Dual 3 Position Semaphore Controller

Revision v2.0 8/23/25

Seth Neumann, sales@modelrailroadcontrolsystems.com

This document describes the Dual 3 Position Semaphore Controller board and how to assemble and install it.

Revision History
V0.1 – first pass – August 1, 2021
V0.2 – Jon Schmidt Comments, Seth Neumann, cleanup
V0.3 – Jon Schmidt Update to sketch 2.4
V1.1 – V 1.1 board, Jon Schmidt sketch 3.2 11/20/23
V2.0 - V1.6 board and 1.3, 1.4 updates, Jon Schmidt Sketch V 8/23/25

CONTENTS

Table of Figures

Table of Tables

1 INTRODUCTION

This board is intended to control Train Order Boards and 2 arm ABS Semaphores, either 2 arm lower quadrant or 2 x 1 arm upper quadrant. Excellent models of these signals are available from Tomar and Showcase Miniatures as of this revision (8/13/25), others are out there. The assumption is that the semaphore has actuator wires for each arm extending down from its base.

Features

- Supports one or two SG90 servos to control two semaphore "arms"
- Operate leads and connections for a white LED are brought to an 8 position 0.100 connector (somewhat different connections on earlier versions)
- A 2.1mm barrel jack is provided if you choose to supply power to the Arduino Nano from a 5V wall wart. Note this power goes directly to the processor and bypasses the on-board regulator.
- 5V power as well as separate 5V for servos may be applied on a 3 position 3.5mm screw terminal. Servo power source is selected on a jumper on the back of the board.
- Each aspect (STOP (form 31), APPROACH (form 19), CLEAR has its own adjustment pot and test button.
- Each input line has an on-board tactile switch button. The tactile switch is used in calibration and testing
- A pair of LEDs is provided to indicate status during configuration

6 lines on the Arduino are not used: D0, D1, (usually TX/RX), A4, A5 (usually SDA/SCL) D8, and D13 (has the onboard LED, could be used for user feedback of some sort).

The servos are positioned on the board so, if equipped with the short horns, they face in and can both be mounted directly below the mast of a 2-arm semaphore or train order board. If placed facing out, the horns can control a pair single arm semaphores protecting an ABS block boundary. The servo may also be mounted off-board (as of this revision we are planning to offer external servo brackets) and all of the usual Radio Control (R/C) adapters and "jewelry" may be used as required for remote mounting or dealing with benchwork obstacles.

This board is based on a design by Seth Neumann, Earl Girbovan and John Plocher for a series of "Introduction to Arduino Programming" clinics that were offered at various NMRA PCR meets from 2017-2019 as well as the NMRA Conventions at Kansas City in 2018 and Salt Lake City in 2019. The code was developed by Jon Schmidt. The board designs are on Seth Neumann's GitHub site https://github.com/SethNeumann/MRCS-Dual-3-Position-Semaphore-Controller and the Code on Jon Schmidt's https://github.com/joneschmidt/SemaphoreControl. If you're looking to modify these designs, check GitHub for the latest versions.

2 IDENTIFICATION AND INFORMATION

2.1 Board Layouts

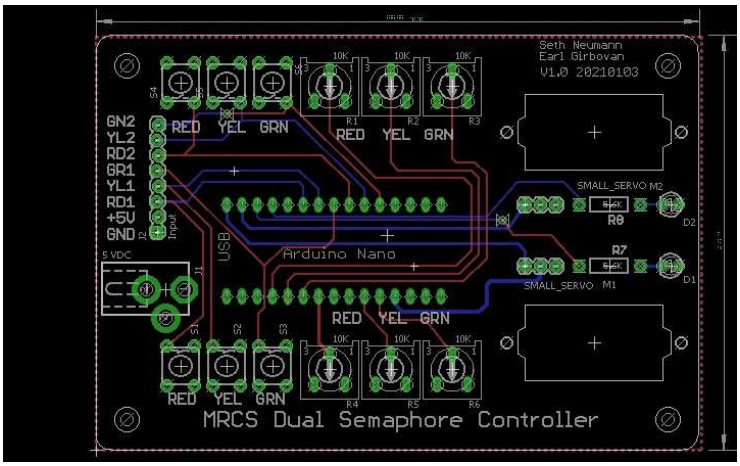


Figure 1- Version 1.0 Board Layout – Note R3 – input power on 8 position screw terminal block.

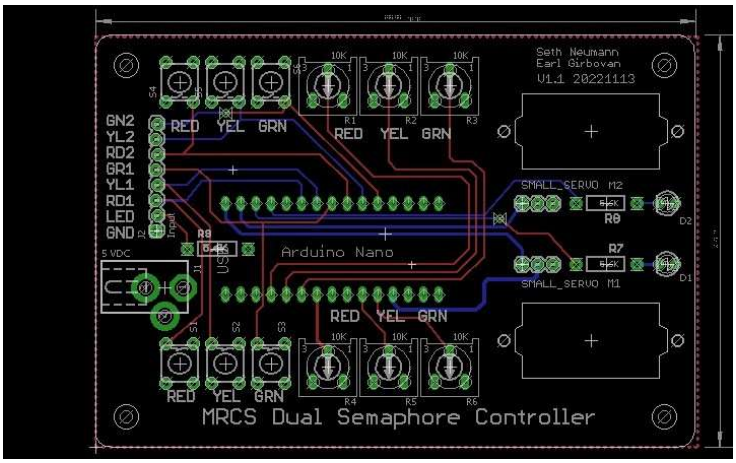
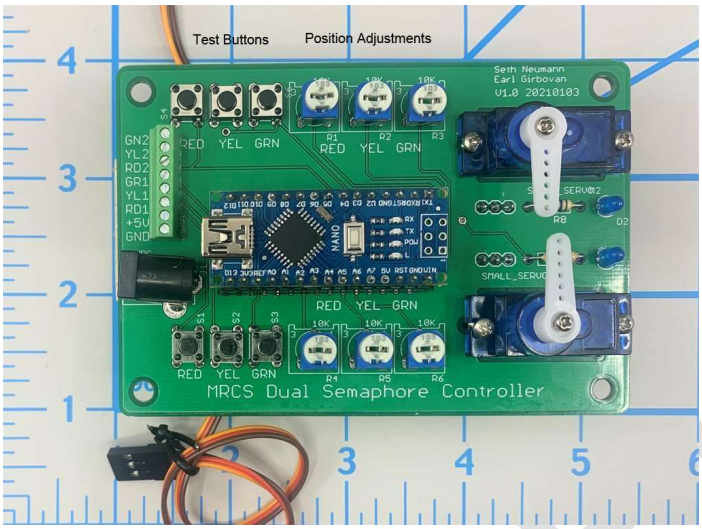


Figure 2- Version 1.1 Board Layout - note R3 – +5 on terminal block labeled as LED Power – no internal limiting resistor



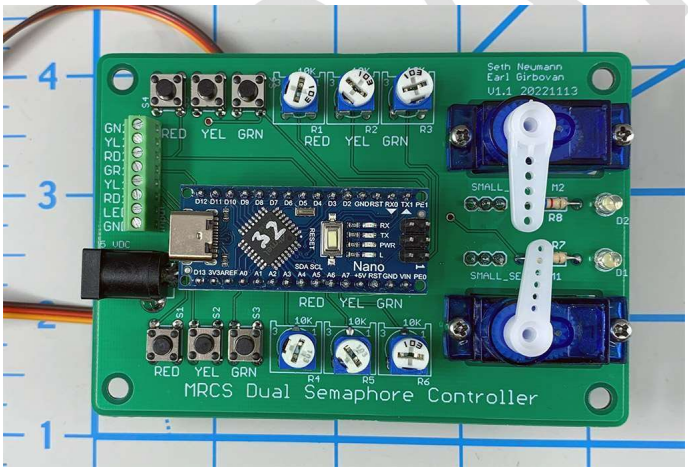


Figure 3 - Photos of V1.0, 1.1

You can't see R3 in the photo, (it's under the USB connector) but the +5 screw has been changed to a current limited output to support a white LED behind the spectacle of the semaphore. This was in response to customer requests and saves the trouble of external limiting resistors.

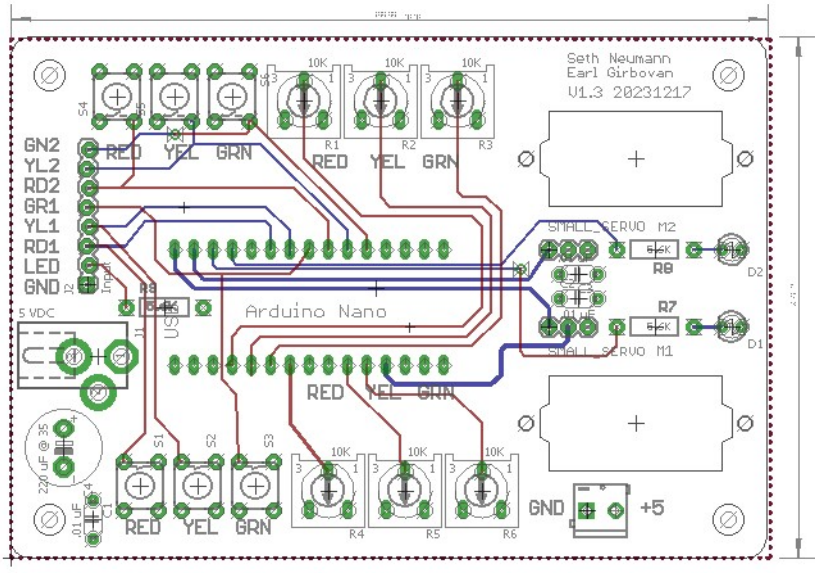


Figure 4 - Version 1.3 on board limiting resistor for LED, Screw terminal for power, filter capacitors

Version 1.3:

- brings the screw terminal power connector back on a separate 3 position 3.5mm (to facilitate heavier feed wire) connector, it is in parallel with barrel jack seen center left. This can be used to daisy chain power to a second 5V device. See notes on power below.
- Jumper to select use of main 5VDC power OR separate Vin for Servos
- Adds filter capacitors on board there is a 220uF electrolytic on the lower left (you can put a larger cap in if it fits) and 3 0.01 uF monolithic caps, one near the electrolytic and two close to the servos. These help to suppress motor noise and keep it out of the Arduino.
- Note the USB end of the Nano (can be mini/micro/or USB-C depending on what's available) is on the left side next to the barrel jack.
- Development and testing was done with version 3.2 of the software, but should be backward compatible.

