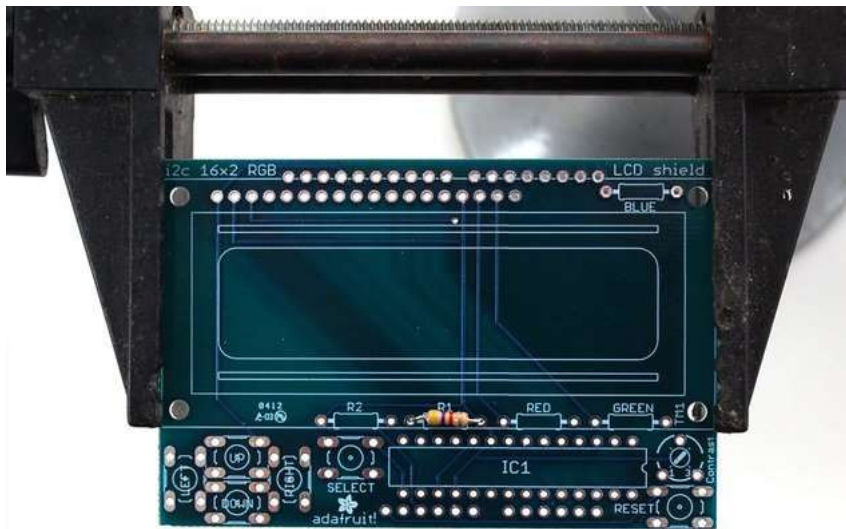
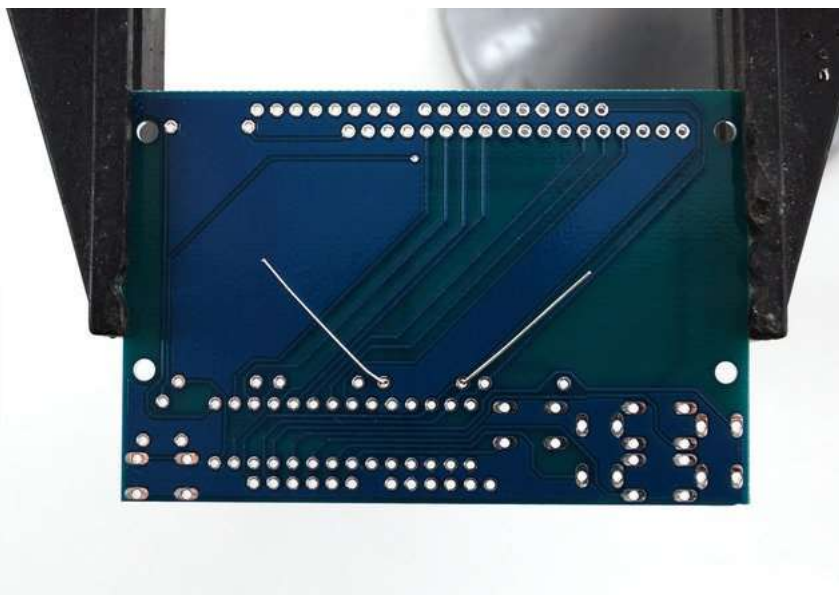


Put the printed circuit board into a vise or board holder, heat up your soldering iron and make sure you're ready to go!



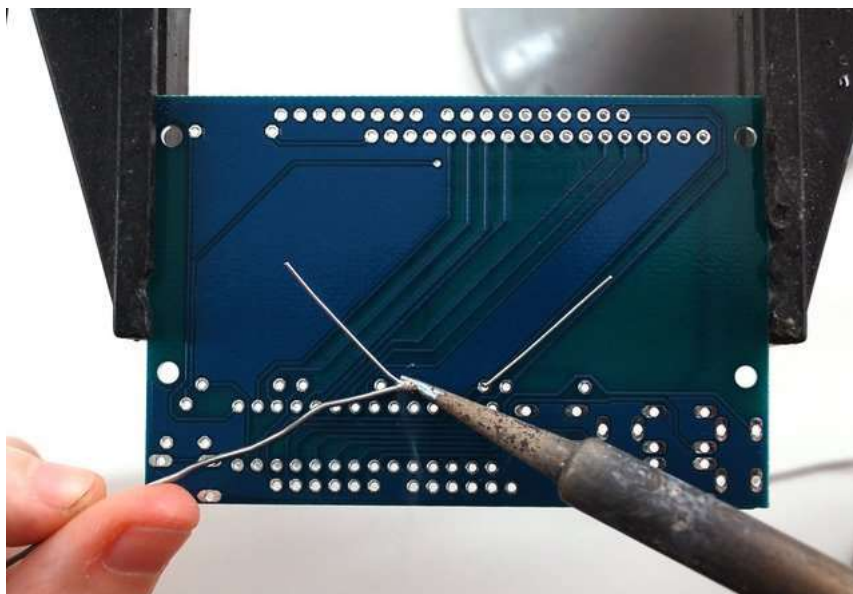
We'll start with the first resistor **R1** - which has yellow, violet, red, gold bands on it. This resistor is part of the interface between the Arduino and the 'port expander' that controls the LCD.

