



Adafruit 16x2 Character LCD + Keypad for Raspberry Pi

Created by Ladyada



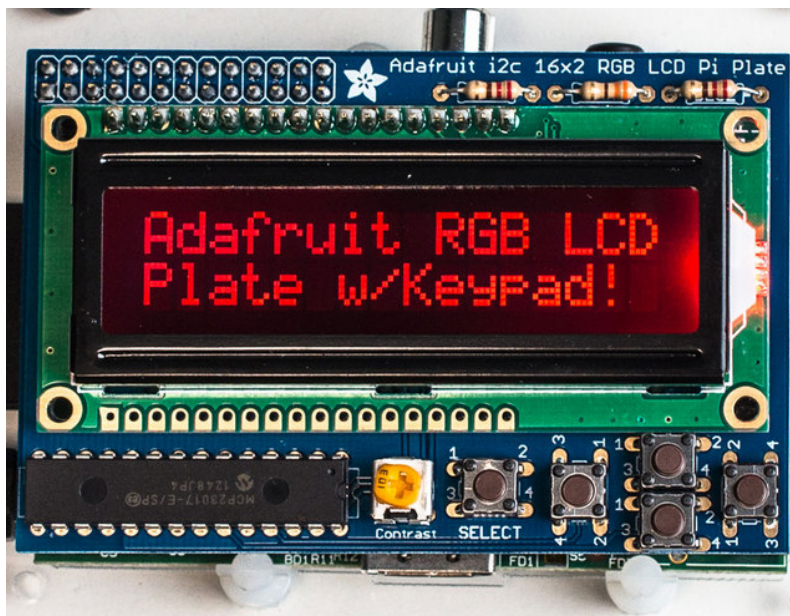
Guide Contents

Guide Contents	2
Overview	3
Parts List	5
1) Resistors	5
2) Potentiometer	5
3) Pushbuttons	5
4) i2c Port Expander Chip	5
5) Male Header Pins	5
6) Printed Circuit Board	6
7) Raspberry Pi Plate Header	6
8) Bumper	6
Additional	6
Assembly	7
Usage	18
Setting up your Pi for I2C	18
Using the example Python code	19
Testing the Library	19
Adjusting Contrast	20
Using the library code	20
Download	21

Overview



This new Adafruit plate makes it easy to use a 16x2 Character LCD. We really like the range of LCDs we stock in the shop, such as our classic [blue & white](http://adafru.it/181) (<http://adafru.it/181>) as well as the fancy [RGB negative](http://adafru.it/399) (<http://adafru.it/399>) and [RGB positive](http://adafru.it/398) (<http://adafru.it/398>). Unfortunately, these LCDs do require quite a few digital pins, 6 to control the LCD and then perhaps another 3 to control the RGB backlight for a total of 9 pins. That's nearly all the GPIO available on a Raspberry Pi!



With this in mind, we wanted to make it easier for people to get these LCD into their projects so we devised a Pi plate that lets you control a **16x2 Character LCD, up to 3 backlight pins**

AND 5 keypad pins using only the two I2C pins on the Pi! The best part is you don't really lose those two pins either, since you can stick i2c-based sensors, RTCs, etc and have them share the I2C bus. This is a super slick way to add a display without all the wiring hassle.



This plate is perfect for when you want to build a stand-alone project with its own user interface. The 4 directional buttons plus select button allows basic control without having to attach a bulky computer.

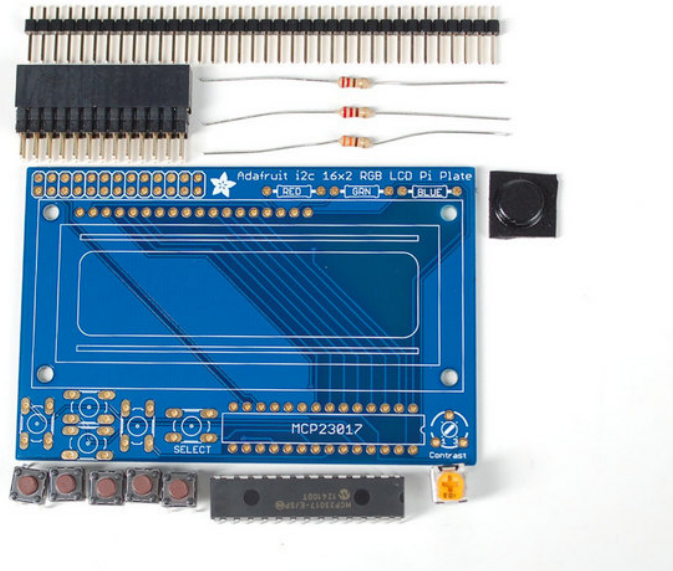
The kit is designed for any Raspberry Pi - version 1 and 2.

