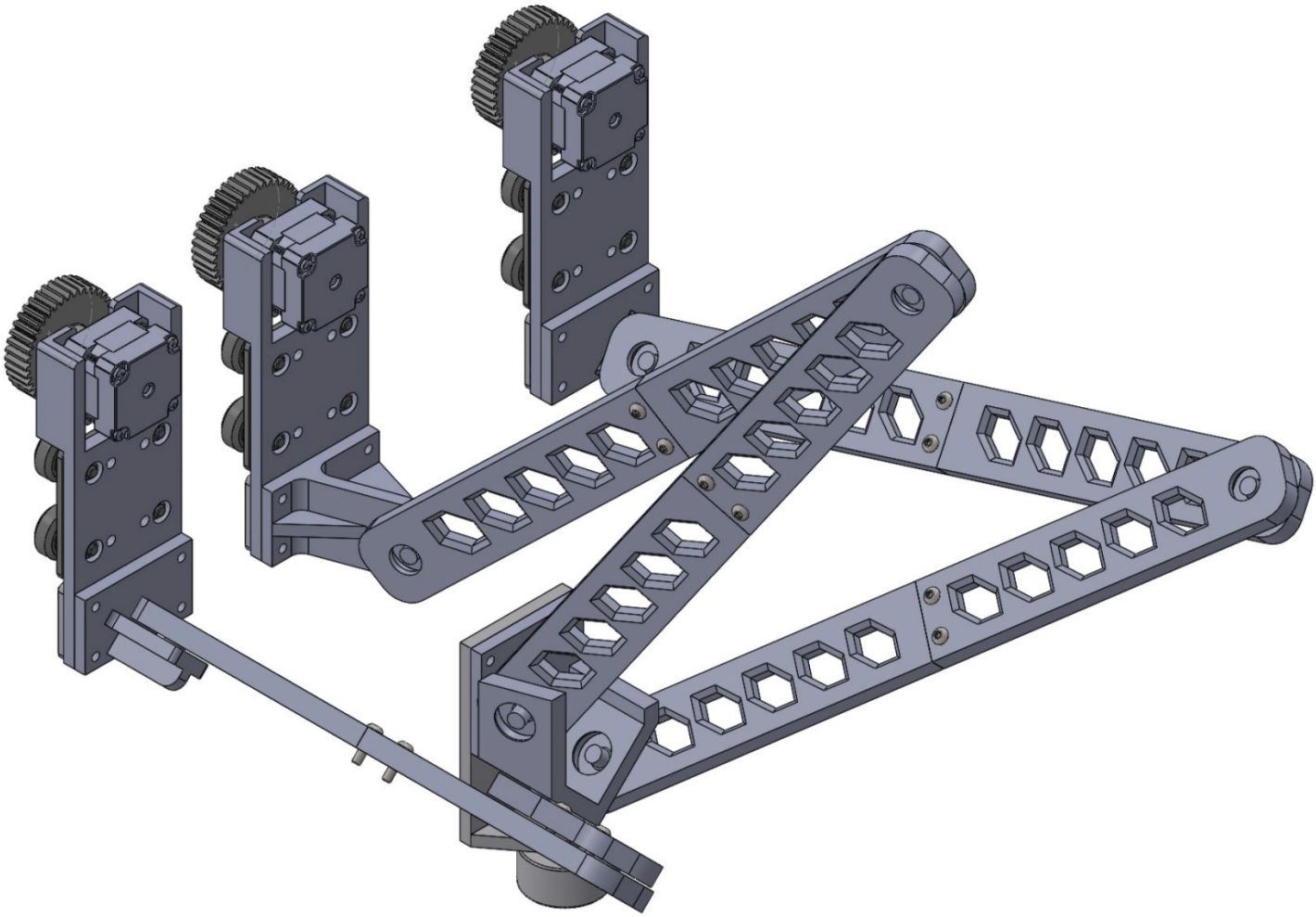
Monorail 3 Axis Robot Documentation

Written and Designed by Matthew Herber

Index:



REV 1.0 (8/19/2021)