Bend the resistor into a 'staple' and slide it into the slot marked **R1** on the PCB. Resistors do not have a direction so you can put it in 'either way' and it'll work fine.



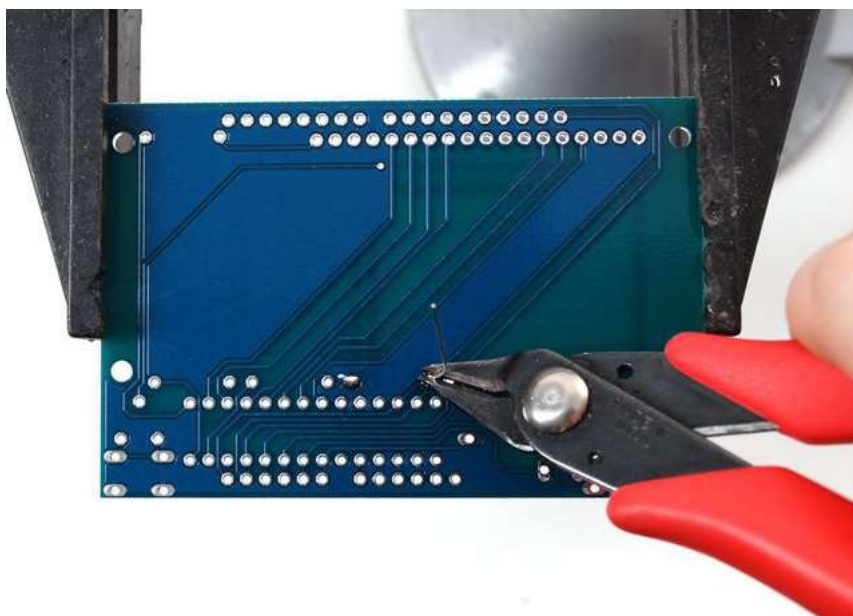
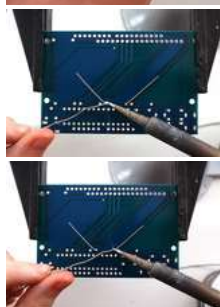
Bend the 'legs' of the resistor out so it sits flat against the PCB and flip it over.

This way the resistor won't fall out while soldering.



With your soldering iron heated up and ready, solder in both leads of the resistor. To do this, heat up the round ring pad and the wire lead at the same time for 2 or 3 seconds, then dip the end of the solder into the heated joint to melt it in.

Then remove the solder and the soldering iron.

