Materials

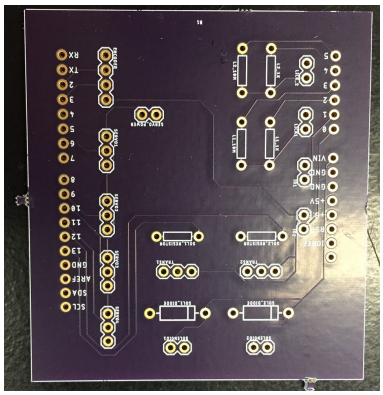
This guide details the process of assembling and using a custom Arduino Uno shield to control hardware in behavioral experiments. The following components are required to complete the assembly of 1 shield:

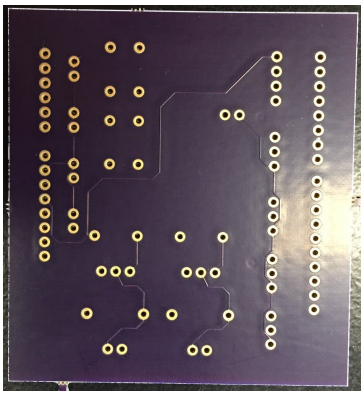
Component	Quantity	Part Number	Vendor
PCB	1	N/A*	Oshpark
4-pin Screw Terminals	1	2137	Adafruit
3-pin Screw Terminals	4	2136	Adafruit
2-pin Screw Terminals	7	2138	Adafruit
Diode	2	1N4004RLGOSCT-ND	Digi-Key
Bipolar Transistor	2	TIP122TU-ND	Digi-Key
1 KΩ Resistor	4	CF14JT1K00CT-ND	Digi-Key
10 MΩ Resistor	2	CF14JT10M0CT-ND	Digi-Key
Straight Header Pins	1	392	Adafruit
Arduino Uno Rev 3	1	A000066	Arduino

*To order a PCB, download the PCB design files here and upload them to www.oshpark.com.

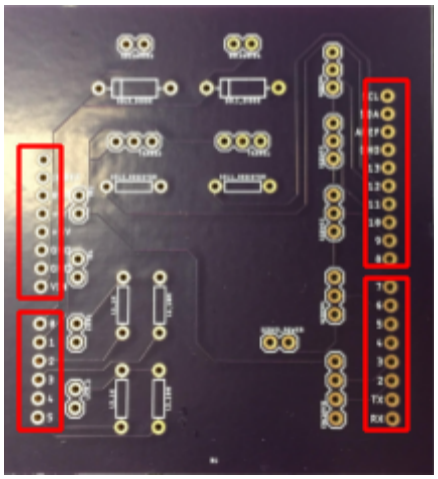
Assembly

1. The Printed Circuit Board (PCB) has 2 sides:

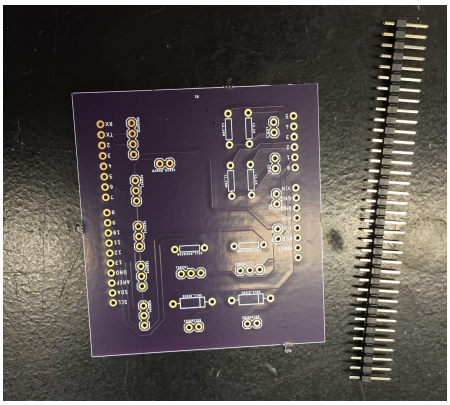




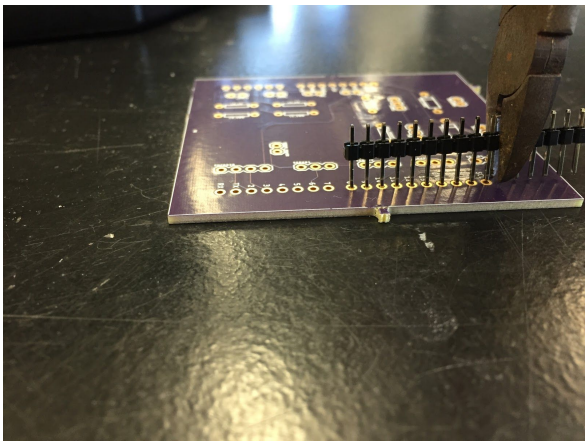
   a. The TOP side of the PCB has a labelled silk screen, and is pictured to the left.
   b. The BOTTOM side of the PCB lacks labels, and is pictured to the right.
2. Soldering the straight header pins