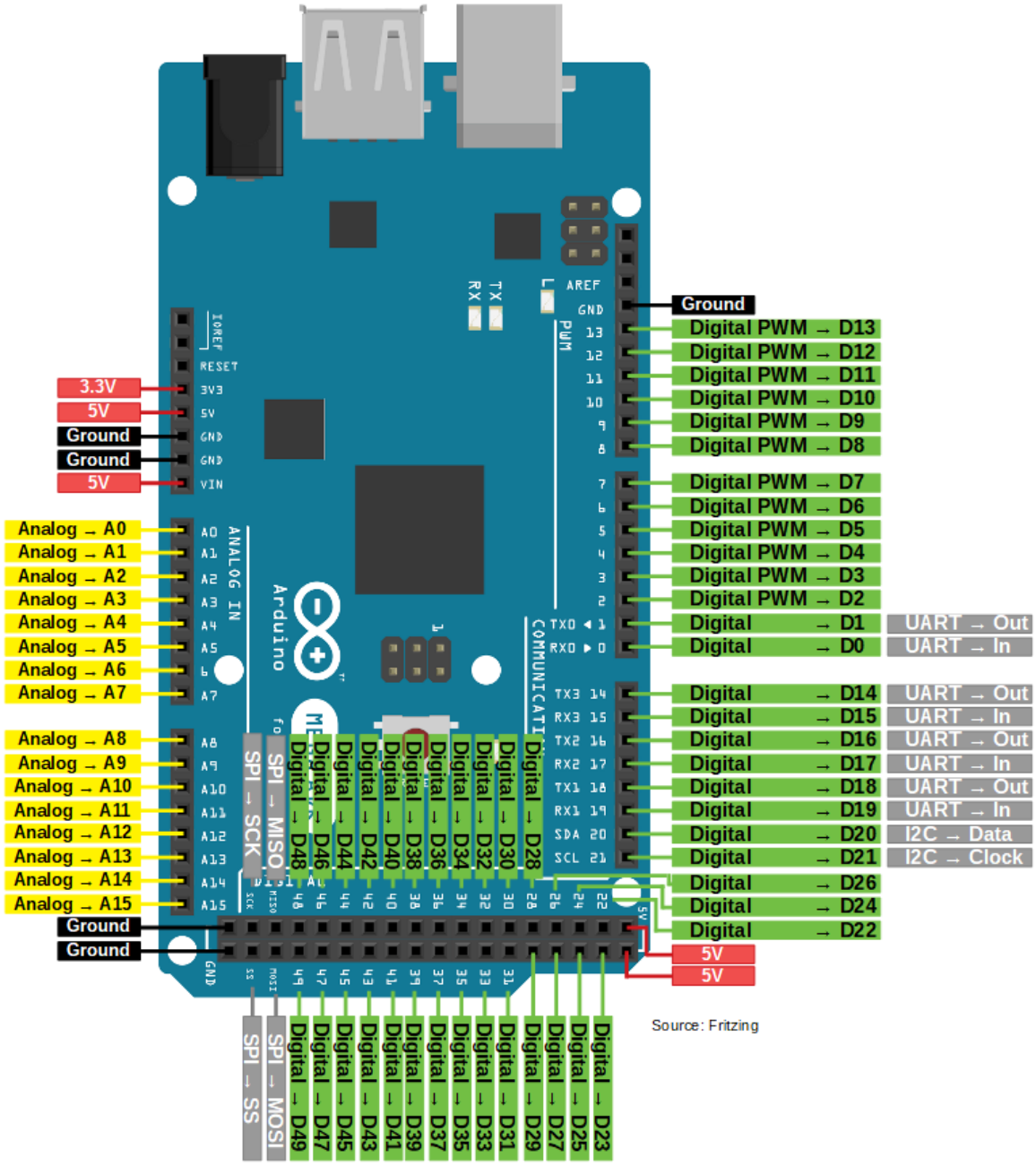
Pin Reservations

Stepper 1
Data: D22, D23, D24, D25
Endstop: D34

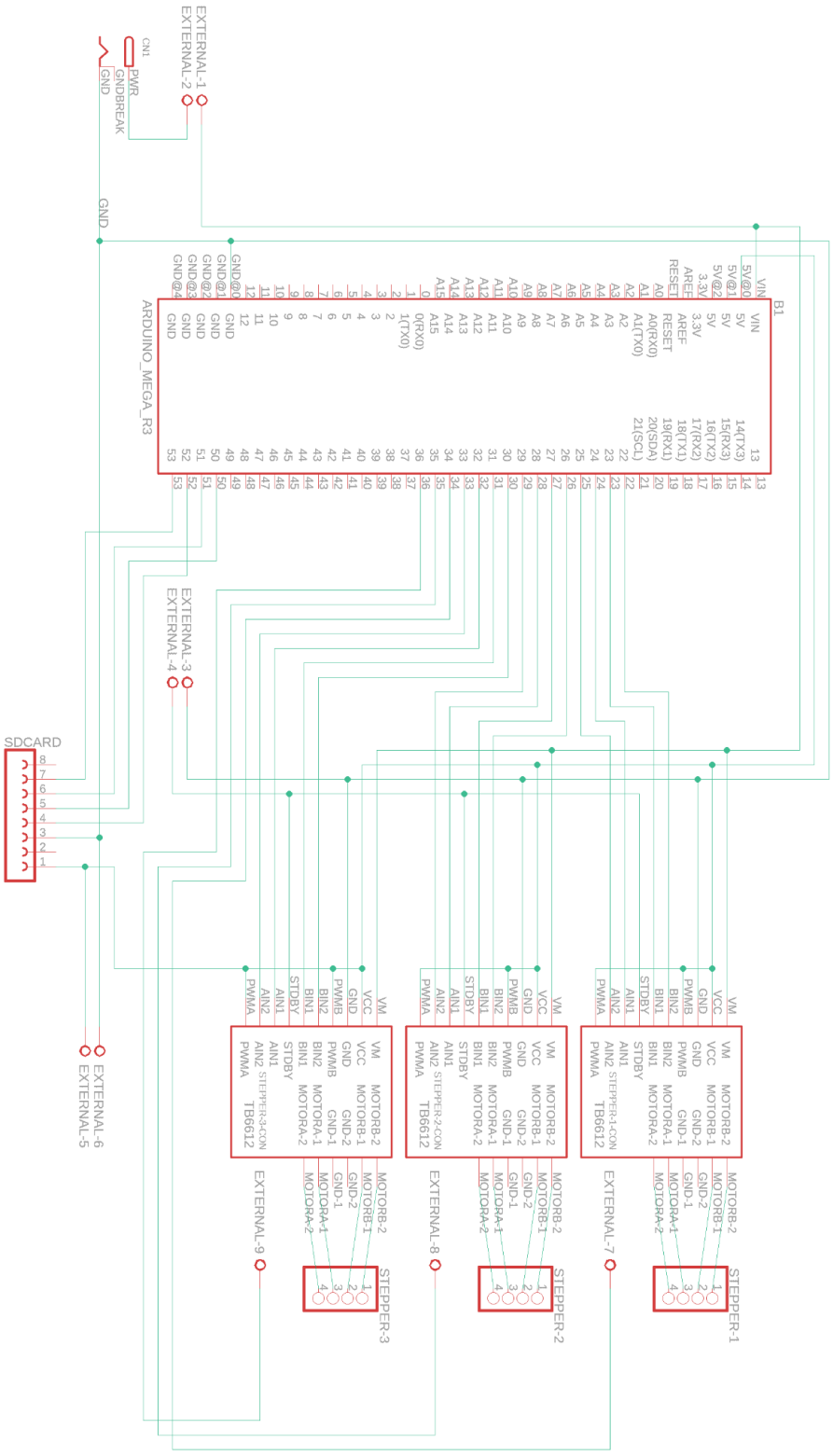
Stepper 2
Data: D26, D27, D28, D29
Endstop: D35

Stepper 3
Data: D30, D31, D32, D33
Endstop: D36

Other:
Electromagnet: D37



Custom Expansion Board Diagram (Board Revision 3.0)



Custom Firmware Command Set

Use # to indicate a comment on said line. To end a line, use ;\. All position commands use MM unless otherwise specified

lmove(x,y,z)
Moves the robot EOAT to the specified X Y Z coordinates in a straight line

jmove(x,y,z)
Moves the robot EOAT to the specified X Y Z coordinates as quick as possible

delay(ms)
Halts program execution for a specified amount of milliseconds

speed(%)
Sets the speed of the robot based on a percentage of the max speed. This change affects all motion moves after it is called

accel(%)
Sets the acceleration of the robot based on a percentage of the max acceleration. This change affects all motion moves after it is called

Carc(x,y,z,r,startAngle,endAngle)
Does a clockwise arc move given a center position, a radius from that center, a start angle and an end angle. always make sure your end angle is larger than your start angle

CCarc(x,y,z,r,startAngle,endAngle)
Does a counter-clockwise arc move given a center position, a radius from that center, a start angle and an end angle. always make sure your end angle is larger than your start angle