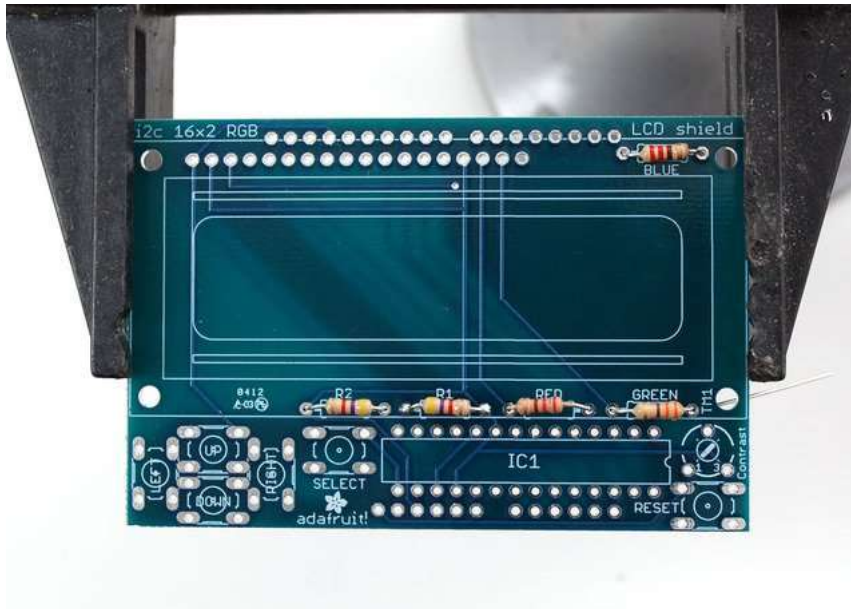


Once the soldering is complete, we can clean up by clipping the leads of the resistor. This keeps them from shorting to something else. Use diagonal or flush cutters to clip the wires right above where the solder joint ends.

Since you did so great with the first resistor, we'll place all of the rest now at the same time. **R2** is a 4.7K resistor just like **R1** (Yellow, Violet, Red Gold) so place that one first

Next you can do the two 220 ohm resistors **RED** and **BLUE** - named because they are the backlight series resistors for the RGB backlights on the LCDs. These resistors are colored **Red Red Brown Gold**. They look very similar to the 330 ohm resistor so check carefully that you have the red-band ones. Use a multimeter if you can to verify the resistance!

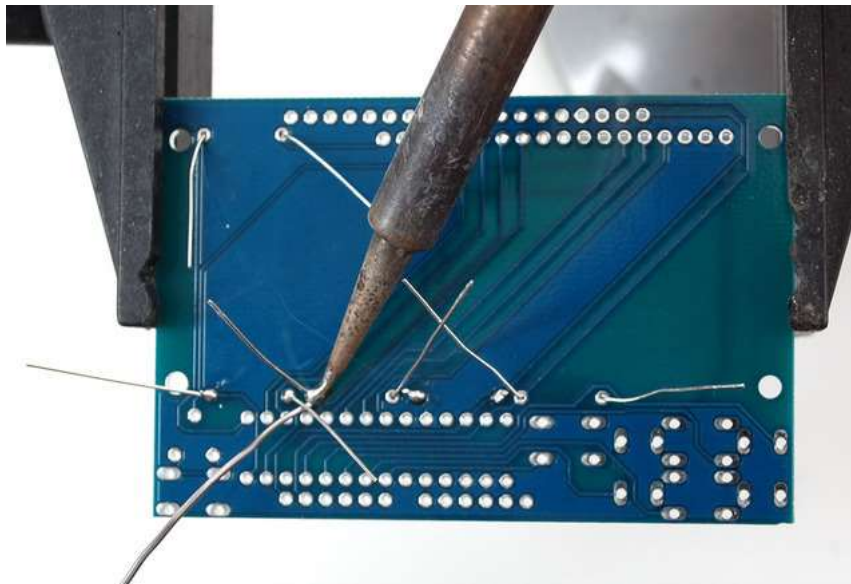
Finally, place the remaining 330 ohm resistor (Orange, Orange, Brown, Gold) into the **GREEN** spot. The green LED is might



brighter than the red or blue in the RGB backlight so a larger resistor is required to balance it out.

