



i2c/SPI LCD Backpack

Created by lady ada

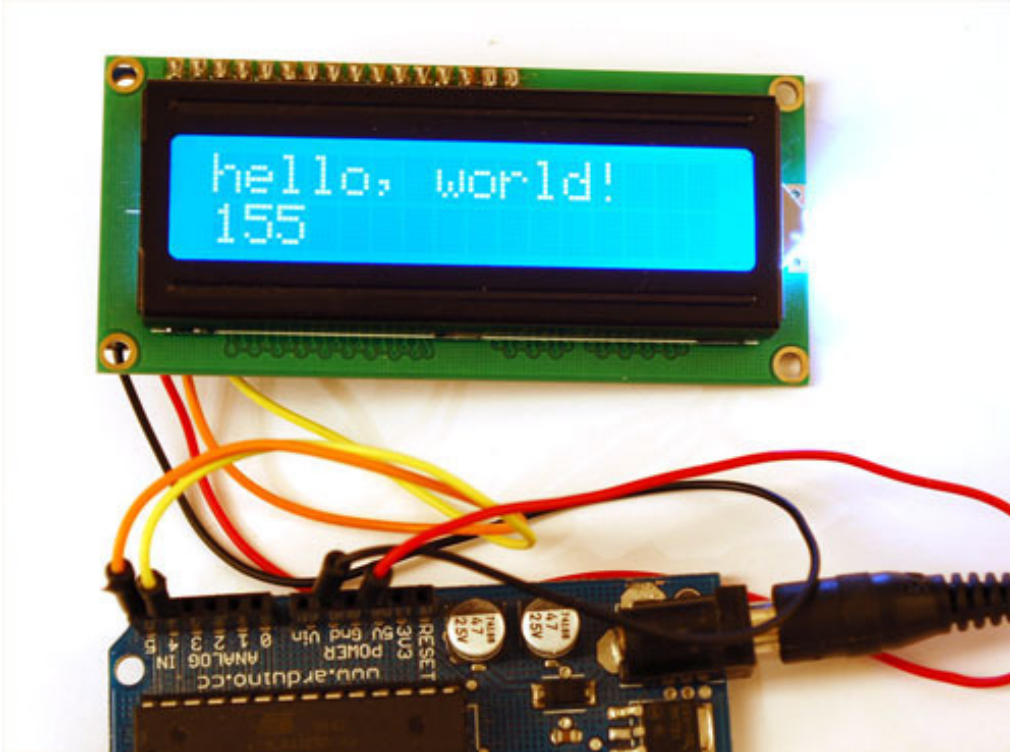


Last updated on 2016-01-21 05:50:09 PM EST

Guide Contents

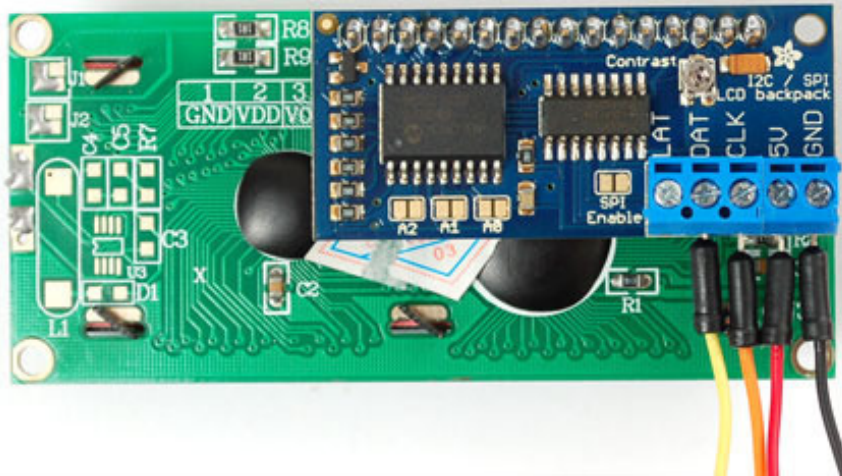
Guide Contents	2
Overview	3
Which LCD to Use?	5
Wait - the backpack has 16 holes, but my LCD only has 14 pins!	5
Assembly	6
Parts Check	6
Terminal Blocks	6
Prepare LCD	7
Attach LCD	9
My LCD only has 14 pins. Which holes do I use?	11
Connect to i2c	12
Install Adafruit_LiquidCrystal	12
Load Demo	13
Changing the I2C Address	14
Connect to SPI	16
Install Adafruit_LiquidCrystal	17
Load Demo	17
Downloads	20
Optimized i2c library	20
Hardware files	20

Overview



LCDs are a fun and easy way to have your microcontroller project talk back to you. Character LCDs are common, and easy to get, available in tons of colors and sizes. [We've written tutorials on using character LCDs with an Arduino \(http://adafruit.it/c8r\)](http://adafruit.it/c8r) (or similar microcontroller) but find that the number of pins necessary to control the LCD can be restrictive, especially with ambitious projects. We wanted to make a 'backpack' (add-on circuit) that would reduce the number of pins without a lot of expense.

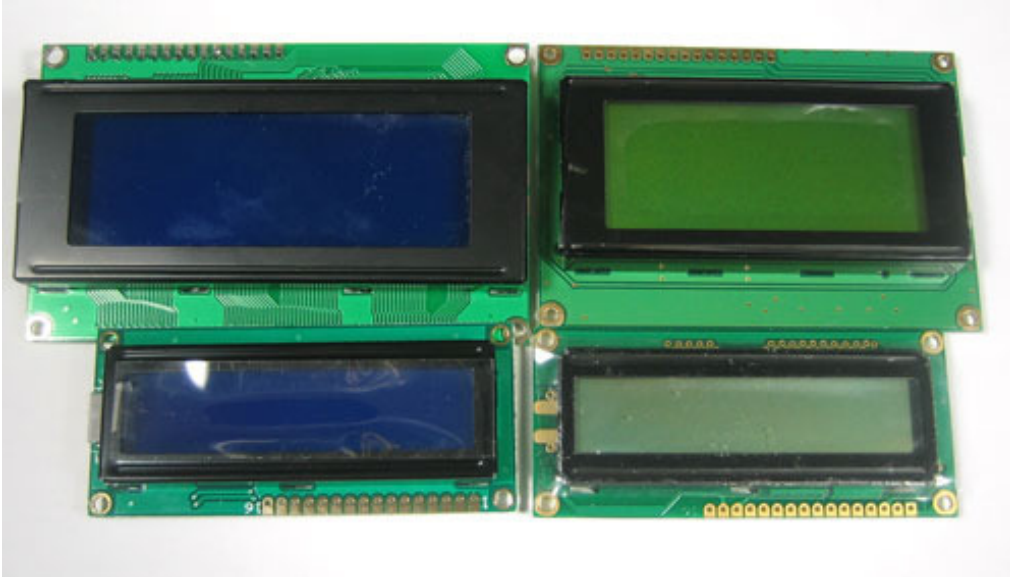
By using simple i2c and SPI input/output expanders we have reduced the number of pins (only 2 pins are needed for i2c) while still making it easy to interface with the LCD. For Arduino users, we provide a easy-to-use library that is backwards compatible with projects using the '6 pin' wiring.