EOAT(0/1)
Turns on or off the EOAT, 0 for off and 1 for on

Sample Program

```
String program = "\
#basic starting program for testing;\
jmove(550,200,100);\
delay(1000);\
jmove(50,50,25);\
jmove(500,50,5);\
jmove(200,100,100);\
jmove(0,0,0);\
delay(1000);\
EOAT(1);\
;\
#Section to test speed command;\
speed(20);\
lmove(300,200,300);\
speed();\
lmove(0,0,0);\
EOAT(0);\
;\
#Section to test accel command;\
delay(1000);\
accel(5);\
lmove(300,200,300);\
accel();\
lmove(0,0,0);\
delay(1000);\
EOAT(1);\
#Rapid section testing individual axis moves;\
#X;\
lmove(200,0,0);\
delay(1000);\
jmove(100,0,0);\
```

Sample Program (cont)

```
#Y;\
lmove(100,100,0);\
delay(1000);\
jmove(100,0,0);\
#Z;\
lmove(100,0,100);\
delay(1000);\
jmove(100,0,0);\
#ALL AXIS;\
lmove(400,50,200);\
delay(1000);\
jmove(100,0,0);\
#Arc Testing;\
Carc(400,200,0,200,360,0);\
lmove(0,0,0);\
CCarc(400,200,0,200,0,360);\
lmove(0,0,0);\
EOAT(0);\
";
```

Basic Code Loop Explanation

The execution of the custom firmware is actually quite simple. There is a string variable stored internally to the code, that contains the entire program. This is done so that in later revisions strings can be read in from a storage device to have different programs stored without requiring a reflash of the Arduino itself. This string is then converted into an array of instructions, using ;\ as a line end marker. This array is looped over for each instruction.