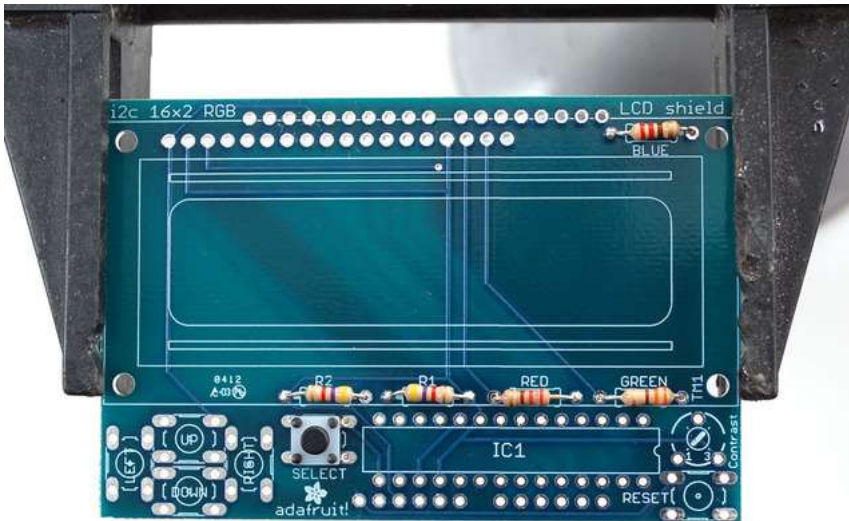
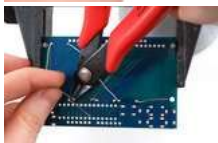
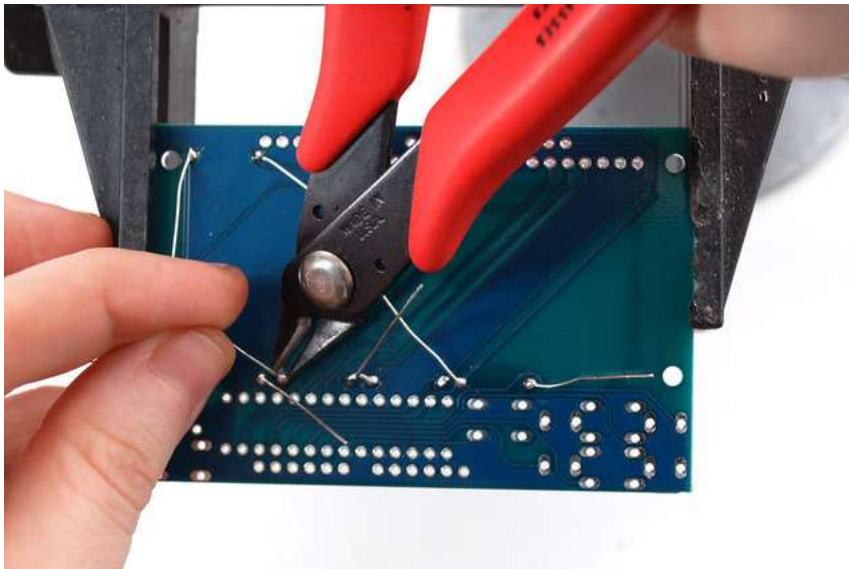


Make sure the resistors all sit nice and flat against the PCB.



Solder the resistors just like you did with the first one.

Clip all the leads.



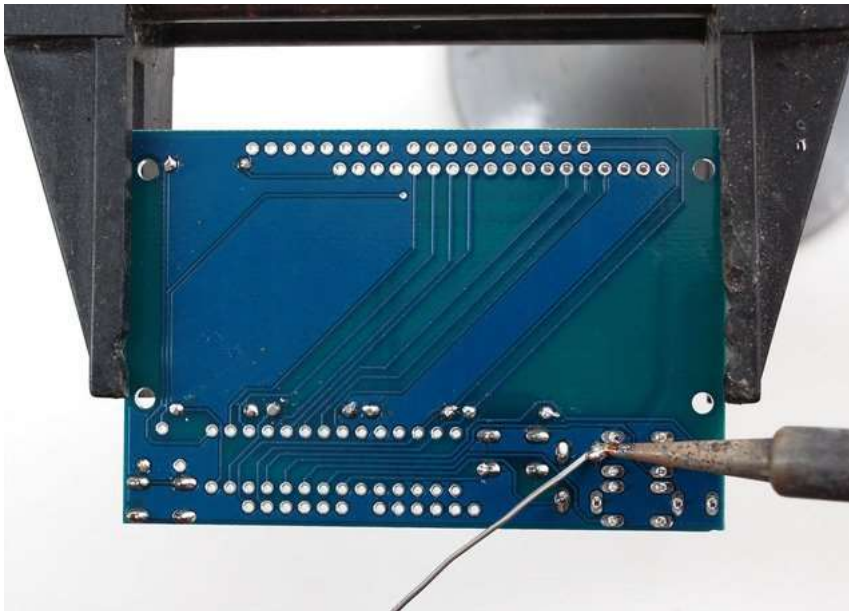
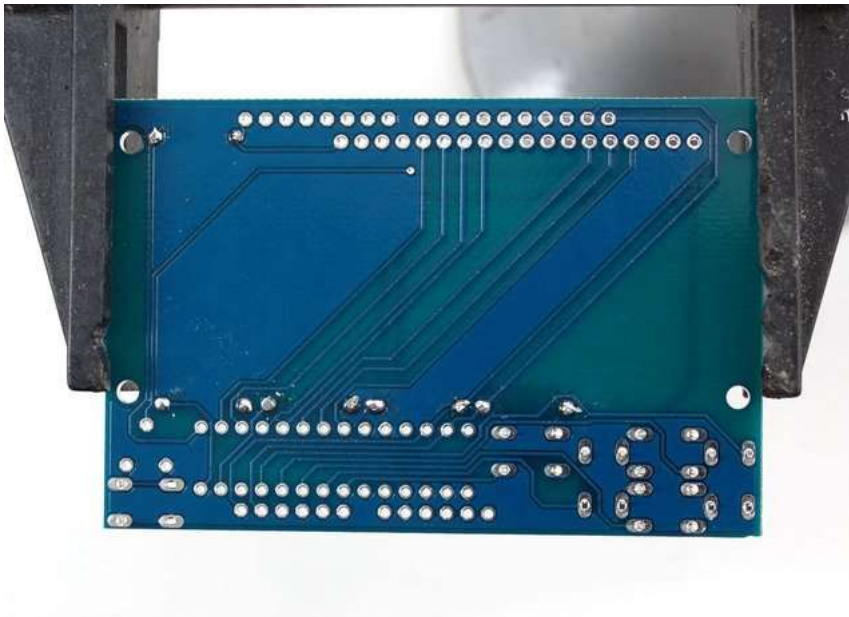
Next up we will place the buttons. These buttons are useful to send a signal to the Arduino (say if you have a basic menu system). We have a 4-way 'direction pad' for up/down/left/right input and a button to the right called SELECT. These 5 buttons should be able to make 'talking' back to your project easy. These are connected to the I2C port expander chip so they require no extra pins on the Arduino, our library does the work of reading whether they are pressed.