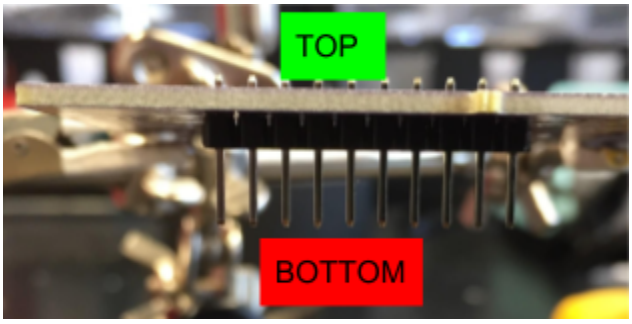
   a. The holes which should be populated by straight header pins are boxed above.



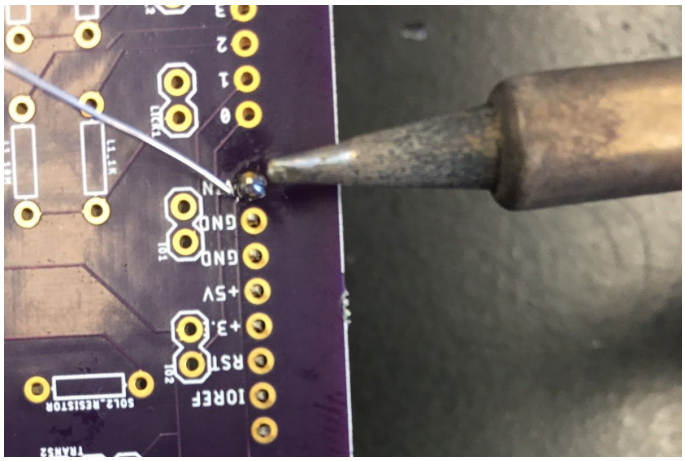
   b. The Straight header pins allow the PCB to interface with the Arduino.



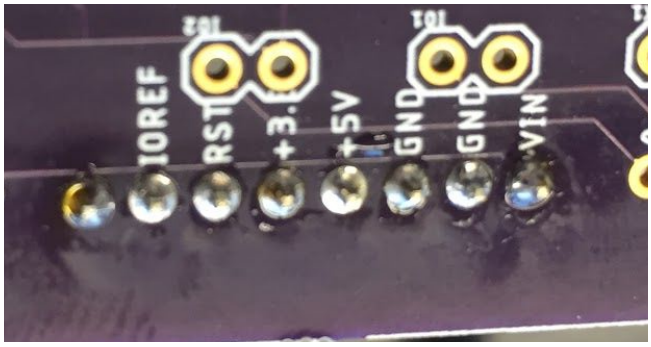
c. Begin by removing a row of pins and lining them up with a corresponding row of holes on the PCB.
d. Clip of the appropriate number of pins using wire cutters.



e. Insert the segment of pins into the PCB's holes such that the plastic midsection touches the BOTTOM of the PCB, and the short sides of the pins protrude out of the TOP of the PCB.
f. It may be helpful to use an adhesive to temporarily mount the pins before soldering. Make sure to use an adhesive that does not produce dangerous fumes when heated.

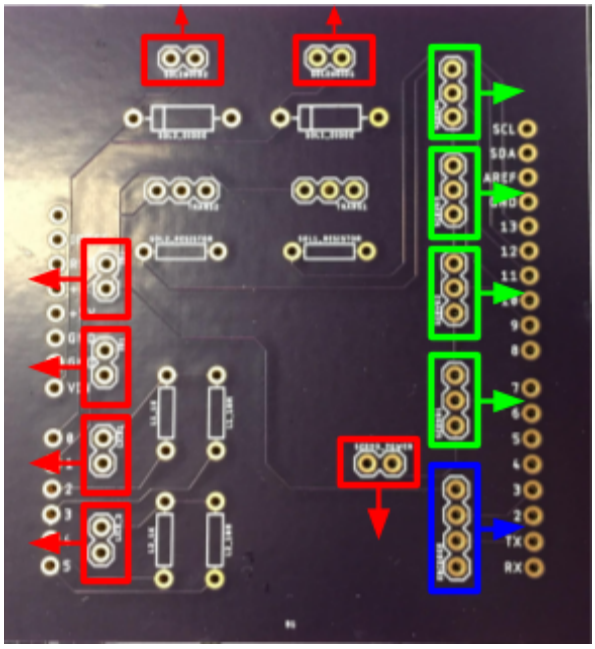


g. Before starting to solder, ensure that you are soldering on the TOP of the PCB, which should have silk screen labels.
h. Solder the header pins by heating up the solder pad for 5-10 seconds, then gently touching a piece of solder to the pin unil solder has flowed across the entire solder pad.



