lighthack: Box 1 Revision: 02
Assembly Instructions

Parts List

Qty	Description
1	Arduino Uno R3
1	HD44780 Display Module
3	Cherry MX Red Key Switch
3	Black Eos-Style Key Cap
2	Rotary Encoder with Panel Mounting Hardware
2	Finger Wheel for Rotary Encoder
1	10 kΩ Potentiometer
1	Red Wire Solid Core 22AWG 2-3'
1	Black Wire Solid Core 22AWG 2-3'
1	White Wire Solid Core 22AWG 2-3'
1	Yellow Wire Solid Core 22AWG 2-3'
1	Blue Wire Solid Core 22AWG 2-3'
3	5-position Wago Wire Connector
1	3-position Wago Wire Connector
1	USB Cable, A to B
10	Machine screws #2-56x3/16"
5	Standoffs #2-56x1/4"
5	Standoffs #2-56x1/2"
1	Enclosure with lid and screws

Tools

Necessary:

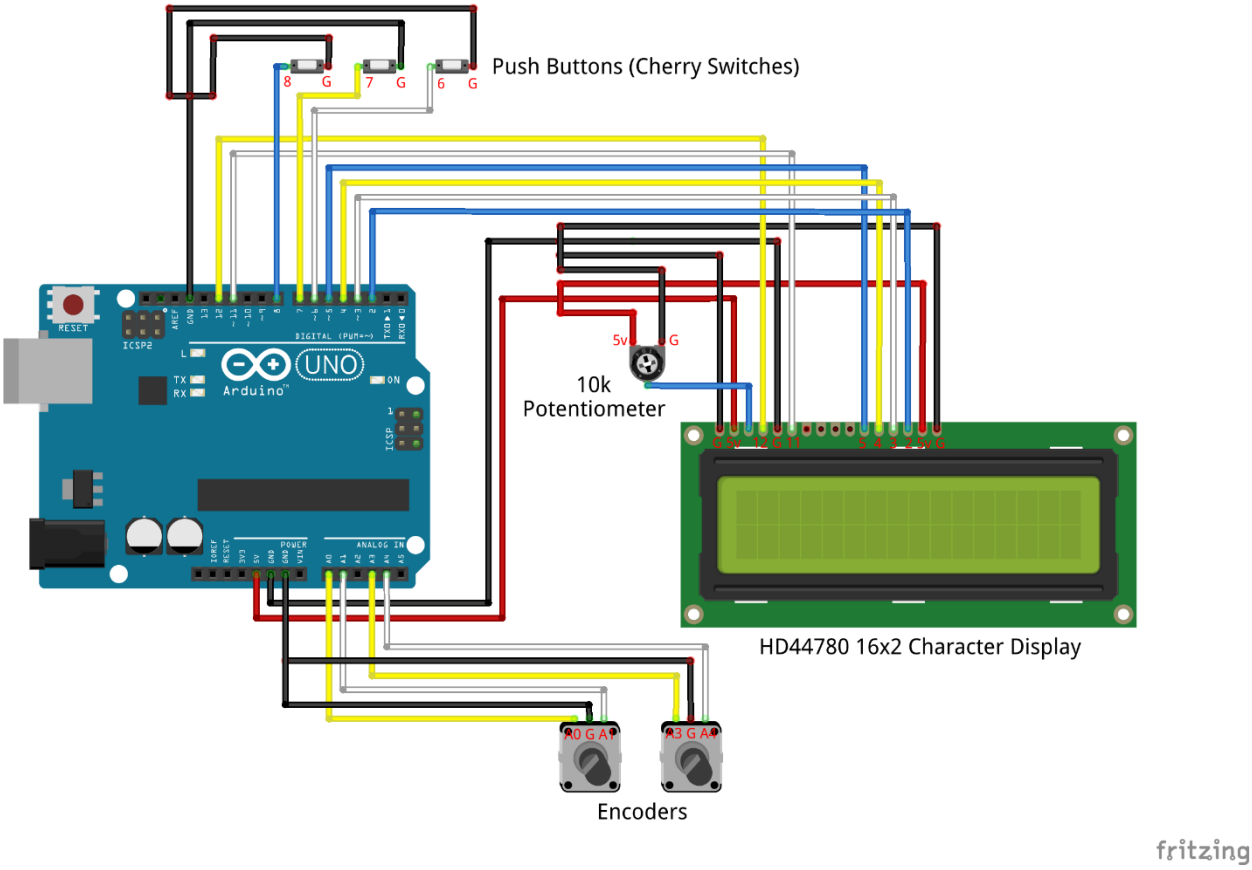
- Wire strippers and cutters
- Soldering iron & solder
- Philips screwdriver
- Drill with 1/8", 1/4", 1/2" bits
- Rasp/File or Chisel

Helpful:

- Needle-nose pliers
- X-Acto knife or other sharp knife
- Electrical tape
- Thick double-sided tape