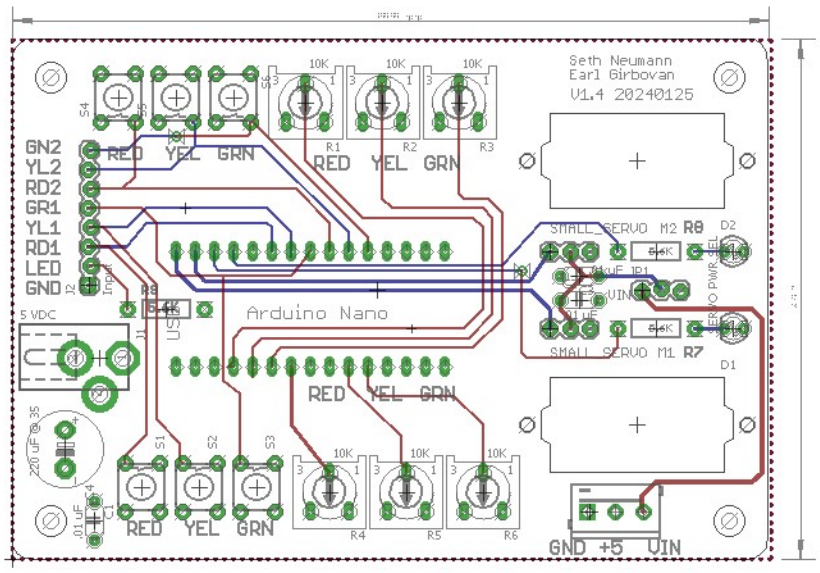


Figure 5 - Version 1.4 – Terminal block for Arduino and Servo power

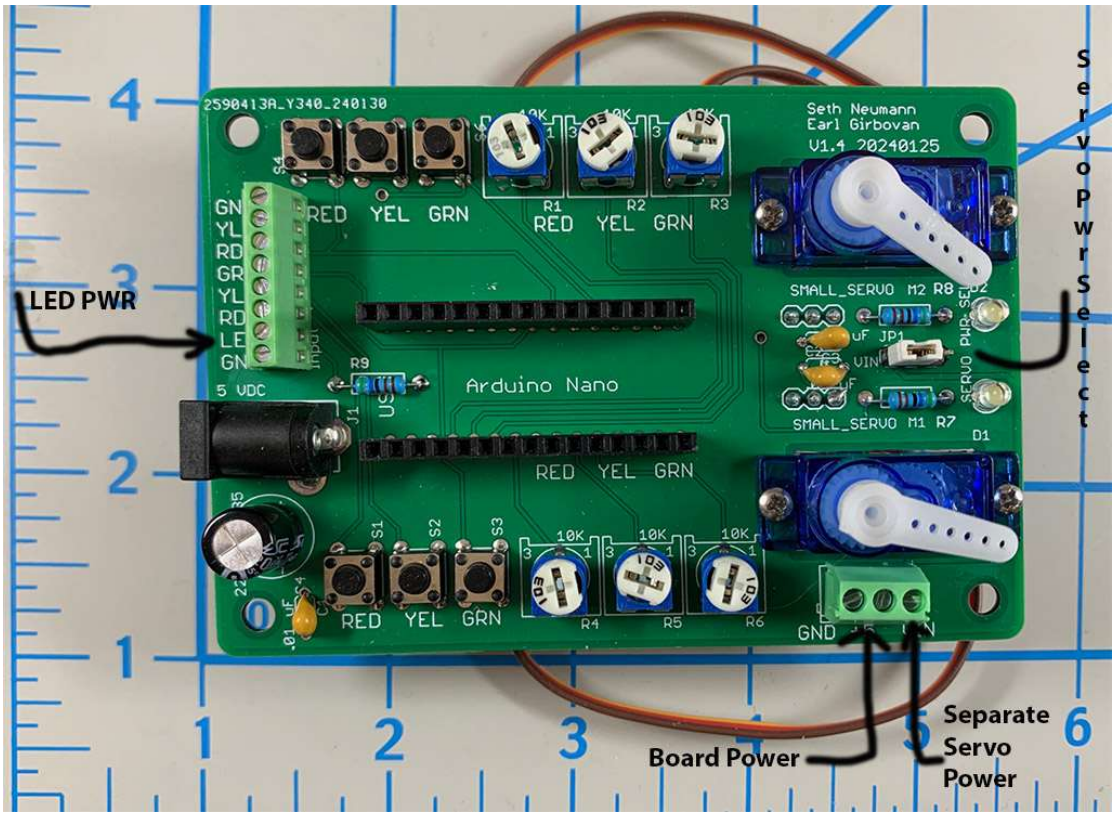


Figure 6 - V1.4 without Nano

Note orientation of USB port, version 3.2 sketch

Version 1.4

- Adds a 3-position screw terminal for GND, 5V for the Arduino and optionally the servos and Vin which is separate power for the servos if desired.
- Use Jumper JP1 to select whether the servo is powered from general 5V or from Vin.
  - Note that in any case a 5V, MINIMUM 1A, preferably 2A supply is available for the source powering the servo.

