#lighthack box 1 Revision: 08

Assembly Instructions

Parts List

ETC PN	Vendor PN	Qty	Description
4201B9001	Mouser 782-A000066	1	Arduino Uno R3
DS221-F	Sparkfun LCD-00255	1	HD44780 Display Module
S846-F	Cherry MX1A-L1NB	3	Cherry MX Red Key Switch
4152A4039		3	Black Eos®-Style Key Cap
L1243-F	Digikey PEC11R-4220F-S0012-ND	2	Rotary Encoder with Panel Mounting Hardware
4201A4003	Adafruit 2055	2	Encoder Knob
4240B7011		1	10 kΩ Potentiometer
4201B7001		1	Red Wire Solid Core 22AWG/0.65mm^2 2-3'/60-90cm
4201B7002		1	Black Wire Solid Core 22AWG/0.65mm^2 2-3'/60-90cm
4201B7003		1	White Wire Solid Core 22AWG/0.65mm^2 2-3'/60-90cm
4201B7004		1	Yellow Wire Solid Core 22AWG/0.65mm^2 2-3'/60-90cm
4201B7005		1	Blue Wire Solid Core 22AWG/0.65mm^2 2-3'/60-90cm
J4630		3	5-position Wago Wire Connector
J4629		1	3-position Wago Wire Connector
W6378		1	USB Cable, A to B
HW0006		10	Machine screws #2-56x3/16" (~5mm)
HW9489		5	Standoffs #2-56x1/4" (~6mm)
4201A4001	Hammond 1591U	1	Enclosure with lid and screws

Tools

Necessary:

- Wire strippers and cutters
- Soldering iron and solder
- Philips screwdriver
- Drill with $^1/_8$" / 3mm, $^1/_4$" / 6mm, $^1/_2$" / 12mm bits
- Rasp/File or Chisel

Helpful:

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

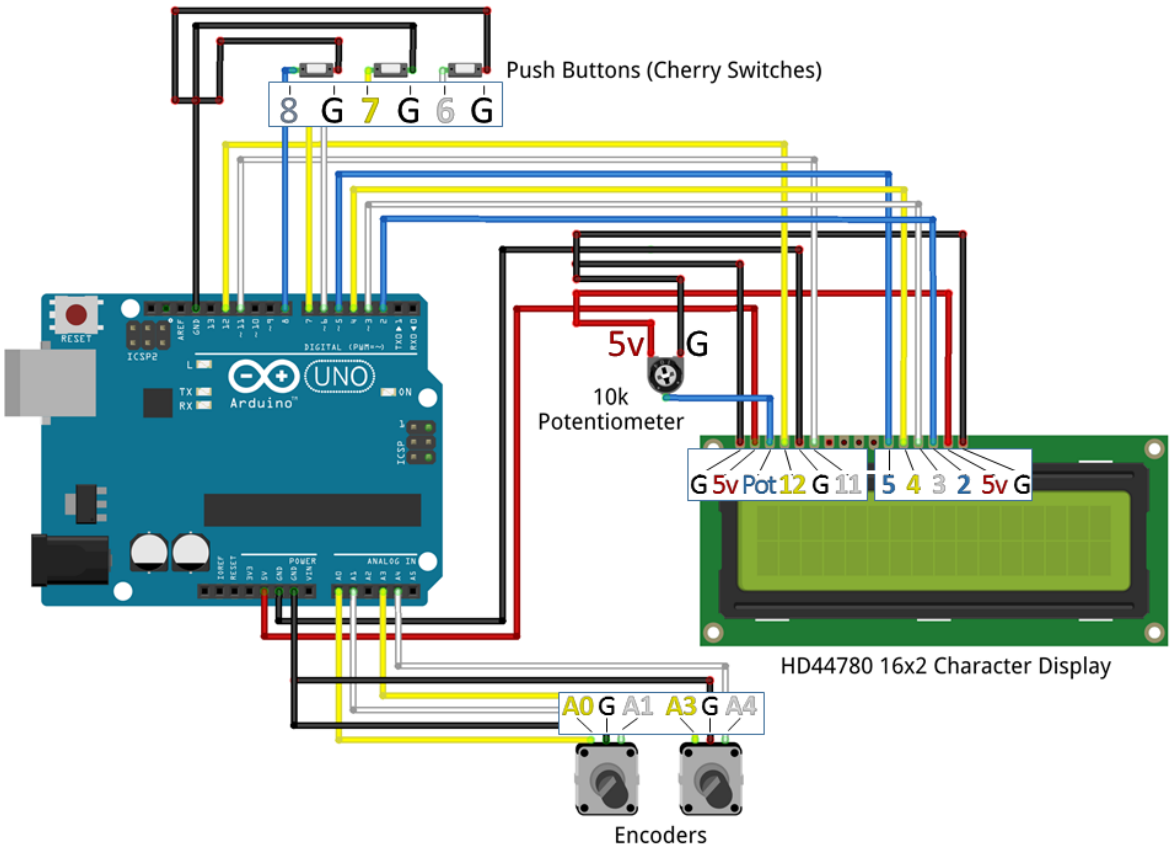
Software

- 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
- Arduino to Eos Test Application. Download from https://github.com/ETCLabs/lighthack

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

Wiring Diagram

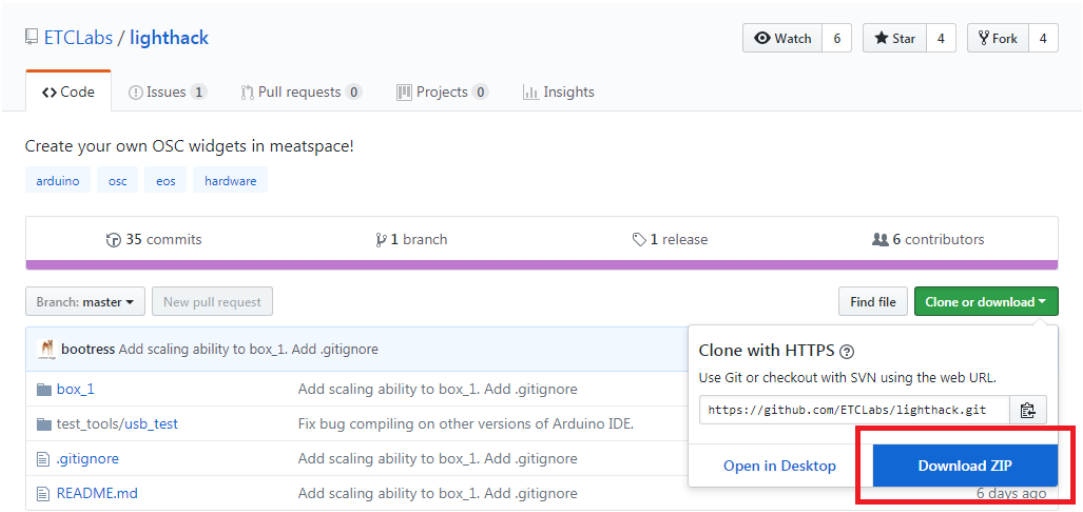
A full page diagram is available for download at https://github.com/ETCLabs/lighthack/tree/master/box_1