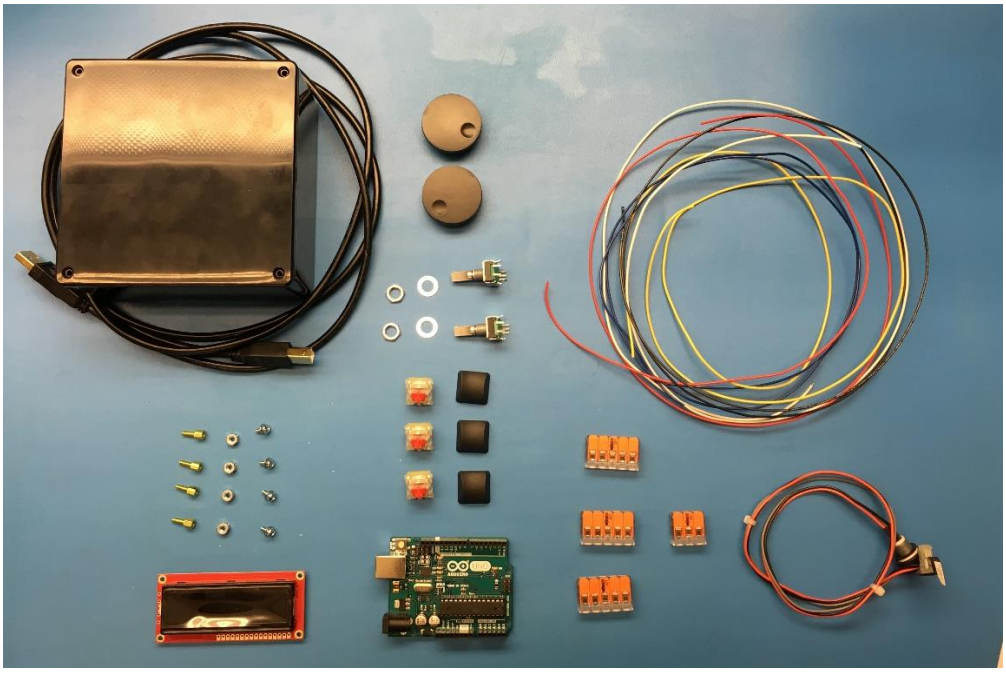
Software

- Arduino Integrated Development Environment (IDE). Download from https://www.arduino.cc/en/Main/Software
- Arduino sketch (code) for Box 1. Download from https://github.com/ETCLabs/lighthack
- Arduino OSC library. Download from https://github.com/CNMAT/OSC



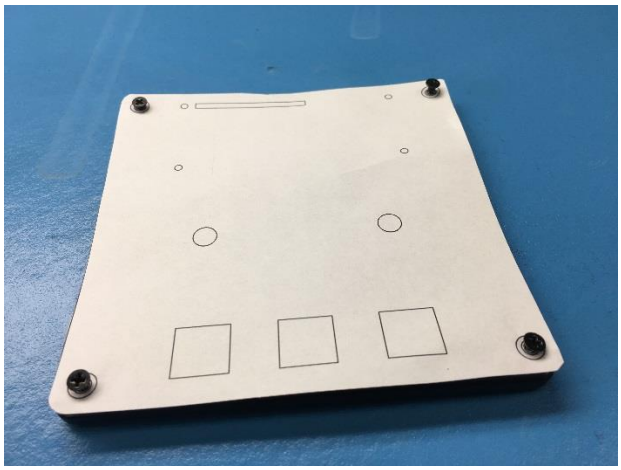
Hook-up diagram

Before you start, verify that all parts are present using the parts list above.



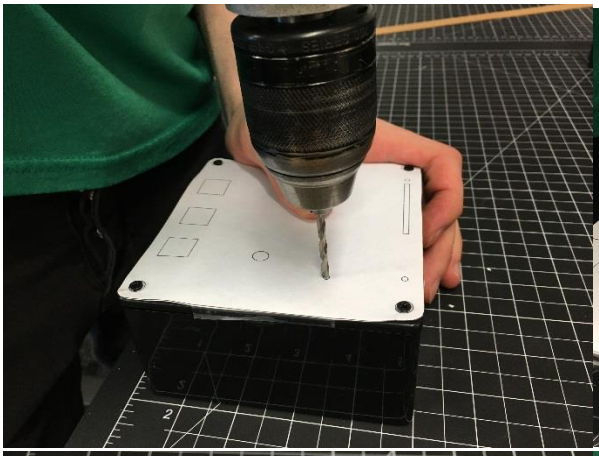
Prepare the enclosure

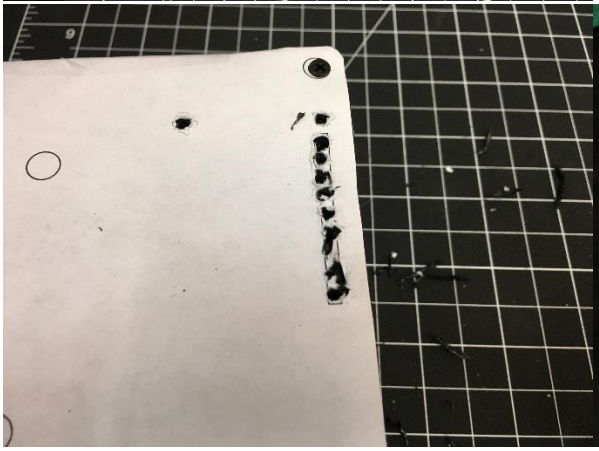
1. Print out the paper template, making sure to disable any "fit to paper" or scaling options so that it prints at the right size. Cut out the template and lay it over the top of the enclosure lid, using the screw holes or tape to make sure it doesn't move.

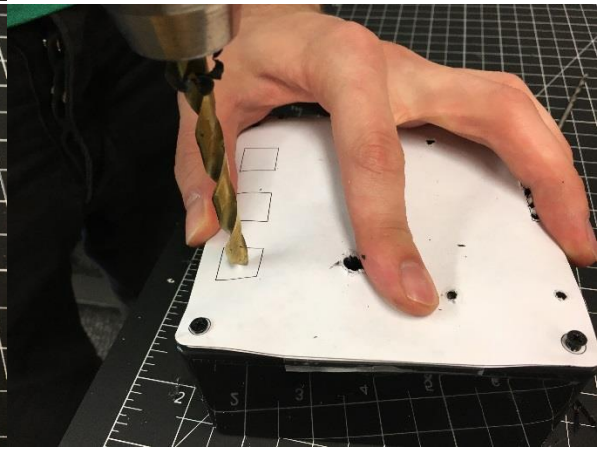


TIP: Don't feel like drilling and filing? Use the provided 3D Printer template to print a "pre-drilled" lid.

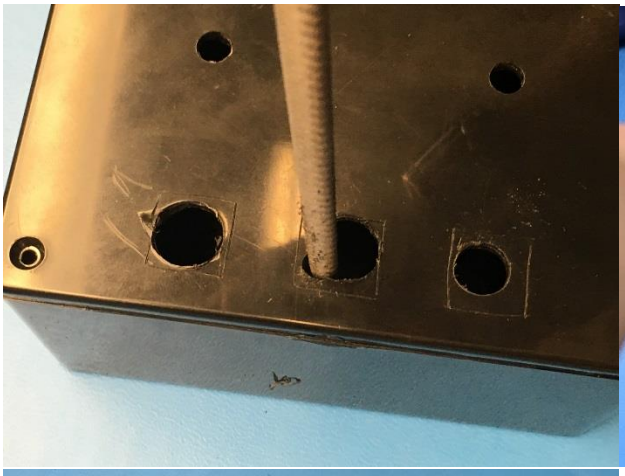
2. Drill out the holes using the proper size drill bits. Drill starter holes in the square cutouts. For the long rectangular slot, you may want to drill a few holes right next to each other as a starting point for a file.