For advanced users, this project can be used for general purpose I/O expansion, the MCP23008 has 8 i/o pins (7 are connected) with optional pullups, the SPI 74HC595 has 7 outputs.

Assembled and tested backpacks are available for purchase for only \$10 in the Adafruit shop!

Which LCD to Use?



This backpack will work with any 'standard'/'classic' **character** LCD. It does not work with **graphic** LCDs. Character LCDs come in sizes ranging from 8x1 (8 characters, one line) to 40x4 (40 characters, four lines). The backpack will also only fit LCDs that have a single line of pins at the top, not the ones that have a 2x10 or 2x8 connector on the side. Those are much rarer these days but just keep a look out for that!

The backpack will work with RGB LCDs (but won't control the RGB backlight, you can do that separately from the LCD control) and it won't work with 40x4 LCDs because they have a second Enable pin.

Wait - the backpack has 16 holes, but my LCD only has 14 pins!

As long as your LCD has a HD44780 driver chip, you're OK. The extra two pins are for the backlight. Your LCD doesn't have a backlight.

We'll show you which holes to ignore in the Assembly section of this guide.

Assembly

Putting together the backpack onto an LCD is a quick process, and should take only a few minutes with a soldering iron.

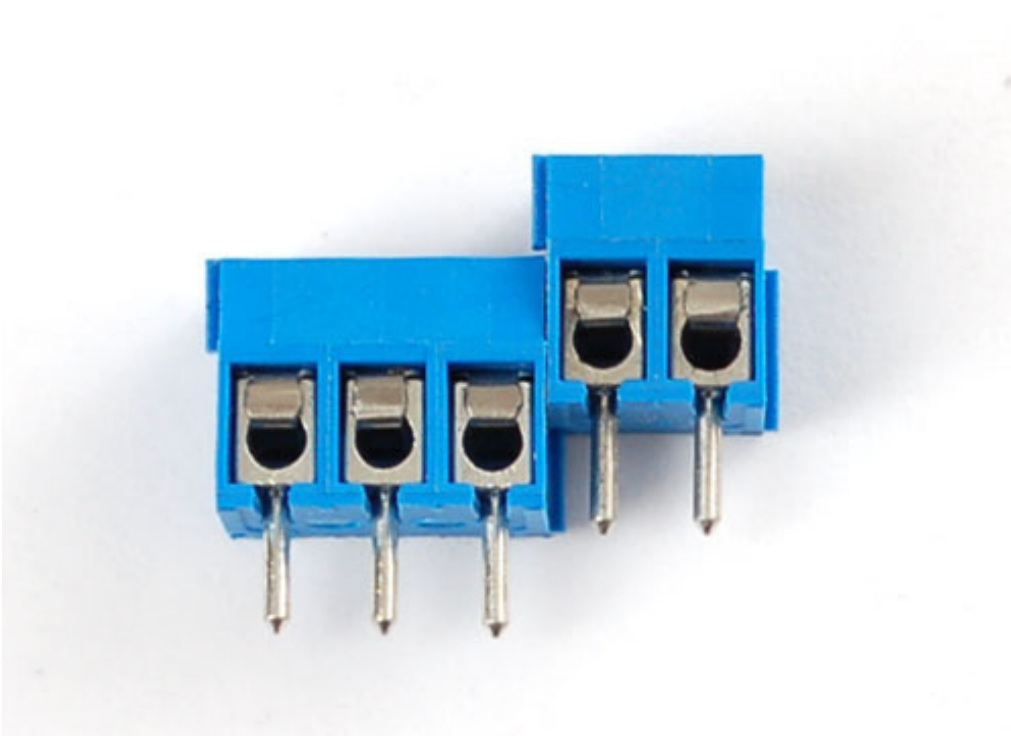
Parts Check

Verify you have everything in the bag, there should be an assembled and tested PCB, a 2-pin and 3-pin 3.5mm terminal block. The backpack does not come with header or an LCD.

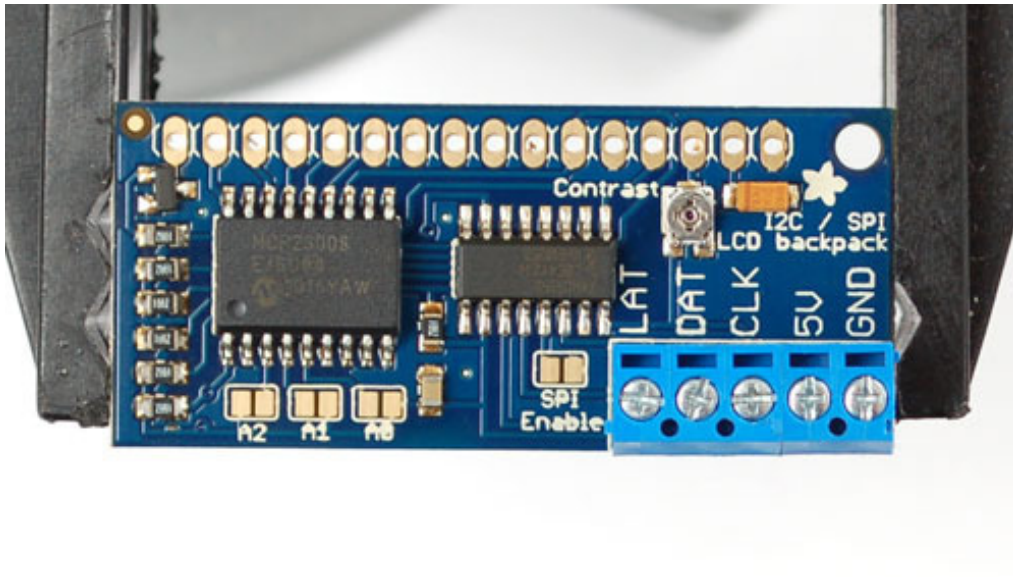
Terminal Blocks

The [terminal blocks](http://adafru.it/cuW) (<http://adafru.it/cuW>) allow you to easily attach and remove the LCD from your wiring, which we think is awfully handy. If you don't want the terminal blocks (they stick out a bit) you can always skip this step.

