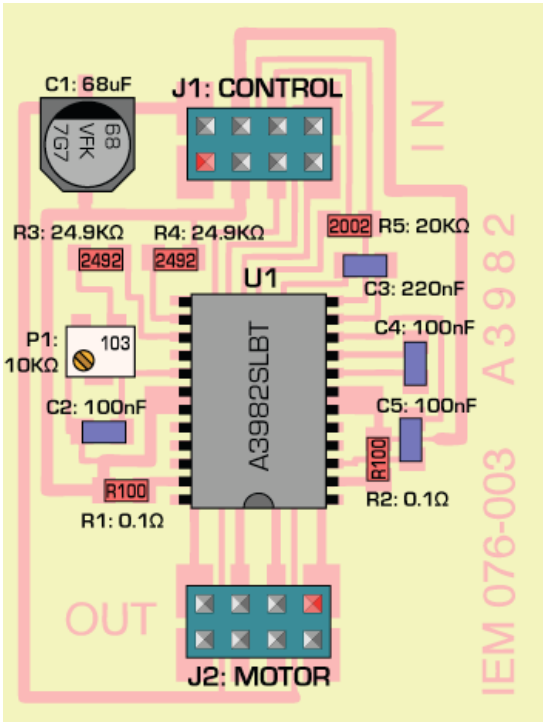
DRIVE ELECTRONICS

Allegro A3967



- 8 uSteps/Step
- 750mA/Phase

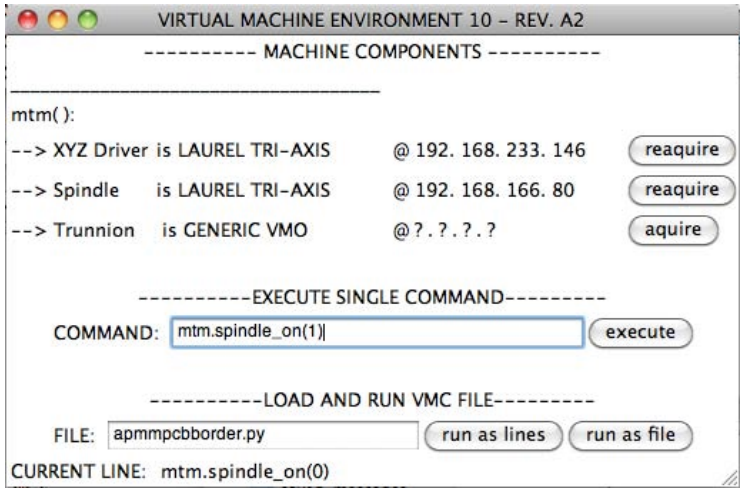
Allegro A3982



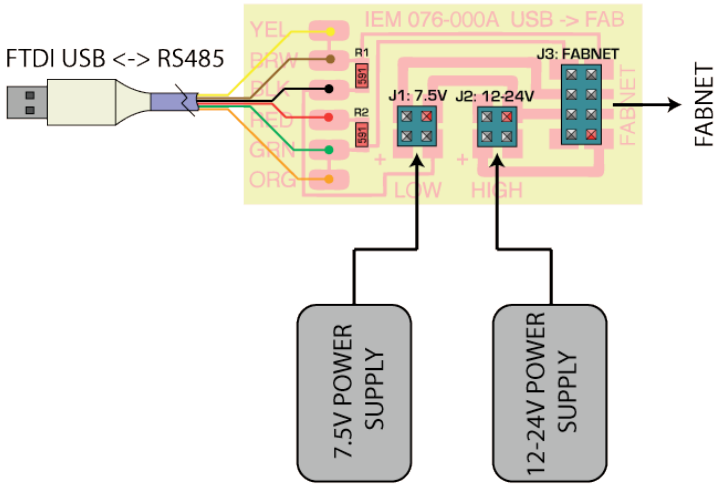
- 2 uSteps/Step