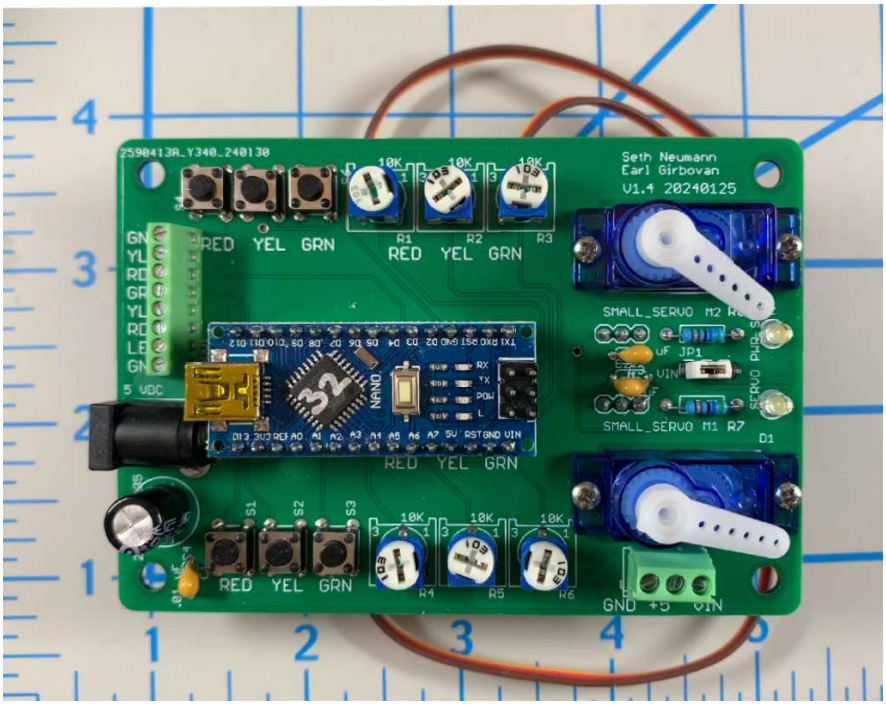


Figure 7 - Version 1.4 with Nano

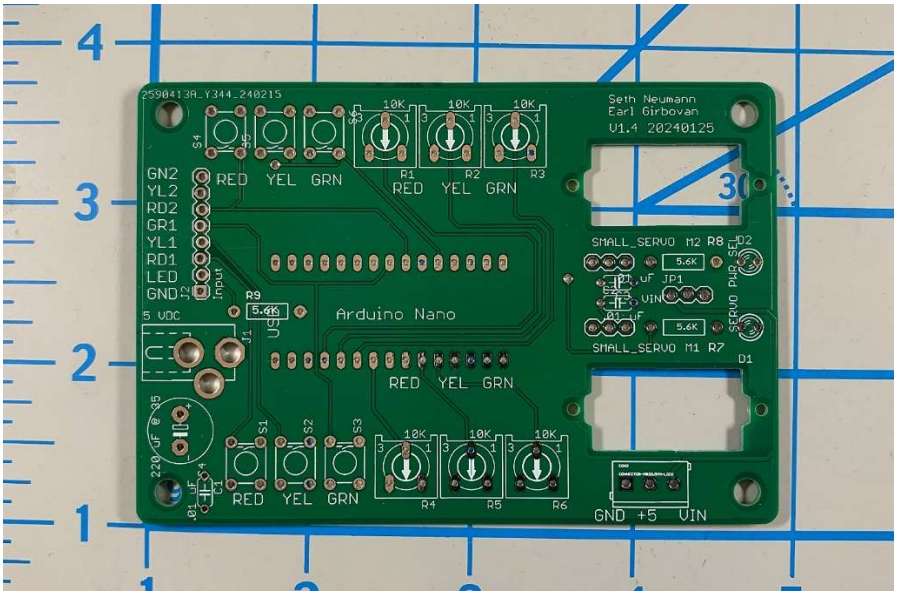


Figure 8 - Version 1.4 Bare Board

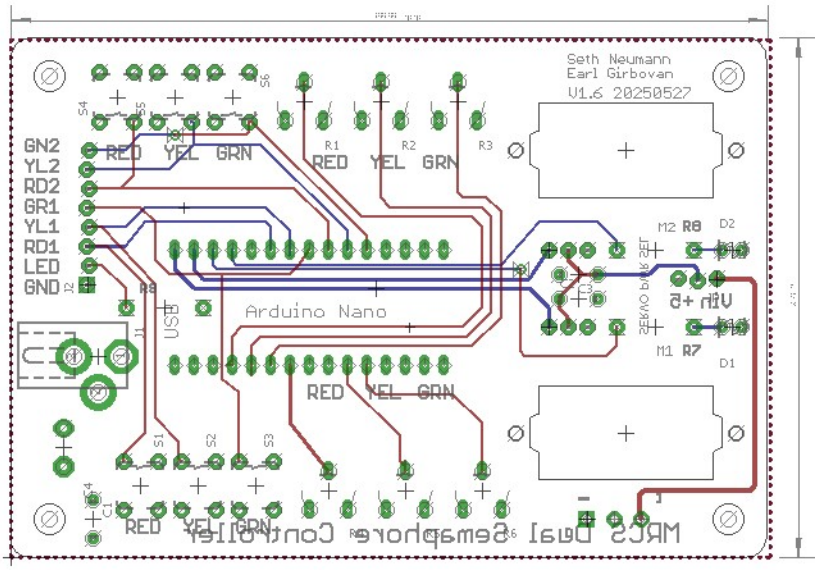


Figure 9 - Version 1.6 mechanical adjustments

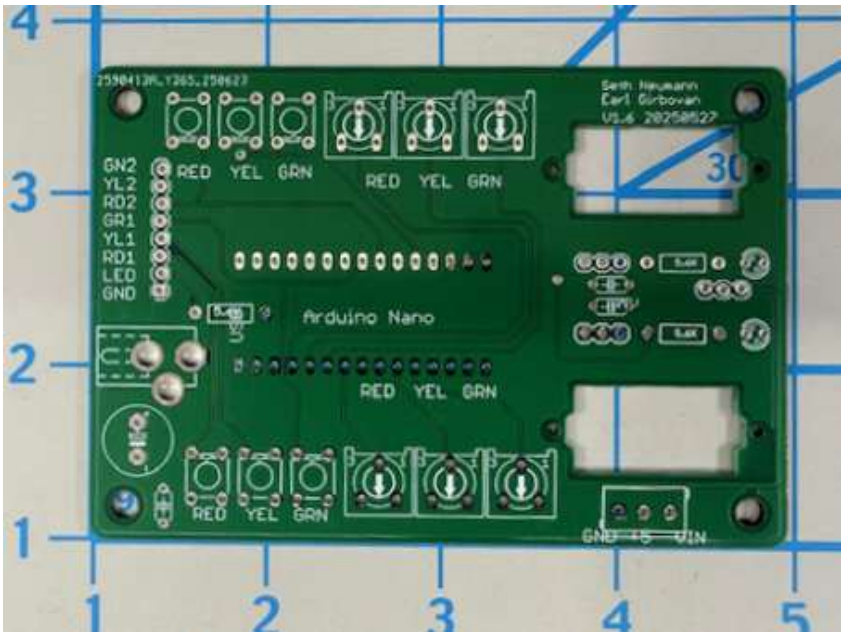


Figure 10 - Version 1.6 bare board top view

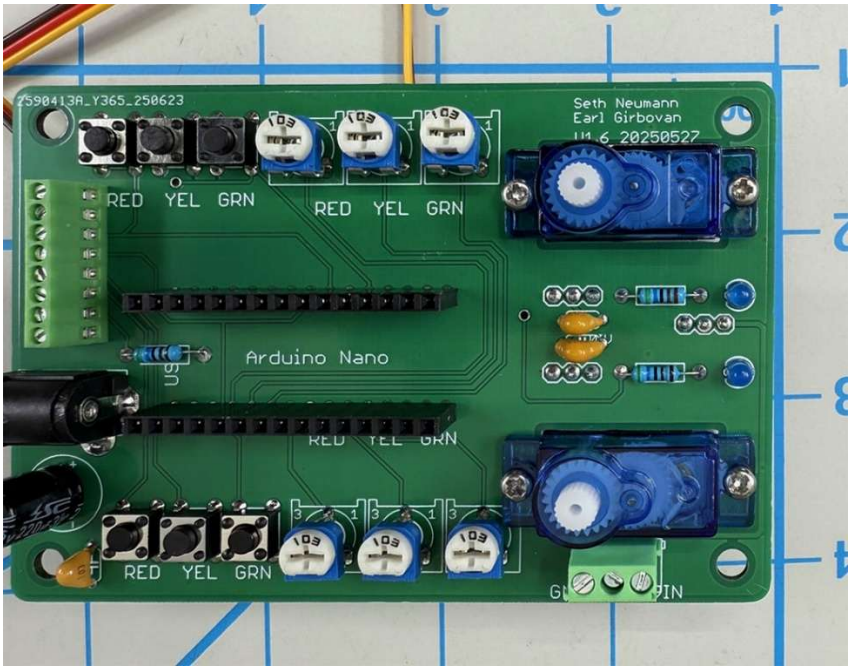


Figure 11 - Version 1.6 top view

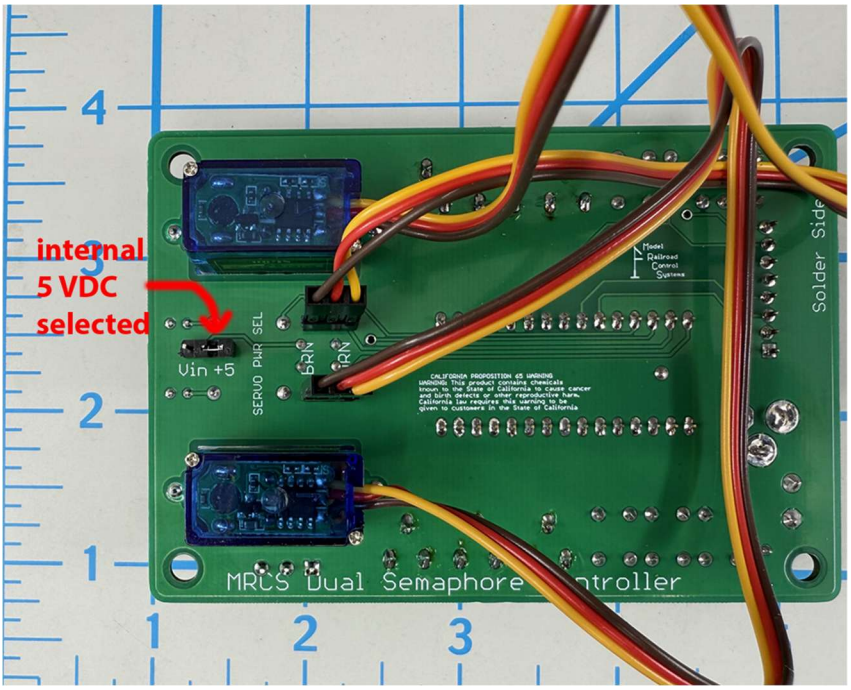


Figure 12 - V1.6 bottom view showing power select jumper

This board is sold as a single unit, assembled, and tested or as bare board. If you are interested in alternate connectors, large quantities, or custom modifications please contact us.

2.1 Bills of Materials:

Qty	Value	Device	Package	Parts	Description
6	10K	TRIM_US-RS3	RS3	R1, R2, R3, R4, R5, R6	POTENTIOMETER
1	5 VDC	CONNECTOR-DC-POWER-RA	DCJ0202	J1	DC POWER JACK
3	5.6K	RESISTORPTH-1/4W	AXIAL-0.4	R7, R8, R9	Resistor
1	Input	CONNECTOR-M08LOCK	1X08_LOCK	J2	Header 8
1	NANO	NANO	NANO	NANO	Arduino Nano
2	SMALL_SERVO	SMALL_SERVO	1X03	M1, M2	Small servo motor
6	SWITCH-MOMENTARY-2PTH	SWITCH-MOMENTARY-2PTH	TACTILE-PTH	S1, S2, S3, S4, S5, S6	Various NO switches- pushbuttons, reed, etc
2	White	LEDLED3MM	LED3MM	D1, D2	

Table 1- Version 1.1 Bill of Materials

Qty	Value	Device	Package	Parts	Description
1	power	CONNECTOR-M023.5MM-LOCK	SCREWTERMINAL-3.5MM-2_LOCK	CON1	PHOENIX CONNECTOR
3	.01 uF	CAPPTH1	C050-024X044	C1, C2, C3	
6	10K	TRIM_US-RS3	RS3	R1, R2, R3, R4, R5, R6	POTENTIOMETER
1	220 uF @ 35	CAP-POLPTH-5MM	E5-10.5	C4	
1	5 VDC	CONNECTOR-DC-POWER-RA	DCJ0202	J1	DC POWER JACK
3	5.6K	RESISTORPTH-1/4W	AXIAL-0.4	R7, R8, R9	Resistor
1	Input	CONNECTOR-M08LOCK	1X08_LOCK	J2	Header 8
1	NANO	NANO	NANO	NANO	Arduino Nano
2	SMALL_SERVO	SMALL_SERVO	1X03	M1, M2	Small servo motor
6	SWITCH-MOMENTARY-2PTH	SWITCH-MOMENTARY-2PTH	TACTILE-PTH	S1, S2, S3, S4, S5, S6	Various NO switches- pushbuttons, reed, etc
2	White	LEDLED3MM	LED3MM	D1, D2	
1	PCB-Dual servo semaphore				

Table 2 - Version 1.3 Bill of Materials

Qty	Value	Device	Package	Parts	Description
3	.01 uF	CAPPTH1	C050-024X044	C1, C2, C3	Monolithic cap
6	10K	TRIM_US-RS3	RS3	R1, R2, R3, R4, R5, R6	POTENTIOMETER
1	220 uF @ 35	CAP-POLPTH-5MM	E5-10.5	C4	Electrolytic Cap
1	5 VDC	CONNECTOR-DC-POWER-RA	DCJ0202	J1	DC POWER JACK
3	5.6K	RESISTORPTH-1/4W	AXIAL-0.4	R7, R8, R9	Resistor
1	CONNECTOR-M033.5MM-LOCK	CONNECTOR-M033.5MM-LOCK	SCREWTERMINAL-3.5MM-3_LOCK.007S	CON2	3 x 3.5mm
2	Nano Connector	CONNECTOR-15 x 0.100	female header 0.100		
1	Input	CONNECTOR-M08LOCK	1X08_LOCK	J2	Header 8
3	3 pin hdr male	JUMPER-3PTH-LOCK	1X03	JP1, Servo1, Servo 2	on back
6	SWITCH-MOMENTARY-2PTH	SWITCH-MOMENTARY-2PTH	TACTILE-PTH	S1, S2, S3, S4, S5, S6	NO switches
2	White or blue	LED3MM	LED3MM	D1, D2	
1	Dual 3 Position Semaphore	PCB Board	69x99		

Table 3 - Version 1.4 Bill of Materials

Qty	Value	Device	Package	Parts	Description
3	.01 uF	CAPPTH1	C050-024X044	C1, C2, C3	
6	10K	TRIM_US-RS3	RS3	R1, R2, R3, R4, R5, R6	POTENTIOMETER
1	220 uF @ 35	CAP-POLPTH-5MM	E5-10.5	C4	
1	5 VDC	CONNECTOR-DC-POWER-RA	DCJ0202	J1	DC POWER JACK
3	5.6K	RESISTORPTH-1/4W	AXIAL-0.4	R7, R8, R9	Resistor
1	CONNECTOR-M033.5MM-LOCK	CONNECTOR-M033.5MM-LOCK	SCREWTERMINAL-3.5MM-3_LOCK.007S	CON2	
1	Input	CONNECTOR-M08LOCK	1X08_LOCK	J2	Header 8
1	NANO	NANO	NANO	NANO	Arduino Nano
1	SERVO PWR SEL	JUMPER-3PTH-LOCK	1X03	JP1	
2	SMALL_SERVO	SMALL_SERVO	1X03	M1, M2	Small servo motor
6	SWITCH-MOMENTARY-2PTH	SWITCH-MOMENTARY-2PTH	TACTILE-PTH	S1, S2, S3, S4, S5, S6	Various NO switches- pushbuttons, reed, etc
2	White	LEDLED3MM	LED3MM	D1, D2	
1	Printed Cirucit Board				