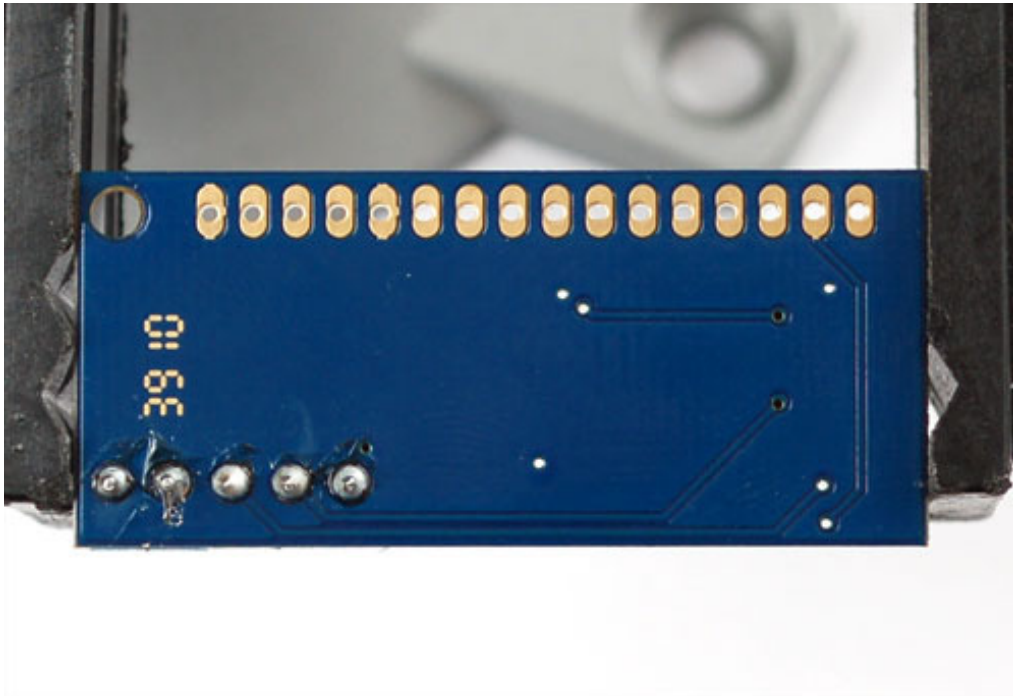
The terminal blocks come in 2 and 3-pin pieces (<http://adafru.it/cuW>) . Slide them together.



Place the blocks over the corner area of the backpack so that the holes stick out (unless for some reason you want them to face the other way)



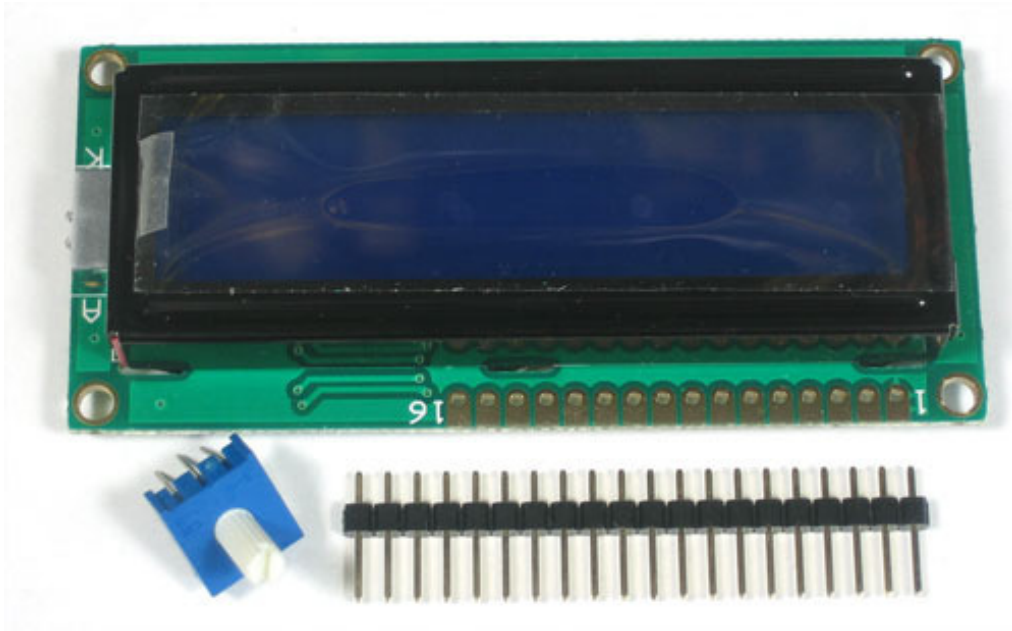
Solder them into place



Prepare LCD

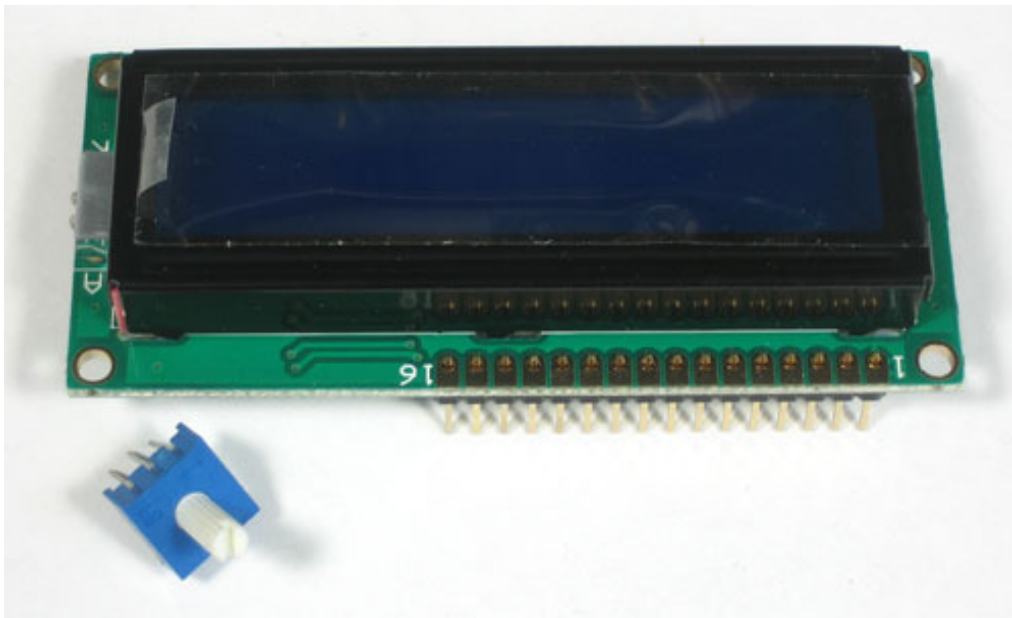
Next we will attach the backpack to the LCD. First we must put header onto the LCD, if you bought the LCD from us, it will come with a stick of header. Otherwise, pick up some [standard 0.1" male header](http://adafruit.it/cuX) (<http://adafruit.it/cuX>)