The button all the way to the right is the RESET button, for when you want to reset the entire system.

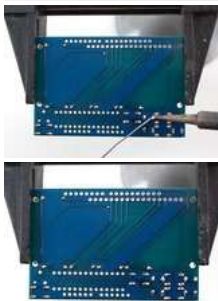
All the buttons are the same, and they should snap nicely into place. Press down onto each button until it snaps in and sits flat against the PCB.



Flip over the PCB and check that all the legs for the buttons are sticking out.

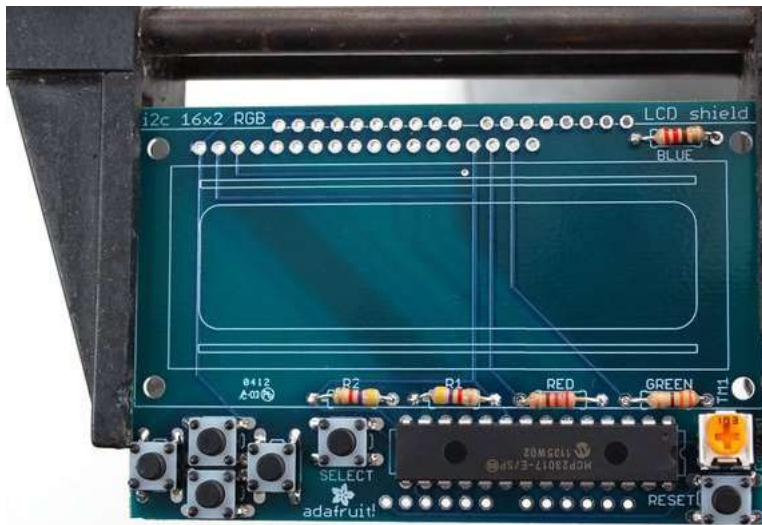


Solder each leg, taking care not to accidentally 'short' two button legs together. The ones for the directional pads are very close!



We're nearly done! Now we will place the I2C port expander chip. Double check that it has the **MCP23017-E/SP** marking on it. This is a 16-pin expander chip, that uses the i2c bus. That means you can send it commands using the i2c pins on an Arduino and control 16 more digital pins! 5 of those pins go to the buttons, 6 go to the LCD control and 3 are used for the backlight (the remaining 2 are unused).