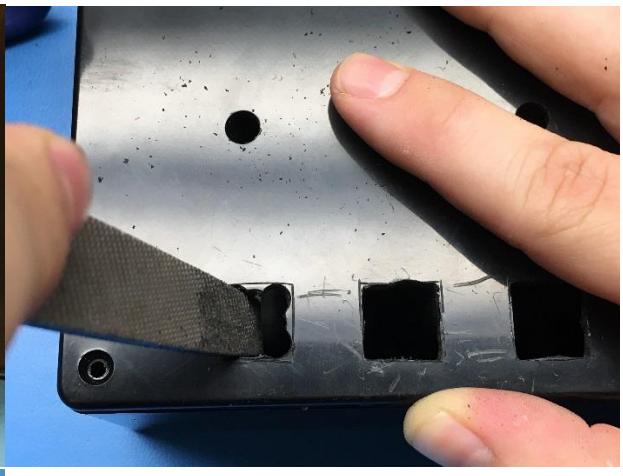






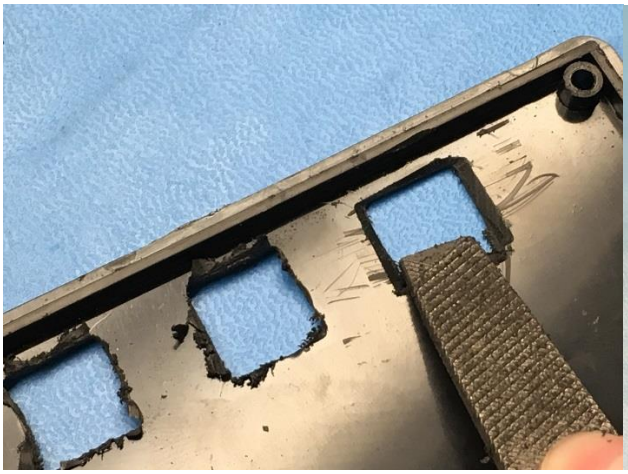
3. Using a square or rectangular file, file out the square holes and slot (or use your preferred method of making square holes). Test fit the Cherry MX Red switches into the square holes as you file to make sure you have the right size.

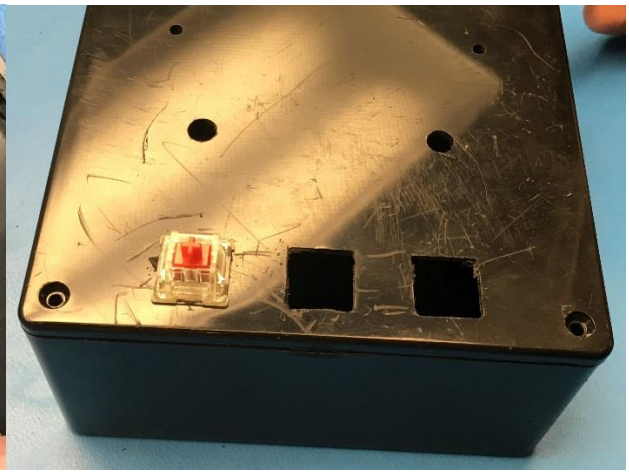


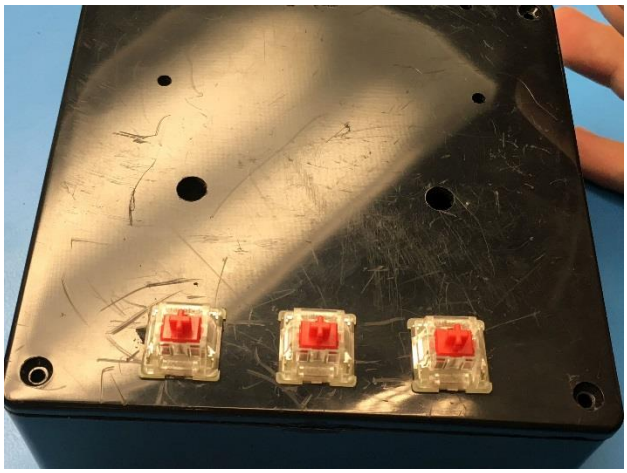


TIP: It may be helpful to trace the edges of the square holes into the plastic with an X-Acto knife, and then remove the paper.

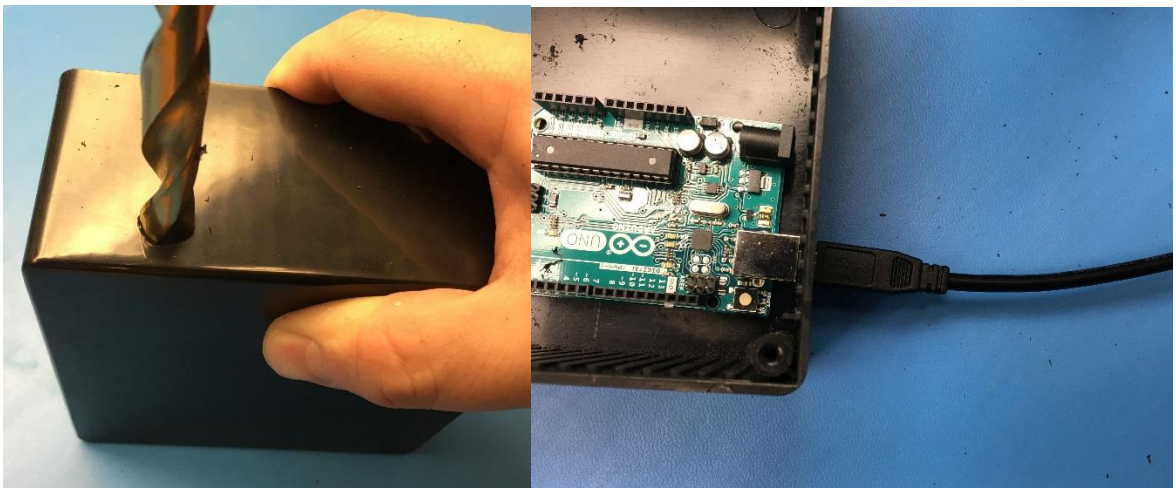
4. From the underside of the lid, use a file to bevel the edges of the square holes. This makes the plastic surrounding the hole thinner so that the Cherry MX Red switches can clip in properly.