These photos shows a 10K potentiometer, you can ignore it

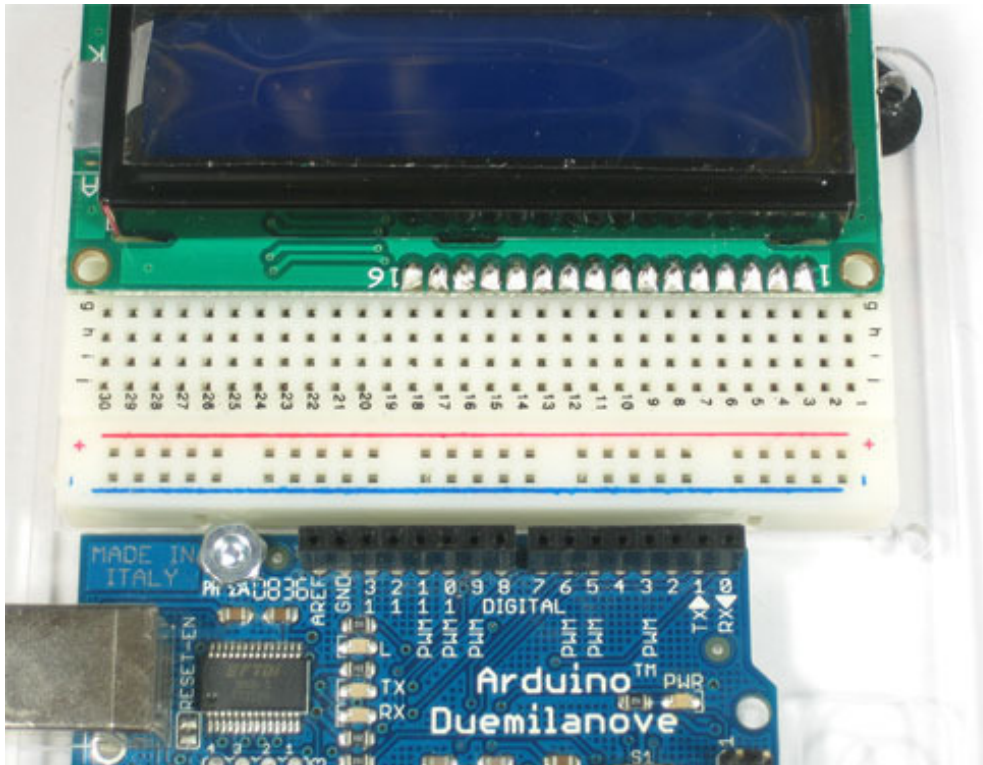


If the header is too long, just cut/snap it short so that it is 16 pins

Next you'll need to solder the header to the LCD. You must do this, it is not OK to just try to 'press fit' the LCD!



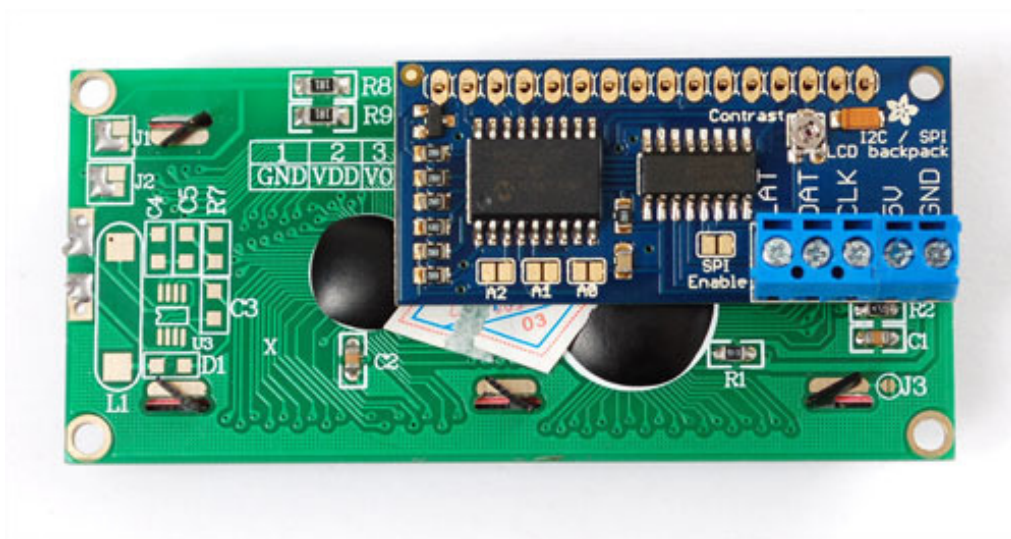
The easiest way we know of doing this is sticking the header into a breadboard and then sitting the LCD on top while soldering. this keeps it steady.