If you want plug in a Cobbler or Gertboard at the same time, check out our [Stacking Header](http://adafru.it/1112), you can fit an IDC cable over it if the Plate is assembled with this part. (<http://adafru.it/1112>)

Parts List

Check to make sure your kit comes with the following parts. Sometimes we make mistakes so double check everything and email support@adafruit.com if you need replacements!

We recently adjusted the kit so the buttons are on the RIGHT side instead of the left. The parts list is otherwise the same, its just a little more stable than before



1) Resistors

There is a total of 3 resistors in this kit. For resistors labeled RED and BLUE on the PCB, they are 1/4W 5% 220 ohm resistors (Red, Red, Brown, Gold). For the resistor labeled GREEN on the PCB, it is a 1/4W 5% 330 ohm resistor (Orange Orange Brown Gold).

2) Potentiometer

There is one 10k trim potentiometer. This part will go in the spot labeled Contrast

3) Pushbuttons

There are a total of 5 x 6mm tactile switch pushbuttons. These will be used in the UP, DOWN, LEFT, RIGHT and SELECT locations on the PCB.

4) i2c Port Expander Chip

There is one of these MCP23017 i2c (16 input/output) port expander chips in the kit. This is how we are able to only use 2 R-Pi pins to run the entire LCD and buttons. [Click here for more info on this chip. \(http://adafru.it/732\)](http://adafru.it/732)

5) Male Header Pins

There is one strip of 36 male header pins in the kit. These will be used to attach the LCD to the PCB.

6) Printed Circuit Board

There will be one PCB in the kit as shown above.

7) Raspberry Pi Plate Header

There will be one extra-tall 26 pin female header for plugging into the Pi

8) Bumper

There will be one rubber bumper as shown above

Additional

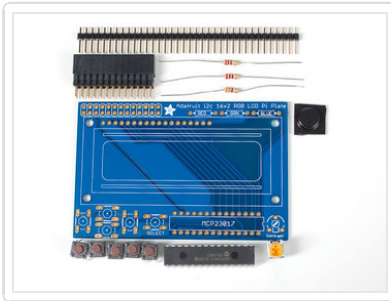
You'll want an LCD to place into the shield. This isn't included by default since a lot of people already have LCDs they may want to use. (We do sell these in packs however so chances are you did get an LCD with your order).



You can also use 16x2 LCDs or even OLEDs that are the same size that do not have an RGB backlight, or have no backlight at all.

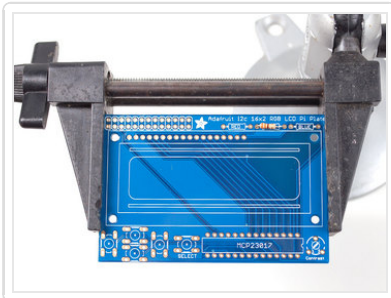
We carry [Negative type \(http://adafru.it/399\)](http://adafru.it/399) or [Positive type \(http://adafru.it/398\)](http://adafru.it/398) LCDs as well as [Blue and White LCDs \(http://adafru.it/181\)](http://adafru.it/181)

Assembly



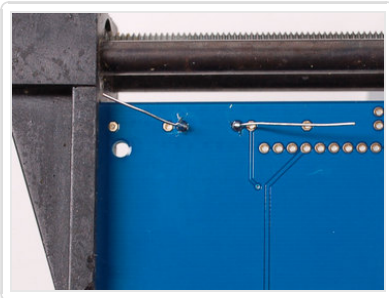
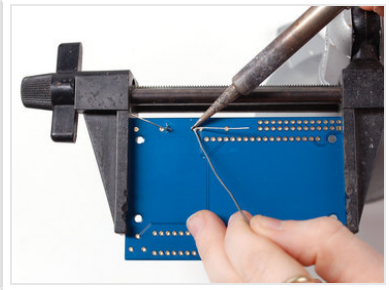
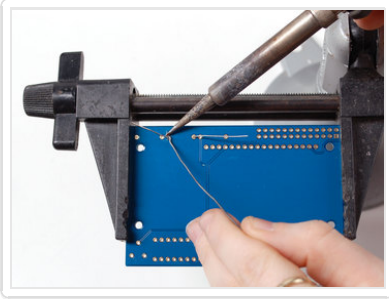
Check the kit against the to verify you have all the parts necessary