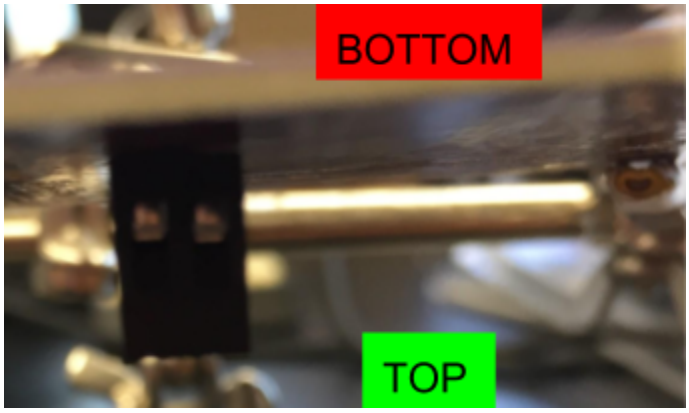
i. A completed row of pins should look like the image above.
j. Repeat steps a-i until all header pins have been soldered onto the PCB.

3. Soldering the screw terminals

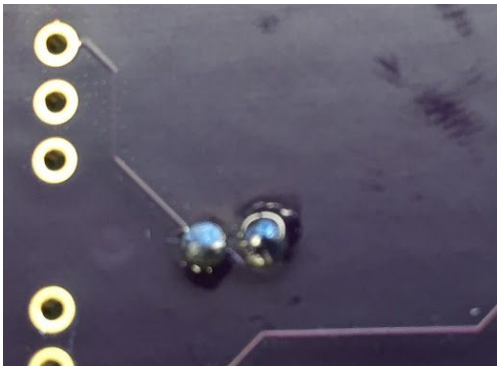


a. The holes which should be populated by 2-pin, 3-pin, and 4-pin screw terminals are boxed above. Arrows indicate the FRONT side of each terminal.

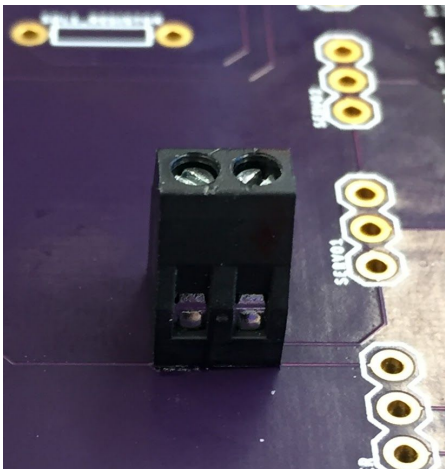
b. Screw terminals are a secure and convenient way to hold wires in place.
c. Each terminal has a FRONT (pictured above on the left) and a BACK (pictured above on the right). The FRONT can be identified by its recessed openings for housing wires. On the board, each screw terminal should be oriented so that the FRONT faces the nearest edge of the board.



d. Place the screw terminal in the corresponding PCB holes such that the pins protrude out of the BOTTOM of the PCB. An adhesive may be helpful.

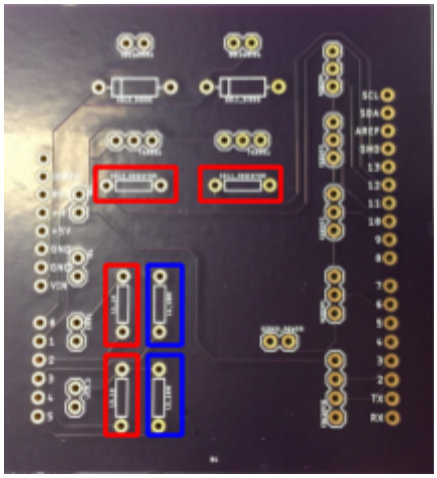


e. Solder the screw terminal pins. Make sure you are soldering on the BOTTOM of the PCB (the side which lacks labels).

