

Open Source Rover: PCB Testing

Author: Eric Junkins, Michael Cox



Jet Propulsion Laboratory
California Institute of Technology

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. ©2018 California Institute of Technology. Government sponsorship acknowledged.

Contents

1 Printed Circuit Boards	3
1.0.1 Printed Circuit Boards	3
1.1 Voltage Divider Board	3
1.1.1 Overview/Schematic	3
1.1.2 Testing	6
1.2 Serial Splitter Board	8
1.2.1 Overview/Schematic	8
1.2.2 Testing	9
1.3 PCB - Logic Shifter Board	12
1.3.1 Overview/Schematic	12
1.3.2 Testing	13

1 Printed Circuit Boards

1.0.1 Printed Circuit Boards

In order to help simplify the electronics of the rover, we designed a few single-purpose Printed Circuit Boards (PCBs). In total there are three boards:

Board	Description/Purpose
Voltage Divider	Distributes the +5V and GND to power the absolute encoders. Also takes the 0-5V signal from the encoders and transforms it to a 0-2V signal that the controllers expect
Logic Shifter	Takes the 3.3V digital signals from the raspberry pi and converts them to a 5V digital signal output
Serial Splitter	Takes the TX/RX serial read/write lines and the +5V and GND power pins from the Raspberry Pi and distributes them to each motor controller

Each of these boards will come unpopulated (without components on it), so you will have to solder the components onto the boards yourself. For tips on how to solder components into the circuit boards, see:

- https://www.youtube.com/watch?v=3R0zE_7Xz_4

1.1 Voltage Divider Board

1.1.1 Overview/Schematic

The purpose of this board is to solve the mismatch between the encoder logic levels and what the motor controller expects as encoder inputs. The absolute encoder outputs a signal from 0-5V based on the position of the encoder shaft, but the motor controllers expect a signal that is 0-2V. To make this transformation, we will create a linear voltage divider.

The following schematic is the electrical diagram for this board. For information about how to read electrical schematics, see:

<https://learn.sparkfun.com/tutorials/how-to-read-a-schematic>

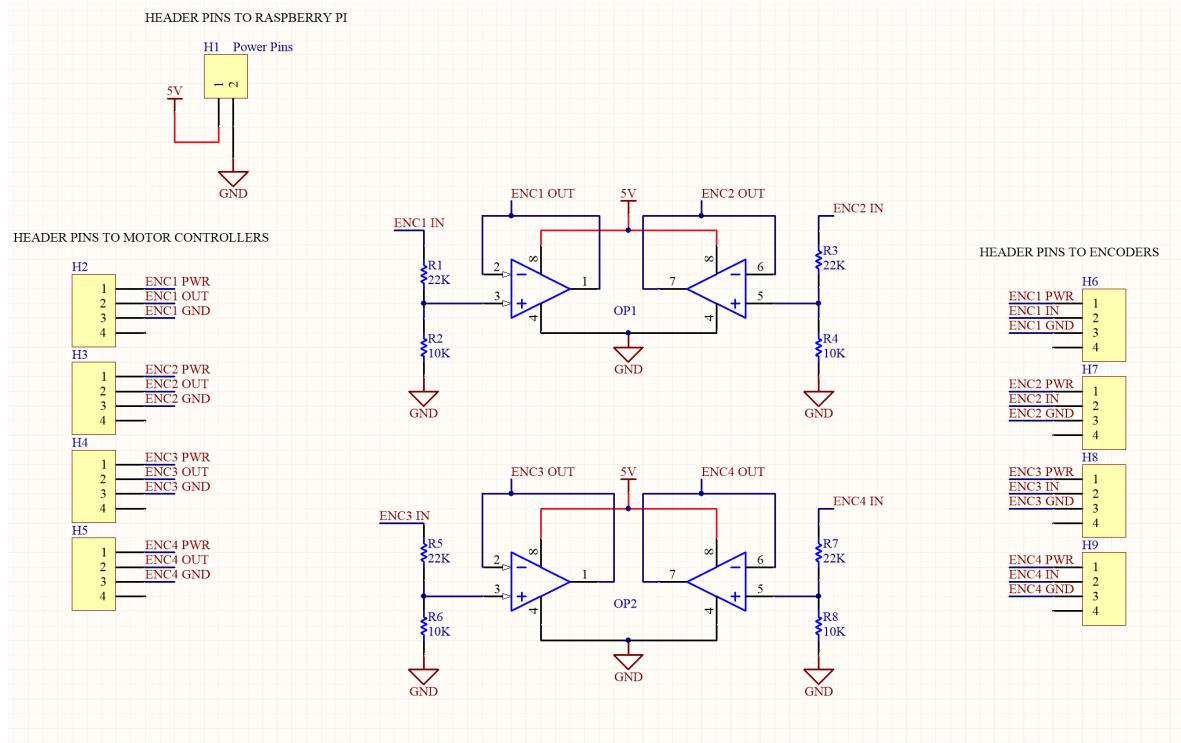


Figure 1: Encoder voltage divider schematic

The operational amplifier (op-amp) chip we chose is a dual channel op-amp, so it contains two separate input/outputs on each chip. The pin-out can be seen in Figure 2. For more information on this op-amp, see its data sheet:

- <http://www.ti.com/lit/ds/symlink/lm358.pdf>

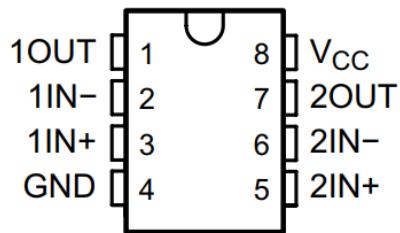


Figure 2: Op-amp pinout

In Figure 1 we can see the electrical schematic for this board. Duplicated 4 times on this board is a voltage divider with a voltage follower after it. This voltage divider with the resistor values of 22K and 10K ohms will give a theoretical voltage division of 0.3125, which is derived from the following equation:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2}$$

However, in the real world we've observed that the voltage division of this chip is actually closer to 0.28. This difference is caused by non-negligible resistances of other electronic components in the system. Any voltage division by a factor of 0.4 or less will be sufficient, as we need to get our 5V signal to no more than 2V. If you wish, you can tune resistor values to get this closer to 2V in your specific application.

The op-amp is used as a voltage follower, allowing you to decouple the resistors and the impedance/capacitance of the motor controllers. This means that the voltage divider as a whole will maintain its voltage division factor despite the changing environment around it. You can see more information at:

- https://en.wikipedia.org/wiki/Voltage_divider
- <http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/opampvar2.html>

1.1.2 Testing

Parts/Tools Necessary

Item	Image	Item	Image
Serial Splitter Custom PCB		LM358P Op-Amp	
0.1 Pitch Header pins		Digital Multimeter	
10K Resistor		Solder Iron	
22K Resistor			

Figure 3

Figure 4 shows how the pins are labeled and should aid in debugging the board and testing the signals at the various pins.



Figure 4: Encoder Voltage Divider board pinout

1. The board will come unpopulated and will look like the board in Figure 3. First, solder all of the header pins, resistors, and op amp chips into the board.

THIS SECTION NEEDS PICTURES ADDED FOR PIN-OUT DEBUGGING. WILL BE ADDED ONCE WE GET THEM FROM SANTA SUSANA

Make sure that the OP-amp is facing the correct direction, there is a half moon on the top of the chip as well as on the silk screen on the board.

2. Once soldered the board should look like Figure 3
3. Apply +5V onto pin 25 and GND on pin 26. This should supply power to the components on the board.
4. Test each of the OP-amp chips by probing from pin 8 to 4 on each chip and verify that they have +5V.
5. Apply +5V onto pin 1 and GND onto pin 3. Verify that there is +5V on pin 13 and 15.
6. Apply +5V onto pin 4 and GND onto pin 6. Verify that there is +5V on pin 16 and 18.
7. Apply +5V onto pin 7 and GND onto pin 9. Verify that there is +5V on pin 19 and 21.
8. Apply +5V onto pin 10 and GND onto pin 12. Verify that there is +5V on pin 22 and 24.
9. Now that we know that each of the encoders will be powered by +5V correctly, we want to test the voltage division. Apply +5V to pin 14 and measure the output at pin 2. This value should read between 1.4V and 2.0V. It is important that it is strictly less than 2V, but slightly lower values are also okay. Resistor values can be changed if desired to make this closer to 2V.
10. Repeat step 8 with +5V on pin 17 and testing pin 5, then +5V on pin 20 testing at pin 8, and +5V on pin 23 testing at pin 11. Once all these are verified, the board is working correctly and is ready for the rest of the project! If any of the tests fail, check that you are using the correct resistor values and that the op-amp chip is functioning correctly and hasn't burned out.