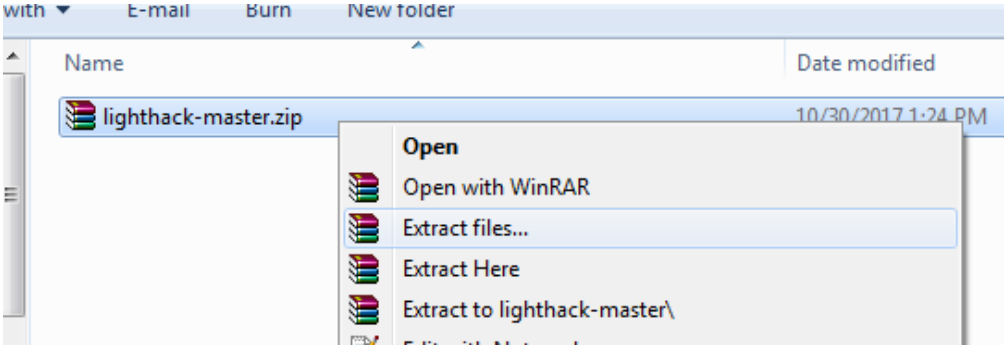


Test the Software

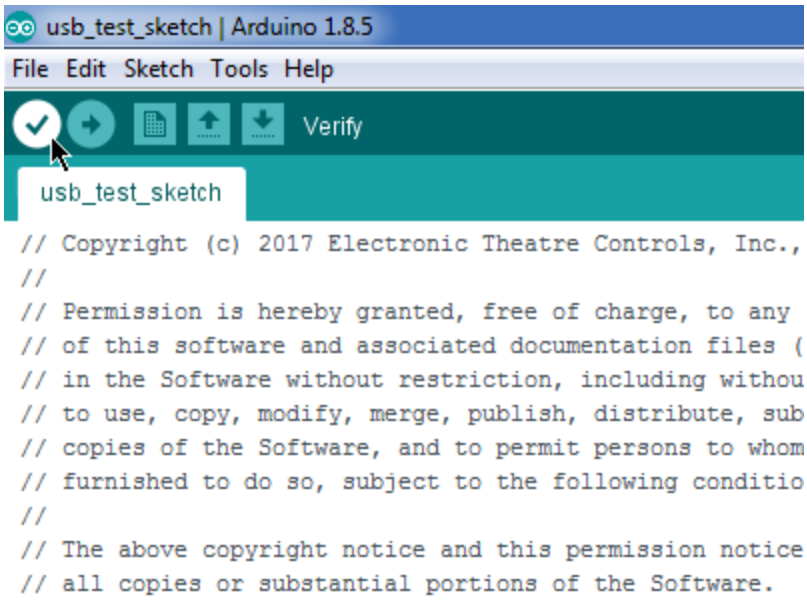
It is recommended that you test the Arduino before starting the assembly of the enclosure.

1. Download the source code from https://github.com/ETCLabs/lighthack, and extract the .zip file (or clone the repository if you're git-savvy!).

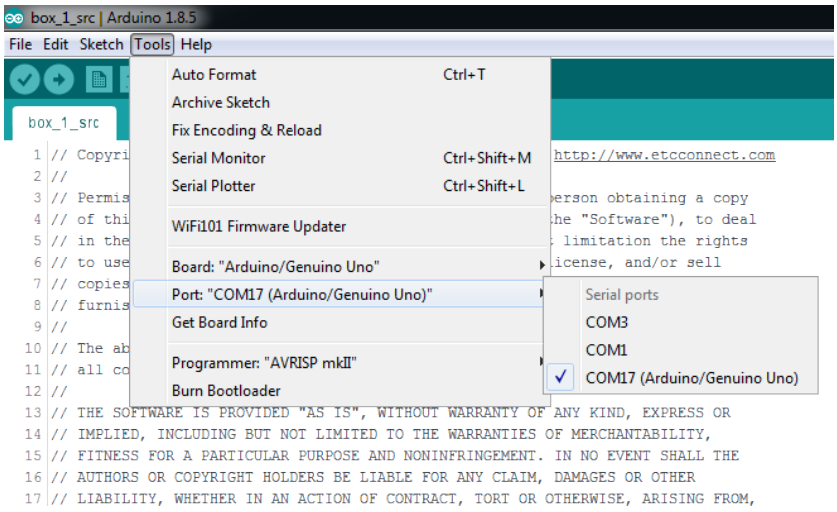




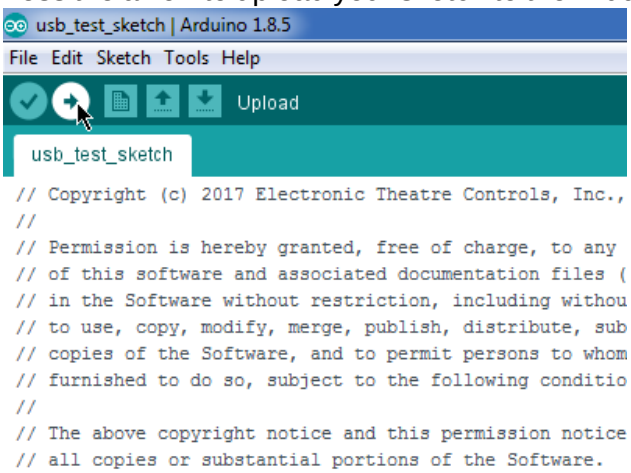
2. Open the Arduino USB Test sketch (lighthack-master\test_tools\usb_test\usb_test_sketch\usb_test_sketch.ino) in the Arduino IDE.
3. Before the sketch will compile, we need to add additional code (called a "library") so that the Arduino knows how to speak OSC. Download the library as a .zip from https://github.com/CNMAT/OSC.
4. In the Arduino IDE, select **Sketch > Include Library >Add .ZIP Library...** and select the OSC zip file you downloaded
5. Press the check mark to verify your sketch.