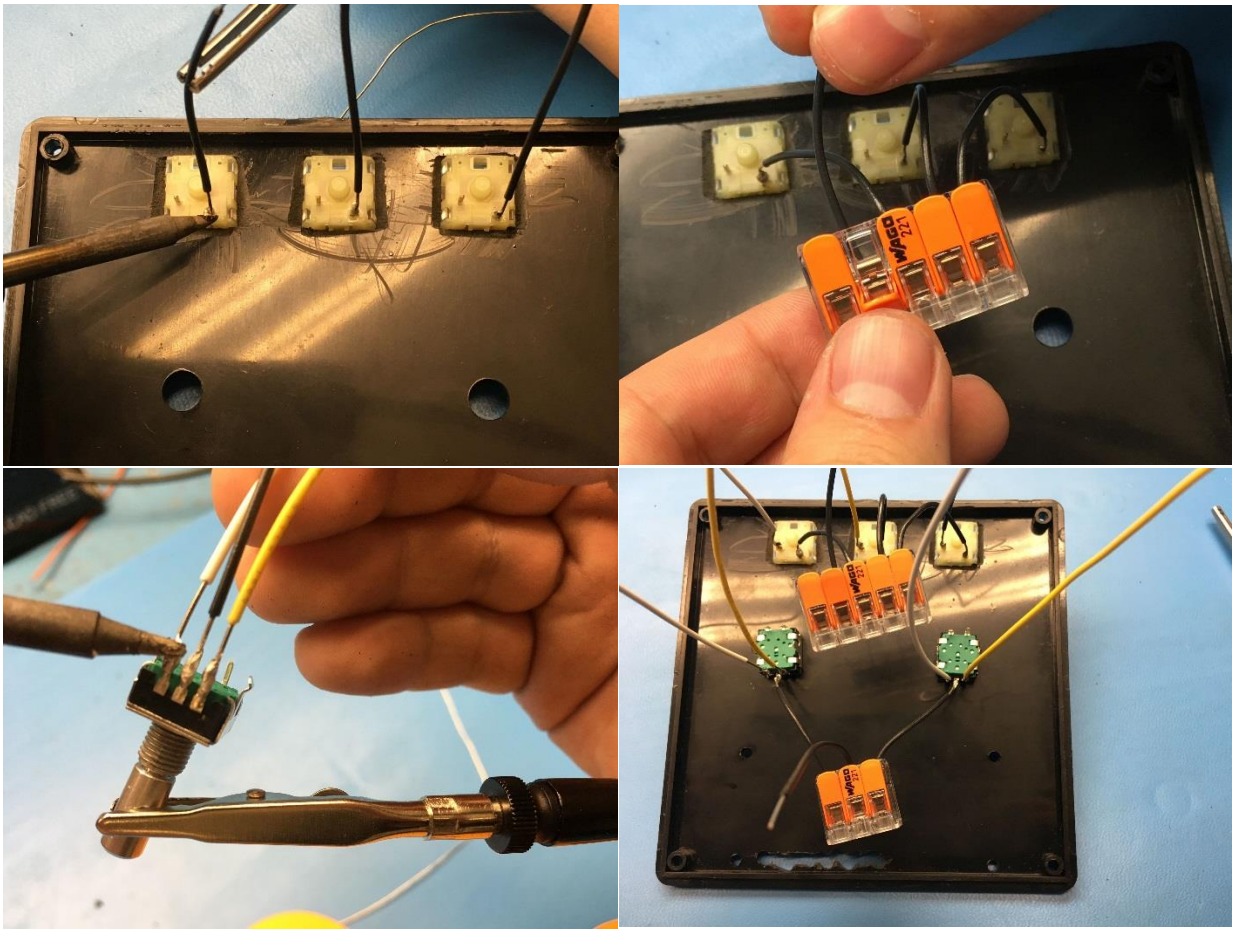
5. Drill a hole for the USB connector. Line up the Arduino with a side of the box in order to estimate the location of the hole. We've found a good location to be 1" in from a corner and ½" up from the bottom. Use a ½" drill bit to drill a hole for the USB connector. Make sure you can connect a USB cable to your Arduino through the hole.



Connect the electronics

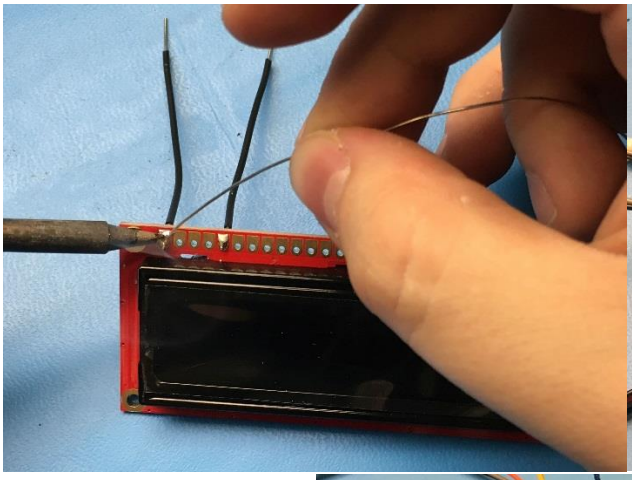
6. Cut appropriate lengths of wire to connect the three buttons and two rotary encoders to the Arduino as shown in the wiring diagram. The wire colors in the wiring diagram are a suggestion. Strip approximately ½" of insulation from either side of each length of wire.

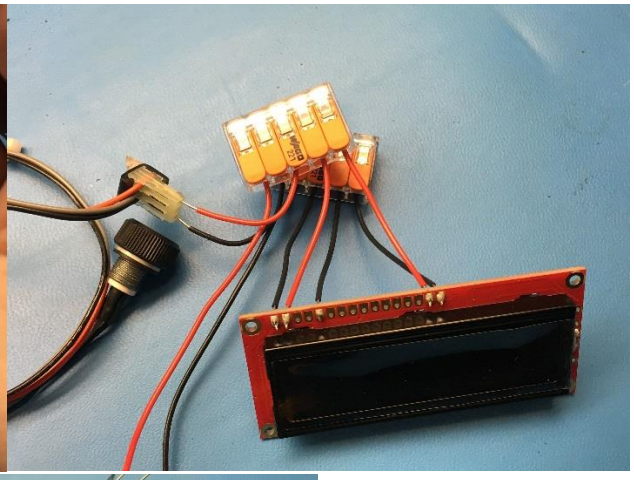
7. Solder one end of each wire length to pins on the rotary encoders and key switches. Use the Wago connectors where multiple black (ground) wires need to be connected together. Only three of the five leads on the encoders need to be connected, as shown. You will want to have the key switches inserted into the enclosure lid already when you do this.