1.2 Serial Splitter Board

1.2.1 Overview/Schematic

This board is to pass the serial read and write lines, as well as +5V/GND from the Raspberry Pi to each of the motor controllers without having to solder wires to split each of these individually.

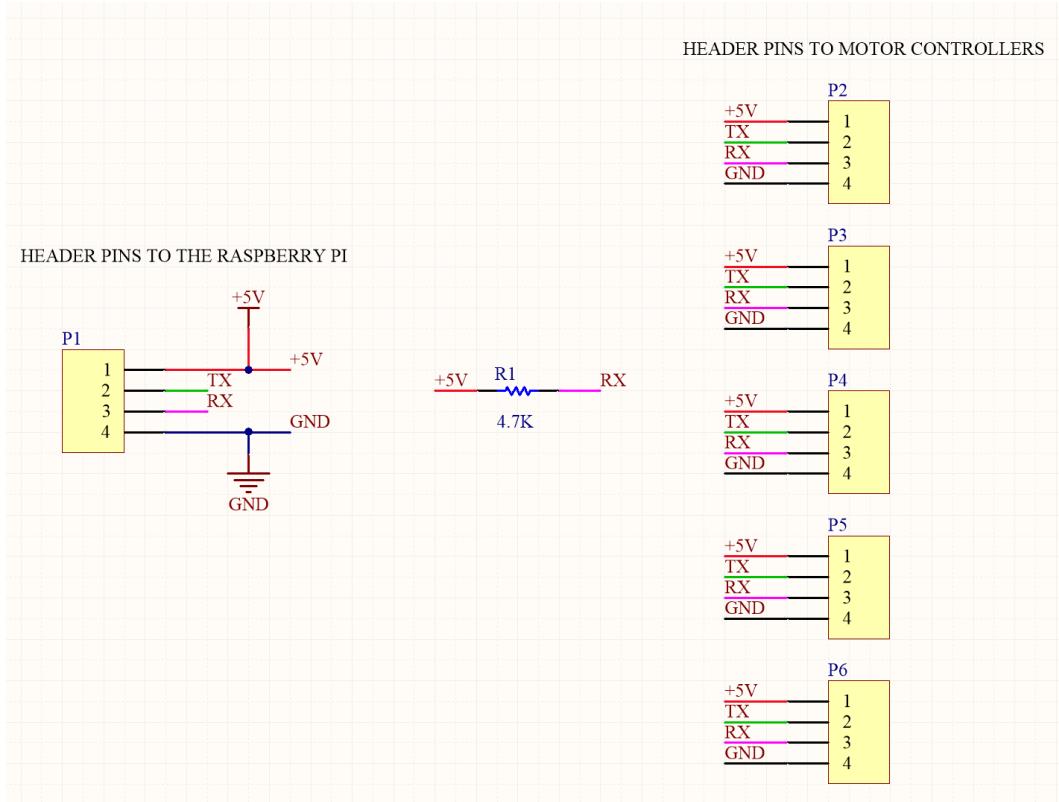


Figure 5: Serial Read/Write schematic

Figure 5 shows the schematic for this board. There is one resistor used as a pull-up resistor as recommended when using multiple serial communication devices from the RoboClaw user manual. +5V and GND are also passed to each motor controller, as these are what is used for the logic battery on the RoboClaw which determines what digital logic and signal power is sent to the encoders.

1.2.2 Testing

Parts/Tools Necessary

Item	Image	Item	Image
Serial Splitter Custom PCB		Digital Multimeter	
0.1 Pitch Header pins		Solder Iron	
4.7K Resistor			

1. The board will come unpopulated and will look like the board in Figure 6. First solder the 4.7K resistor and all of the header pins into the board.