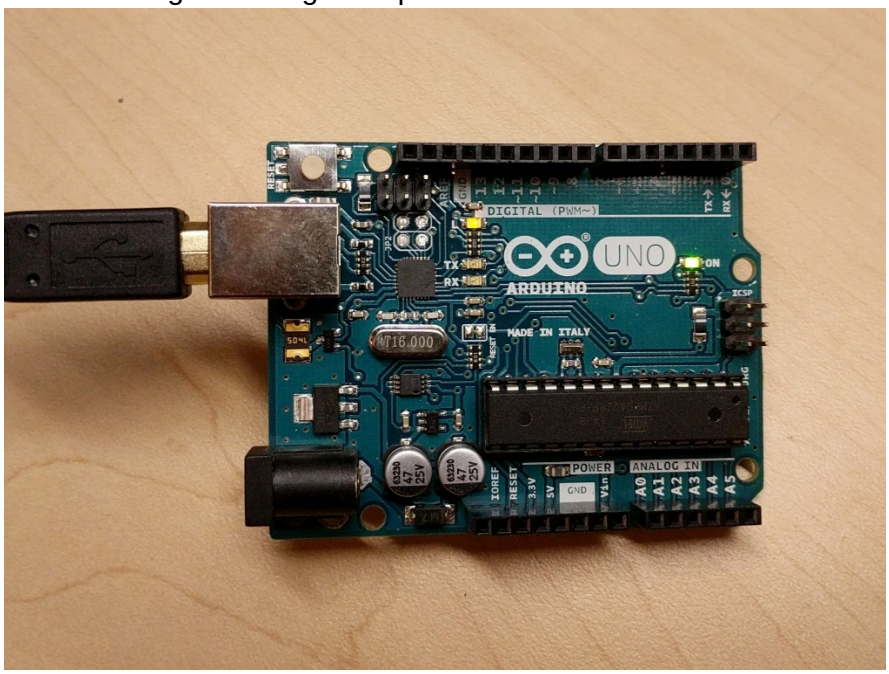
6. Connect the Arduino to your computer using the USB cable.
7. The Arduino IDE will automatically detect your Arduino Uno and select it. You can verify this in the Tools menu:



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



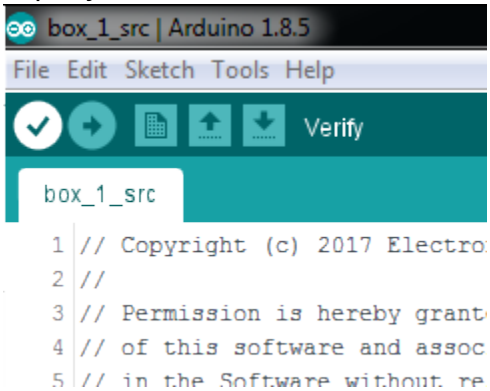
9. After the sketch has been loaded, the Arduino's lights should have the L and ON lights solid and the TX light blinking once per second:



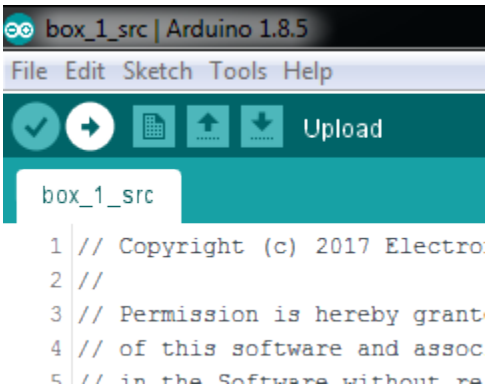
10. Launch Eos software, open Tab 99 Diagnostics, and look for messages that read:

    "OSC USB Device Handshake Complete [OK]"
    "OSC USB Device Handshake Initiated [ETCOSC?]"
    The TX and RX lights should blink in sequence once per second.
    Press "Incoming OSC" and you should see repeating messages of "[OSC Packet] /eos/ping," and a number counting up

    These messages indicate that your Arduino and Eos are talking to each other. The RX LED on your Arduino should also start blinking.
11. Exit the Eos software.
12. Now, load the main box_1_src.ino code. If you've added the OSC library properly, the verification will be successful.



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

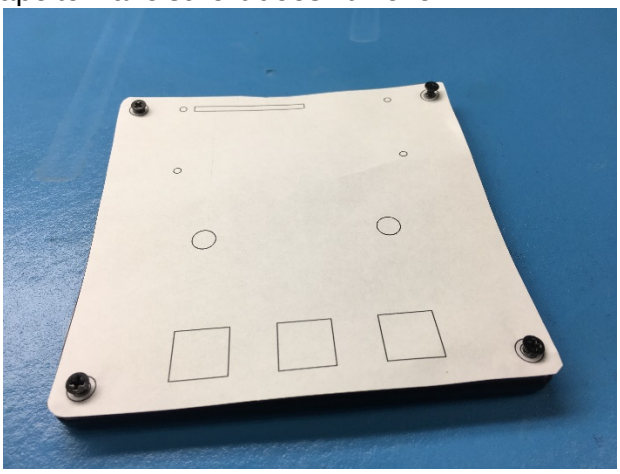


14. After the sketch has been loaded, start Eos again and look for the same Handshake messages. Note that in this mode, the Arduino does not send the repeating Ping messages, and the Arduino's TX/RX LEDs will only blink once.

If text does not appear, follow the troubleshooting steps at the end of this document.