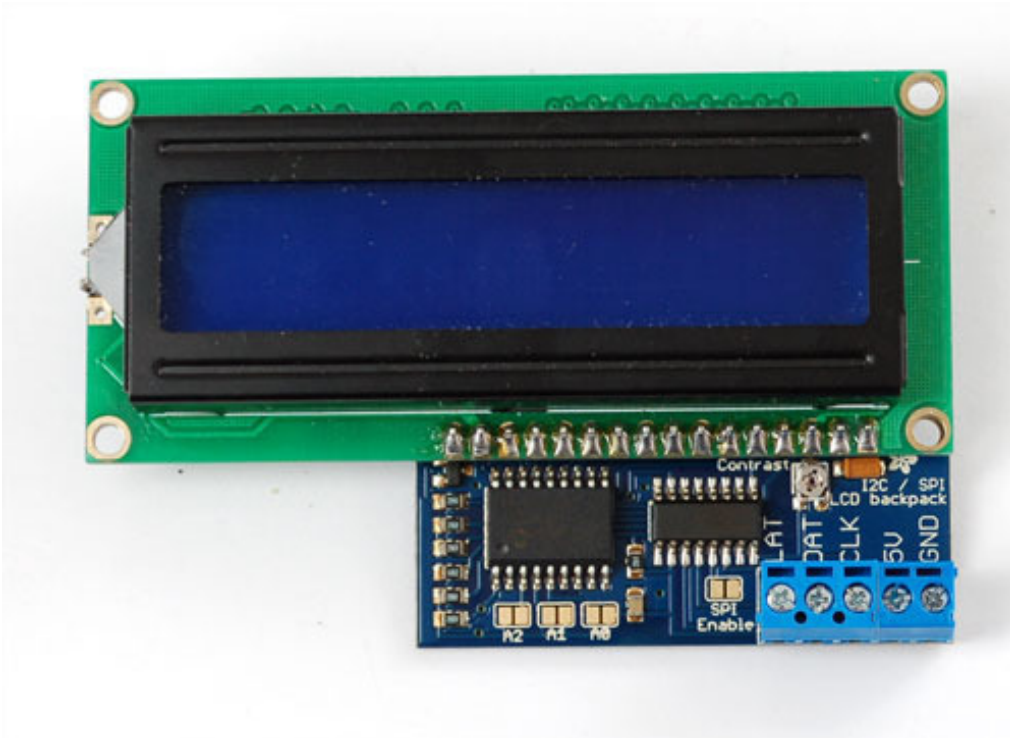
Attach LCD

Now we will attach the backpack. We will show how to do this in a permanent fashion. If you think you would like to remove and replace the LCD at some time, you can use [a piece of 16-pin long 0.1" female header](http://adafruit.it/cuX) (<http://adafruit.it/cuX>) as a socket but be aware it will stick out a lot.

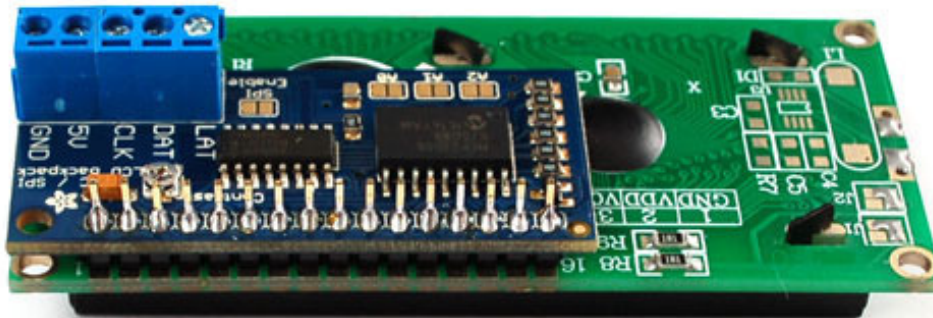
There are two options, you can tuck the backpack behind the LCD



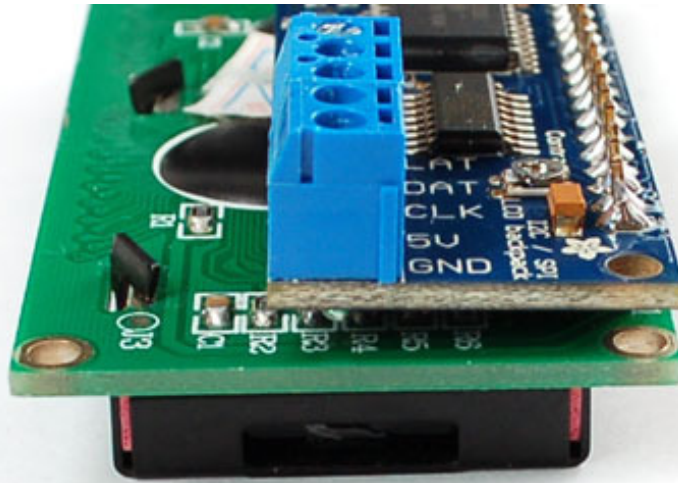
Or solder it so it's to the side, this way it's thinner.



Solder the header to the backpack.



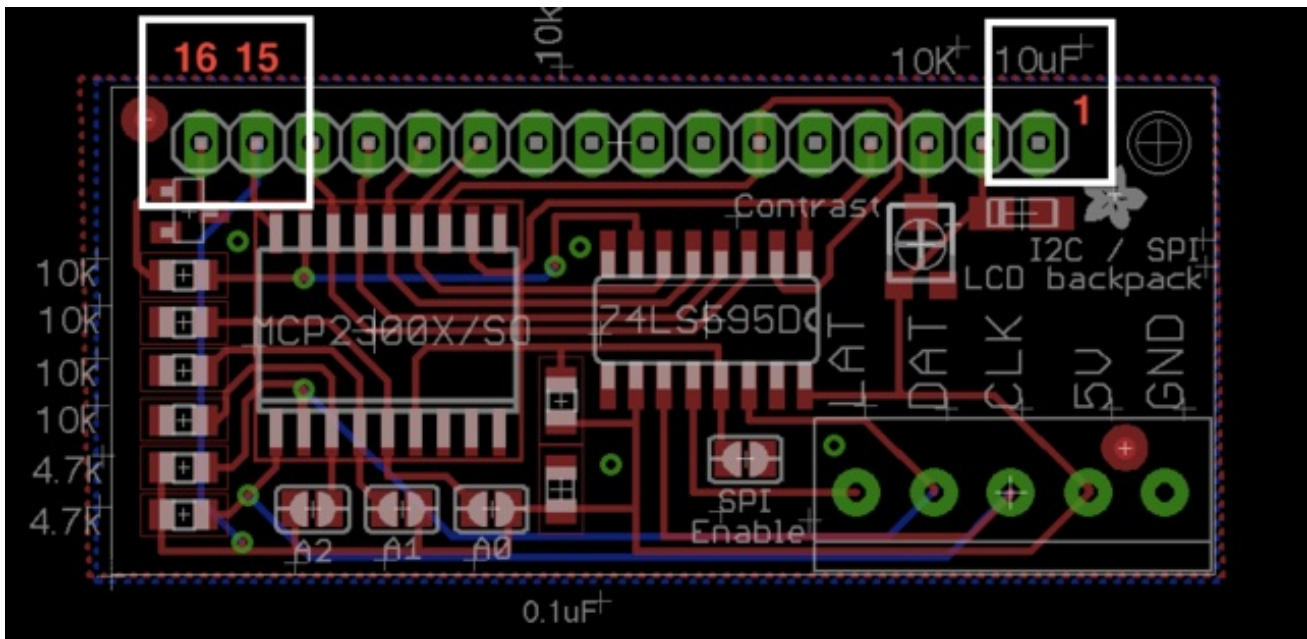
Make sure that as you solder the first pin, the backpack PCB isn't leaning against the LCD, where the terminal blocks could short against some components. You can put some electrical tape or foam tape behind to avoid this if you think it could be an issue.