Table 4 - Version 1.6 Bill of Materials

2.2 Schematic Drawings

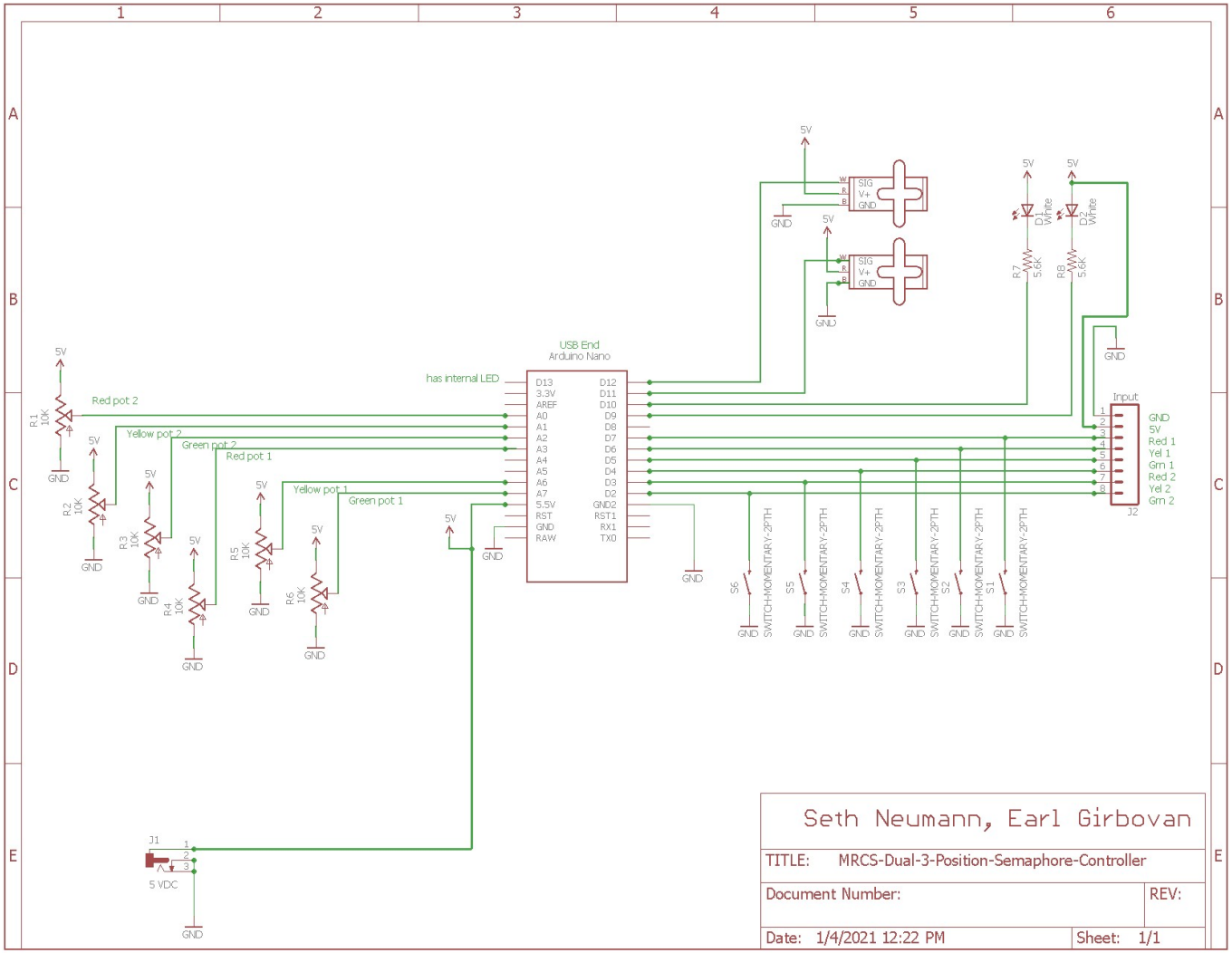


Figure 13 - V1.0 Schematic

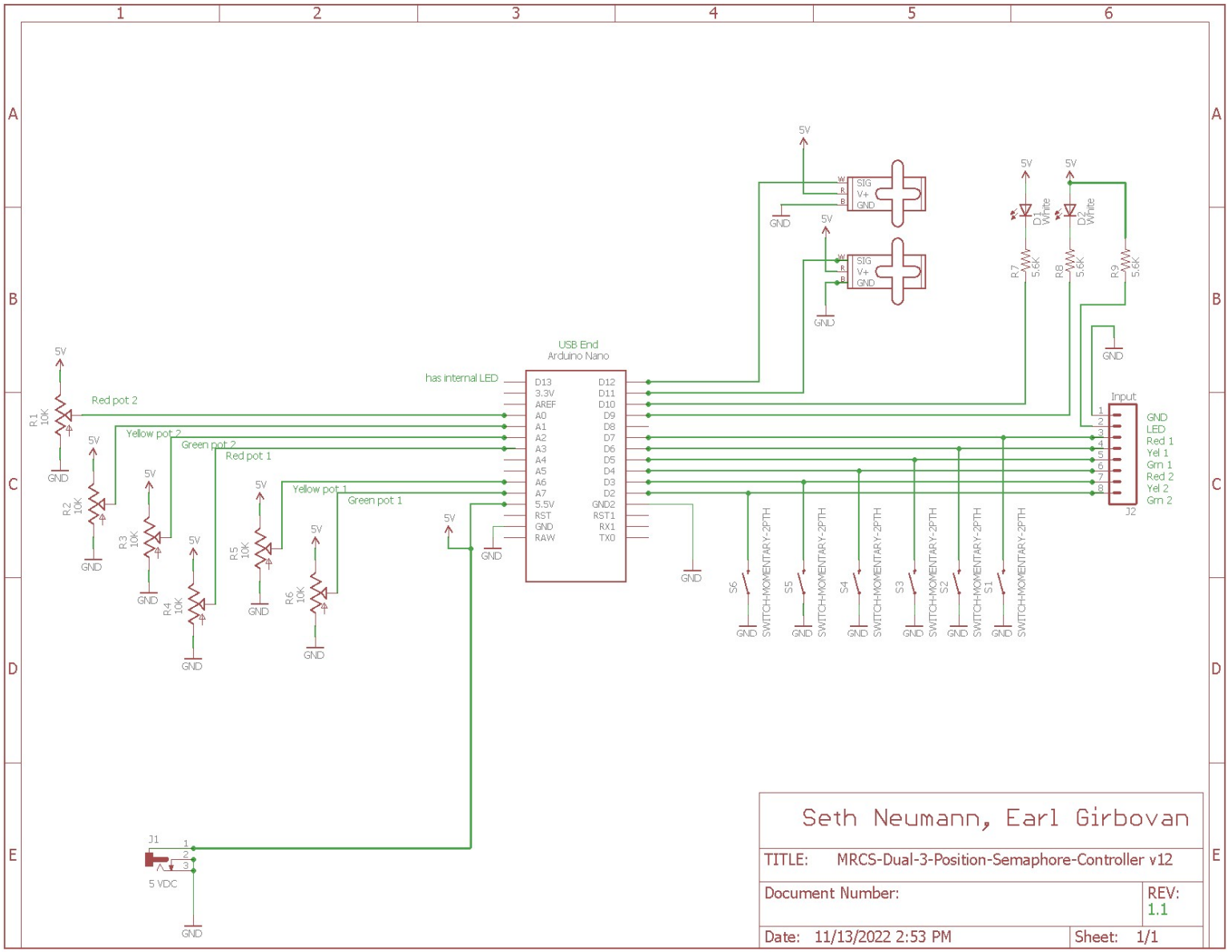


Figure 14 - Version 1.2 Schematic

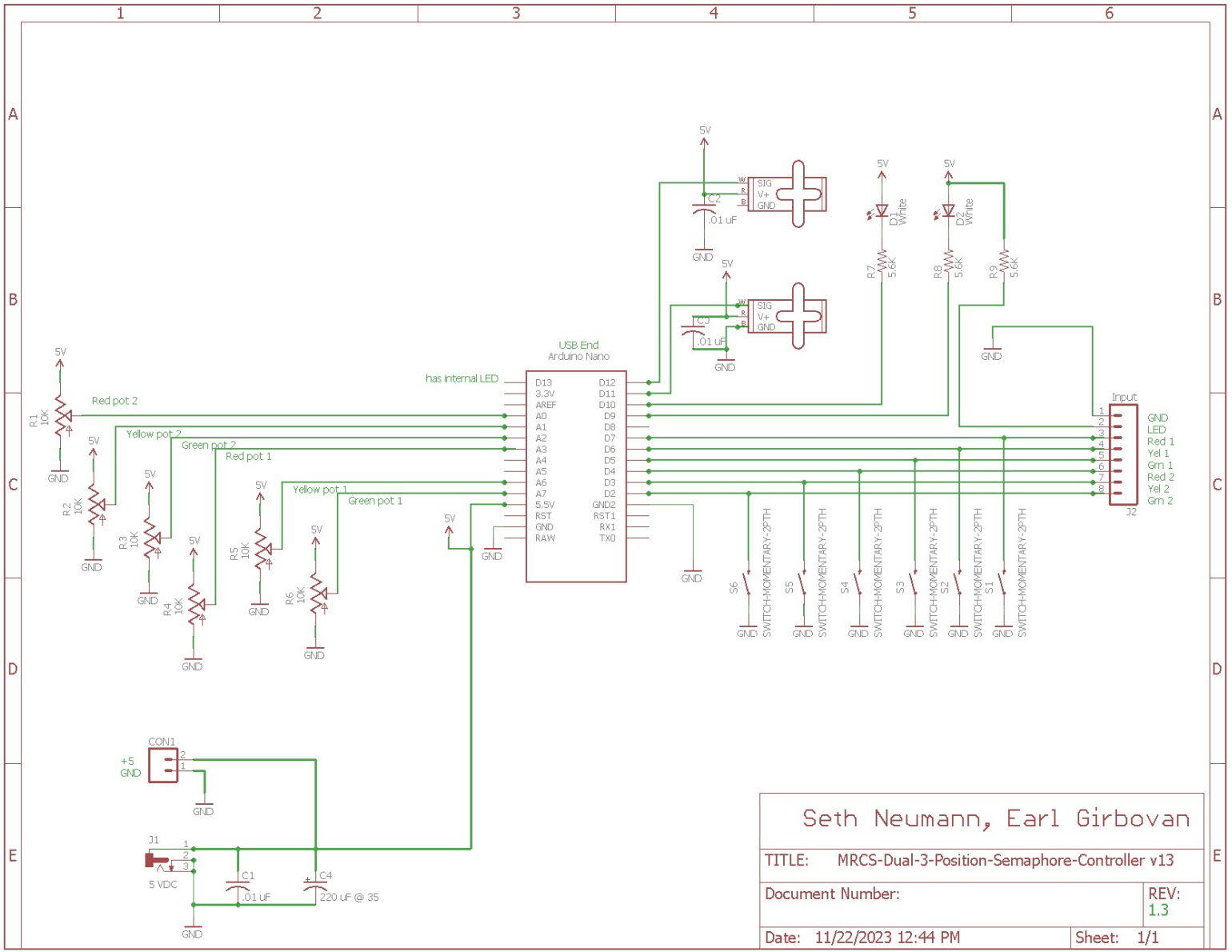


Figure 15 - Version 1.3 Schematic

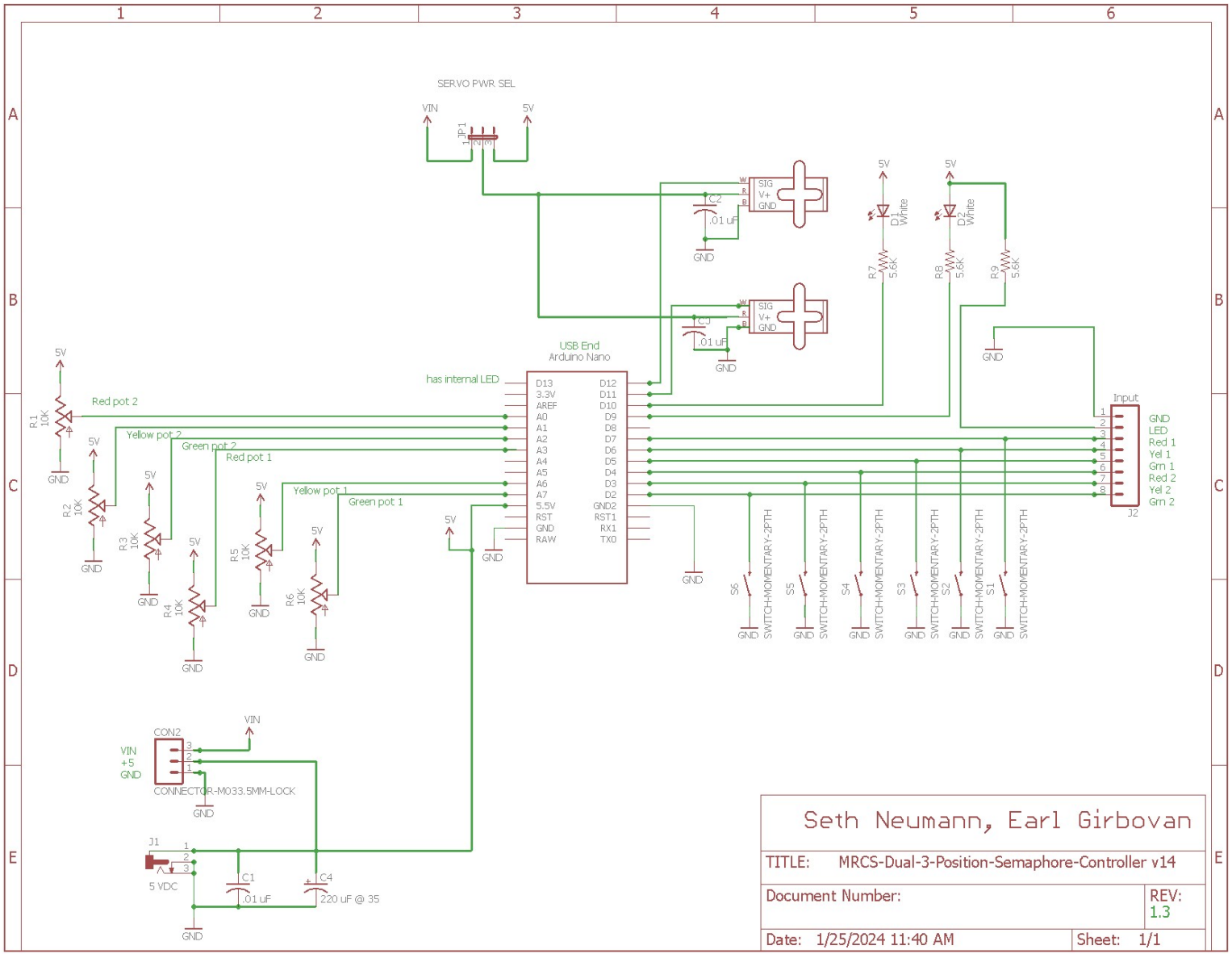


Figure 16 - Version 1.4 Schematic

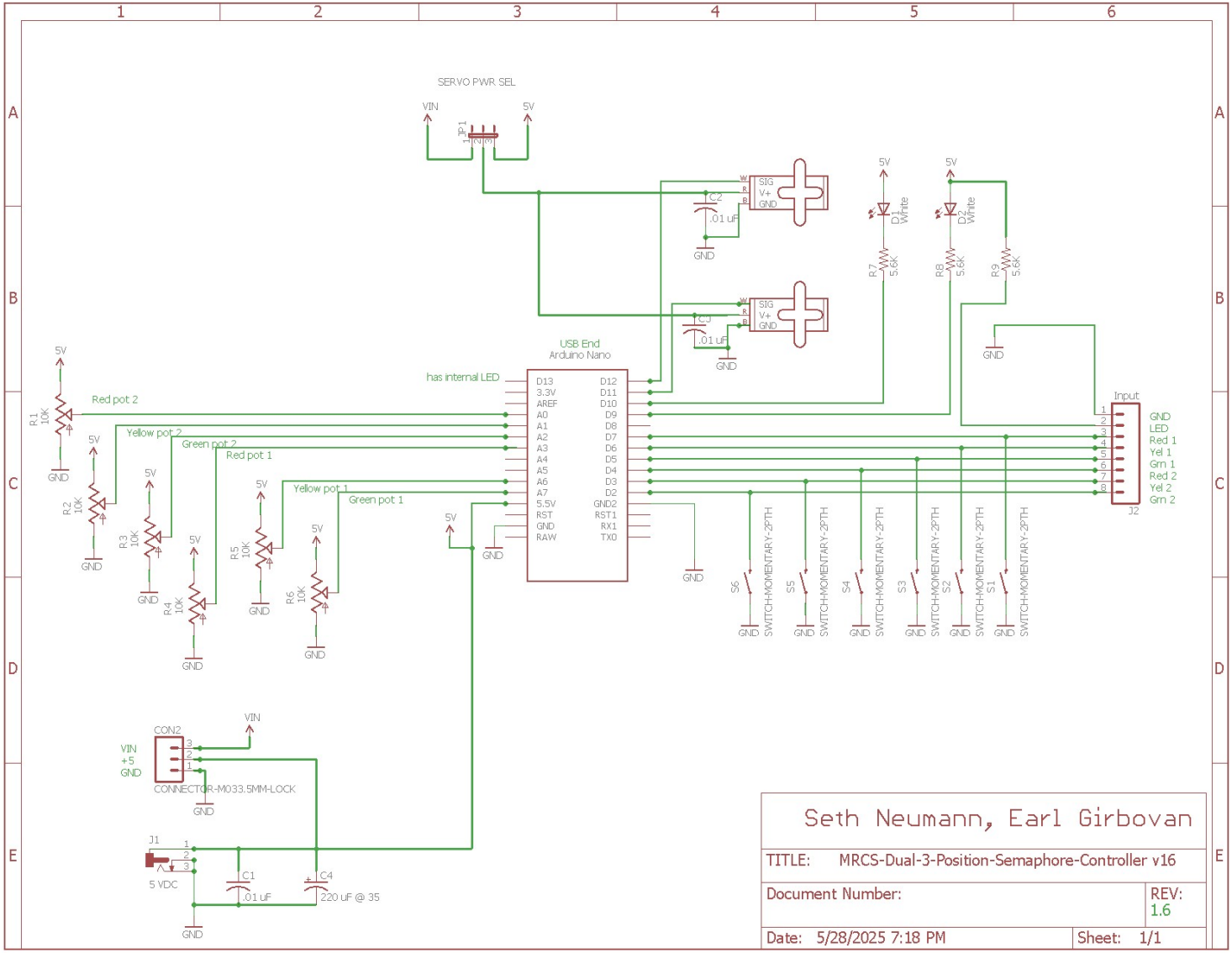


Figure 17 - Version 1.6 Schematic

3 OPTIONS

3.1 Connectors

The 8-position connector is on 0.100 centers (staggered slightly to hold the connectors in place during assembly). While our standard connector is the 0.100 screw terminal, you may substitute any 0.100 connector you prefer. If you are ordering an assembled and tested unit from MRCS and you would prefer a different connector, please contact us at sales@modelrailroadcontrolsystems.com and we'll provide a quotation.

3.2 LEDs

I use blue LEDs with 5.6K limiting resistors for the on board LEDs (used as indicators of programming status), there is no magic to this: I usually have extra blue LEDs because of the packaging I buy my LEDs in, but any color will work fine. The supply is 5V. In case you are new to building up electronics, here's a handy table of typical LED colors and values, the more experienced buider will have his/her own preferences and whatever you have on the bench is probably fine!

Color	Forward Drop	Typical Current	Suggested Resistor Value*
Red	1.5V	10mA	330 ohms
Yellow	1.8V	10mA	330 ohms
Green	1.8V	10mA	330 ohms
White	3.0V	0.5 mA	5.6 K ohms
Blue	3.0V	0.5 mA	5.6 K ohms

Table 5 - Suggested LED limiting Resistor Values

* Using nearest standard value

The formula is supply voltage (5) – Forward Drop (1.5v for Red) = 3.5/current in A, = .01 = 350 ohms.

3.3 Servos

The board is set up for SG90 servos which are adequate for our purpose and inexpensive and have a range of a little less than the specified 180 degrees (note you can buy 360 degree SG90s: don't!). You can use any other analog servo you like but it probably won't fit in the holes in the board. There is no reason not to mount the servos off board if you need to. We are developing dual brackets for this purpose.

3.4 Optional LED power

The change in version 1.1 is power in series with R3 is provided for a LED to illuminate the spectacle of the semaphore. The standard value of R3 is 5.6K, but you change it to use a grain of wheat bulb or some different value that looks good to you with the LED you are using. If you don't want to illuminate the spectacle, you can use a jumper and the "LED" terminal becomes +5 which can be used to power the board or to provide +5 to something else if you're using the barrel jack for power. Starting version 1.

3.5 Optional External Servo Power (Version 1.4 and up)

One customer encountered problems with “random” resets which proved to be the result of inadequate current supply. The servos instantaneously draw about 1A and lower current (higher impedance) supplies will sag under load (the sketch only operates one servo at time to help control this) so you need a minimum of a 5V 1A supply (note many wall warts do not actually supply their rated current with the voltage sagging, so check!). We made a couple of changes to protect against this, adding capacitors C1 and C4 and providing the option of a separate supply for the servos. Use a common ground and connect your servo power to “VIN” on CON2 and set the selector jumper on JP1 to Vin. If you are powering everything from barrel jack and using the 5V supply for the servos, you can omit CON2.

4 ASSEMBLY

4.1 Detailed Assembly Instructions

The components are through-hole technology with wire leads. A lead bender is a useful tool for forming the leads at 90 degrees for easy insertion into the pad holes. See “suggested tools” below if you are new to electronic assembly.

The general rule is to install the lowest components first, working towards components that are higher off the board. This enables you to support the low components as you solder them. Most of the components will stay in place as you flip the board over to solder but, if not, use a small piece of cardboard to hold them in place as you flip the board over. Use 0.015 solder unless otherwise indicated to help control the flow of solder onto the work.

[ ] Resistors

> [ ] Install R7, R8, R9 (R9 only in v 1.1 and higher). Use your lead bender to make 0.400 bends. There is no polarity but for consistency, I put the gold “tolerance” band towards the Arduino. R9 is for the external LED to illuminate the spectacle. If you are not illuminating, you can insert a wire jumper see the LED table in section 3.2 above if you want to use a different color LED or a you can use a Grain of Wheat bulb and a suitable resistor.

[ ] LEDs

> [ ] install 3mm LEDs D1, D2 align the flats on the bottom (Arduino) side. See section 3.2 above for typical values if you choose not to use Blue LEDs.