- 2A/Phase

THE NETWORK

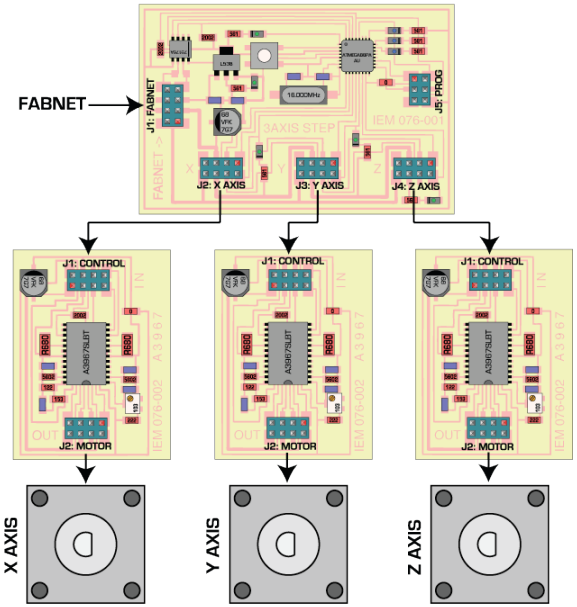
Virtual Machine Environment



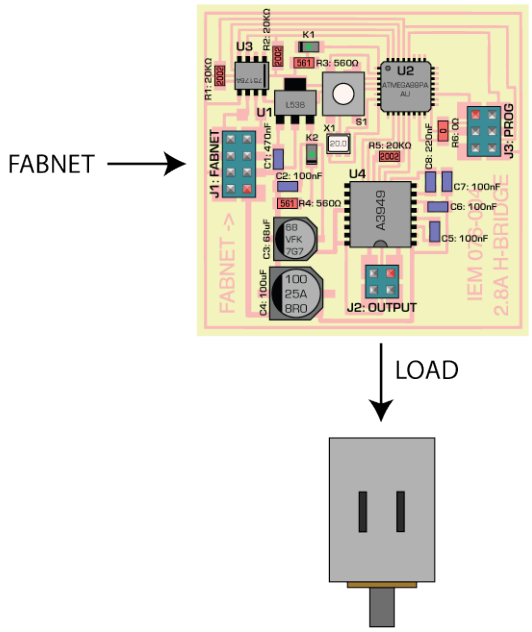
NETWORK + POWER



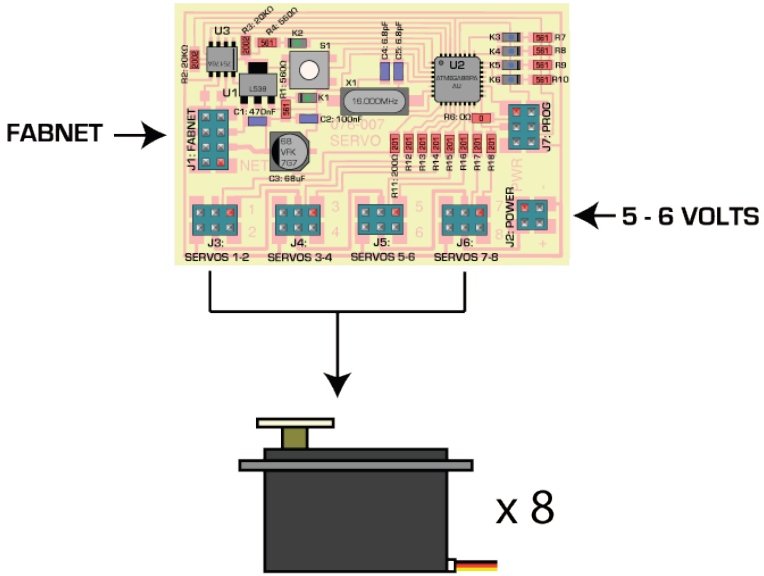
Stepper Motor Control