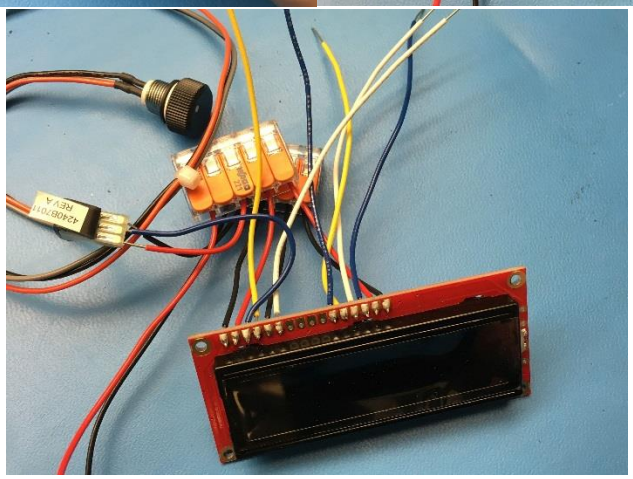


8. Cut appropriate lengths of wire to connect the display and contrast adjustment potentiometer to the Arduino as shown in the wiring diagram. Strip approximately ½" of insulation from either side of each length of wire.

9. Solder one end of each wire length to the display as shown. Connect the middle pin of the potentiometer to the display as shown.