[ ] Tactile Switches

> [ ] install switches S1 – S6. They should snap into the holes and stay put as you turn the board over to solder them

[ ] Pots

> [ ] install variable resistors (Potentiometers) R1 -R6. These should also snap into the holes and stay put as you turn the board over to solder them

[ ] 15 position female headers for Arduino Nano

> [ ] install the female headers, use a Nano (they usually come with the male headers installed) to hold the female headers in alignment. A properly grounded electronic soldering iron (see tools below) will not harm the Nano, but experienced experimenters will have a dead Nano around to use as a fixture.

[ ] Screw Terminals

[ ] Install the 8 position screw terminal with wire openings pointing off the board.

[ ] Install the 2 or 3 position 3.5mm screw terminal for DC power for version 1.3 and up

[ ] Install the DC Power Jack (if used)

[ ] Use 0.031 solder for this. Fill the holes with solder so there are no gaps.

[ ] Install the Male Headers

[ ] install the 3 position male servo headers, on the bottom of the board facing down

[ ] install the 3 position male power selector header, on the bottom of the board facing down (Version 1.4 and up)

[ ] Install the Servos (if mounting on board)

[ ] you can mount them in either orientation but it's hard to capture semaphore actuator wires if the servo shaft is towards the top.

[ ] slip the servo cable through the mounting hole, and gently wiggle the servo into the hole (foil label may come off in the process) and push it firmly against the board. The secure with 2 #2 x ¼ or metric 2.6 x 6 sheet metal screws. You may want to hit the tip of the screw with a file to make it less sharp when you're done

THIS WOULD BE A GOOD TIME TO MAKE SURE YOU DON'T HAVE ANY SHORTS!

Use your bench supply with a banana jack to 2.1 mm power adapter and apply 5V. (a 5V wall wart with your volt-ammeter in series will do also, a few clip leads will be handy.) You should be drawing 3-5 mA current and the output of the supply should hold at 5V. If it's drawing more current, look for solder bridges or pins that haven't been soldered. If less, make sure the solder connection to the barrel jack is good.

[ ] install the Nano

Insert the Nano into the two female headers. (I'm assuming you've already programmed it) with the USB connector facing towards the DC power Jack (the board says "USB"). Make sure the headers are lined up (not spilling over one end). Try powering up again, you should be drawing about 50 mA.

[ ] Plug the servos in

Note the colors marked on the board: Orange to Orange and Brown to Brown.

4.2 List Of Handy Tools For The Assembler

- Lead Bender
- Side Cutters
- Small Needle nose Pliers
- Temperature controlled solder iron. Don't skimp here, this is a very useful tool for everything you do in model railroad electronics, get one with replaceable tips, the finer the better, Weller (such as WE1010 NA) and Hako (FX888D) make very nice irons that balance well in your hand for <$150 both available from Digikey and Amazon.
- 0.015 solder for fine pitch items like 0.100 connectors (also handy on decoders)
- 0.031 solder for larger items like the power connector
- Isopropyl alcohol 91% or 99% for cleaning left over flux off the board
- Bench power supply – this gives you precise control of the voltage and allows you to measure and limit current as you test. You can see if you are drawing the correct amount of current, if not that's an indication that something is wrong. These come and go so I'm not including a link but an Amazon or AliExpress search on "bench power supply" should provide many capable options that won't break the bank.
- Digital Volt-Ohm-Milliamp meter. This really comes down to features like quality of the probes and how the stand works, even Harbor Freight has very functional meters for as little as $10.

5 Software And Programming the Arduino Nano

The Arduino Nano is programmed using Jon Schmidt's "semaphore3.2" [check Jon's GitHub site https://github.com/joneschmidt/SemaphoreControl to be sure you have the most current version of the sketch] sketch using the free Arduino Integrated Development Environment (IDE). You can get it at https://www.arduino.cc/en/software . The IDE is available for Windows, Mac and Linux. There are many tutorials and guides on how to download and There are many tutorials and guides on how to download and install the Arduino IDE if it doesn't just work for you. Note that most commodity Nanos use the CH340 serial chip and you may need to find and download the driver. They usually are supplied by the Asian vendors who make the chips and look pretty bare bones, but they work!

When you start the IDE, go to the "tools" menu:

5.1 Windows

I've found the following settings to work on my Windows 10 and 11 machines.