DC Motor Control



RC Servo Control



GENERAL PACKET FORMAT

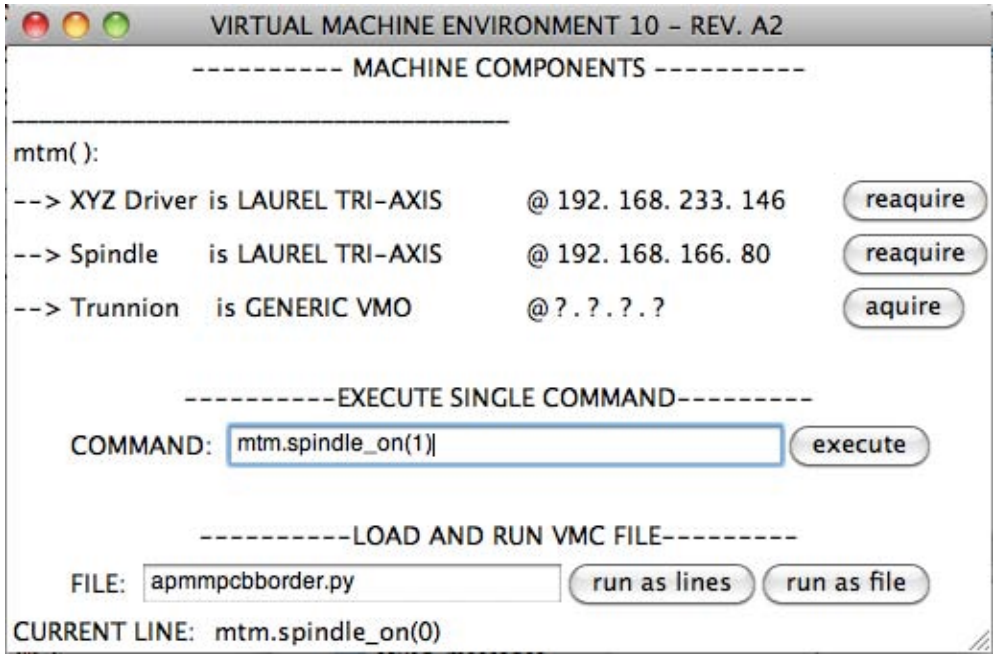
[
[UNICAST|MULTICAST]
[SOURCE IP][SOURCE PORT]
[DEST. IP][DEST PORT]
[LENGTH]
[----PAYLOAD----]
[CRC CHECKSUM]
]