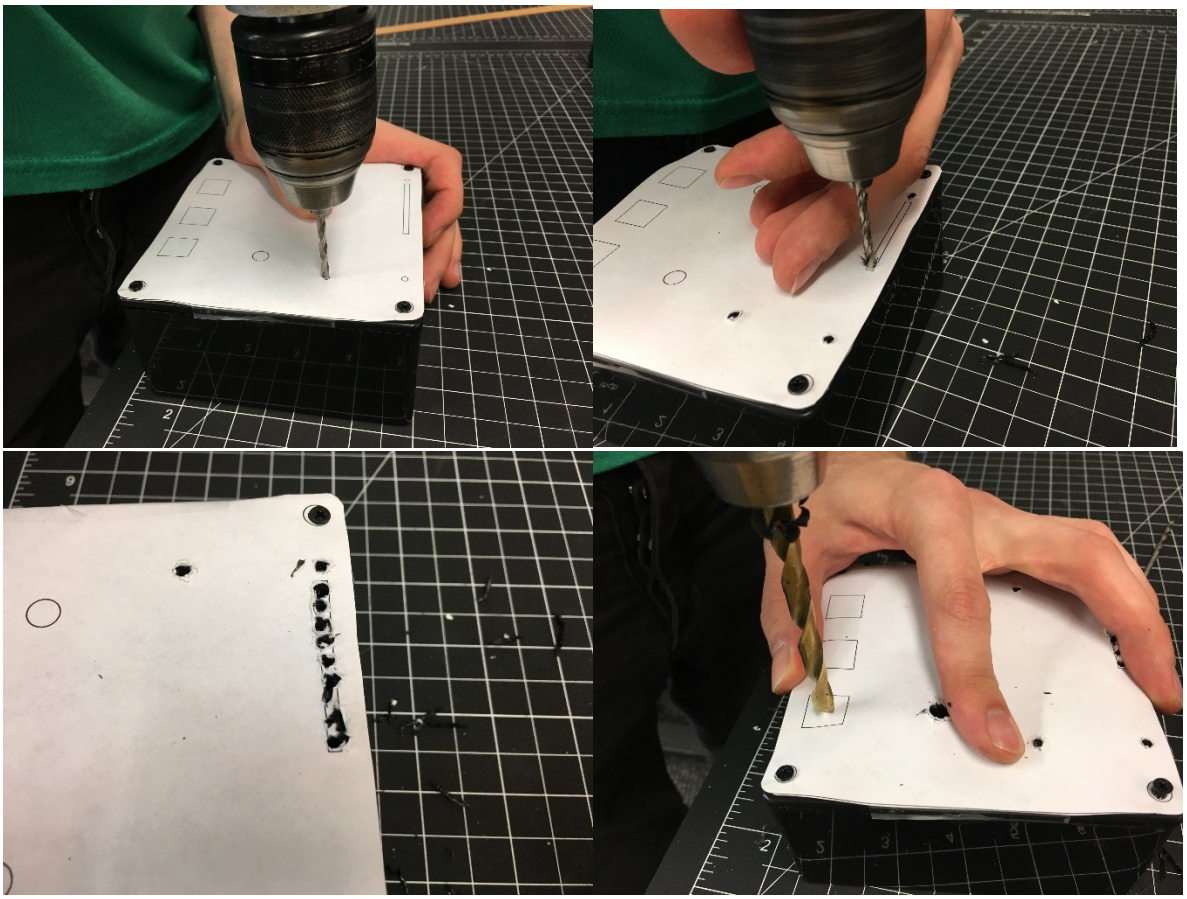
Prepare the Enclosure

15. Print out the paper template, making sure to disable any "fit to paper" or scaling options so that it prints at actual size. The template is found at https://github.com/ETCLabs/lighthack/tree/master/box_1.
16. 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.

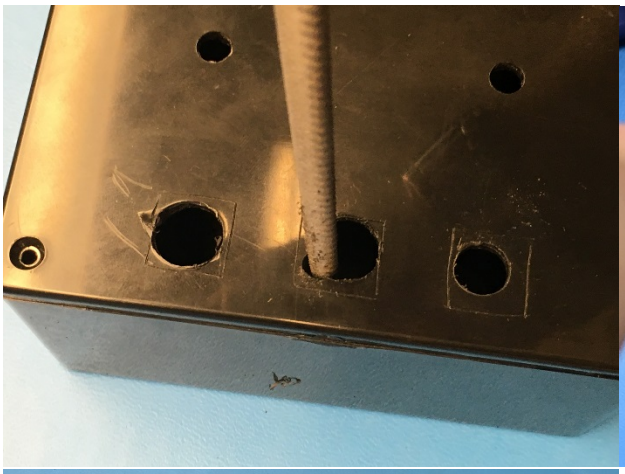


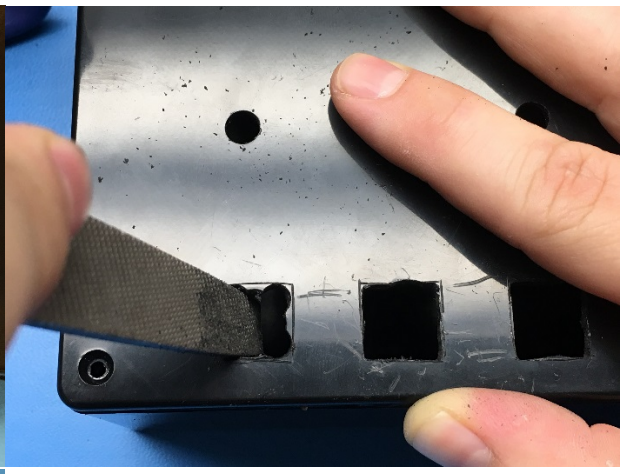
TIP: Don't feel like drilling and filing? Use the provided 3D Printer template to print a "pre-drilled" lid.

17. 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.
18. Using this template, the LCD display will be mounted to the outside of the lid. Skip to Step 13 if you want to follow the template. If you would like to mount the display to the inside of the box, cut a rectangular hole in the enclosure lid slightly larger than the face of the LCD screen.