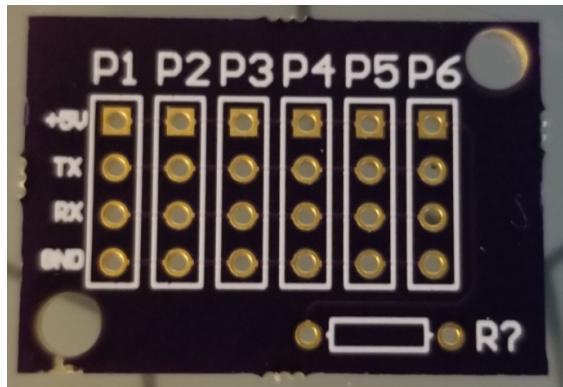


Figure 6: Unpopulated Serial Splitter Board

2. Once soldered on the board should look like Figure 7

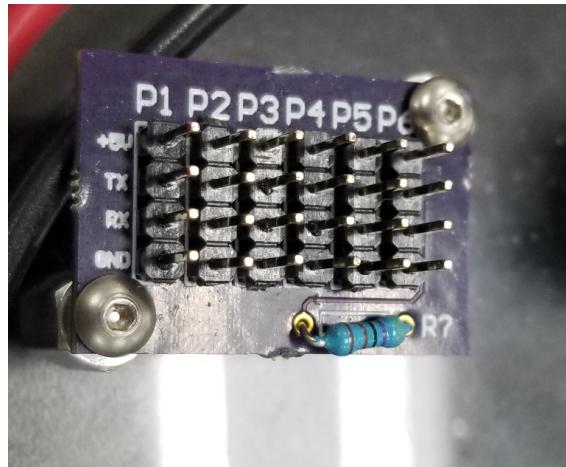
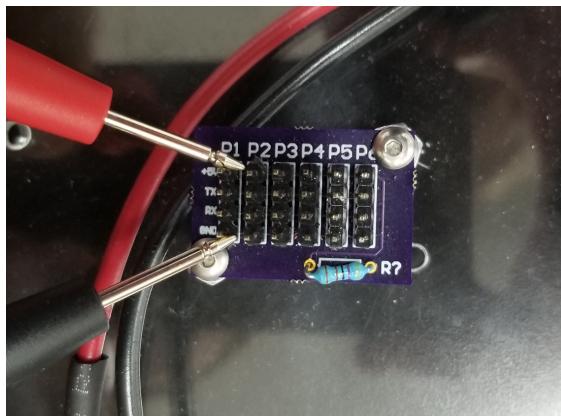
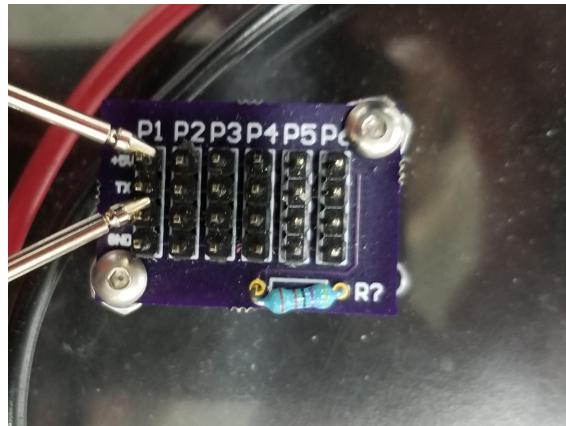


Figure 7: populated Serial Splitter Board

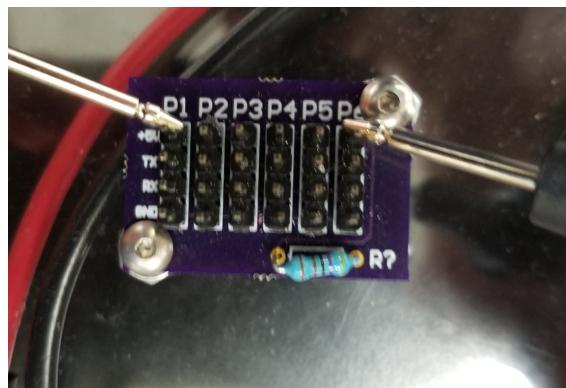
3. We first want to test and make sure that each of the rows are independent of each other. Using the resistance setting on your DMM, make sure that there is no resistance between any pins in the first row (+5V) and any pins in the 2nd (TX) or 4th (GND) row. Said another way, you should be testing the resistance between the +5V pin and the TX and GND pins for each column.



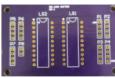
4. Now, test the resistance between row 1 and row 3 (the +5V row and RX row). This should be equal to the resistance of R1, which is 4.7K Ohms. Repeat this test for each column.