MOTION CONTROL

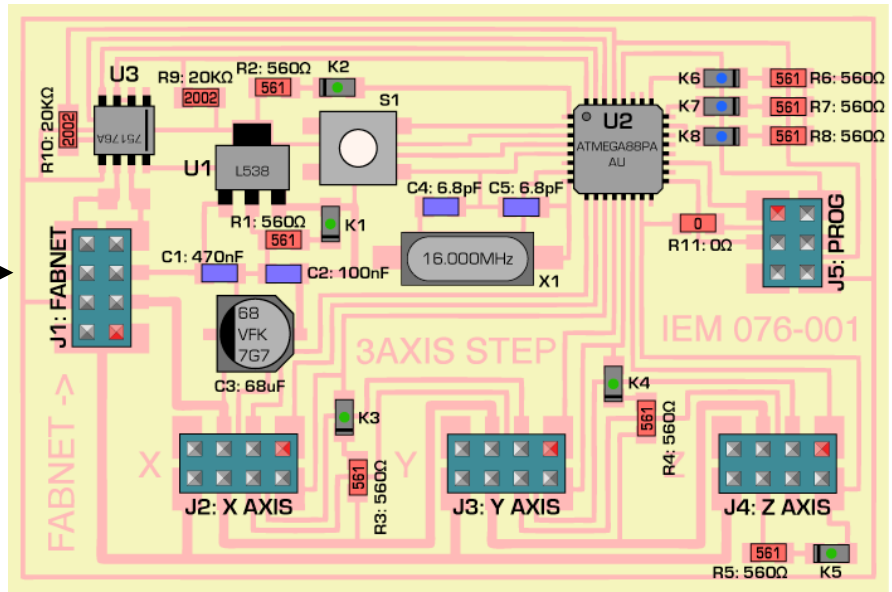
WWW

http://mtm.cba.mit.edu/fabinabox/vm/076-001e.py

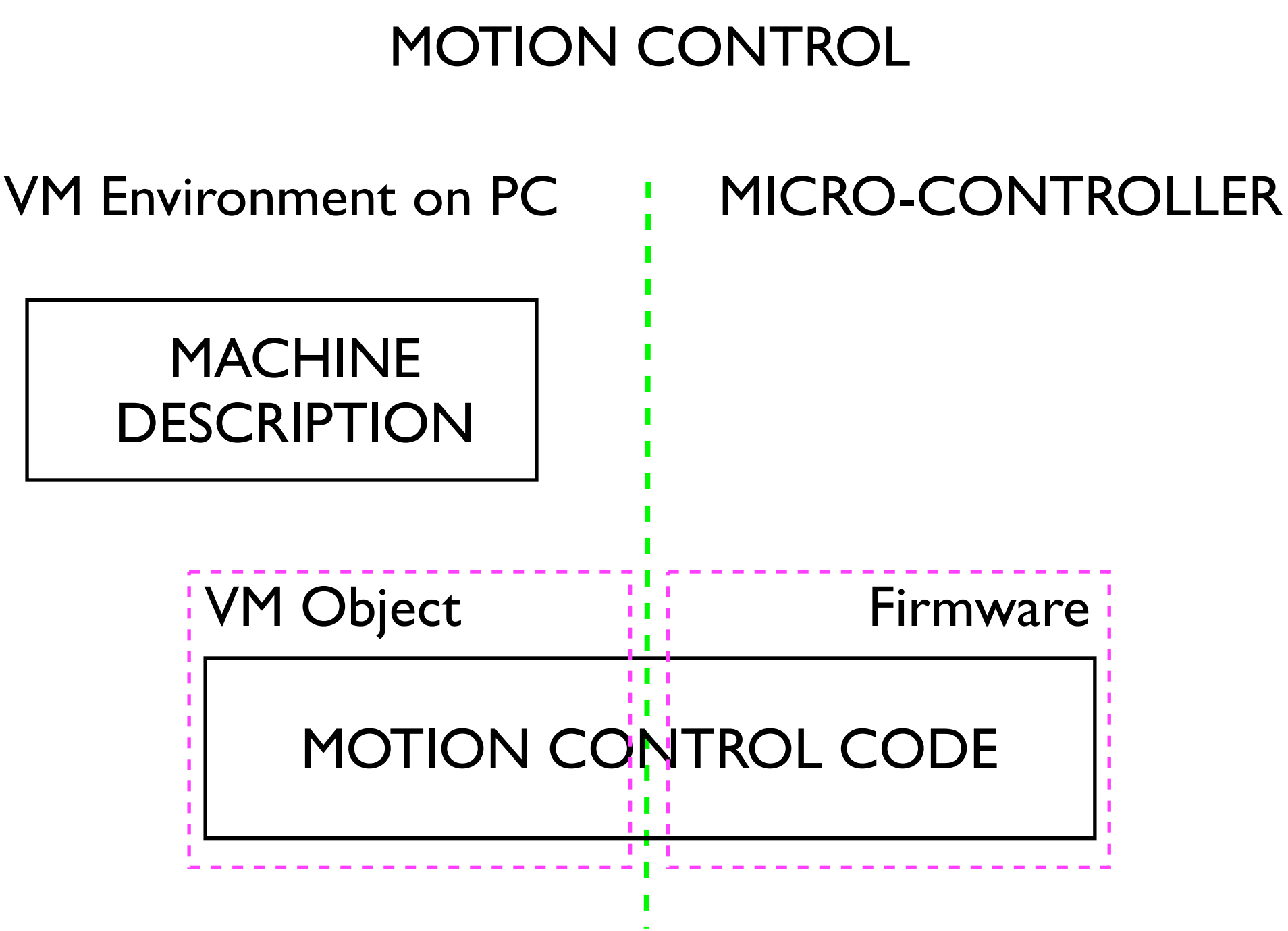
Virtual Machine Object



NODE URL: http://mtm.cba.mit.edu/vm/076-001e.py