We recently adjusted the kit so the buttons are on the right side instead of the left. The parts list is otherwise the same, its just a little more stable than before



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 **GREEN** - which has orange, orange, brown, gold bands on it. This resistor acts as the backlight control resistor for the green backlight pin.

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

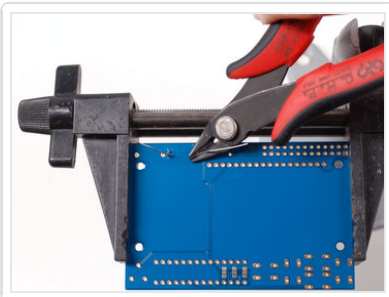


Bend the 'legs' of the resistor out so it sits flat against the PCB and flip it over.

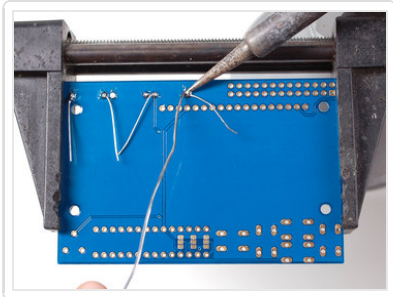
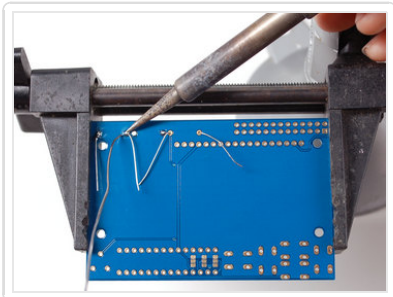
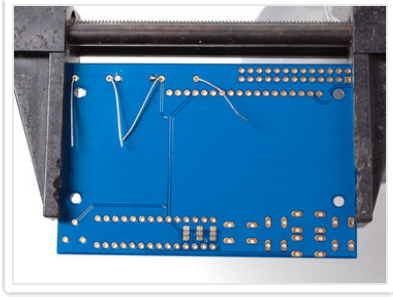
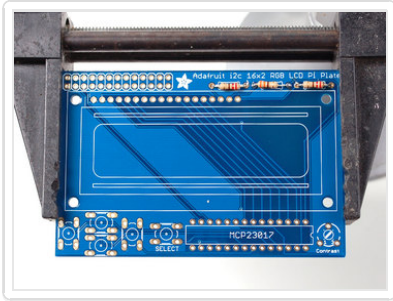
This way the resistor won't fall out while soldering.

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.

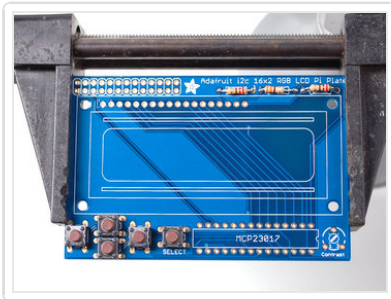
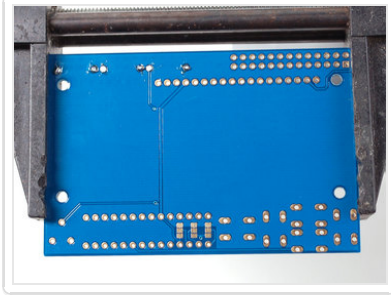
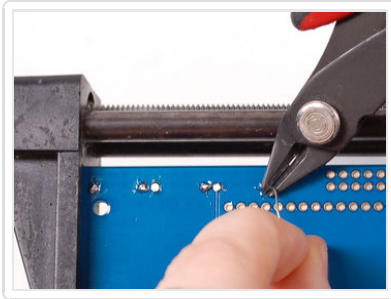


Since you did so great with the first resistor, we'll place all of the rest now at the same time.

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**.