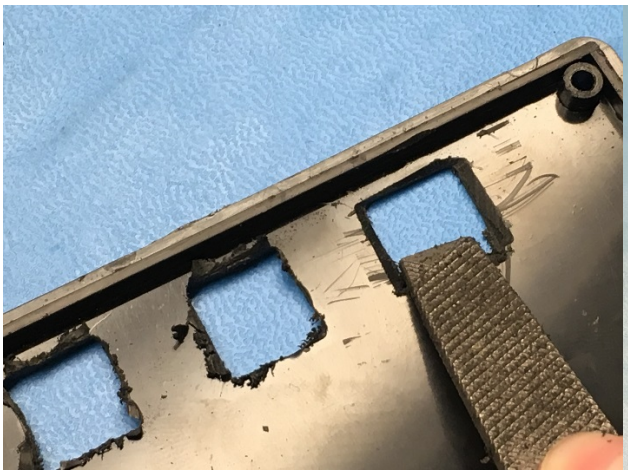
19. Using a square or rectangular file, file out the square holes and slot (or use your preferred method of making square holes).
20. Test fit the Cherry MX Red switches into the square holes as you file to make sure you have the right size.

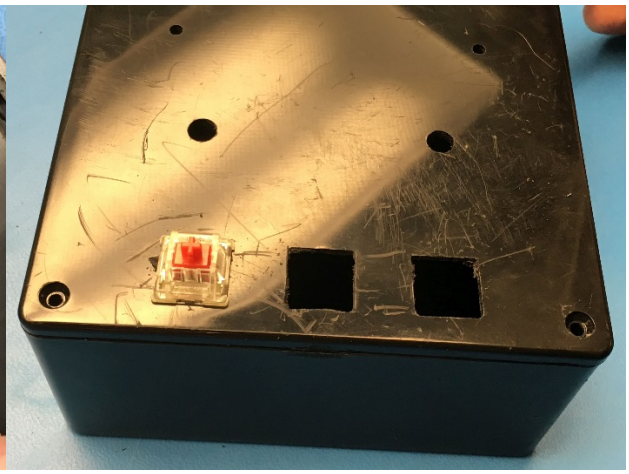


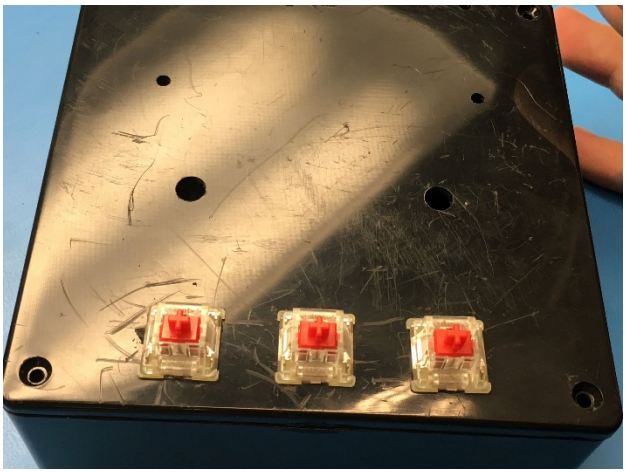


TIP: It may be helpful to trace the edges of the square holes into the plastic with an X-Acto knife, and then remove the paper.

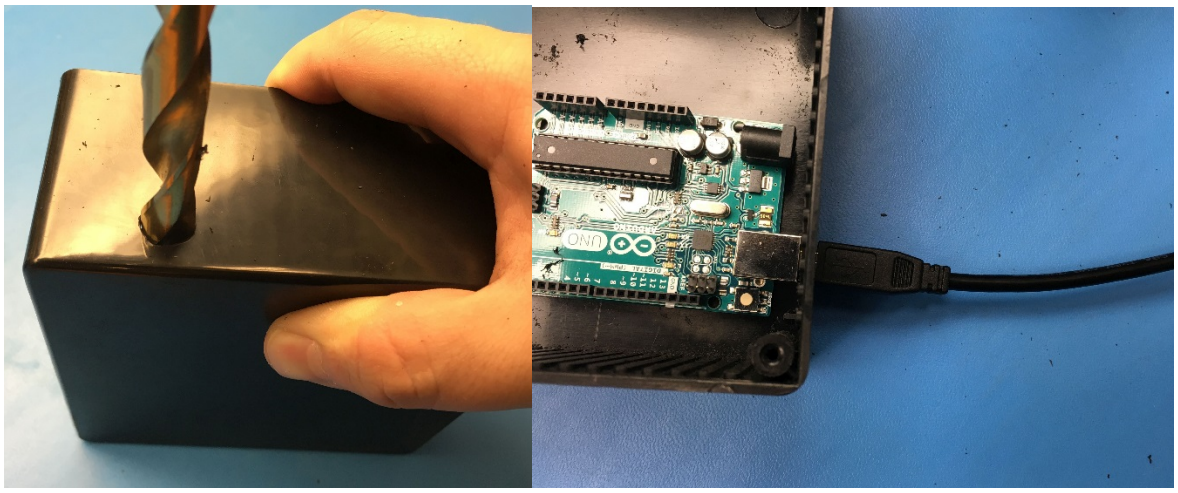
21. From the underside of the lid, use a file to bevel the edges of the square holes. This makes the plastic surrounding the hole thinner so that the Cherry MX Red switches can clip in properly.