USING LOCAL VMO: 076-001e.py



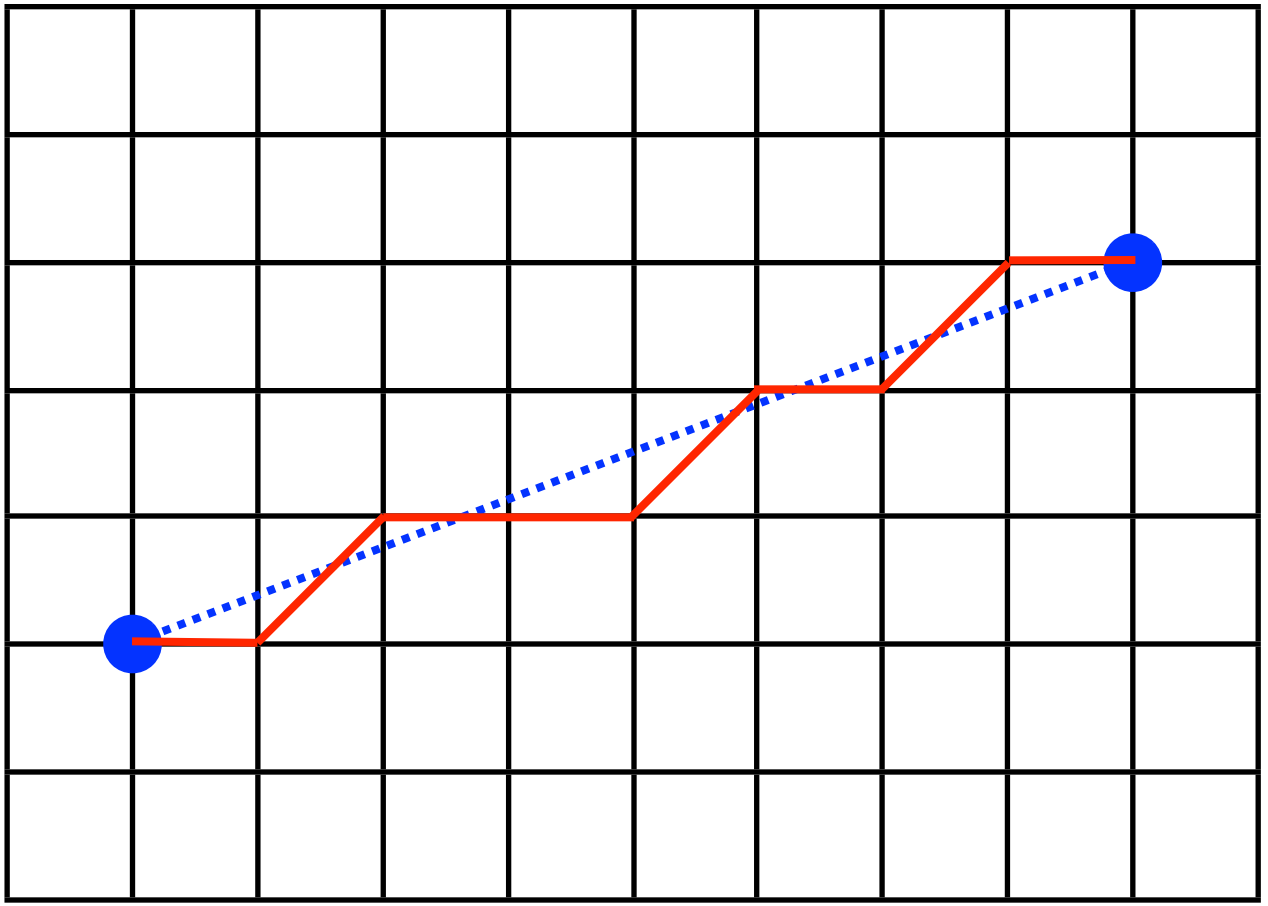
Control Firmware


MOTION CONTROL
VM Environment on PC
MICRO-CONTROLLER
MACHINE DESCRIPTION
VM Object
Firmware
MOTION CONTROL CODE