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

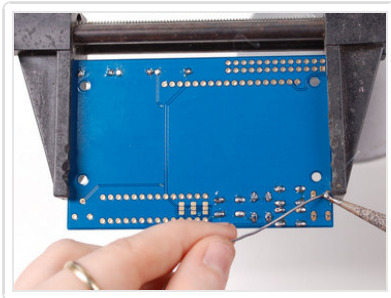
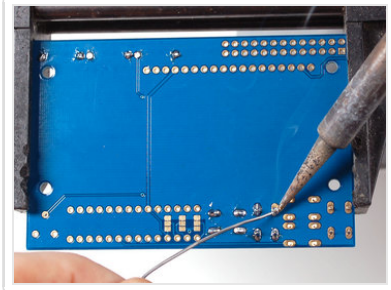
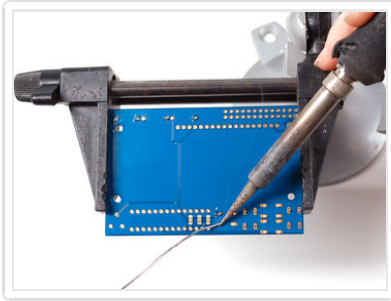
Clip all the leads.



Next up we will place the buttons. These buttons are useful to send a signal to the Pi (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 Pi, our library does the work of reading whether they are pressed.

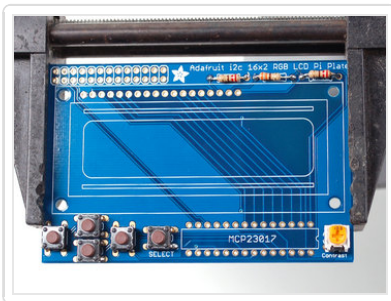
All the buttons are the same, and they should snap nicely into place. Press down onto each button until it snaps in and sits flat against the PCB.

We recently adjusted the kit so the buttons are on the RIGHT side instead of the left. The buttons snap in the same but they're on the right



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

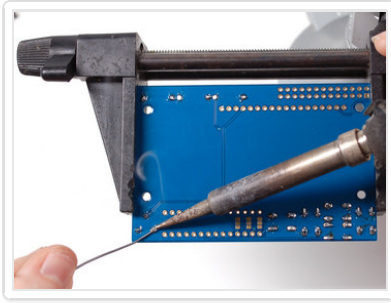
Solder each leg, taking care not to accidentally 'short' two button legs together. The ones for the directional pads are very close!



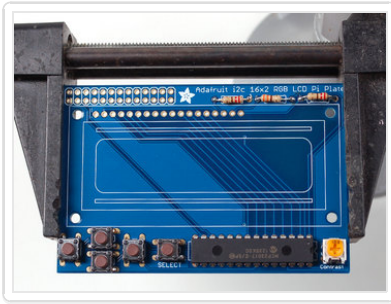
Next, place the 10K potentiometer (the orange-faced thing with three legs) into the spot above the RESET button. It will only fit one way. This is the contrast potentiometer which will adjust how dark the characters appear. All displays are slightly different so you'll adjust this once the display is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

We recently adjusted the kit to have the potentiometer in the center rather than the right, goes in the same way, just in the middle!



Flip over the PCB and solder in the three legs of the potentiometer

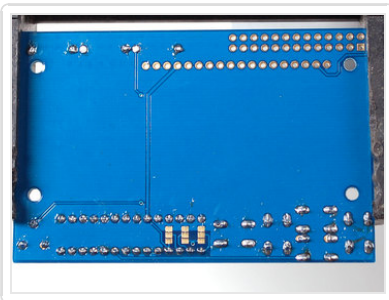
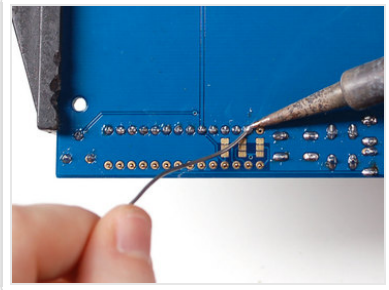
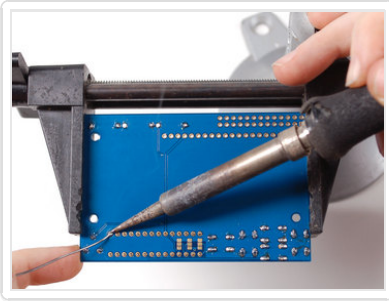