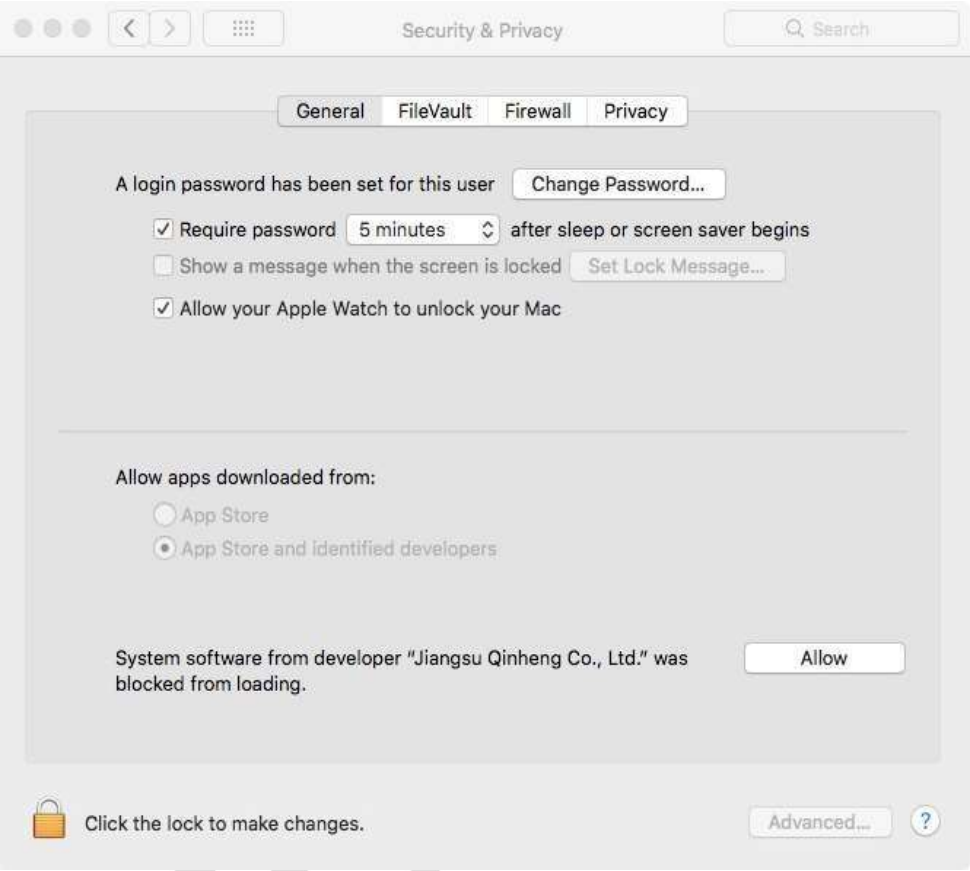
> Go to Board, mouse right and select "Arduino AVR Boards" and select "Arduino Nano"
> Go to Processor, mouse over to the right and select "ATmega328P (old bootloader)"
> Go to Programmer, mouse over to the right and select "AVRISP mkii"
> Go to Port and be sure it's not COM1 (if that's the only choice you probably have to install the CH340 driver, or at least reboot to make your machine sees it).

5.2 Mac

A Mac friend suggests these for the Mac:

> Note that as of this writing (August 2025) I am informed that Mac comes CH340 drivers so this procedure may not be necessary.
>
> Download the Arduino IDE from https://www.arduino.cc/en/Main/Donate
> double click the arduino-1.8.5-macosx.zip (or whatever is most current) file and drag Arduino.app into your Applications folder
> Download serial driver from https://github.com/adrianmihalko/ch340g-ch34g-ch34x-mac-os-x-driver
> double click the ch340g-ch34g-ch34x-mac-os-x-driver-master.zip file
> double click the CH34x_Install_V1.4.pkg file to begin installation

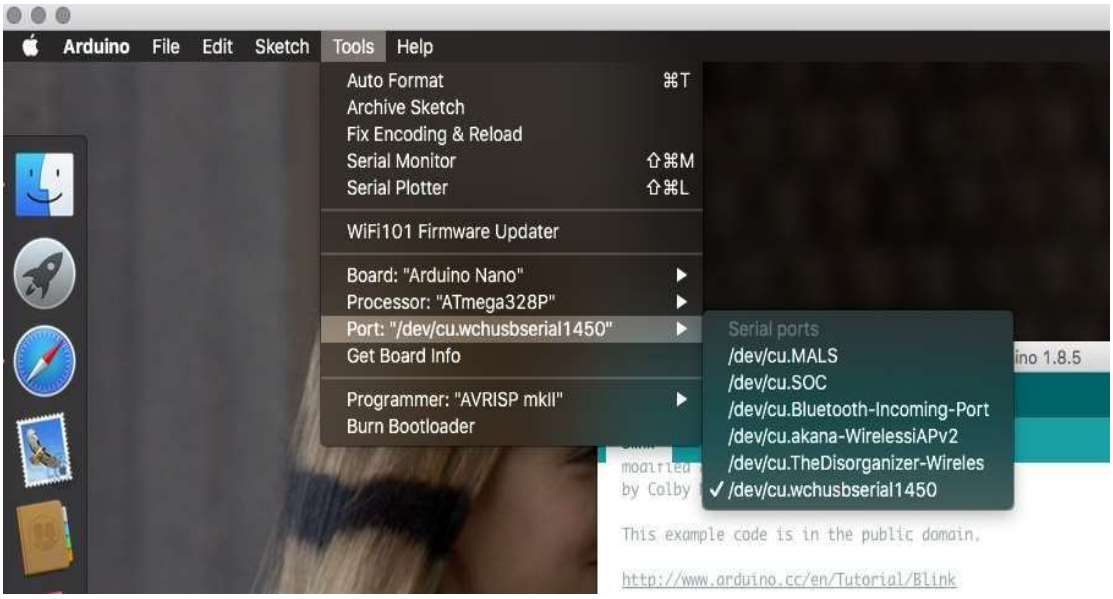
click Open Security Preferences



Click Allow

Restart your Mac – it takes a little time to rebuild the driver cache – don't panic!

Run Arduino.app from your Applications folder. There will be a prompt whether you "really want to do this". Say ok - because hey - we've got a semaphore to program!



To prove to myself this worked I followed the instructions at https://www.arduino.cc/en/Guide/ArduinoNano

“Open your first sketch”. Once you upload it the board red light will blink regularly.

If you’re using Linux, you can probably figure this out for yourself.

Since the D0, D1 serial data lines on the Nano are not connected to anything, it’s OK to leave the Nano on the Dual 3 Position Semaphore Controller while reprogramming.

6 USING AND MODIFYING THE SEMAPHORE SKETCH

6.1 Design

Control two servos for driving semaphores. The board contains 3 variable resistors (pots) for each servo. These pots are labelled Red/Yel/Grn for the 3 possible positions for a blade. There are also 3 pushbuttons, labelled Red & Yel & Grn, for a blade. The buttons and pots are aligned along the left and right of the board. The left set controls the left servo, the right, the right servo. The connector block on the bottom of the board connects to the external buttons or switches or logic inputs for driving the semaphore indication.

6.2 Setup

The servo will follow the relative positions of the pots for each requested indication. If Red is requested the current setting of the Red pot will be used to position the servo and blade. If Yellow, the Yellow pot is used. Green, Green. Therefore, some care must be used to set the pots appropriately so that the blades fall into the correct position. This tuning of the pots/blades can happen during normal operation. However, there is a special period during a power-up or reset which can be used. Note that there is a define, *#define UseMidYellow true*, which sets the blade yellow position midway between Red and Green, so the Yellow pot may be ignored.

Setup at Power-up or Reset

At power-up or reset, the board goes into a mode where the pots may be changed to set blade positions. During this time the LEDs on the board will flash in a moderate alternating pattern. If no pot positions are changed for 15 seconds, the board transitions into normal mode and the LEDs will flash in a slower pattern.

However, a change in the position of any pot will cause the corresponding servo & blade to move to that position. This allows a period in which each blade position may be set. A change in any pot will also reset the 15 second timer, so the user may take as much time as needed to set blade positions. If no pots are changed in 15 seconds, the board progresses to normal operating mode.

We recommend that for initial installation, all the pots be set to mid-point and the blades connected. From that point, the red/yellow/green positions can be set as desired. Rapid flashing of the LEDs is an indication that more than one input signal was active at the same time, possibly as a result of an accidental cross-connection.

The blades will move to each position Red/Yel/Grn at the beginning of the transition into normal mode. Then they will move to the requested position based on input signals. In normal mode the pots can still be changed. Each time a request is received to change the blade indication, the corresponding pot is read and the blade is positioned accordingly.

Software controls:

```
// signals/servos – don't change

#define NumSigs 2

// FullArc - the maximum degrees of arc to travel

#define FullArc 180
```

The servos can be driven to 180 degrees of motion. This is unnecessary as smaller changes are useful for driving a semaphore. The potentiometer values are mapped to this FullArc range.

```
// MinPotChg - the minimum pot change we will act on #define MinPotChg 5
```

The software doesn't act on a change to the potentiometers unless that change exceeds this value. The normal pot range is 0 - 1023.

```
// HoldMS - time before looking for a new signal for that semaphore
#define HoldMS 1500
```

The software doesn't act on a change to a semaphore until this time limit from the last change.

```
// latch = false – if true, hold last active button status
#define Latch false
```

Set false the software will scan repeatedly the input switches. This assumes that the switches will constantly present their state. Some panels use an enabling momentary signal to enact the signal change. By setting this Latch to true the software will maintain the last presented state of the inputs until a new state is detected.

```
// mirror = true - upper moves counter-clockwise, lower moves clockwise
#define Mirror true
```

The servos are in pairs. Depending on how the card is mounted in relation to the semaphore, you may wish the servos to move in the same direction, or in opposing directions. Mirror = false will cause the servos to move in the same direction. Mirror = true will cause the servos to move in opposite directions.

```
// debugging
#define Trace 0
// main loop delay in milliseconds – make large for debugging
#define LoopDly 1000
```

Setting Trace to non-0 will enable messaging to the console for debugging purposes.

Version 2.2 Update 5/2/2021:

Added code to produce a “bounce” in the semaphore action when the blade returns to the Red position. The BounceArc controls how much of a bounce. A factor of 3 says that the bounce will be one third of the Green-Red arc, and decrease by 1/3 until the blade comes to rest. A factor of 0 means no bounce. The BounceDly controls the speed of the bounce.

```
// Chg 2.2 vvvvvvvvvvvv
// BounceArc - factor for arc of bounce
//   - (green-red)/Bounce
//   - 0 means no bounce
#define BounceArc 3
//   BounceDly - delay between moves
#define BounceDly 300
// Chg 2.2 ^^^^^^^^^^^^
```

Version 3.2 Update 11/22/2024:

Fix to the code for the Red bounce such that it bounces from the last position, not from Green.

Added code to create an “overrun” in the movement to Yellow position. This is to account for linkage slack in some custom signals.

```
// YelOverun <> 0 - overrun (underrun) the yellow position (degrees) and return (+-0)
#define YelOverun 0
// YelOverunDlyMS - time for overrun to take effect
#define YelOverunDlyMS 300
```

7 TESTING

Testing your Dual 3 Position Servo controller is straight forward:

1. I recommend setting BounceArc to 0 while setting the position as it’s a little easier to see what’s happening, you can reset it later without affecting your blade position settings. You may also want to set HoldMS to 500 so it is a little more responsive to changes.
2. Power up and starting with one blade set Red, then Green for the ends of travel and then adjust Yellow as desired if UseMidYellow is false. Use the pots and buttons to set red, yellow, and green. Remember the board won’t look for a change for the time HoldMS after the Setup period
3. Repeat for the second arm.

8 INSTALLATION AND CONNECTIONS

8.1 Power Connections

Power is 5 Volts DC, a servo can draw up to 0.25A when moving, the Dual 3 Position Semaphore Controller will only move one servo at a time, so the maximum current is about 0.3A, to be safe use a supply rated for a minimum of 1.0A. (We've seen problems with spontaneous resets with supplies supplying less than 1A – beware not all "wall warts" can supply their rated current). The Dual 3 Position Semaphore Controller has a 2.1mm barrel jack for DC power (typically from a wall wart, but you could tap off a local 5V bus or use our "Buckeroo" power distribution board to derive 5V from a 12 V bus. You can also supply 5V to the 5V and ground terminals of the 8 position screw terminal block (V1.0 or make the mod described above, or the green terminal block on revision 1.4 or above) on the Dual 3 Position Semaphore Controller. If you have multiple Dual 3 Position Semaphore Controllers in the same area, you can daisy chain from one to another assuming the power supply can source enough current.