FIRMWARE

- PACKET HANDLING
- BRESENHAM LINE ALGORITHM
- CIRCULAR QUEUE
- ANTI-BACKLASH

BRESENHAM LINE ALGORITHM

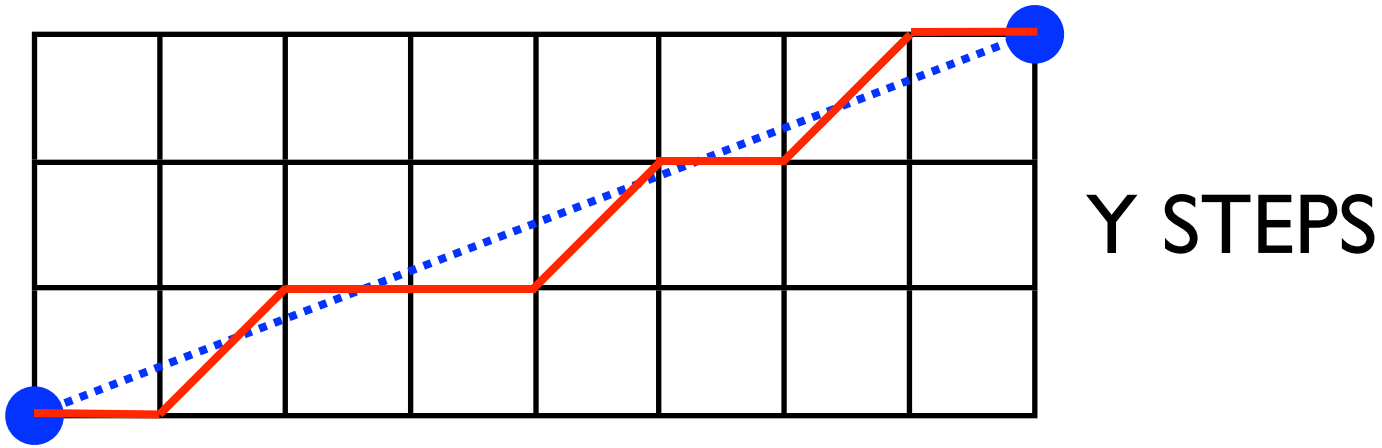


RULE #1: Always step along major axis.

RULE #2: Step in minor axis if error > 0.5 steps.

BRESENHAM ALGORIGTHM... THE TRICK:

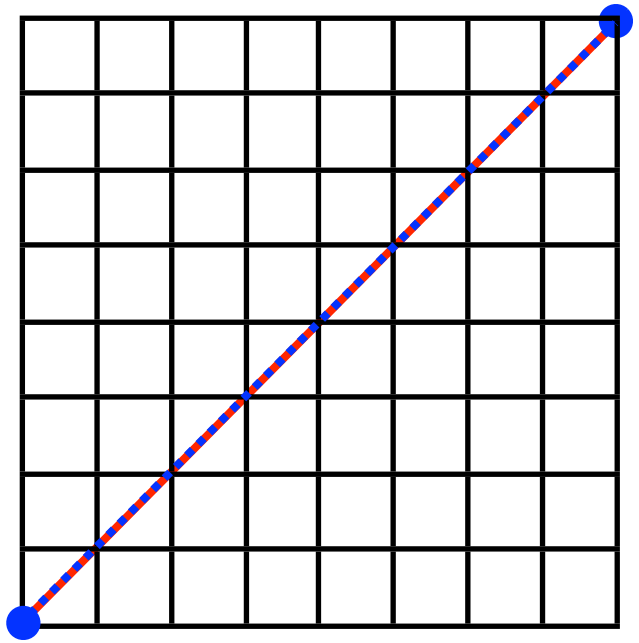
How do you know when the error is > 0.5?
Don't you need to know the slope?



ALWAYS STEP IN X AXIS.
SLOPE = (Y STEPS) / (X STEPS)
ERROR = (Y STEPS)
ACCUMULATE ERROR EACH X STEP.
WHEN ERROR > 0.5(X STEPS), TAKE A STEP IN Y.
OR, WHEN 2 x ERROR > X STEPS.

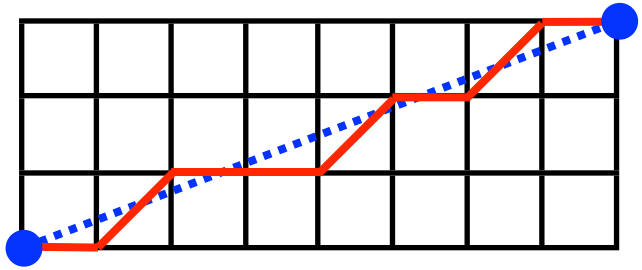
VIRTUAL MAJOR AXIS
FOR DISTRIBUTED MOTION CONTROL

mtm.move(8,3,1)