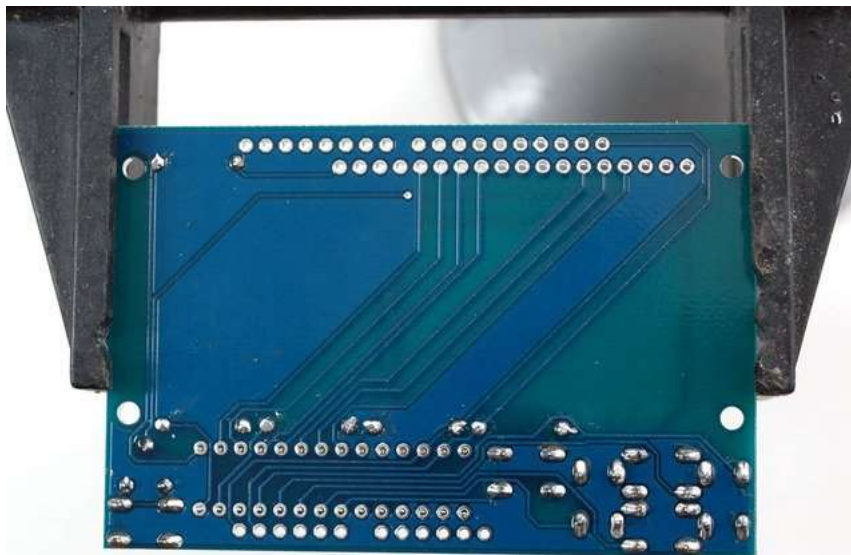
Unlike buttons or resistors, chips do have a direction and the must be put in the right way! First, use a flat table to carefully bend the legs of the chip so they are parallel. Then slip it into the silkscreened outline so that the notch at the end of the chip is on the *right*. Click the image to the left to make absolutely



sure you've got it in the right way. Once you are sure, press the chip into place.

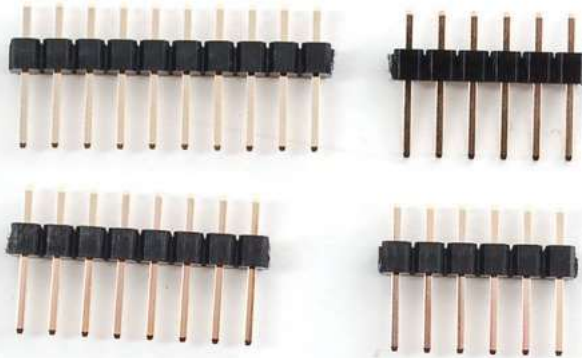
Next, place the 10K potentiometer (the orange-faced thing with three legs) into the spot above the RESET button. It will only fit one way. This is the contrast potentiometer which will adjust how dark the characters appear. All displays are slightly different so you'll adjust this once the display is soldered in.

The kit may come with two potentiometers - a big blue one for breadboarding the LCD and a smaller orange one for the shield kit. You can throw away or recycle the blue one, use only the orange one here!



Flip over the PCB and solder in the three legs of the potentiometer and then all 28 pins of the port expander.





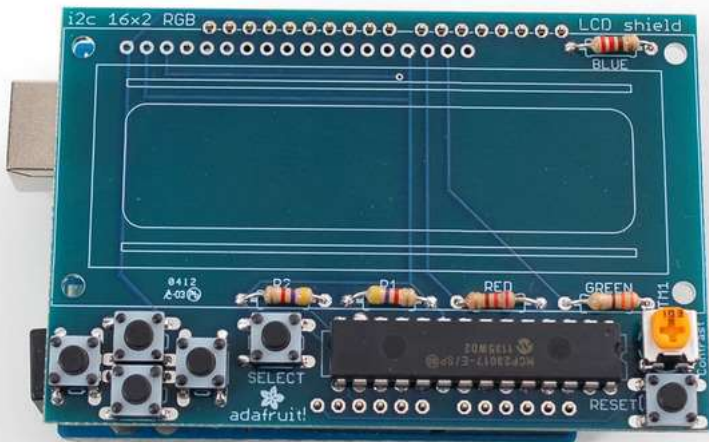
Take a stick of 36 pin header and break off a 10 pin piece, an 8 pin piece and two 6 pin pieces.