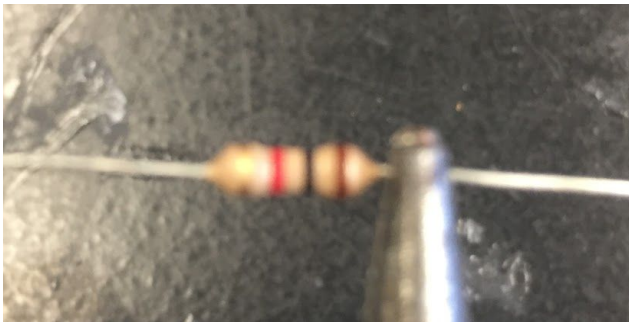


f. When finished, the screw terminal should rest on the TOP of the PCB.
g. Repeat steps a-e for every 2-pin, 3-pin, and 4-pin screw terminal.

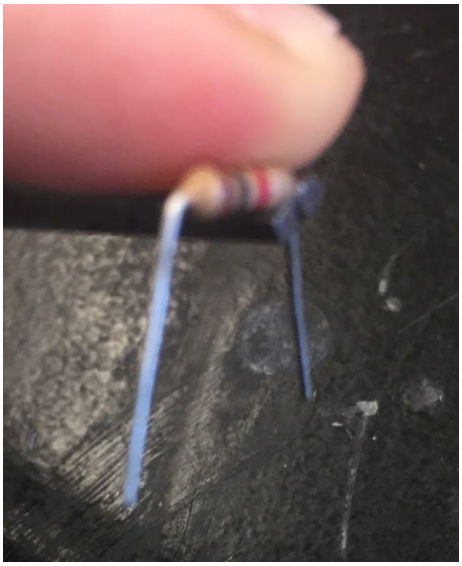
4. Soldering the resistors



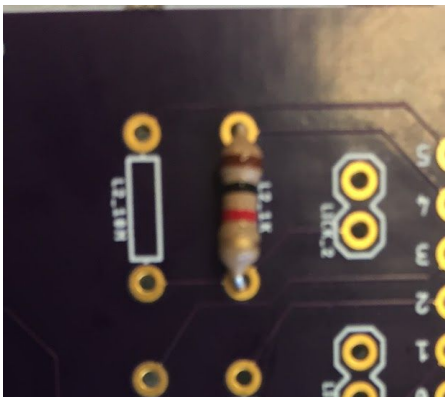
a. The holes which should be populated by 1 KΩ and 10 MΩ resistors are boxed above.



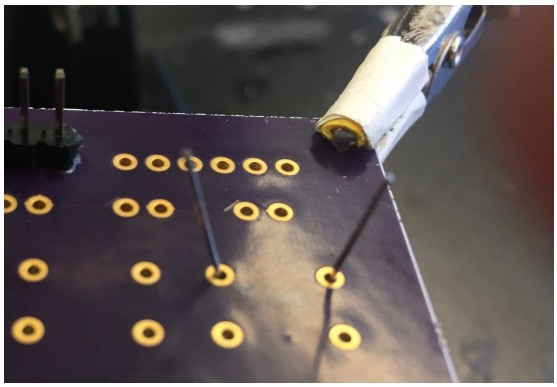
b. In several parts of the PCB, resistors are used to modulate input currents.



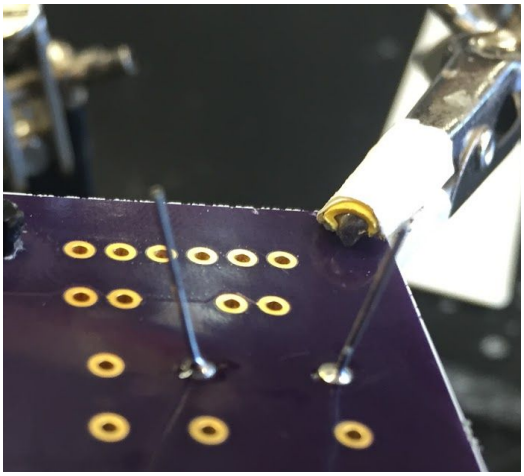
c. Use a pair of pliers to bend the resistor leads at 90 degree angles. Make the bends as close to the resistor body as possible.