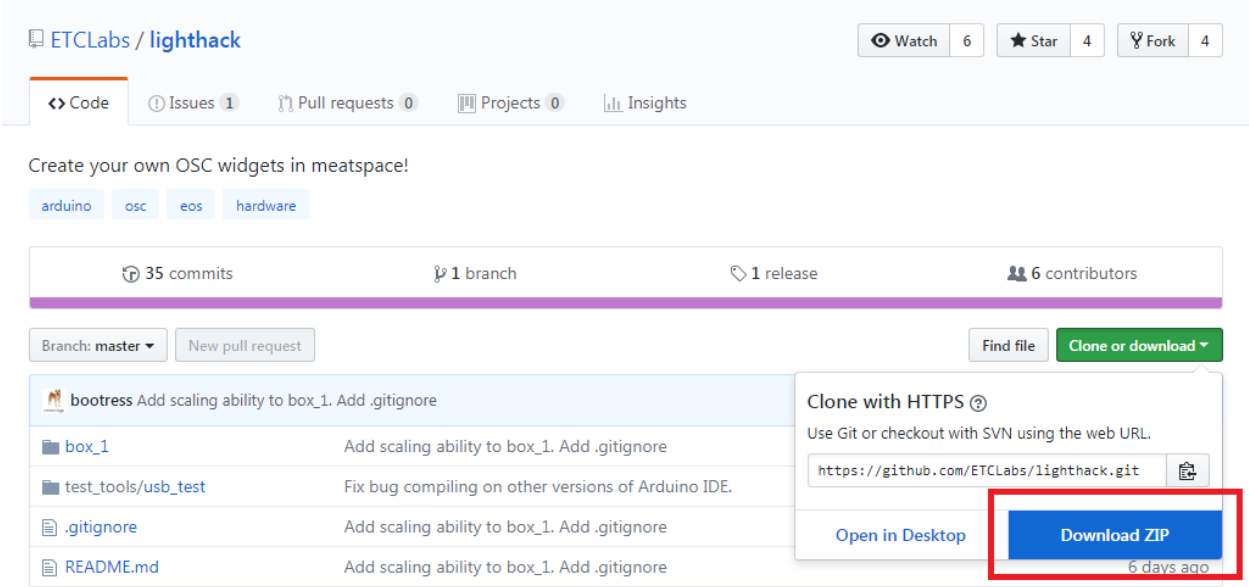


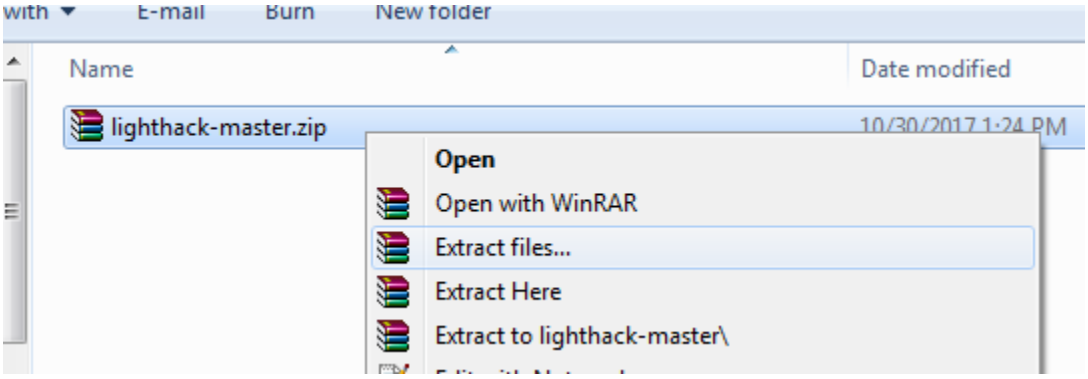


10. Connect the free end of each wire to the appropriate pin on the Arduino as shown in the wiring diagram.

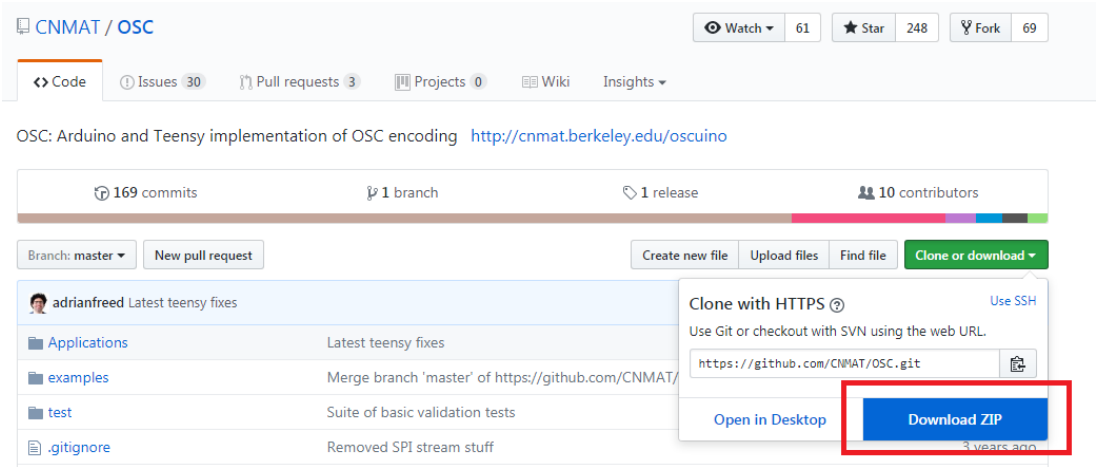
Test the software

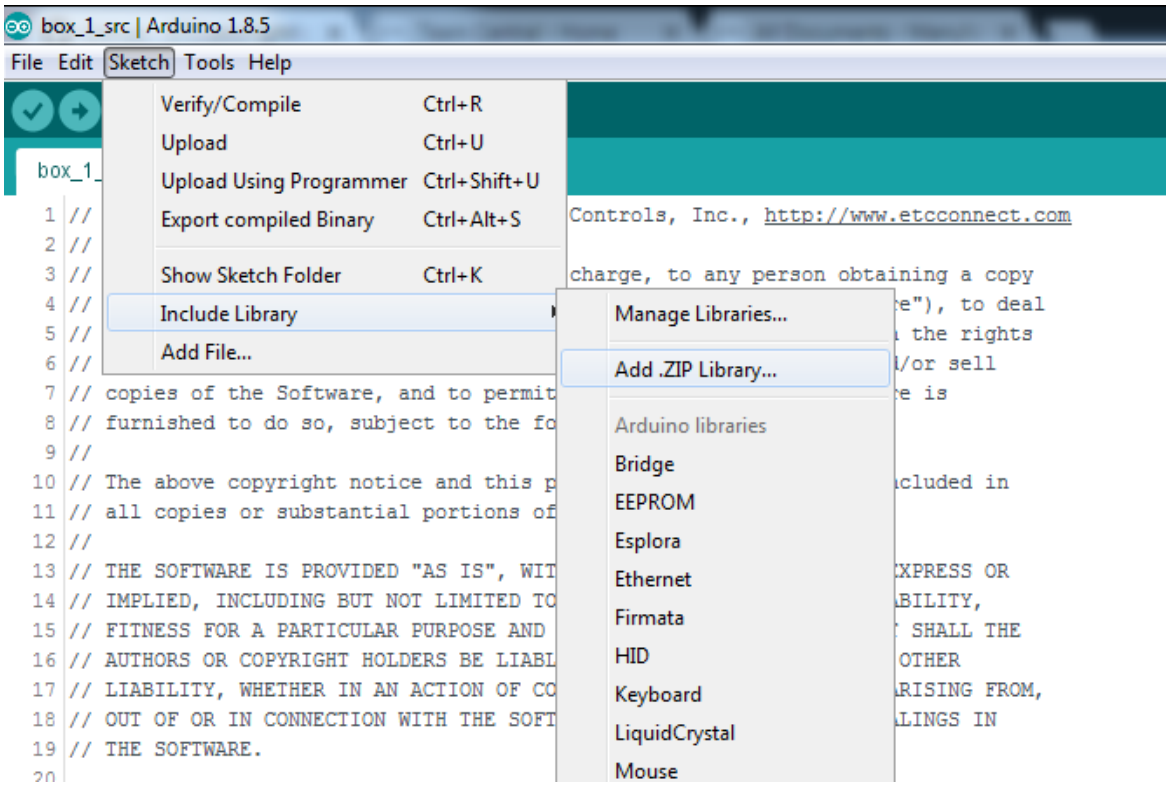
11. Now that your electronics are assembled, you'll want to test the device before finishing with the enclosure. First, let's make sure we have all the software we'll need. Download the source code from https://github.com/ETCLabs/lighthack, and extract the .zip file (or clone the repository if you're git-savvy!).





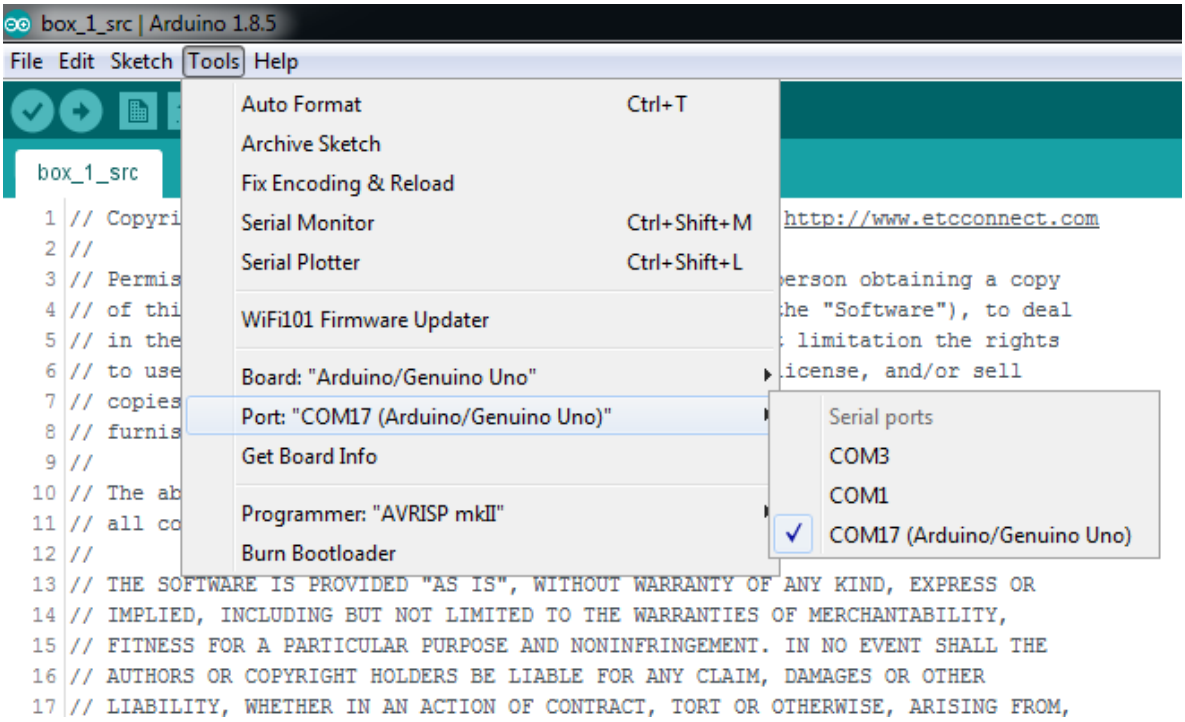
12. Open the Arduino sketch (box_1/box_1_src/box_1_src.ino) in the Arduino IDE. Before the sketch will compile, we need to add additional code (called a "library") so that the Arduino knows how to speak OSC. Download the library as a .zip from https://github.com/CNMAT/OSC. Then, in the Arduino IDE, select Sketch->Include Library->Add .ZIP Library... and select the zip file you downloaded.