5. Every row should be connected all the way across, so verify that the first pin is connected to each of the other pins in the row using the diode setting on your DMM. If all of these tests have passed, then the board is all ready to go! If not, check your solder connections and make sure there isn't any short on the board and then rerun the above tests.



Parts/Tools Necessary

Item	Image	Item	Image
3D printed Head		20 AWG black wire	
Logic Shifter Board		20 AWG red wire	
74LVC245 Logic Level Shifter		30 AWG colored wire	
Female/Female Jumper wire		Digital Multimeter	
0.1 Pitch Header pins		Solder Iron	
Raspberry Pi 3B			

1.3 PCB - Logic Shifter Board

1.3.1 Overview/Schematic

This board is made to handle the mismatch between the digital logic voltage level of the Raspberry Pi and the LED matrix. Digital logic is typically at either 3.3V or 5V. The GPIO pins on the Raspberry Pi will output at the 3.3V logic, and the LED matrix is made to read at 5V digital logic levels. In order to have these signals make sense to each other we use a device called a logic shifter, which will take the 3.3V signals and turn them into 5V signals.

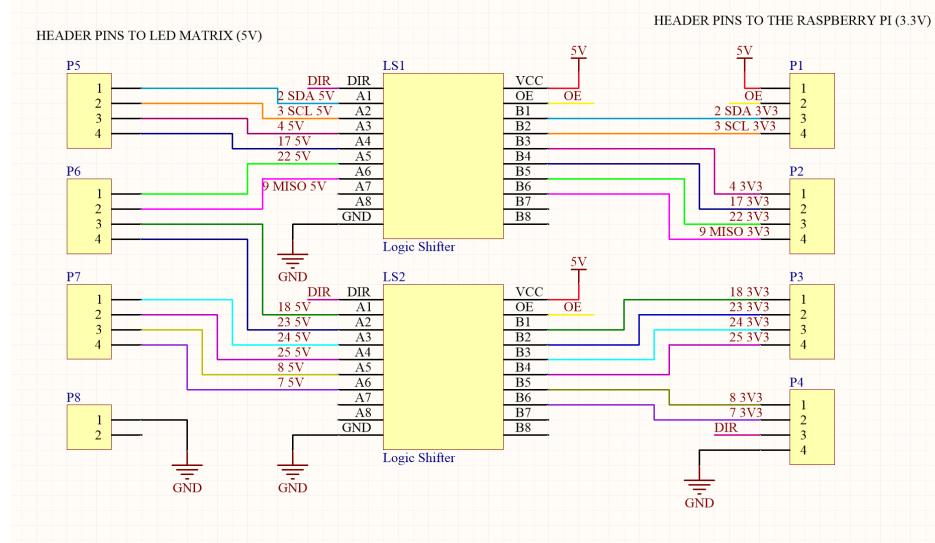


Figure 8: Logic Shifter schematic

In Figure 8 the wire labels correspond to the name of the GPIO pin the signal comes from on the raspberry pi. All **pin numbers that lead to the Raspberry Pi** represent **GPIO pin numbers, NOT physical pin numbers**. For instance, for header P2, pin 1 is labelled with "4 3V3". This should be connected to **GPIO pin 4** on the **Raspberry Pi (physical pin 7)**. The data sheet for the logic shifter IC chip can be found at:

- <https://cdn-shop.adafruit.com/datasheets/sn74lvc245a.pdf>

1.3.2 Testing

The board will come initially unpopulated of components and will look like this:

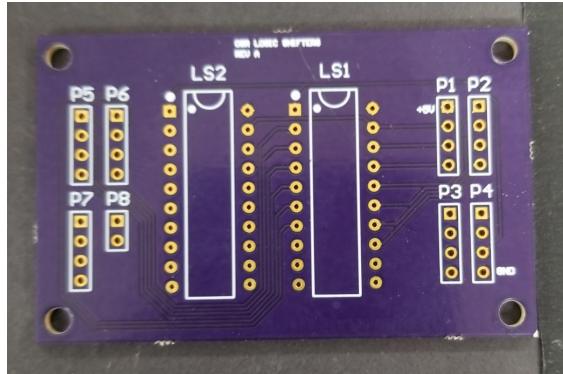


Figure 9: Logic Shifter Board

1. We need to populate the board and test that it is working properly before fully integrating it into the rest of the system. Break off the 0.1 pitch header pins in the correct segments to fit onto the board, and insert the logic shifting chip. The chip might need the pins slightly pushed one direction in order to fit in. Solder all components into the board.