8.2 Servo Control Connections

Setting the position manually with a switch

If a train order operator is controlling an order board, single pole, double throw, center off switch provides a simple interface. The center position is RED, “STOP” or “Stop and sign for a form 31 train order,” one of the “ON” positions is YELLOW, “Slow down to pick up a form 19 train order on the fly” and the other ON position is GREEN, PROCEED. The center pole of the switch is wired back to the same ground as the Dual 3 Position Semaphore Controller.

8.2.1 Toggle Switch Control

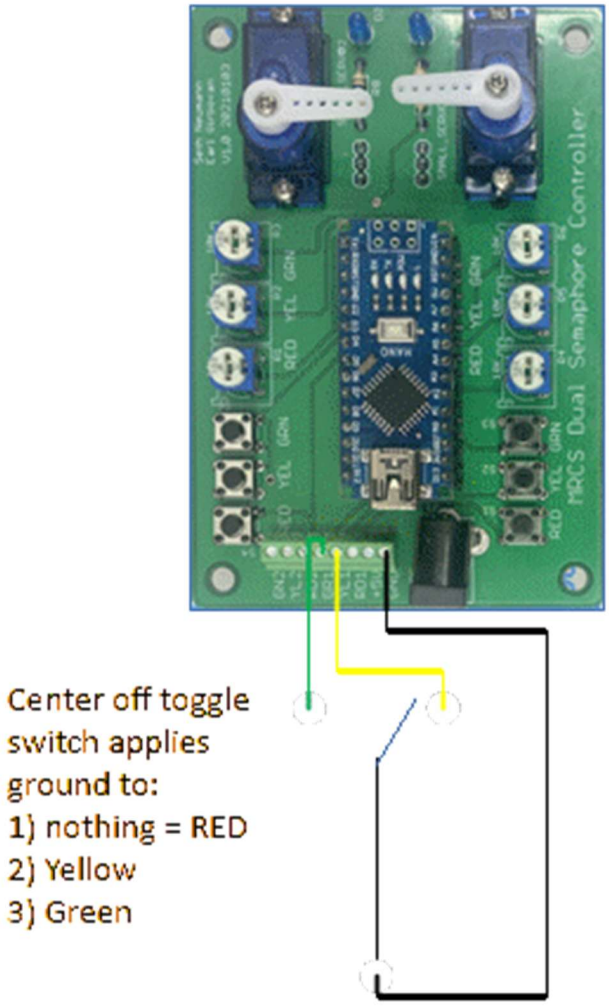


Figure 18 - Toggle Switch Control

8.2.2 Using the Dual 3 Position Semaphore Controller in an Automatic Block Signal (ABS) System

A good example is Dennis Drury's Dual ABS Signal Controller (DABSC) which reads block occupancy detectors and drives the controller. The DABSC has red, yellow and green outputs and they are connected Yellow out from the DABSC to Yellow in on the Dual 3 Position Semaphore Controller and Green out on the DABSC to Green in on the Dual 3 Position Semaphore Controller.

JRMI or Dr Chubb's BASIC Code can do the same for an ABS or APB system, the Dual 3 Position Semaphore Controller looks just like a 3-color signal to the logic except you don't need to connect the RED output as RED is the default position.

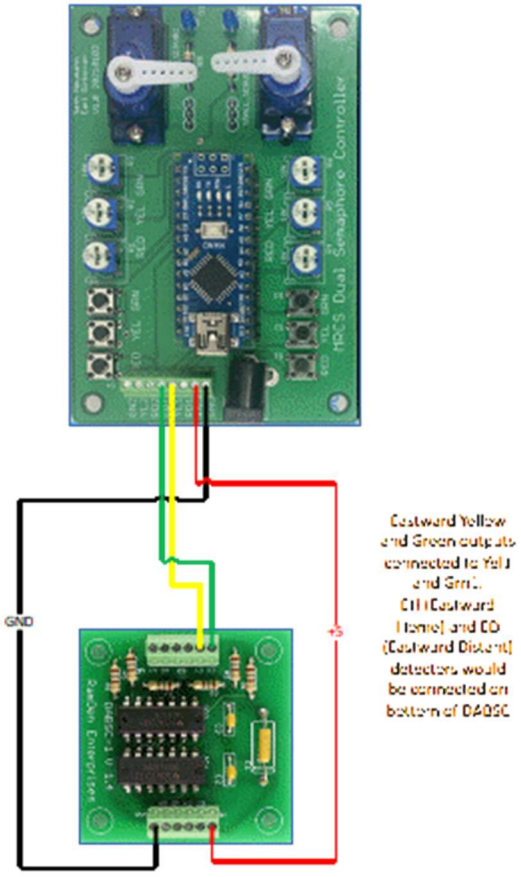


Figure 19 - DABSC Control

8.2.3 DCC Control

George Sinos suggested using an NCE Light-It to control the Dual 3 Position Semaphore Controller (D3PSC).

- the Light-It has 3 outputs, (Wht)Grn/Yel/Red and a 5 volt source. The Light-It is designed to drive LEDs directly and includes 330 ohm resistors in series with each output. Since there is no (official) ground reference, I tied the 5V to the 5V screw terminal on the D3PSC to provide a reference voltage (we're not drawing any power from the Light-It).
- Then I tied the Yel output of the Light-It to the Green1 terminal of the D3PSC (I'll explain later) and the Red output of the Light-It to the Yellow1 terminal of the D3PSC.
- I had previously programmed the Light-It as mobile decoder, address 1000 and kept the configurations for its 3 outputs as effect 1: "steady on when the corresponding function button is pressed" so Green is function 1 on the cab and Yellow is function 2. The D3PSC defaults to RED, so there is no need to connect to the Red1 terminal.
- So why didn't I use the GRN output from the Light-It? It turns out NCE has installed a white LED on the (Wht)/Grn output as an indicator to help with programming, and because of the resistor value selected, the GRN output can only get down to about 2.5 volts when connected to the Nano, which does not ensure reliable detection of a "logic low" input. However, the Yel and Grn outputs will pull the voltage down below 0.4V which is well within spec. The Light-It outputs can be re-mapped. You can also cut a trace and remove the LED from the circuit but then you are programming blind
- NCE also supports Accessory Decoder address and Signal address, which would be handy for an ABS-with-semaphores layout. I didn't try these options, but I expect they work fine.
- My test setup powers the D3PSC from a bench power supply and the DCC system is an NCE Power Cab, powered by the included wall wart. It is possible that some DCC systems may be referenced to ground and have incompatible voltages resulting in possible damage to the D3PSC or Light-It, so check with a voltmeter to see if there is any significant voltage between the 5-volt terminals of the D3PSC and Light-It before connecting.

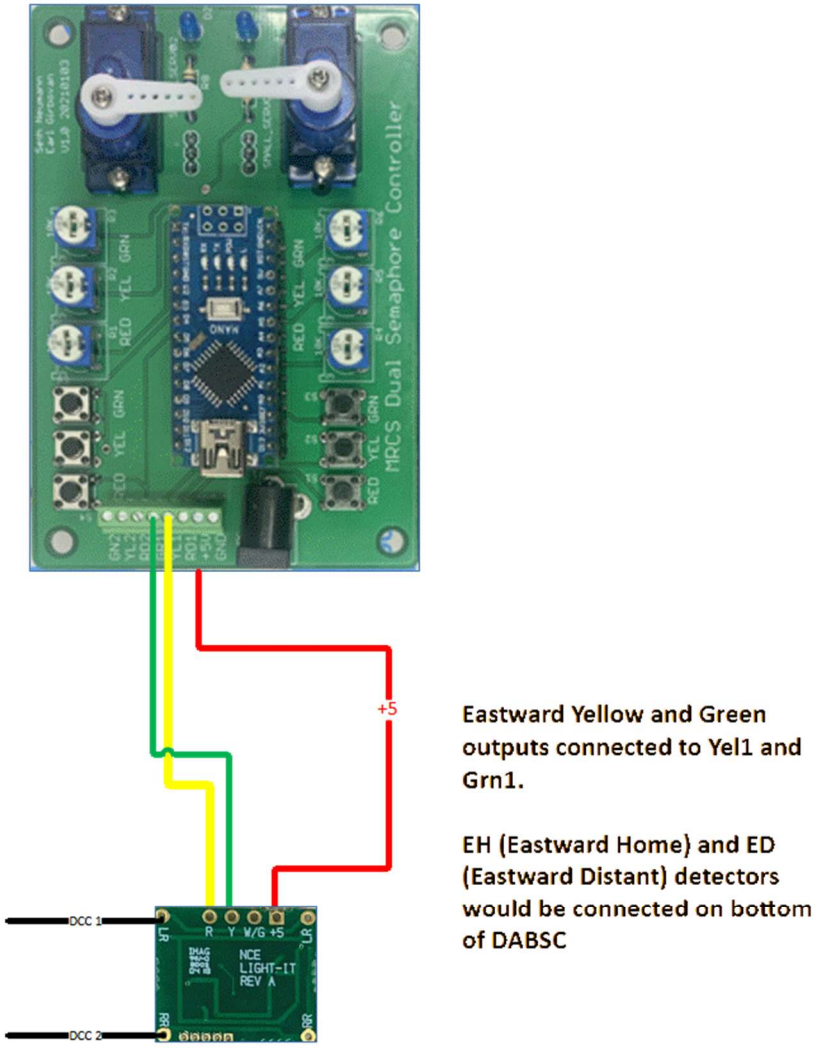


Figure 20- NCE Light-It

8.2.4 Quick Install with Showcase Miniatures Train Order Board

1. I Started with a handy piece of 1x3.
2. Cut it into 6”, 6” and 1’ pieces.
3. Drilled a 3/8” hole with a Forstner bit in the middle of the 1’ piece for the base for the Showcase Miniatures #2365 magnetic semaphore base (note: once the actuator wires for the semaphore are installed it will be tricky to remove the semaphore from the base.) The Forstner bit provides a hole with flat bottom for the base, easing alignment. The 2365 is a handy magnetic base that will release if the mast is bumped and, although the semaphore linkage may limit this movement, in any case it is more resilient than just gluing the semaphore in.
4. I screwed the 6” pieces into the bottom of the 1’ piece with the Forstner hole, then drilled a (~1/4”) hole thru the Forstner hole for the actuator wires to clear. You could cut a slot for this but it’s a lot more trouble and isn’t needed to support the mast.