That's it!

My LCD only has 14 pins. Which holes do I use?

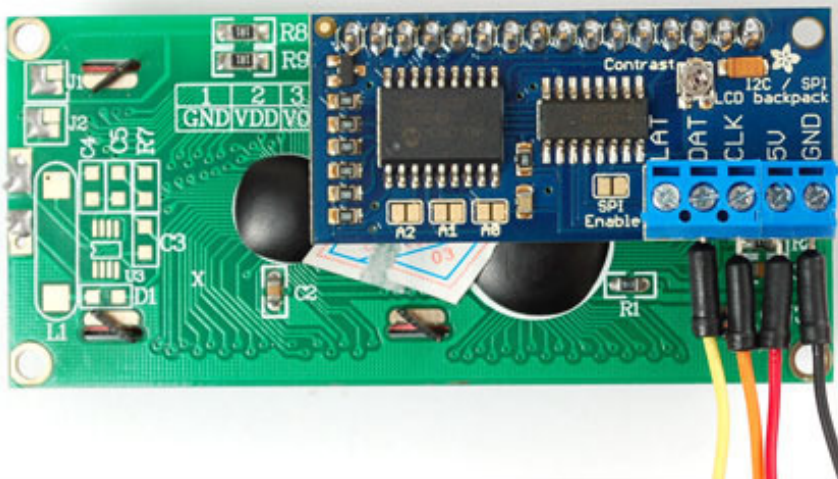
The extra two pins are for the backlight. Your LCD doesn't have a backlight. Here's a picture of the backpack board layout, with pins 15 and 16 located. Just leave those empty. Cut the pin header down to 14 pins instead of 16, and attach your LCD to holes 1-14 in the backpack.



Connect to i2c

The first option we'll show is how to use the i2c interface on the backpack. We'll be showing how to connect with an Arduino, for other microcontrollers please see our MCP23008 library code for the commands to send to the i2c i/o expander. I2c is nice because it only uses two pins, and you can put multiple i2c devices on the same two pins. So for example, you could have up to 8 LCD backpacks+LCDs all on two pins! The bad news is that you have to use the 'hardware' i2c pin. You can't change those pins and you can't use them for reading analog data. If you absolutely need those two pins, use SPI (see next section)

For this we'll need to connect four wires: GND, 5V, CLK (clock) and DAT (data).



Connect:

- **DAT (SDA)** to I2C Data (on the Uno, this is **A4** on the Mega it is **20** and on the Leonardo digital **2**)
- **CLK (SCL)** to I2C Clock (on the Uno, this is **A5** on the Mega it is **21** and on the Leonardo digital **3**)

Install Adafruit_LiquidCrystal

To begin reading sensor data, you will need to [download Adafruit_LiquidCrystal from our github repository \(http://adafru.it/kZe\)](http://adafru.it/kZe). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit_LiquidCrystal
Library

<http://adafru.it/kZf>

Rename the uncompressed folder **Adafruit_LiquidCrystal** and check that the **Adafruit_LiquidCrystal** folder contains **Adafruit_LiquidCrystal.cpp** and **Adafruit_LiquidCrystal.h**

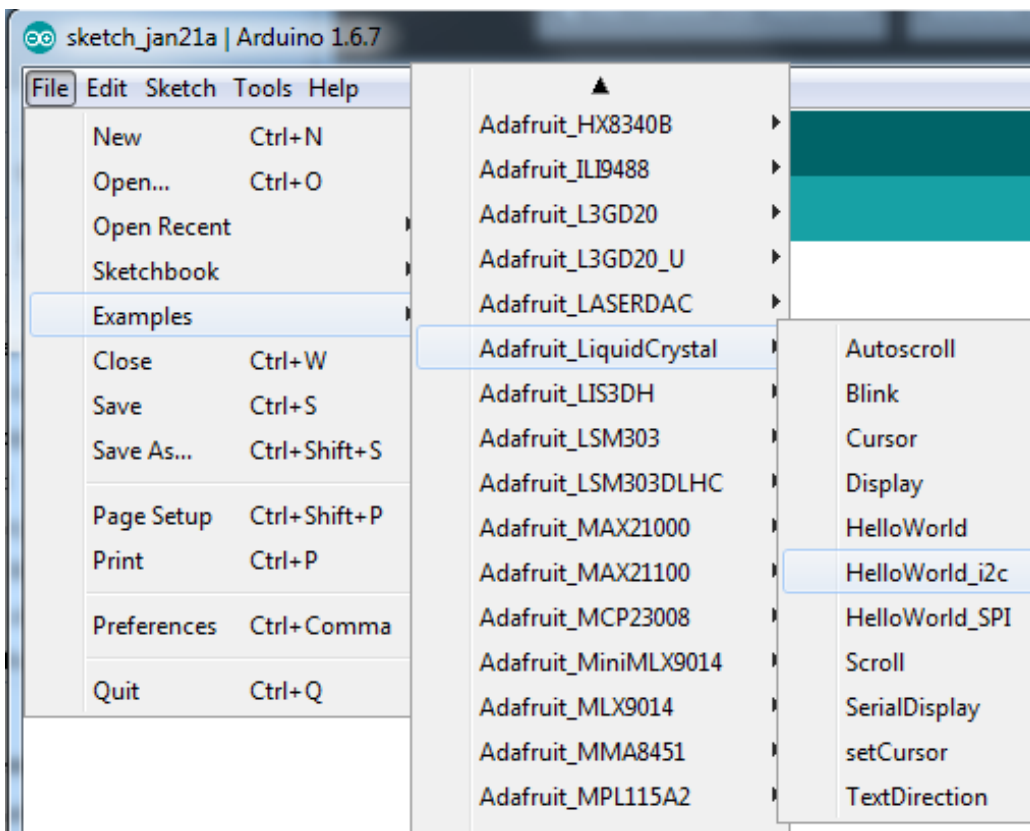
Place the **Adafruit_LiquidCrystal** library folder your **arduinofolder/libraries/** folder. You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:

<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<http://adafru.it/aYM>)

Load Demo

Restart the IDE and load up the **Adafruit_LiquidCrystal->HelloWorld_i2c** demo



Upload the sketch. You should see the backlight turn on when the Arduino resets. If you don't see any characters, adjust the **Contrast** trim potentiometer with a mini-screwdriver until you see the text clearly

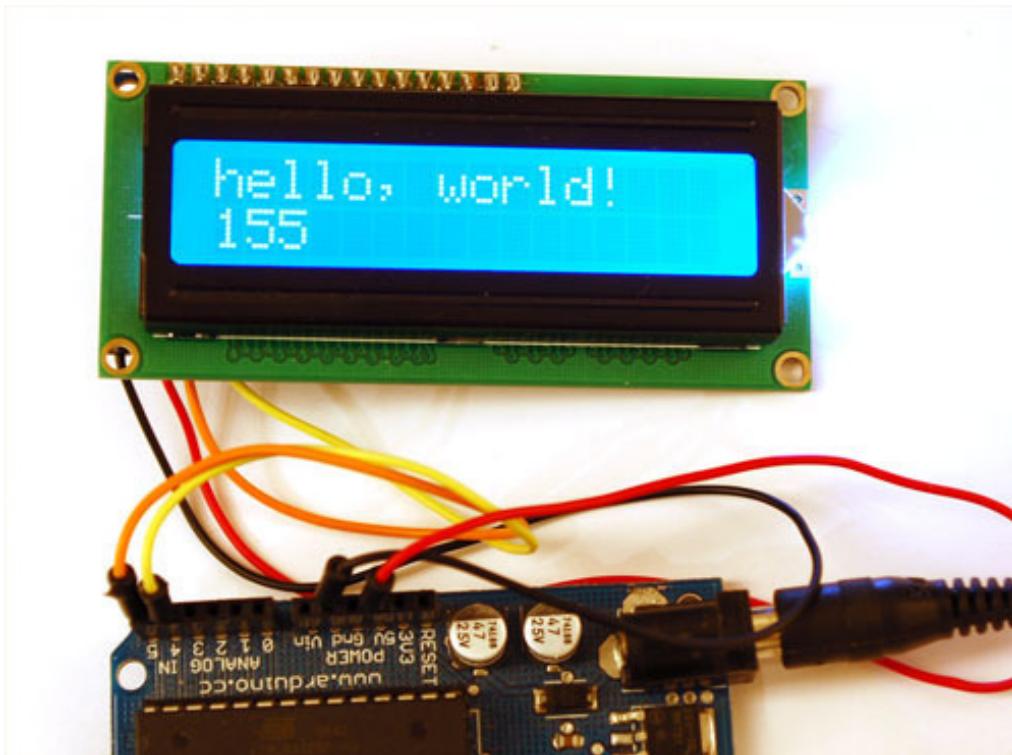
The default HelloWorld sketch blinks the backlight as well as updating the text.

If you see the backlight blinking that means your connection to the i2c port is OK but the contrast is too low or too high, or the LCD data pins are not solidly connected.

Check the contrast first by gently twisting the mini trim potentiometer, if that doesn't help, recheck your soldering and resolder all 16 of the LCD pins!

Once you're done, you can remove the blinking LED backlight code:

```
lcd.setBacklight(HIGH);  
delay(500);  
lcd.setBacklight(LOW);  
delay(500);
```



Changing the I2C Address

If you want to have more than one MCP23008 device (like more than one backpack+LCD) each one needs to have a unique 'address'. You can set the address by jumpering the **A0** **A1** and **A2** solder jumpers. By default, no jumpers are soldered, giving an address of **0**. If you want to have an address of **3** you would solder **A0** (bit 0) and **A1** (bit 1) for an address of "011" = 3 in binary. Then in the code change:

```
// Connect via i2c, default address #0 (A0-A2 not jumpered)  
LiquidCrystal lcd(0);
```


to

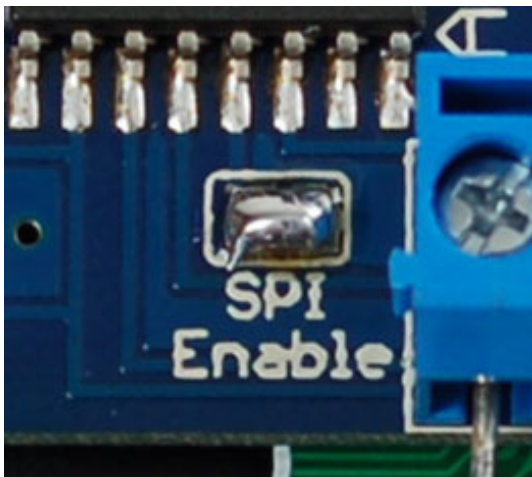
```
// Connect via i2c, address #3 (A0&A1 jumpered)  
LiquidCrystal lcd(3);
```

Connect to SPI

Another option for connecting is to use **SPI**, which is a simpler protocol. The good news about SPI is that its very simple and you can use **any** 3 pins to connect. You can share the **data** and **clock** pins with another device as long as they remain outputs, the **latch** pin should only be used for the backpack. So if you wanted 3 LCDs, for example, they would all have the same **data** and **clock** pins, but the **latch** pin would be different, for 5 pins total

