Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

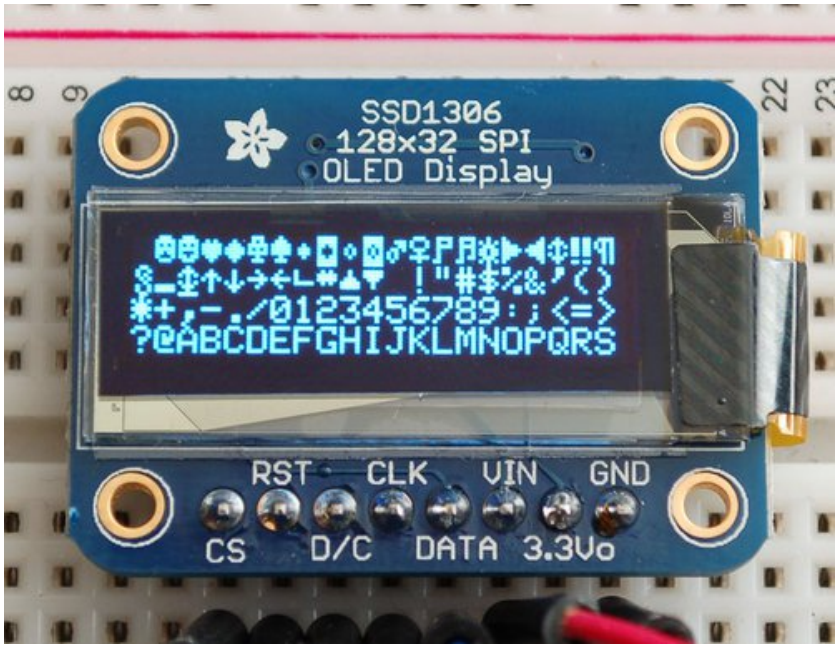
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Monochrome OLED Breakouts