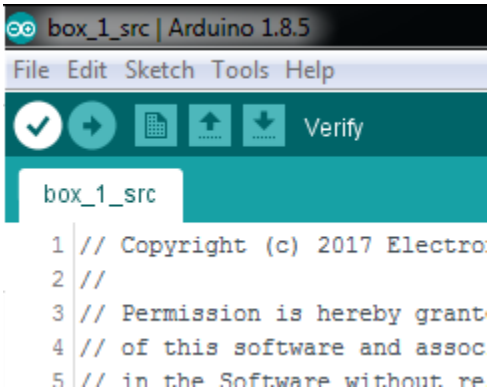


13. Connect the Arduino to your computer using the USB cable.

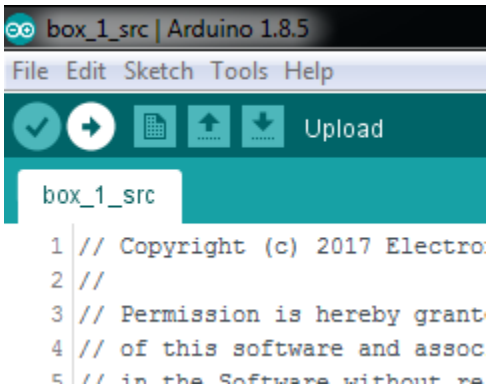
14. The Arduino IDE should automatically detect your Arduino Uno and select it. You can verify this in the Tools menu:



15. Press the check mark to verify your sketch. If you've added the OSC library properly, the verification should be successful.



16. Press the arrow to upload your sketch to the Arduino.

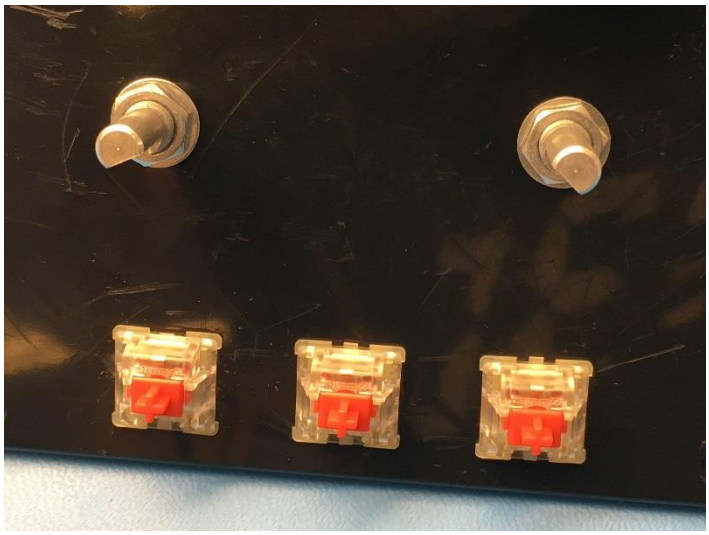


The sketch should start running and you should see text appear on the display. Turn the contrast potentiometer until the display text is readable. If text does not appear, follow the troubleshooting steps at the end of this document.

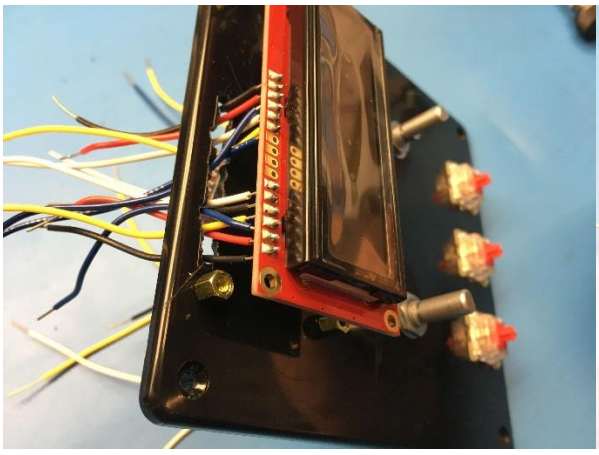
17. Open ETCnomad, or connect the Arduino to a Win7e Eos console. Test the device by patching a moving light, selecting it and using the encoders to modify the pan and tilt values. Check to make sure that the "next" and "last" buttons cycle through the channels and the "fine" button switches between coarse and fine adjustment.

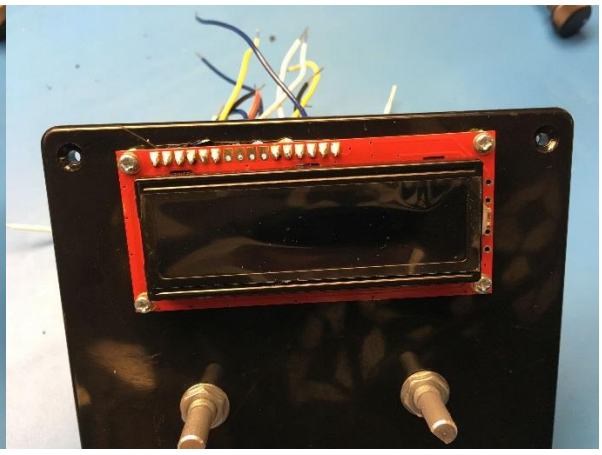
Finish the assembly

18. Now that your module has been tested, it's time to put it in the enclosure. Insert the two encoders through the holes from the inside of the lid. Use the included panel mounting nuts and washers to secure the encoders.