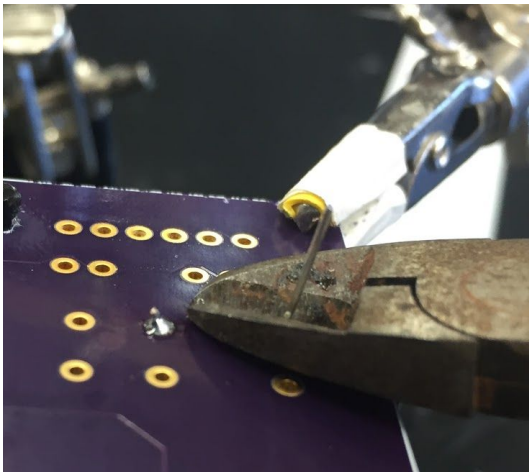


d. Place the resistor inside the corresponding holes in the PCB. The resistor body should rest on the TOP of the PCB.



e. Flip the board over and bend out the resistor leads to keep it in place.

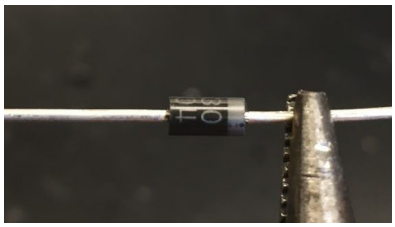




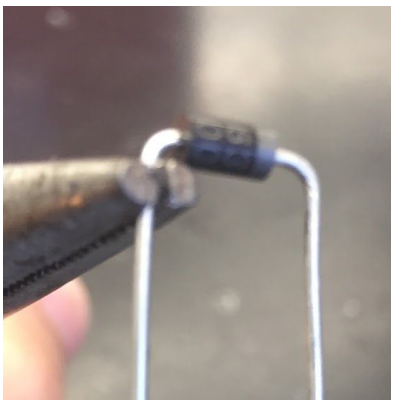
f. Solder the resistor terminals in place and clip away excess length using wire cutters.
g. Repeat steps a-h for all 1 KΩ and 10 MΩ resistors.

5. Soldering the diodes

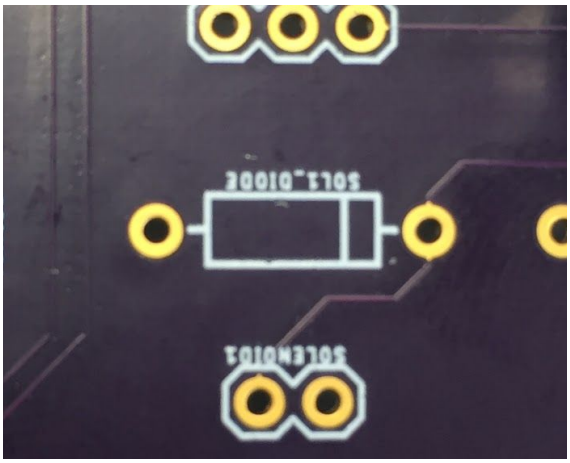
a. The holes which should be populated by diodes are boxed above.

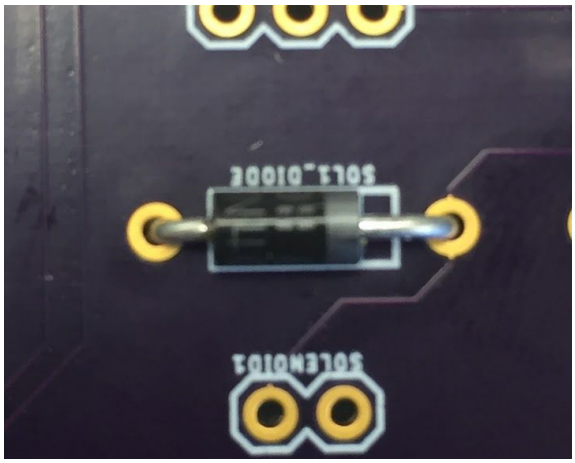


b. The PCB uses a diode to safely dissipate currents induced in the solenoid valve when it is turned off abruptly.