Created by lady ada

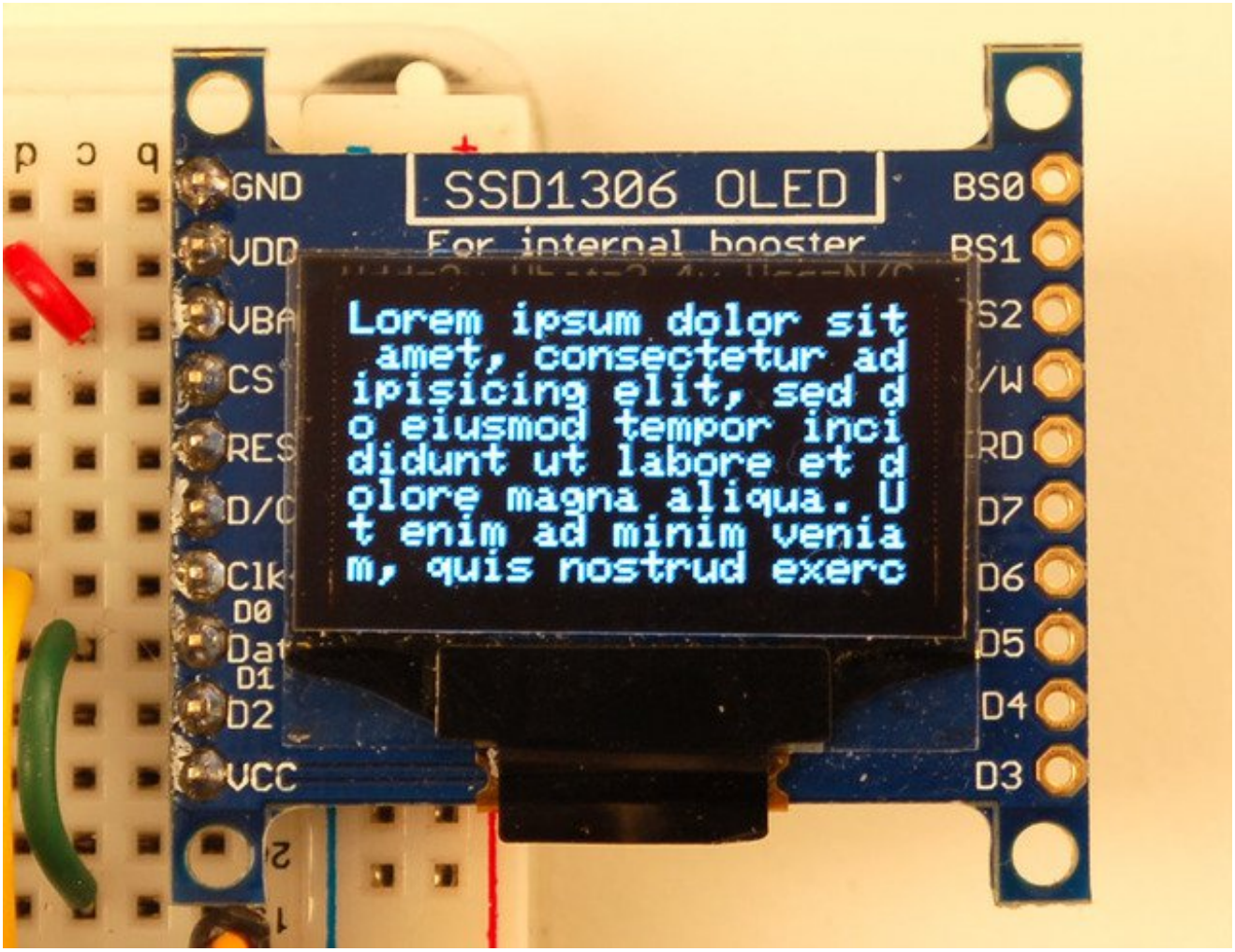


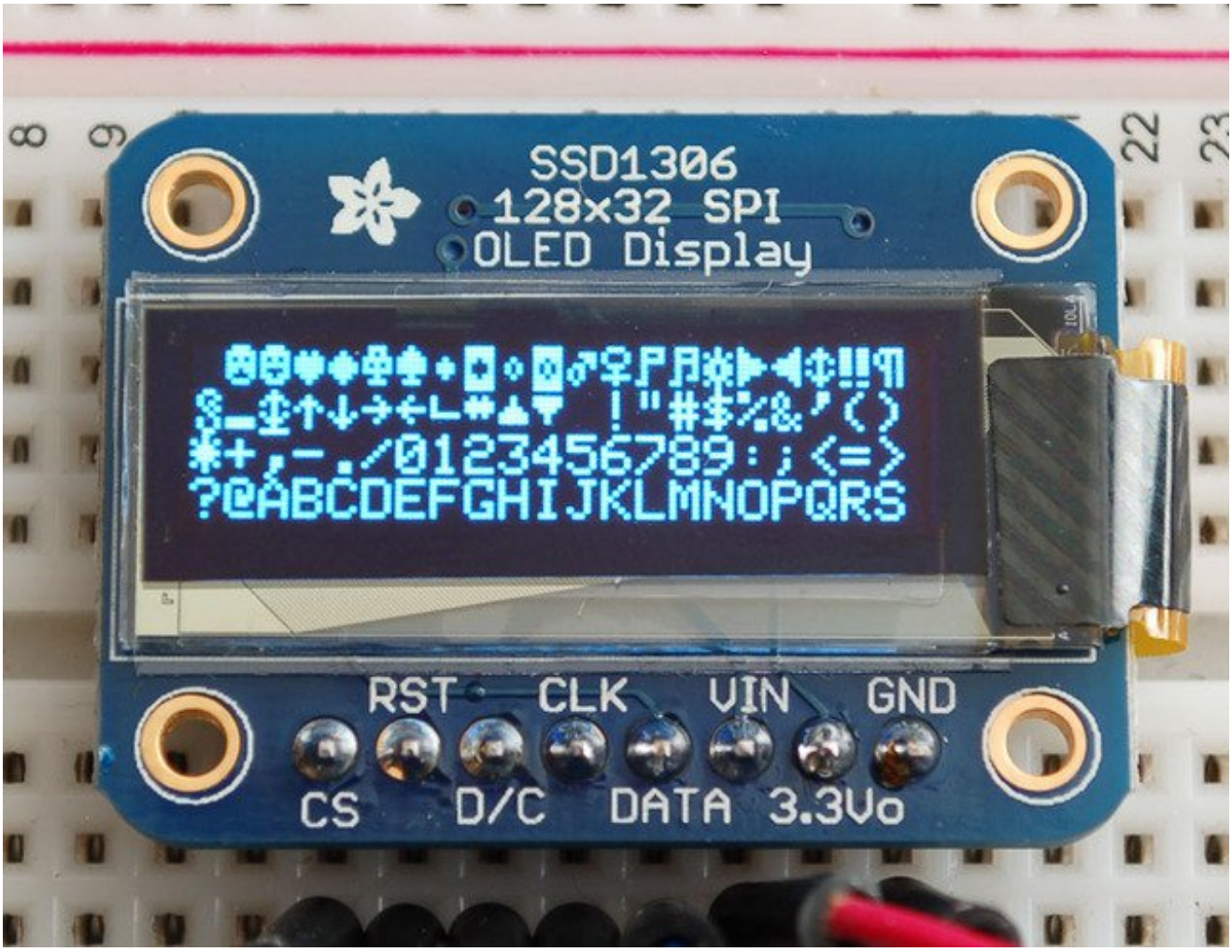
Last updated on 2016-10-13 07:46:03 PM UTC