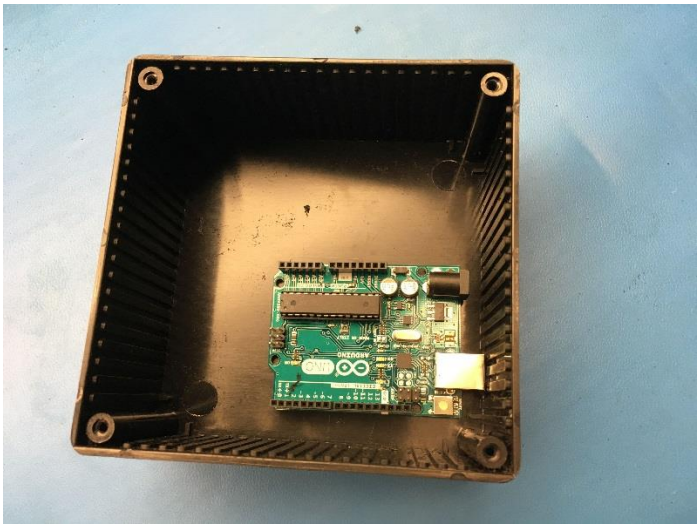


19. Disconnect the screen from the Arduino and feed its wires through the rectangular slot. Secure the screen in place using eight #2-56 machine screws (four from the top, and four from the bottom) and either the ¼" or ½" standoffs. Extra screws and standoffs are provided.

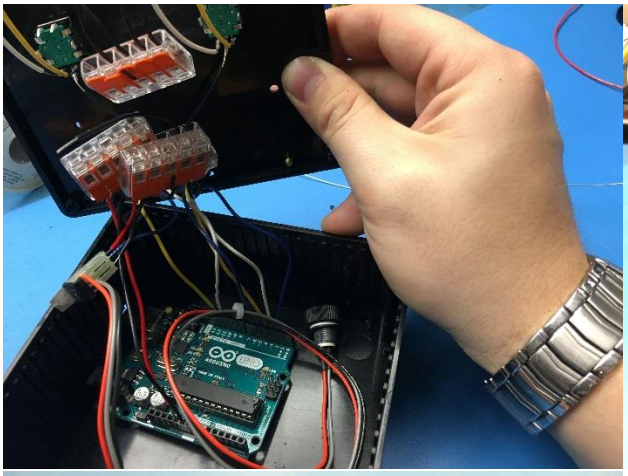


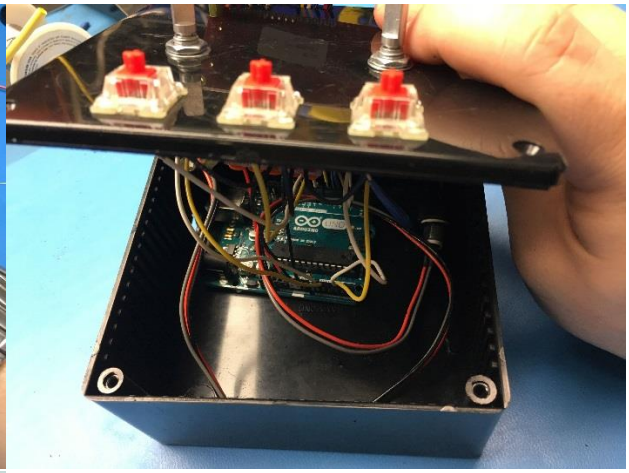


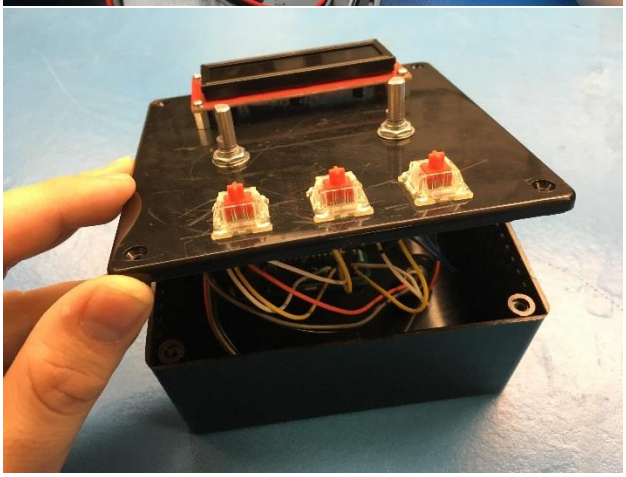
20. Place the Arduino into the enclosure such that the USB connector is aligned with the hole you drilled earlier. Secure the Arduino using double-sided tape or your preferred method.



21. Reconnect all of the wires to the Arduino, referring to the wiring diagram.







TIP: You may want to use some electrical tape for cable management.

22. Secure the face plate to the enclosure using the four black screws.

23. Press-fit the keycaps and encoder finger wheels onto the key switches and encoders.

24. Re-test the module to make sure everything still works. Congratulations! You're done.

Troubleshooting

Problem	Possible solution
Arduino software displays an orange error message when I try to verify using the check mark icon	Verify that the OSC library is installed using the procedure above.
Device mostly works but a button or an encoder is not working	Check the solder connections for cold or broken joints. Check the connections to the Arduino. Make sure the button or encoder is connected to the proper Arduino pins as shown in the wiring diagram.
One or both encoders are reversed	Look near the top of the source code (box_1_src.ino) for lines that say "#define PAN_DIR FORWARD" and "#define TILT_DIR FORWARD". To reverse the direction of an encoder, set the corresponding line to REVERSE, i.e. "#define PAN_DIR REVERSE".
Screen "glitches" or displays strange characters	Check the screen's solder connections for cold joints. Check the screen's connections to the Arduino. Make sure no connections are loose or disconnected and that the connections are as shown in the wiring diagram.
Screen is all black or all white	Make sure the contrast potentiometer is connected to the screen as shown in the wiring diagram. Make sure the Arduino is powered on and the code is loaded. Arduino power can be verified by a green "ON" LED located next to the Arduino UNO logo. Turn the contrast potentiometer until text displays on the screen.
Device is not recognized by Eos	Make sure the version of Eos or ETCnomad you are running supports OSC-over-USB. Make sure the Arduino driver pack is installed on the machine running ETCnomad. Use the usb_test sketch from the Github repository to narrow the problem.