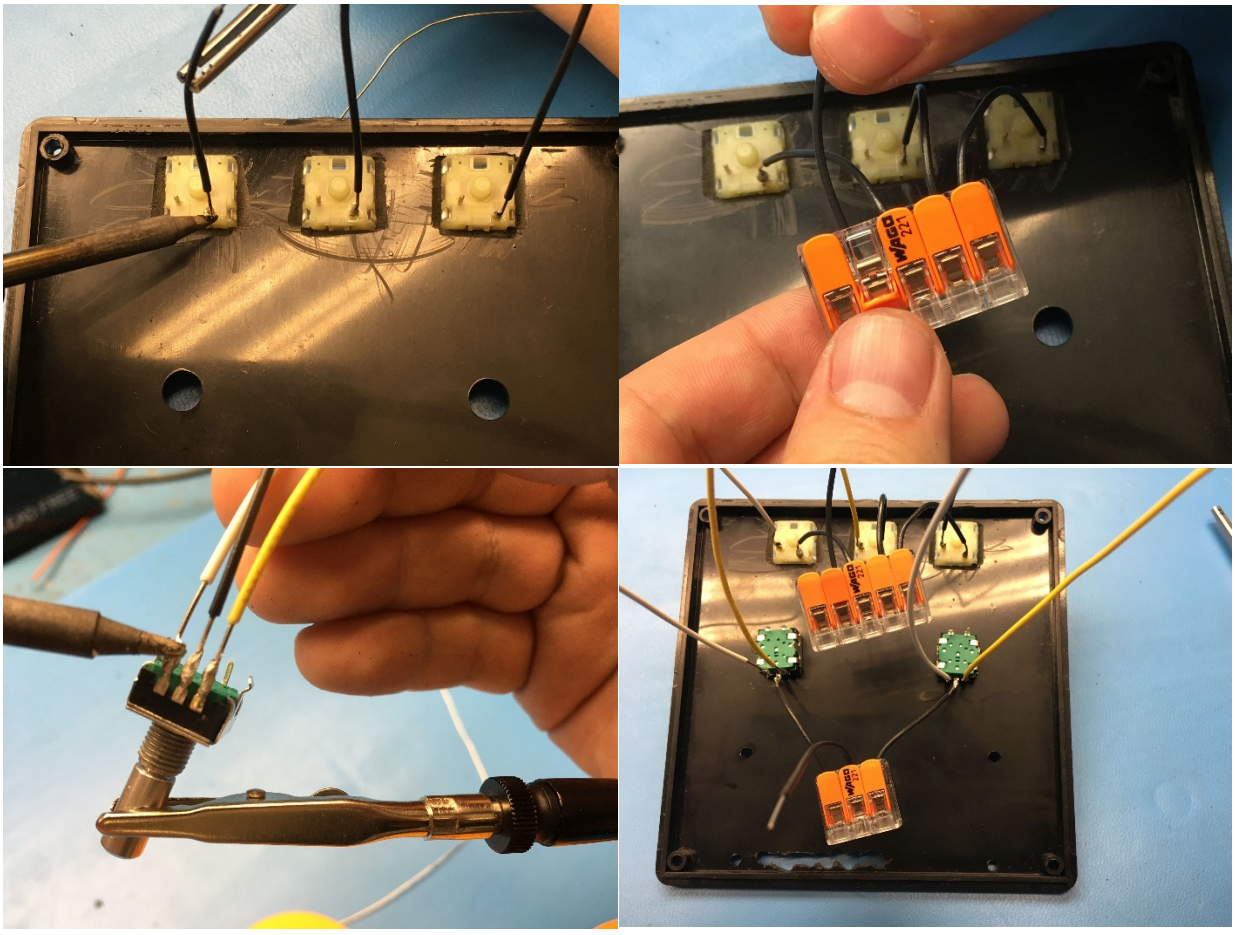
22. Line up the Arduino with a side of the box in order to estimate the location of the hole. We've found a good location to be 1"/25mm in from a corner and ½"/12mm up from the bottom. Use a ½"/12mm drill bit to drill a hole for the USB connector. Make sure you can connect a USB cable to your Arduino through the hole.
23. Drill a hole for the USB connector. If desired, add a second 1/2" hole to mount the potentiometer.



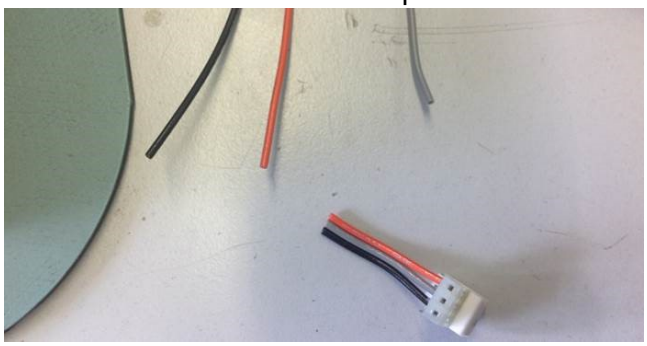
Connect the Electronics

24. 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. The wire colors in your kit may vary.
25. Strip approximately ½"/12mm of insulation from either side of each length of wire.

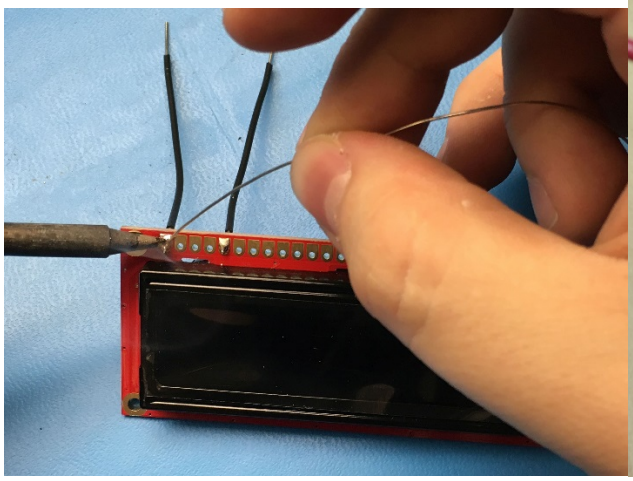
26. Solder one end of each wire length to pins on the rotary encoders and key switches.
27. Use the Wago connectors where multiple black (ground) wires need to be connected together. Only three of the five leads on the encoders need to be connected, as shown. You will want to have the key switches inserted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

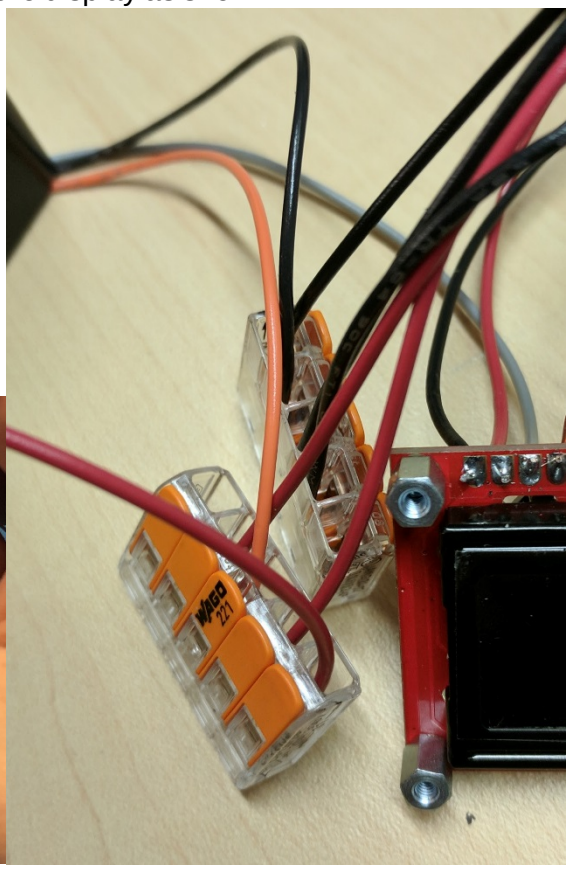


28. Cut appropriate lengths of wire to connect the display and contrast adjustment potentiometer to the Arduino as shown in the wiring diagram.
29. Cut off the connector from the potentiometer.