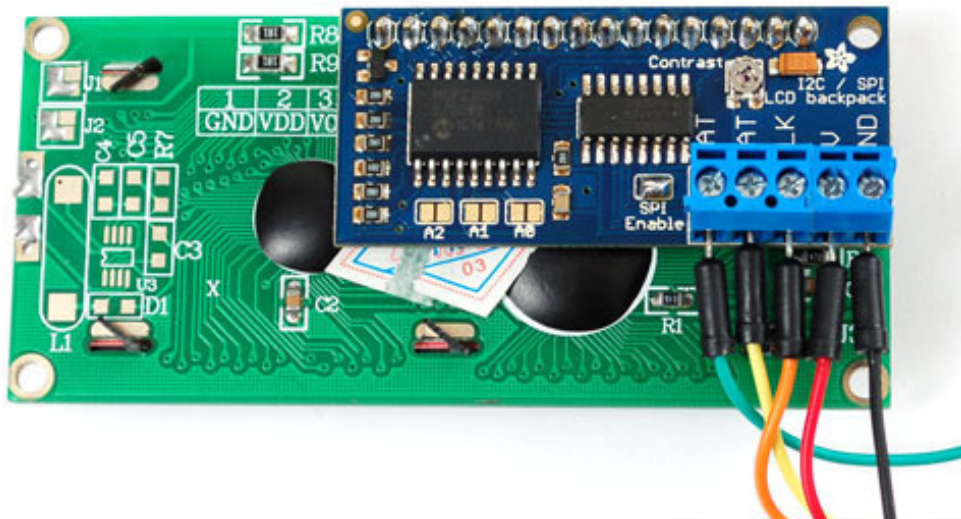
This library does not use the Hardware SPI library, which means you can use any 3 pins! However, if you are using the hardware SPI port (such as for Ethernet, WiFi, an LCD, etc. etc) you cannot share those pins with this LCD!

The first thing you will need to do is to enable SPI. To do this, solder the **SPI Enable** solder jumper by heating up the pads with a soldering iron and soldering a blob onto both pins:



This will switch the backpack over to SPI mode instead of i2c. If you want to go back to i2c, use wick or a solder sucker to remove the jumper

Next we will connect 5 wires, **5V**, **GND**, **DAT**, **CLK**, and **LAT**. To match the example, **CLK** goes to to Digital 2, **DAT** to Digital 3, and **LAT** to Digital 4. Once we have the example sketch running you can of course change these to anything you'd like. Connect **5V** and **GND** to the 5v and Ground Arduino power pins



Install Adafruit_LiquidCrystal

To begin reading sensor data, you will need to [download Adafruit_LiquidCrystal from our github repository \(http://adafru.it/kZe\)](http://adafru.it/kZe). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit_LiquidCrystal
library

<http://adafru.it/kZf>

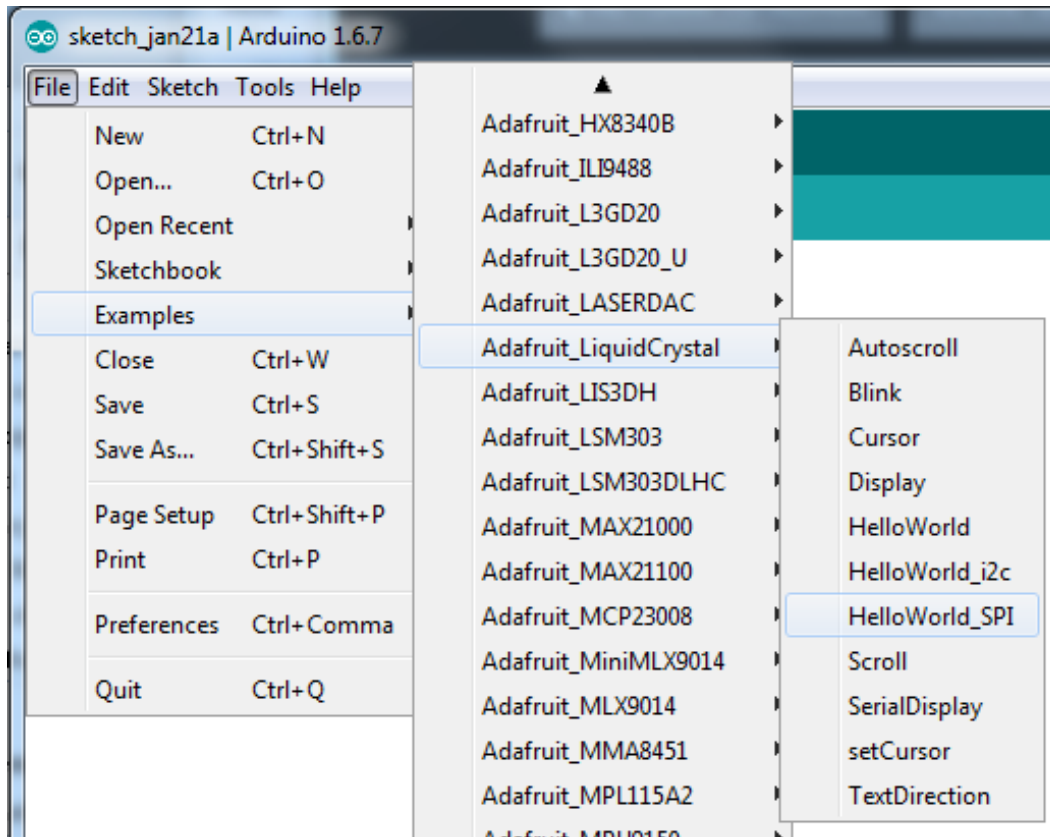
Rename the uncompressed folder **Adafruit_LiquidCrystal** and check that the **Adafruit_LiquidCrystal** folder contains **Adafruit_LiquidCrystal.cpp** and **Adafruit_LiquidCrystal.h**

Place the **Adafruit_LiquidCrystal** library folder your **arduinofolder/libraries/** folder. You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<http://adafru.it/aYM>)

Load Demo

Restart the IDE and load up the **Adafruit_LiquidCrystal->HelloWorld_SPI** demo



Upload the sketch. You should see the backlight turn on when the Arduino resets. If you don't see any characters, adjust the **Contrast** trim potentiometer with a mini-screwdriver until you see the text clearly

The default HelloWorld sketch blinks the backlight as well as updating the text.

If you see the backlight blinking that means your connection to the SPI port is OK but the contrast is too low or too high, or the LCD data pins are not solidly connected.

Check the contrast first by gently twisting the mini trim potentiometer, if that doesn't help, recheck your soldering and resolder all 16 of the LCD pins!



Once you're done, you can remove the blinking LED backlight code:

```
lcd.setBacklight(HIGH);  
delay(500);  
lcd.setBacklight(LOW);  
delay(500);
```

Downloads

Optimized i2c library

FalconFour posted up a [ultra-optimized high-speed i2c library](http://adafru.it/aP1), check it out for better performance! (<http://adafru.it/aP1>)

Hardware files

Eagle CAD format schematic and board layout file (as well as a PNG of the schematic) are all available in the [GitHub repository](http://adafru.it/aP2) (<http://adafru.it/aP2>)