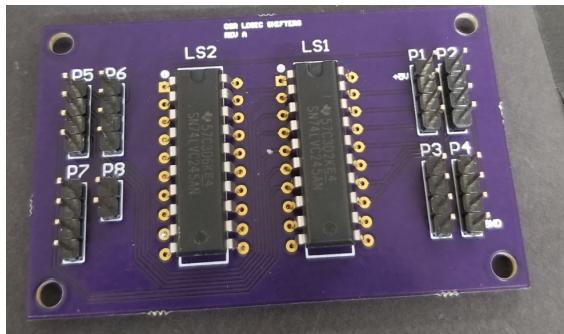


Figure 10: Populating the board

2. Now, you will need to run connections to the board from the Raspberry Pi. Figure 8 shows which pins on the board are connected to which pin on the Raspberry Pi, with the exception of the purple wire, which will be using to test the signals and drive it from the 3.3V DC power pin on the Raspberry Pi. Figure 11 shows which of these we will need to start connecting the board to test.

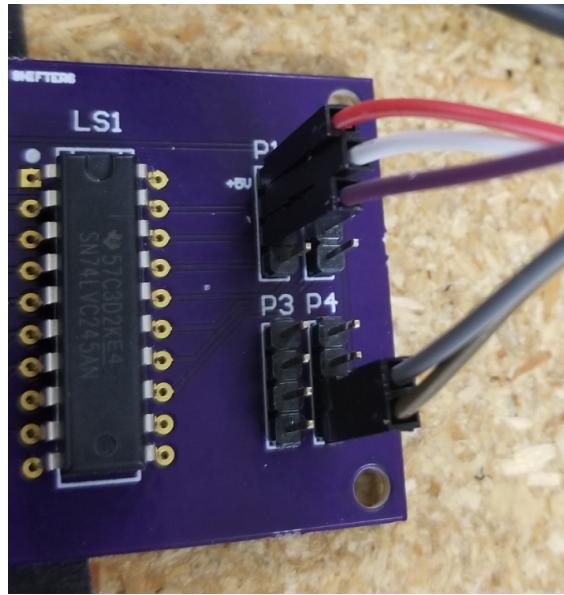


Figure 11: Board Connections

Red	+5V
Brown	GND
White	GPIO12
Grey	GPIO10
Purple	3.3V (pin#01)