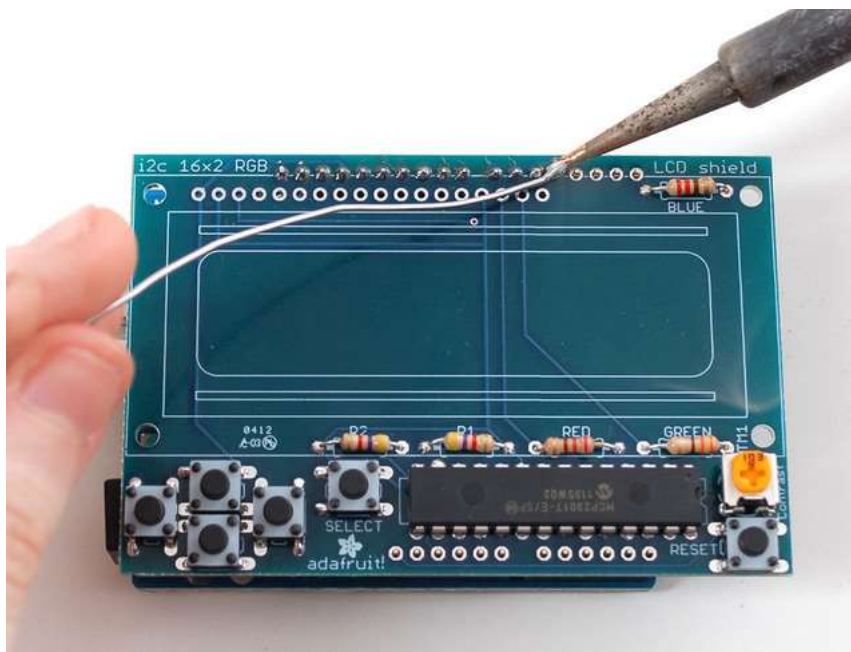
Place the header onto your Arduino, if you have an R2 or earlier arduino, there will be two pins that 'hang over' past the AREF breakout. These two extra pins are used for R3 and later Arduino classics and Mega's to make connecting to the I2C pins easier. We suggest keeping those two pins in case you ever upgrade your 'duino.

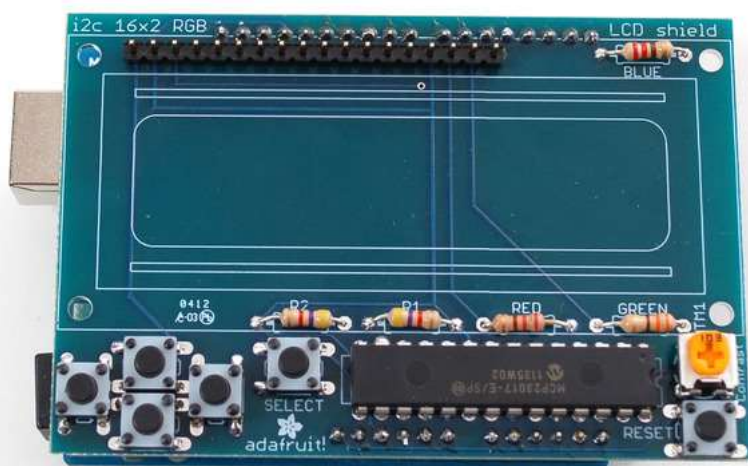
Place the shield right on top. It should fit perfectly where all the short ends of the header fit into the holes of the shield.

To make it sit flat, you may need to clip the legs of the buttons down - they will interfere with the DC jack, lifting the shield a bit.



Solder all of the header pins. Even though they aren't used for passing signals, they will make the mechanical connection to the Arduino strong.





Next, check if you have an RGB LCD (with 18 pins) or a Monochrome display (not RGB, with 16 pins).

If you have an RGB 18 pin display break off a 18-pin strip of header and place it with the long ends down into the LCD placement slot.

If you have a Monochrome 16 pin display break off a 16-pin strip of header and place it with the long ends down into the LCD placement slot so that the **two right-most pins** are empty.