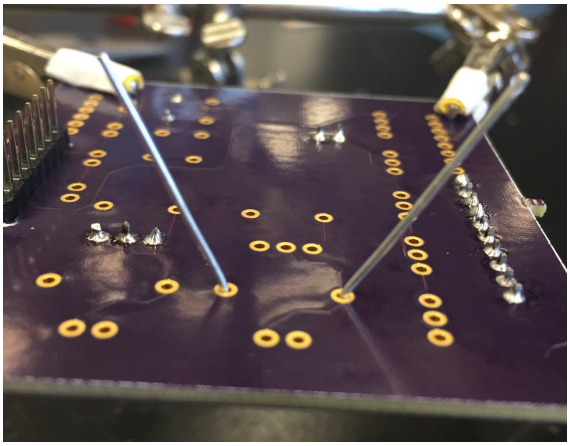


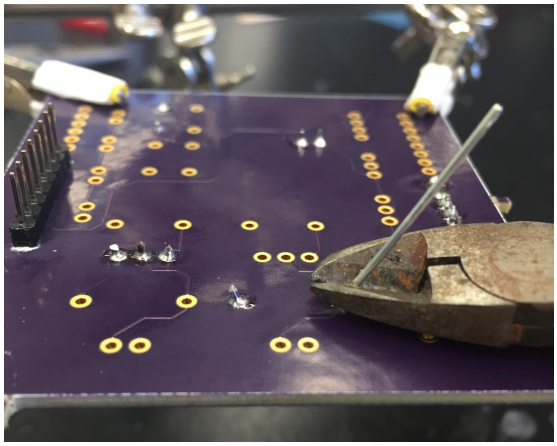
c. Like with the resistor, bend the leads of the diode at 90 degree angles using pliers. Since the diode's body is smaller, leave around 3 mm of horizontal wire on either side of the body.





d. Insert the diode into its designated holes in the PCB. Since the diode is a polarity sensitive component, it will not work if installed backward. Make sure the gray rectangle on the diode aligns with the corresponding diagram on the silk screen, as pictured above.

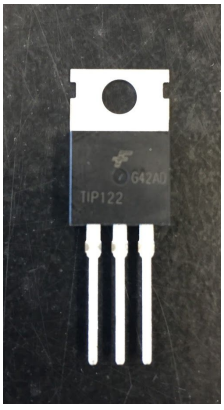




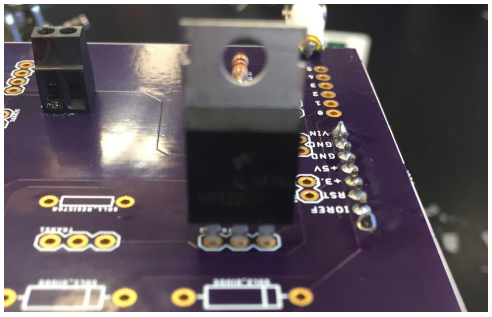
e. Bend the leads of the diode outward to hold the component in place.
f. Solder the two leads.
g. Trim off excess length using wire cutters.
h. Repeat steps a-g for both diode slots

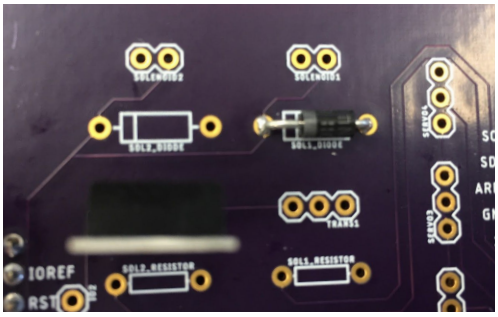
6. Soldering the transistors

a. The holes which should be populated by transistors are boxed above.

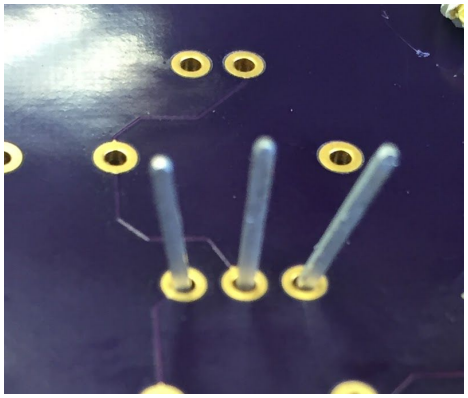


b. The PCB uses a transistor to amplify voltage applied to the solenoid valve.