We're nearly done! Now we will place the I2C port expander chip. Double check that it has the **MCP23017-E/SP** marking on it. This is a 16-pin expander chip, that uses the i2c bus. That means you can send it commands using the i2c pins on an Pi and control 16 more digital pins! 5 of those pins go to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

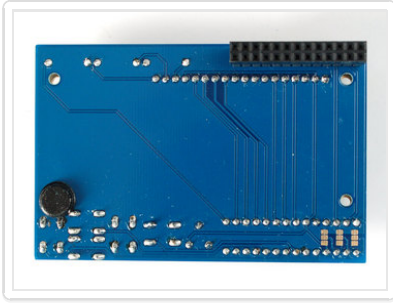
Unlike buttons or resistors, chips do have a direction and the must be put in the right way! First, use a flat table to carefully bend the legs of the chip so they are parallel. Then slip it into the silkscreened outline so that the notch at the end of the chip is on the *right*. Click the image to the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

We recently adjusted the kit so the chip is more to the left, its the same alignment, just shifted over

Flip over the PCB and solder in the 28 pins of the port expander.

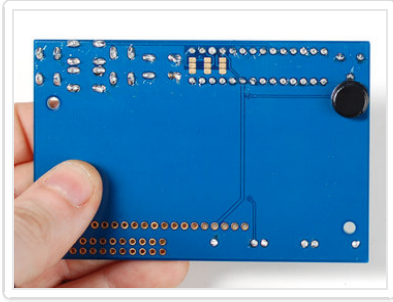


Next we will attach the header, there is a standard 'extra tall' header included in the kit. However, if you want to attach a cobbler or GertBoard, etc to the plate, you may want to opt for one of our stacking headers, they're extra long so you can plug in an IDC cable on top!

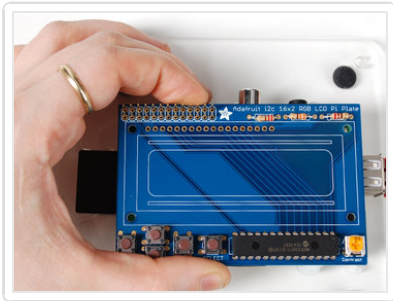


Before we solder in the header, remove the bumper from the backing and attach it on the underneath so its right over the Ethernet jack when the PCB is plugged in.

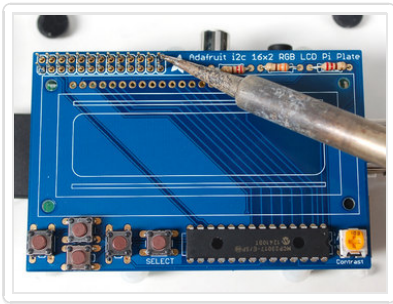
On kits where the buttons are on the left, put it right below the contrast pot, as shown. This will keep the pot from touching the Ethernet jack.



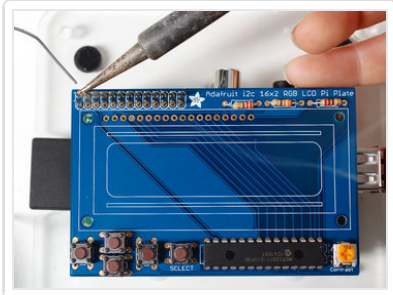
On kits where the buttons are on the right, put it right next to the buttons (it shouldnt overlap any button legs) sitting flat



Place the 2x13 header onto your Pi, and slide the plate on top, it should fit perfectly.

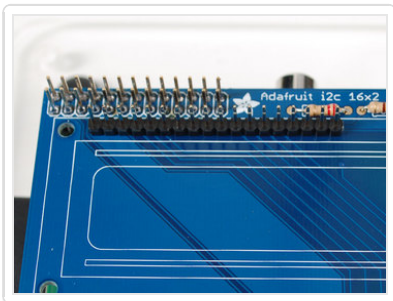
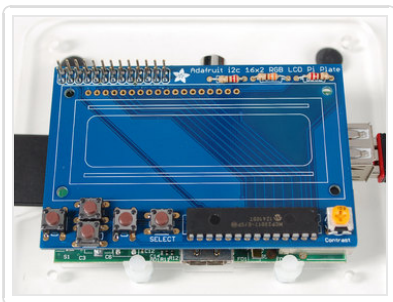


Now we will solder all 26 pins for the plate - this will send power and data between the two boards and also provide a mechanical stability. You may need to hold the plate down with tape to get it sitting flat against the Pi.