30. If you are mounting it to the box, thread on the rubber washer and locking ring, string the wire through the box, then tighten with the nut. This has to be done before soldering.
31. Strip approximately ½"/12mm of insulation from either side of each length of wire.
32. Solder one end of each wire length to the display as shown.
33. Connect the middle pin of the potentiometer to the display as shown.





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

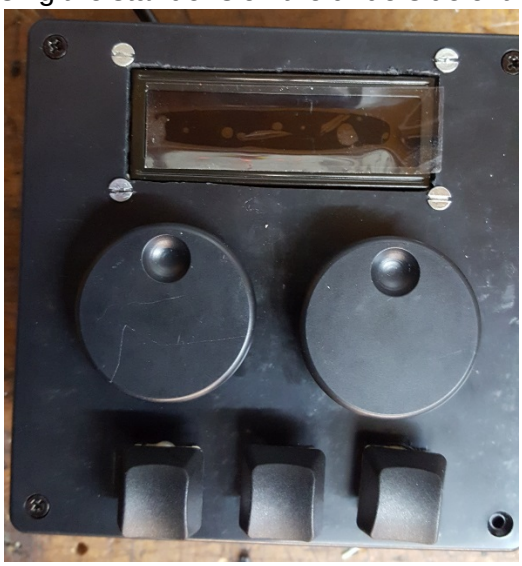
35. Before mounting everything in the enclosure and finishing the assembly, the electronics should be tested. Connect the Arduino to a Win7 Eos Family console or a computer running ETCnomad.
36. Test the device by patching a moving light, selecting it and using the encoders to modify the pan and tilt 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

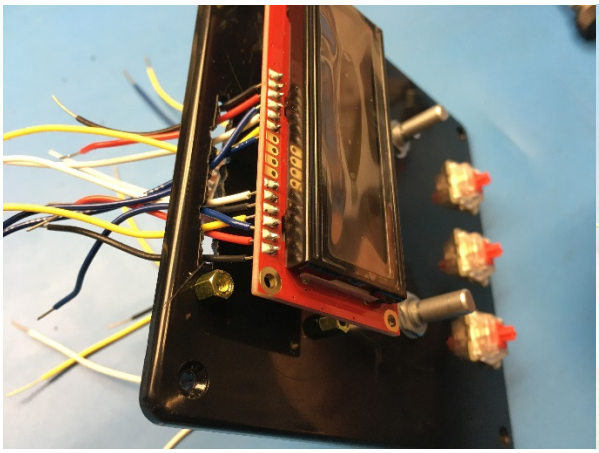
Now that your module has been tested, it's time to put it in the enclosure.

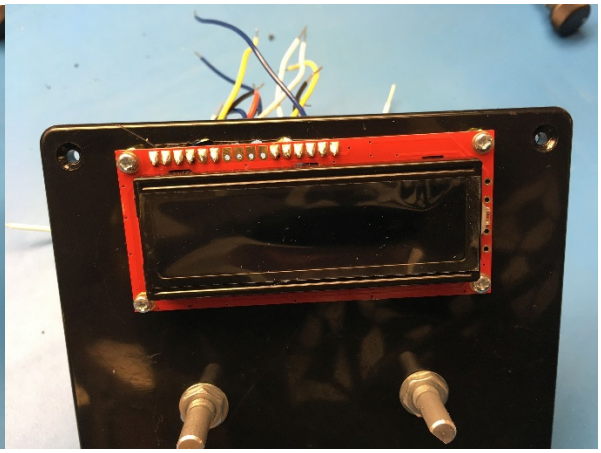
37. Insert the two encoders through the holes from the inside of the lid.
38. Use the included panel mounting nuts and washers to secure the encoders.

39. Disconnect the screen from the Arduino and feed its wires through the rectangular slot. If mounting to the inside of the box, you would instead mount the LCD screen inside the lid using the standoffs on the underside of the lid.

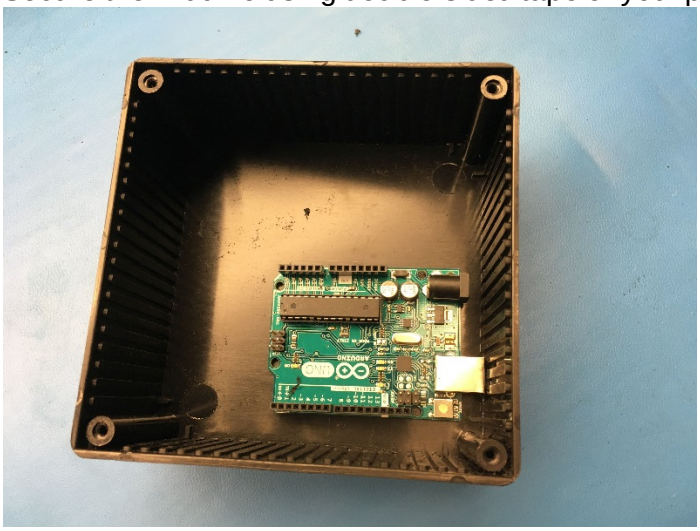


40. Secure the screen in place using eight #2-56 machine screws (four from the top, and four from the bottom) and either the ¼" (~6mm) or ½" (~12mm) standoffs. Extra screws and standoffs are provided.