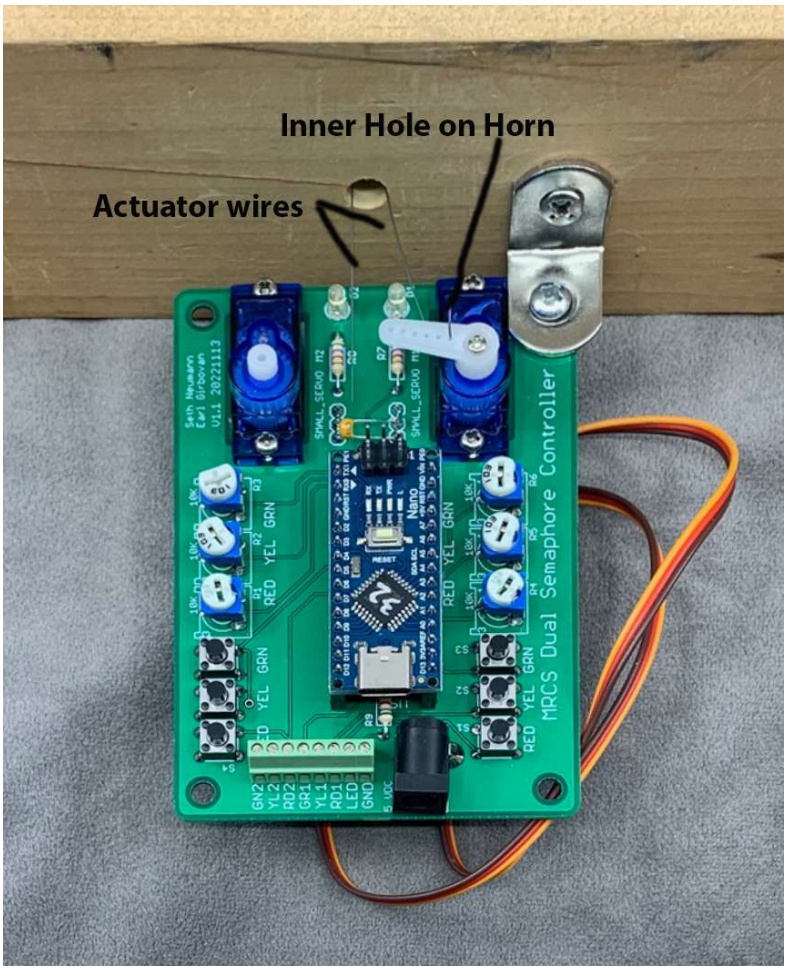


Figure 21- close up of board with actuator wires

5. I Installed a corner bracket to hold the Dual Servo Semaphore Controller to the bottom of the 1' cross piece. Note the ones I found on Amazon are just a little too big, so either get smaller corner brackets that fit or make your own from 1/2" angle or strip stock.
6. I installed the PC board about ½" below the 1' cross piece with the servo horns centered below the hole. Note in figures 1 and 3 that only the right horn is installed, the procedure is the same for the other horn and actuator.
7. I used one of the double ended horns which had a hole very close to the hub on the first servo, I cut off the other end of the horn as it was just in the way. (Walter from Showcase Miniatures recommends "Hobbypark Adjustable D2.1mm Pushrod Connector Linkage Stoppers RC Model Airplane Replacement" for easy adjustment of the actuator wire. This is an Amazon link from December 2023, so you may need to hunt around for something similar, but RC linkages are your friend, just beware of backlash in all the bell cranks and similar connections.)
8. Apply power (5V DC regulated, >= 1A) and adjust the horn and RED travel so the horn is about horizontal, this will be the RED position and the blade will be horizontal, then make green about 15 degrees higher (pushing the rod up makes the blade dip). The green will want to be about 75 degrees down. By default the yellow is halfway between Red and Green, but you set it manual adjustment, if you're using a LED or GOW bulb to illuminate the roundels.
9. Turn power off while you are connecting the semaphore's actuator.
10. Now bend the actuator wire (you are controlling the opposite side of the semaphore from the servo) so it goes in the closest hole to the hub. (see item 7 above)or use the pushrod connector linkage.
11. Power the Dual Servo Semaphore Controller back up and make final adjustments to the positions to your preference.
12. Note that if the Dual Servo Semaphore Controller is constantly going back into program mode (rapid flashing of the LEDs) after the servo moves, you probably have either a noisy power supply or your supply is not putting out enough current. Use at least a 1A regulated supply (we offer a 5V 1A supply but we stock 2A, in any case if you need one from us, please send me a note and we'll make sure you get a 2A supply!).
13. The latest versions of the board, V1.3-V1.6, have additional filter capacitors on board, but this is not a substitute for a stiff power supply!

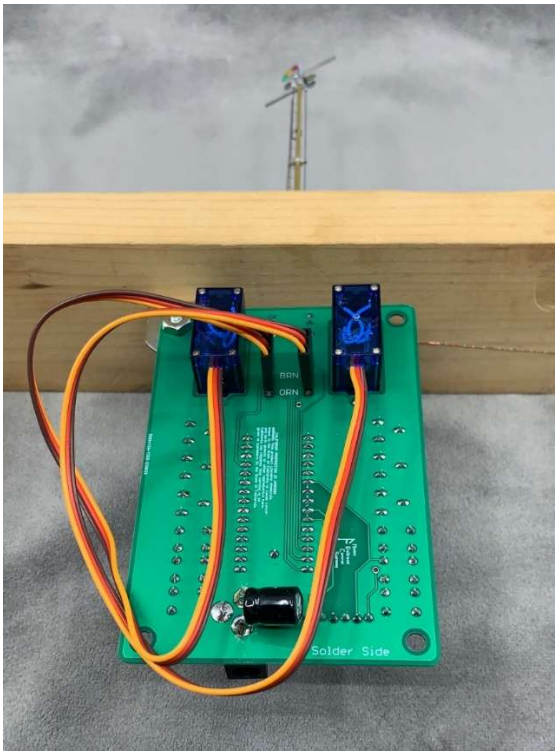


Figure 22- back side of PC Board - the capacitor at bottom is the prototype for V1.3

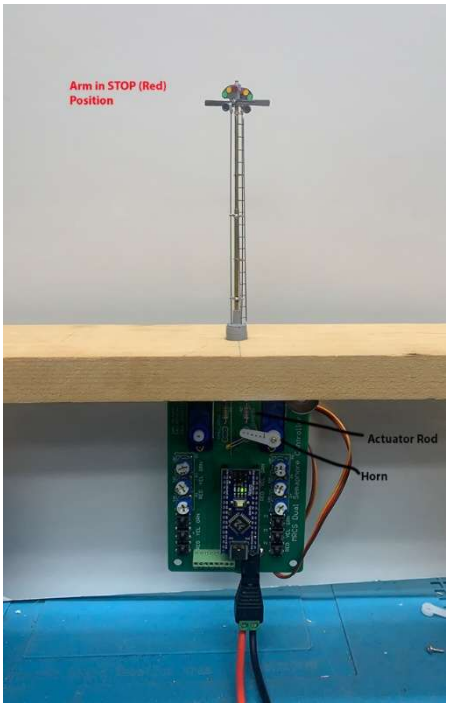


Figure 23- Showcase Miniatures Train Order Board with Dual Semaphore Controller

8.2.5 Hobbypark Adjustable Pushrod Connector Linkage Stoppers D1.3mm

As of Version 1.6 we've been including a pair of pushrod connector linkage stoppers. These are available from the usual on line marketplaces in packages of 25 including a 1.5mm hex key. We only supply the linkages. I recommend acquiring a 1.5mm hex key and 4mm hex nut driver to install these. You'll also need a #47 drill for the threaded stud on the stopper.

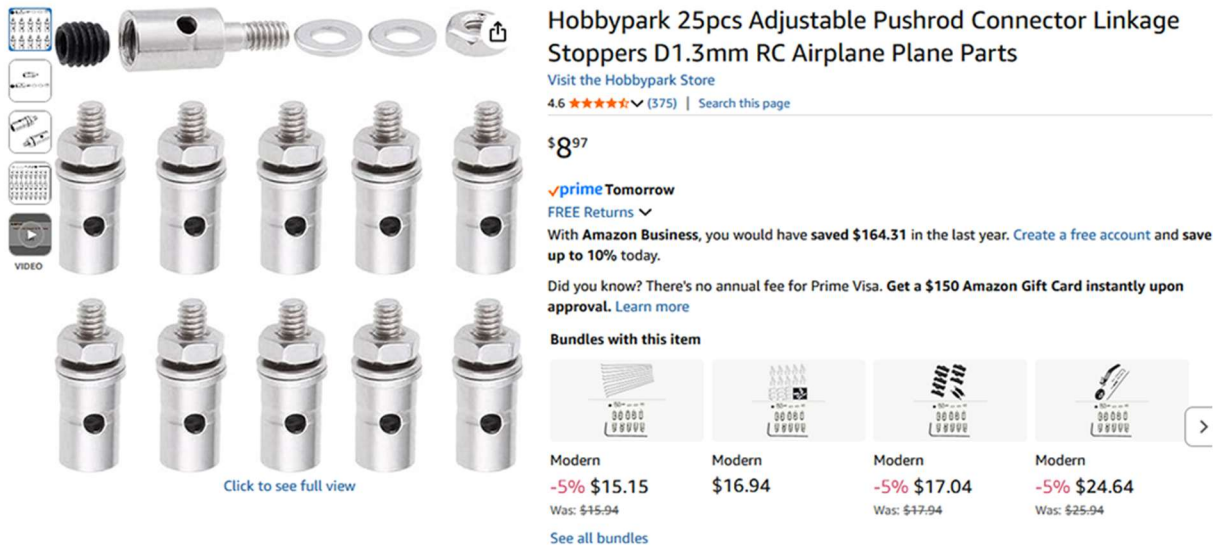


Figure 24 - Amazon Page for Pushrod Connector Linkage Stoppers

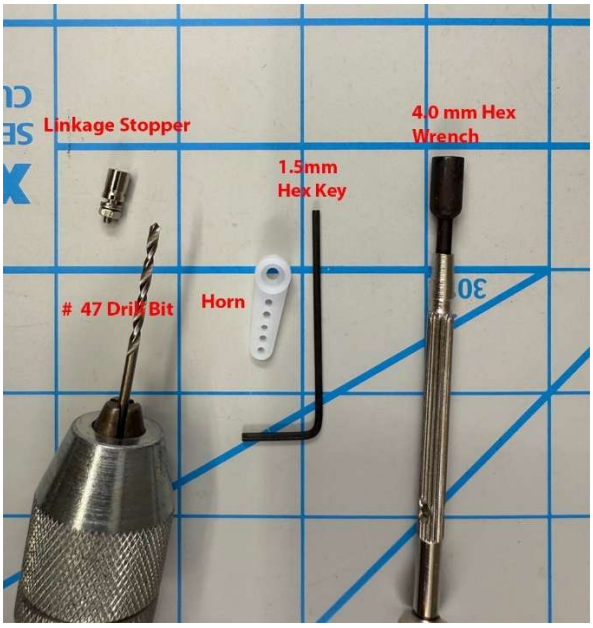


Figure 25 - Parts and Tools

Figure 26 - Handy Metric Tool Kit from HobbyPark

The tool kit above has nice hex keys with handles, but they may be too big for confined spaces under the layout. The bright metallic tool has 4mm metric socket for the nuts. Metric screw, socket, hex etc kits can be found at any of the tool vendors on line or at train shows or r/c hobby dealers.

You will want to use the single sided horns and use the inner or 2nd hole, depending on clearance (you may be able to get in closer by filling the little gussets on the horn). The stud is a little larger than the holes in the horn, so you'll want to drill it out with a #47 drill, I used a pin vise, but a rotary tool will work.

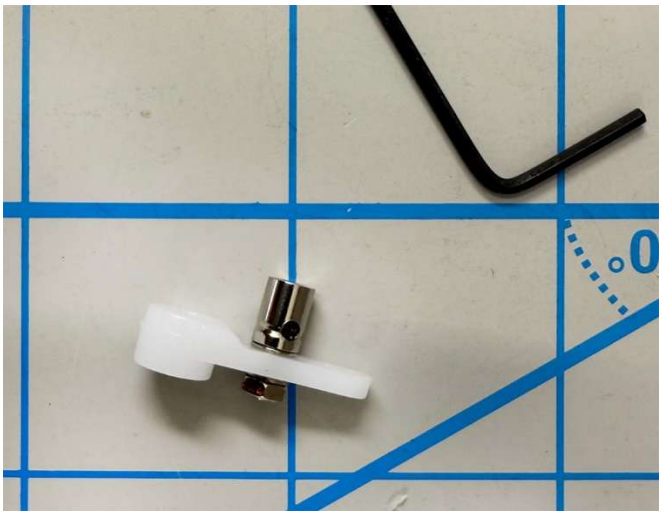


Figure 27 - Stopper Installed

Align the hole so it's more or less parallel with the actuator wire from the semaphore. Tighten the nut with the 4mm wrench. Put the horn on the servo and run the servo to one end or the other so you can start the alignment process. When you're done screw the horn into the servo.

Tighten the hex screw in the barrel of the stopper with the 1.5mm kex key.

Figure 28 - Stopper with Actuator