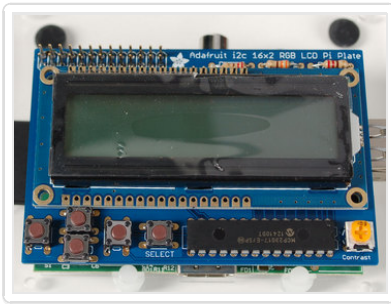
Solder in one pin, we suggest the top right, then check if the plate is sitting flat. If not, heat up the joint with one hand while stabilizing with the other until its right.

Then solder the other 25 pins



Finally, we place the LCD. Break off a piece of 18 or 16 pin header from the stick in the kit. If you have an RGB display, you'll need 18, for a blue&white you'll need 16

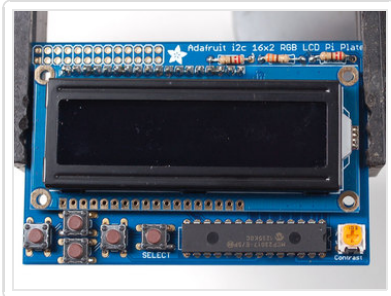
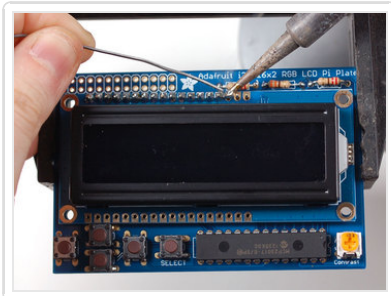
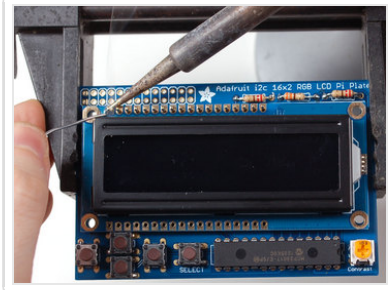
Place the header into the remaining slot with the long ends of the pins sticking UP. If you have only 16 pins, leave the two rightmost pads empty



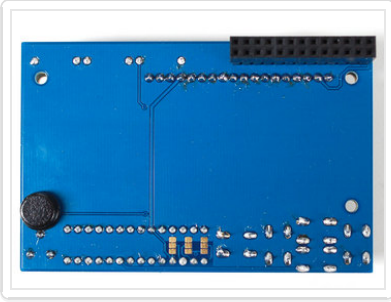
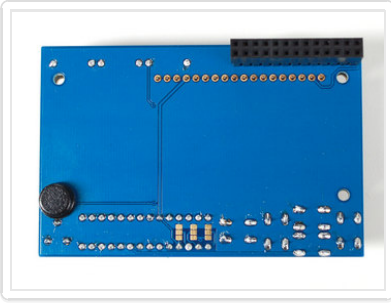
Slide the LCD over the header so that it is perfectly centered over the four holes and the silkscreen.

The LCD should 'snap' in against the buttons

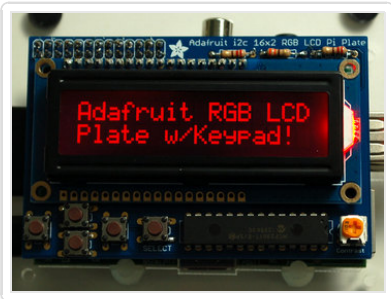
Solder all the pins!



Flip over the assembly, carefully. Then solder in the 16 or 18 pins of header



That's it! You're done soldering, now you just have to run the code in the Usage section next.



Usage

The backlights will not turn on until you have the code running so don't be alarmed if nothing lights up when you plug in the plate! If you have a 'positive' style LCD, you may see some squares on the first line, that's normal

Setting up your Pi for I2C

For a more basic introduction to setting up I2C on your Pi then you may wish to take a diversion to this Adafruit tutorial: <http://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c> (<http://adafru.it/aTI>)

If you're using Occidentalis, you can skip this next step, but for Raspbian users, edit `/etc/modules` (**sudo nano /etc/modules**) and add:

```
i2c-bcm2708  
i2c-dev
```

to the end of the file. Then save and reboot to enable the hardware I2C driver.