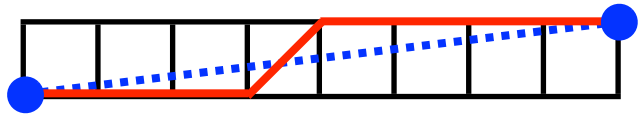
X STEPS = 8

2-AXIS CONTROLLER



Y STEPS = 3

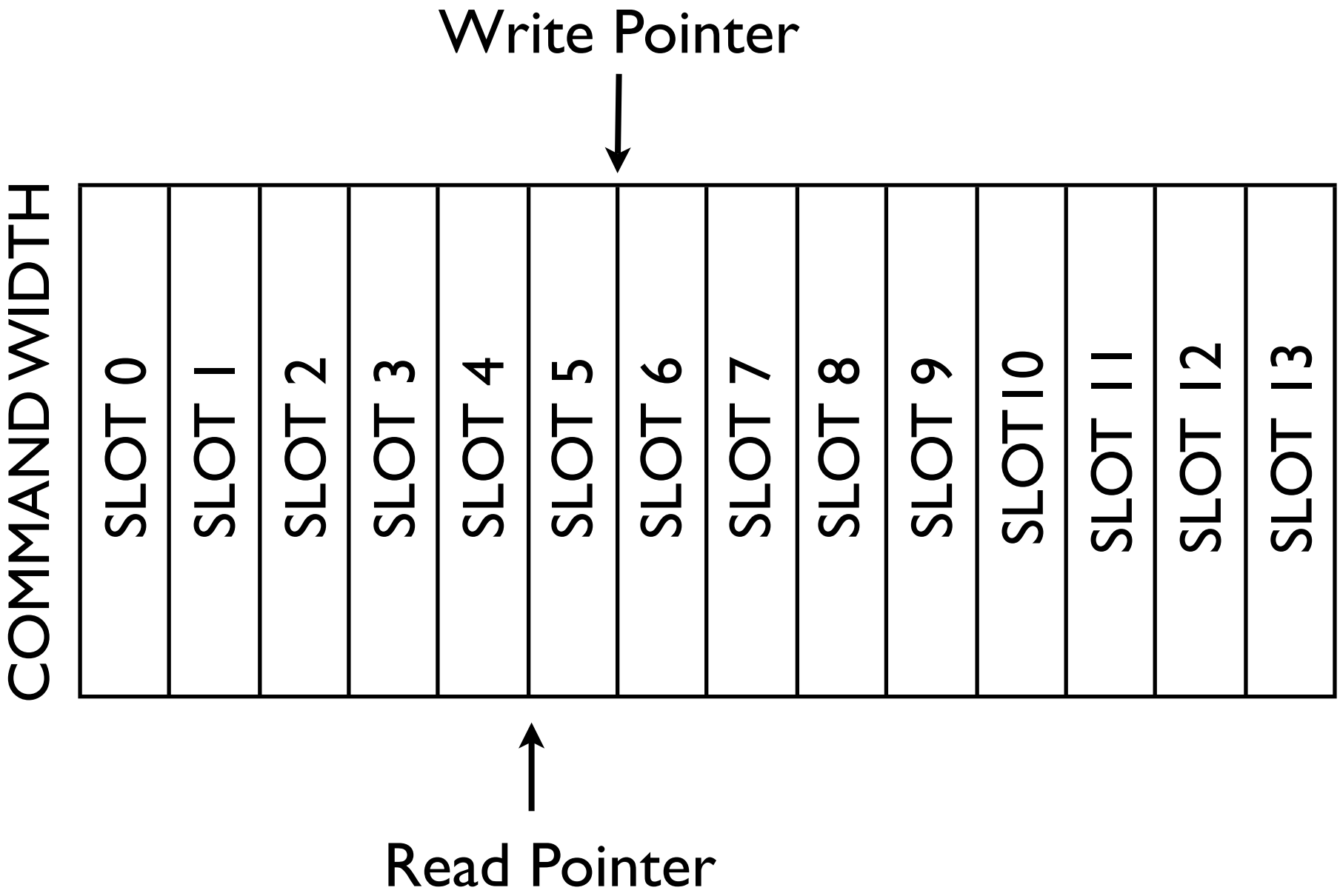
1-AXIS CONTROLLER



Z STEPS = 1

VIRTUAL STEPS = 8
= MAXIMUM STEPS

CIRCULAR BUFFER: ACCOUNTS FOR RAPID SERIES OF MOVES, SUCH AS WHEN GOING AROUND AN ARC.



3 AXIS MOTION CONTROL

mtm.move(x,y,z,rate)

VME

Machine Hardware Description
i.e. step angle, linear reduction,
control system topology, etc...

vmo.spin(x_steps, y_steps, z_steps, move_time)

VMO

PC-BASED FIRMWARE
Does complex calculations such as
floating point and division.

[key, dir., max_steps, x_steps, y_steps, z_steps, counter_top]

MCU

MCU-BASED FIRMWARE
Does timing-critical calculations and
I/O control.

mtm.cba.mit.edu/fabinabox/