3. To begin testing make sure that both chips are being powered correctly with +5V.

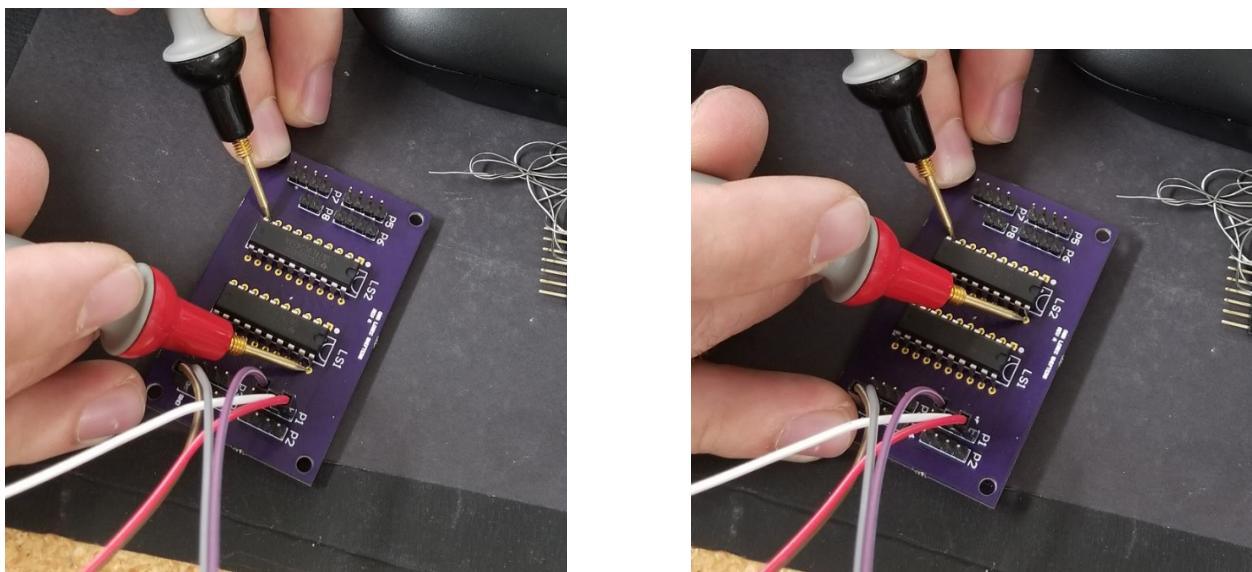


Figure 12: Populating the board

4. We want to make sure we set the direction to flow from bus B (right side of chip) to bus A (left side of chip), which means that both the DIR and the OE pins should be low, 0V

FUNCTION TABLE			
INPUTS		OPERATION	
\overline{OE}	DIR		
L	L	B data to A bus	
L	H	A data to B bus	
H	X	Isolation	

Figure 13: Chip bus direction logic

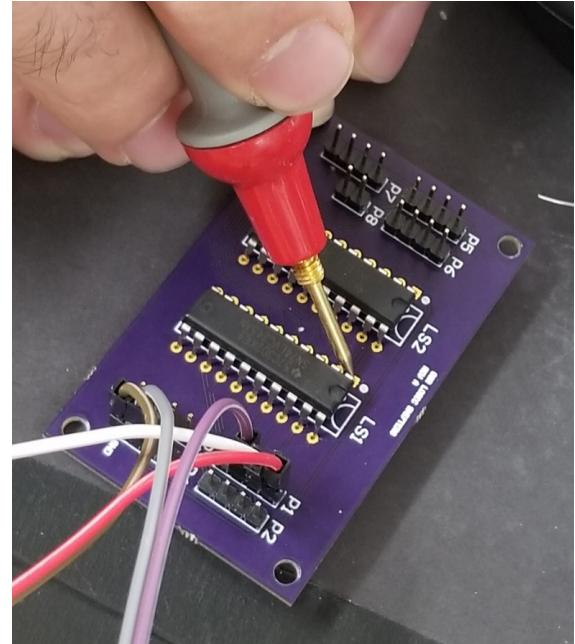
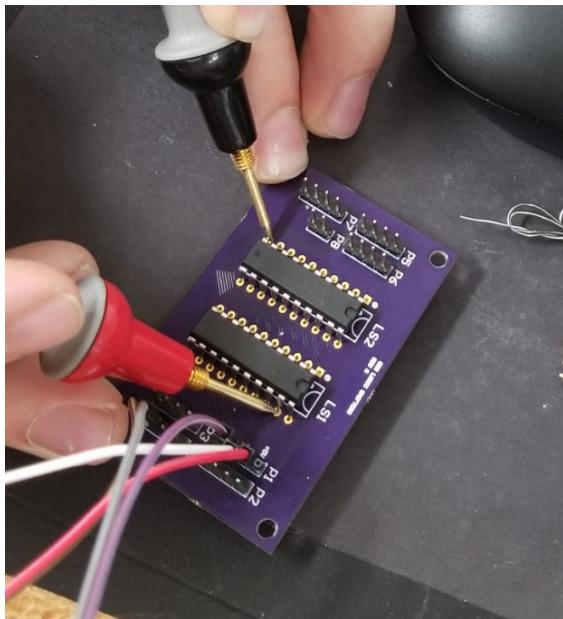


Figure 14: Populating the board

5. Now we can test the signal. Starting with the 3.3V plugged into the GPIO 2 pin (3rd pin down on the P1 set of headers) test the input pin on the chip; it should read 3.3V. Then test the output; it should read 5V.