Before you can get started with I2C on the Pi, you'll need to run through a couple quick steps from the console. Just enter the following commands to add SMBus support (which includes I2C) to Python:

```
sudo apt-get install python-smbus  
sudo apt-get install i2c-tools
```

`i2c-tools` isn't strictly required, but it's a useful package since you can use it to scan for any I2C or SMBus devices connected to your board. If you know something is connected, but you don't know it's 7-bit I2C address, this library has a great little tool to help you find it:

```
sudo i2cdetect -y 0 (if you are using a version 1 Raspberry Pi)  
sudo i2cdetect -y 1 (if you are using a version 2 Raspberry Pi)
```

```
raspberrypi
About these Buttons  Send "exit<enter>"
pi@raspberrypi ~ $ sudo i2cdetect -y 0
 00:  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20: 20  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi ~ $ _
```

This will search /dev/i2c-0 or /dev/i2c-1 for all address, and if an Adafruit LCD Plate is connected, it should show up at **0x20**

Once both of these packages have been installed, you have everything you need to get started accessing I2C and SMBus devices in Python.

Using the example Python code

The LCD Pi Plate Python code for Pi is available on Github at <https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code> (<http://adafru.it/aOg>)

The easiest way to get the code onto your Pi is to hook up an Ethernet cable, and clone it directly using 'git', which is installed by default on most distros. Simply run the following commands from an appropriate location (ex. "/home/pi"):

```
sudo apt-get install git
git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git
cd Adafruit-Raspberry-Pi-Python-Code
cd Adafruit_CharLCDPlate
```

Testing the Library

Once the code has been downloaded to an appropriate folder, and you have your LCD Plate properly connected, you can start with the basic demo display, which is run by simply executing the library file:

```
sudo python Adafruit_CharLCDPlate.py
```

If you have a rev 2 (512MB) Pi, or if you're not getting anything displaying, it might be due to the I2C bus number change in the Pi hardware. Edit `Adafruit_CharLCD.py` using a command like "nano `Adafruit_CharLCD.py`" and change the line

```
lcd = Adafruit_CharLCDPlate(busnum = 0)
```

to

```
lcd = Adafruit_CharLCDPlate(busnum = 1)
```

Adjusting Contrast

The plate uses a character LCD with an external contrast potentiometer. The first time you use it, adjust the potentiometer in the bottom right until you see the text clearly. If you don't upload code to the Pi, some boxes may appear instead, or you may see nothing at all.

Using the library code

Interfacing with the python example code is fairly easy! Inside the `Adafruit_CharLCDPlate` folder you'll find a **`testLCD.py`** python script. The script does a few things, at the top it imports all the sub-modules. You'll need to have the **`Adafruit_I2C.py`**, **`Adafruit_MCP230xx.py`** and **`Adafruit_CharLCDPlate.py`** python files to be in the same directory so be sure to copy them to your final destination.

Next it initializes the plate with **`lcd = Adafruit_CharLCDPlate()`** - this creates the `lcd` object and starts communicating with the plate to set up the LCD and buttons

After initialization, you can clear the display with **`lcd.clear()`** and write text with **`lcd.message("text goes here")`** don't forget that its only 16 characters per line, and it does not automatically wrap lines. To insert a newline use the special `\n` character such as shown in the command: **`lcd.message("Adafruit RGB LCD\nPlate w/Keypad!")`**

Next you can set the backlight with **`lcd.backlight(lcd.COLORNAME)`** where **`COLORNAME`** is RED, YELLOW, GREEN, TEAL, BLUE, VIOLET for RGB LCD's or for monochrome LCDs, just use ON and OFF

Finally, you can ask the plate which buttons are pressed with **`buttonPressed(lcd.BUTTONNAME)`** where **`BUTTONNAME`** is **LEFT RIGHT UP DOWN** or **SELECT**. This is not an interrupt-driven library so you can't have an interrupt go off when a button is pressed, instead you'll have to query the button in a loop.

That's it! If you want to make detailed messages, [use the python string formatting commands to create a string \(http://adafru.it/aUi\)](http://adafru.it/aUi) and then write that string with **`message()`**

Download