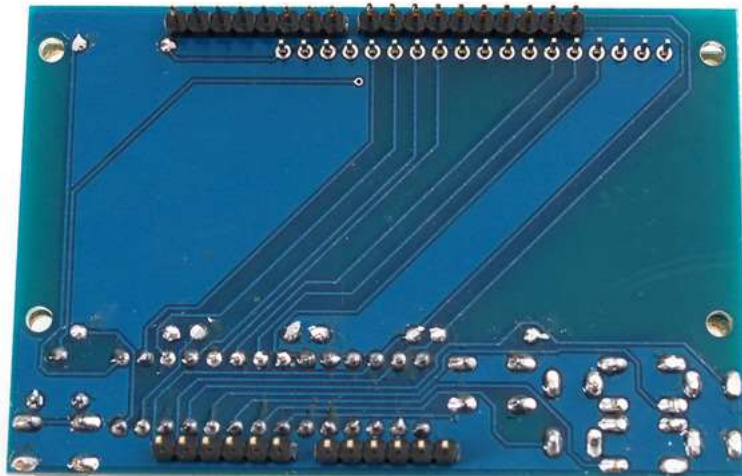
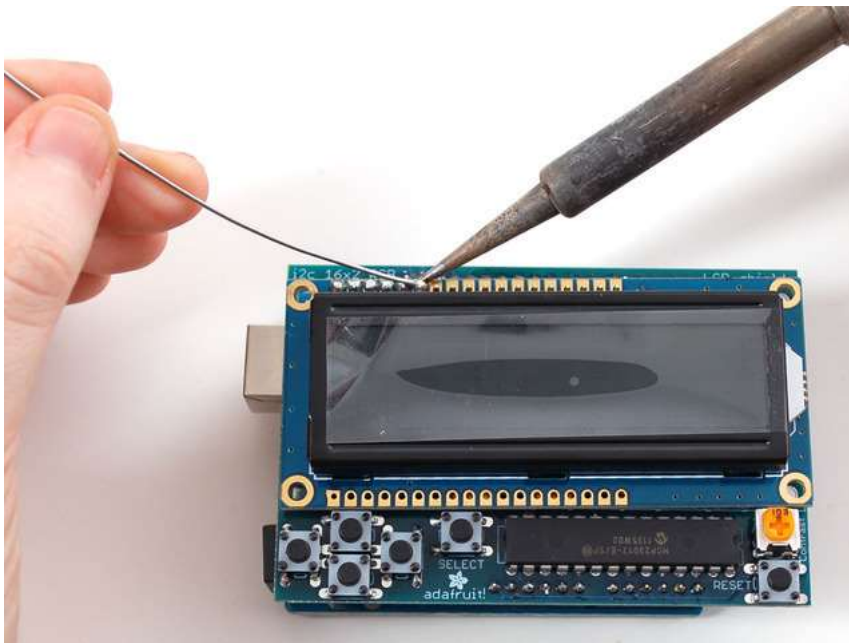


Now you can take the 16x2 LCD of your choosing and place it on top. For the RGB LCDs we stock, there are two connection rows, just make sure it matches the outline silkscreen on the shield and you should be good to go.

If your LCD does not have an RGB backlight or it has no backlight, you may have to shorten the length of header (see the previous step) until it matches and make sure the LCD fits over the silkscreen outline.

If you already soldered all 18 pins but you have a 16 or 14 pin LCD, simply cut off the right-most pin tops until it fits.

Solder all the pins to the LCD display.

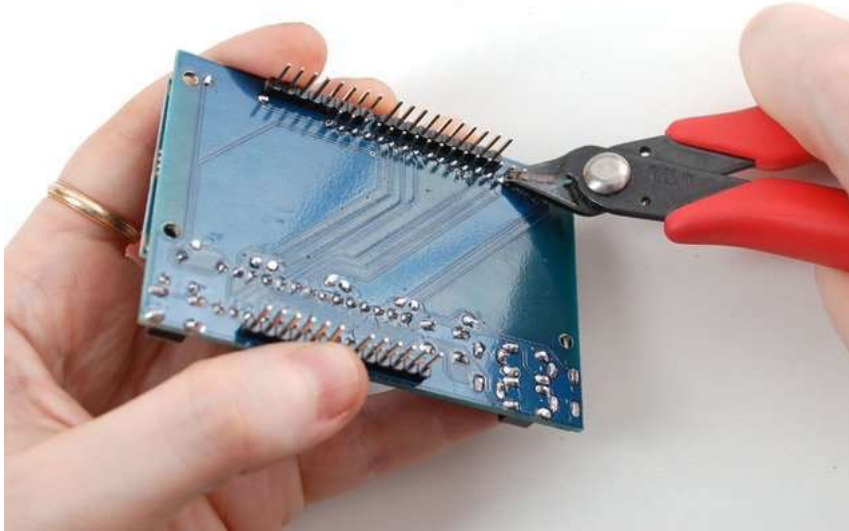


Flip over the shield and place it flat on the table so the long leads of the header stick out.

Solder each of the header legs.



Finish up by carefully clipping the leads. For some Arduinos, the long leads may interfere with headers or parts on the Arduino board so we suggest just clipping them all. Do it carefully as these are a little difficult to cut and may fly at you. We suggest doing it over a trash can.



You're done!

Check out the next page for how to use the shield and libraries.