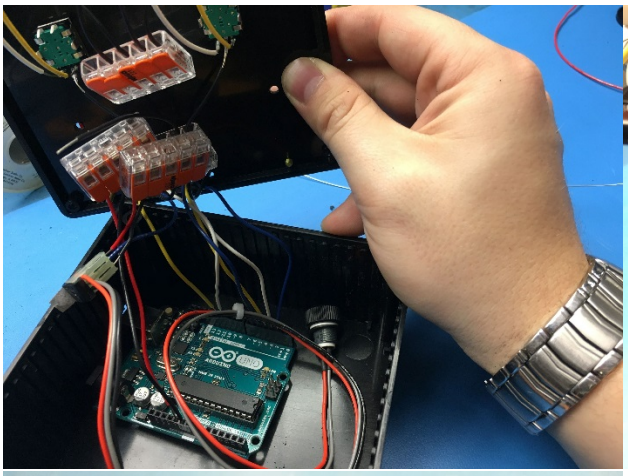


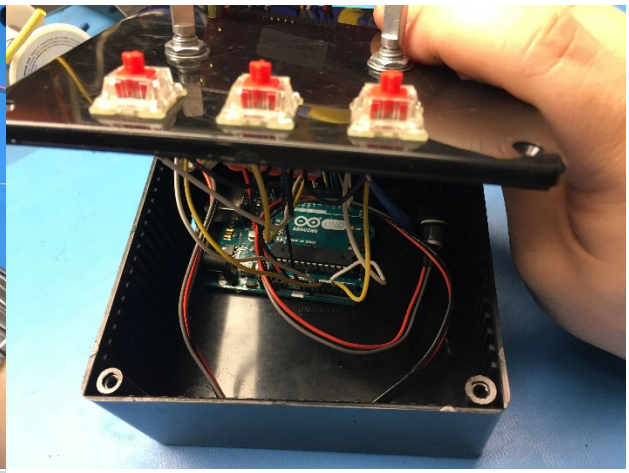


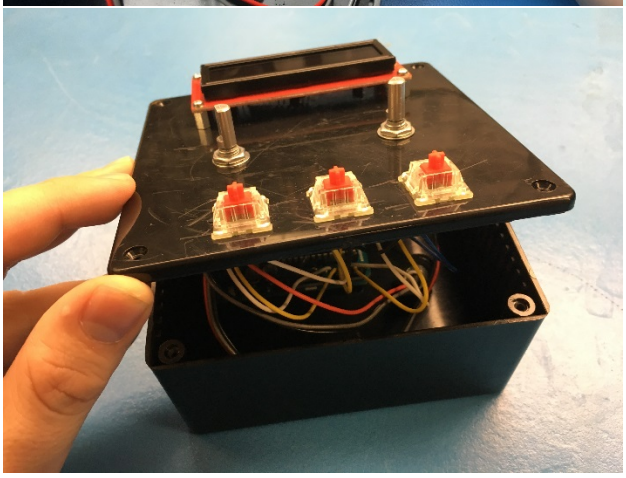
41. Place the Arduino into the enclosure such that the USB connector is aligned with the hole you drilled earlier.
42. Secure the Arduino using double-sided tape or your preferred method.



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



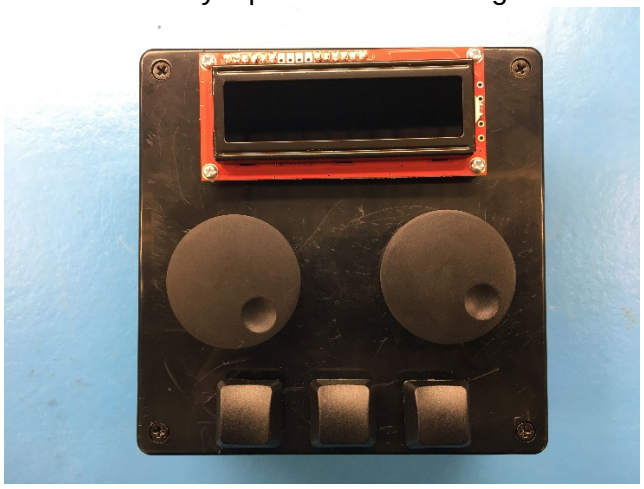




TIP: You may want to use some electrical tape for cable management.

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

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

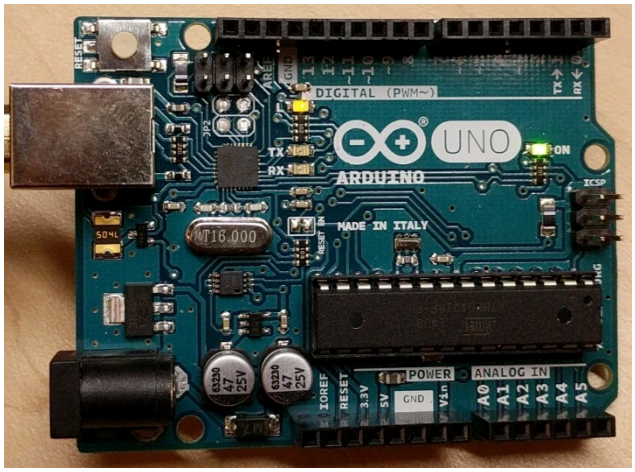


46. Re-test the module to make sure everything still works. Congratulations! You're done.

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 discussed in the Test the Software section.
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 Family device	Make sure the version of Eos or ETCnomad software you are running supports OSC-over-USB. Make sure the Arduino driver pack is installed on the device running ETCnomad software. Use the usb_test sketch from the Github repository to narrow the problem.

Arduino LED Guide



LEDs to be aware of:

- ON (green): power presence
- L (amber): presence of code
- TX (amber): transmitting data
- RX (amber): receiving data

When you plug the Arduino in for the first time:

- On: solid
- L: possibly blinking
- TX: off
- RX: off

When you are loading sample or test code to the Arduino:

- On: solid
- L: will go off
- TX: flashing then solid
- RX: flashing then solid

When you have loaded the USB test sketch, but are not connected to a compatible Eos Family device:

- On: solid
- L: solid
- TX: heartbeat
- RX: off

When you have loaded the USB test sketch, and are connected to a compatible Eos Family device:

- On: solid
- L: solid
- TX: heartbeat
- RX: hearbeat

When you have loaded the Box 1 sketch, but are not connected to a compatible Eos Family device:

- On: solid
- L: will go off
- TX: flashing then solid
- RX: flashing then solid

When you have loaded the Box 1 sketch, and are connected to a compatible Eos Family device:

- On: solid
- L: solid
- TX: flashes when you move pan/tilt or hit next/last on the box
- RX: flashes when you move pan/tilt on Eos Family device