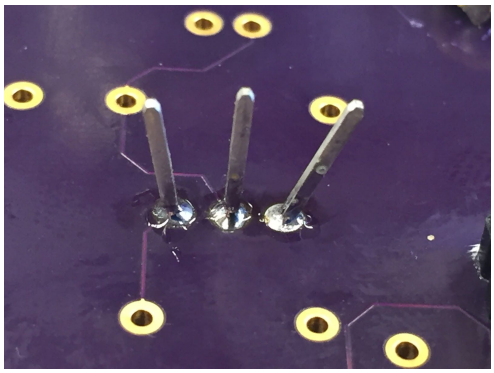


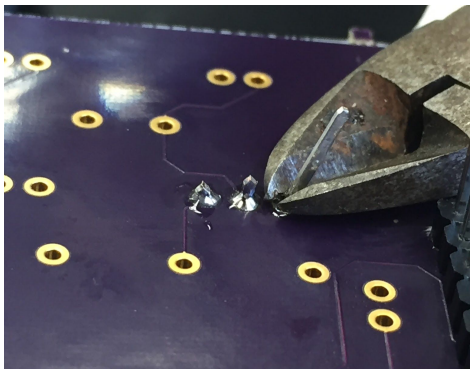


c. Gently place the Transistor into its corresponding holes on the PCB. Note that the transistor will not work if oriented improperly. As pictured above, the gray plate on the back of the transistor should face **away** from the diodes.



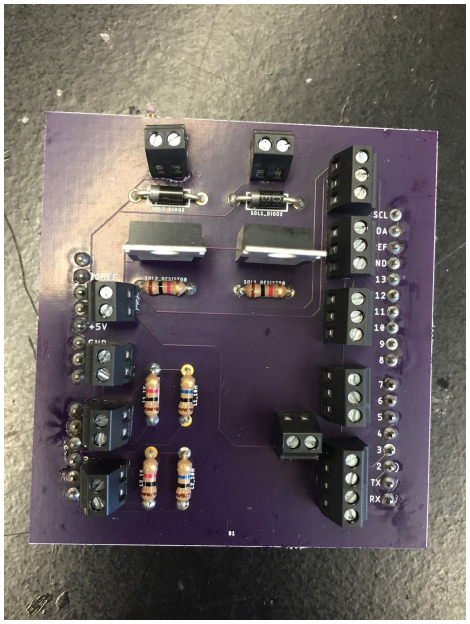
d. Flip the board over, and bend the outside leads outward to keep the transistor in place.

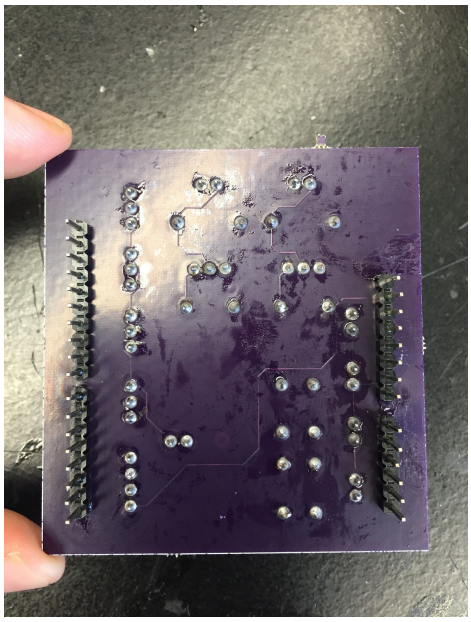




e. Carefully solder the leads in place.
f. Trim excess length with wire cutters.
g. Repeat steps a-e for both transistors.