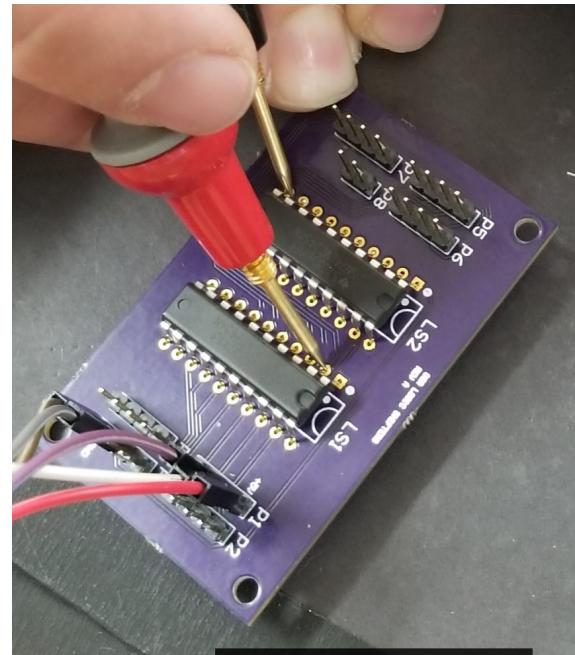
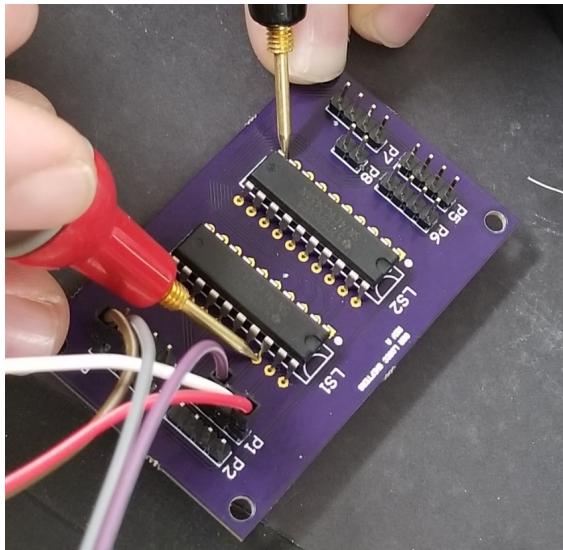


Figure 15: Populating the board

6. Repeat this process for the rest of the pins, following along with the circuit diagram in Figure 8, making sure each channel of the shifter is working correctly.

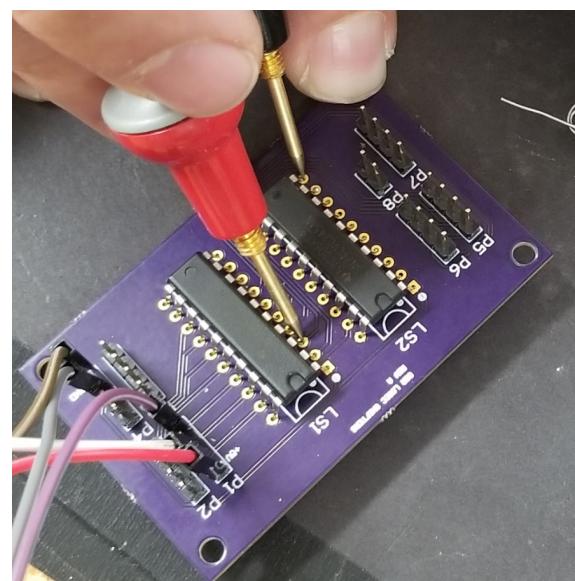
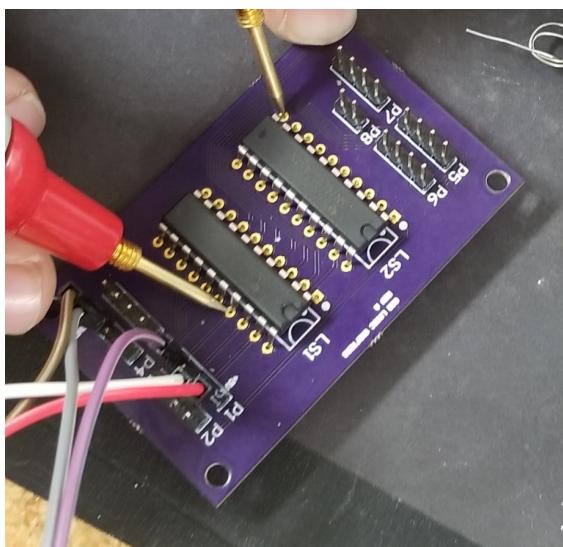


Figure 16: Populating the board