7. Finished Board

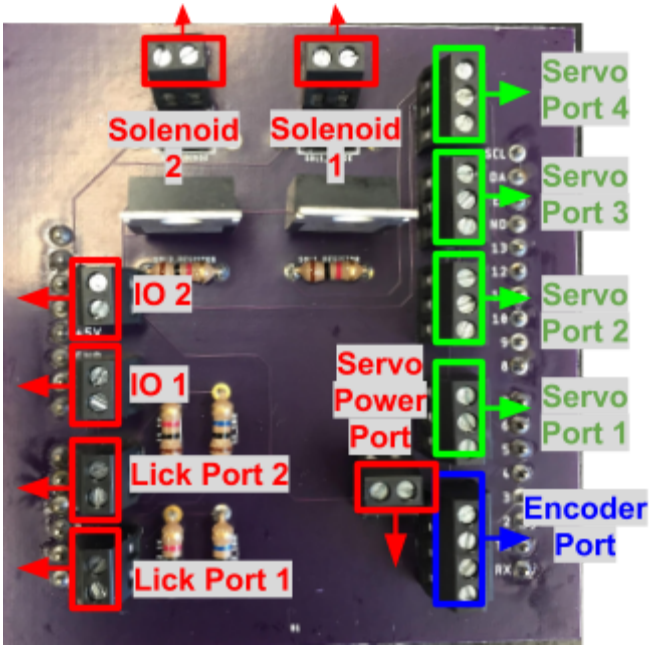




a. Completed PCB viewed from the top and bottom.

Use

1. Board Functions and Functional Diagram



   a. As in the previous diagram, arrows indicate the FRONT sides of screw terminal.
   b. Board Conventions
      i. When a screw terminal is viewed from the FRONT, the leftmost opening is a power supply, the rightmost opening is ground, and all openings in between are for digital or analog signals (Power -> Signal -> Ground).
   c. Servo Power
      i. The servo power port provides the 5V power and ground lines to each of the 4 servo motor ports on the board. An external power supply is needed for the servos, since it is possible to overdraw the Arduino's power supply if the servos are in frequent use.
      ii. To use, strip the insulation off of a 5V power supply, coat the ends in solder, and secure them in the servo power port according to the previously mentioned convention.
   d. Servo Ports
      i. Servo ports follow the global board convention, so the servo input wires should be plugged in in the order (Power -> Signal -> Ground) when viewed from left to right.
      ii. When plugged in and powered, each servo can be commanded to a particular position by sending a PWM signal from a particular arduino pin. The Port-Pin correspondences are as follows:
         1. Servo Port 1: D6
         2. Servo Port 2: D9
         3. Servo Port 3: D10
         4. Servo Port 4: D11

```
>>
>> servo1 = Servo(arduinoBoard,'D6')
```

iii. example Initialization code

e. Solenoid Ports
   i. When activated, the solenoid ports produce a potential difference equal to the V_{in} voltage supplied through Arduino's barrel connector. To operate a 12V solenoid, a 12V barrel connector must be plugged into the Arduino.
   ii. When plugged in and powered, each solenoid can be opened or closed by applying a digital output to a particular pin. Port-Pin correspondence is as follows:
      1. Solenoid Port 1: D13
      2. Solenoid Port 2: D12
   iii. For both pins, a "high" digital output will open the solenoid, while a "low" digital output will close the solenoid.
f. Encoder Port
   i. A four-pin screw terminal is provided to support reading from quadrature encoders. In accordance with the global board convention, the leftmost opening is a 5V power supply, the rightmost opening is a ground line, and the center two openings collect signals from the A and B channels of a quadrature encoder.
   ii. The input channels relay encoder signals to pins D2 and D3 on the arduino.

```
>>
>> enc = rotaryEncoder(arduinoBoard,'D2','D3');
```

   iii. Example initialization code
g. Lick Ports
   i. The board contains two "lick ports" which support capacitive sensing of an animal's licking behavior.
   ii. In accordance with global board convention, the left opening is grounded. A wire from this ground channel should be in contact with the animal or its immediate surroundings.
   iii. The right channel is connected to an analog output pin. Port-Pin correspondence is as follows:
      1. Lick Port 1 (output): A5
      2. Lick Port 2 (output): A3
   iv. This analog output channel should be wired to the needle or spout used to dispense the animal's reward. Therefore, when the animal licks, the analog output channel will form a complete circuit with the ground channel.
   v. An additional analog input pin is associated with each port. This pin reads "high" when the circuit is incomplete, and reads "low" when the circuit is

complete. By continuously monitoring this pin, licks can be detected. Port-Pin correspondence is as follows:

1. Lick Port 1 (input): A4
2. Lick Port 2 (input): A2

h. Additional IO Ports
   i. In addition to the previously described ports, the board contains two general purpose Input-Output Ports.
   ii. In accordance with board convention, the left opening is grounded, while the right connects to an analog input/output pin. Port-Pin correspondence is as follows:
      1. IO 1: A1
      2. IO 2: A0
   iii. The additional IO ports can be used for operating small devices like LEDs.