Guide Contents

Guide Contents 2
Overview 3
Power Requirements 6
OLED Power Requirements 6
5V- ready 128x64 and 128x32 OLEDs 6
0.96" 128x64 OLED 7
Arduino Library & Examples 9
Install Adafruit SSD1306 Library 9
Install Adafruit GFX 9
Run Demo! 10
Create Bitmaps 12
Wiring 128x64 OLEDs 16
Solder Header 16
I2C or SPI 17
Using with I2C 17
Using with SPI 18
Wiring 128x32 SPI OLED display 21
128x32 SPI OLED 21
Wiring 128x32 I2C Display 23
128x32 I2C OLED 23
Wiring OLD 0.96" 128x64 OLED 26
128x64 Version 1.0 OLED 26
Downloads 30
Software 30
Datasheets 30
Files 30
Schematic & Fabrication Print for 0.96" OLED 30
Schematic & Fabrication Print for 1.3" OLED 32
Schematic & Fabrication Print for 0.91" 128x32 I2C 34
Schematic & Fabrication Print for 0.91" 128x32 SPI 36

Overview


SSD1306 OLED
For internal booster
GND
VDD
CS
RES
D/C
Clk
D0
D1
D2
VCC
BS0
BS1
RD
D7
D6
D5
D4
D3
Lorem ipsum dolor sit
amet, consectetur ad
ipisicing elit, sed d
o eiusmod tempor inci
didunt ut labore et d
olore magna aliqua. U
t enim ad minim venia
m, quis nostrud exerc



This is a quick tutorial for our 128x64 and 128x32 pixel monochrome OLED displays. These displays are small, only about 1" diagonal, but very readable due to the high contrast of an OLED display. Each OLED display is made of 128x64 or 128x32 individual white OLEDs, each one is turned on or off by the controller chip. Because the display makes its own light, no backlight is required. This reduces the power required to run the OLED and is why the display has such high contrast; we really like this miniature display for its crispness!

The driver chip, **SSD1306** can communicate in multiple ways including **I2C**, **SPI** and **8-bit parallel**. However, only the 128x64 display has all these interfaces available. For the 128x32 OLED, only SPI is available. Frankly, we prefer SPI since its the most flexible and uses a small number of I/O pins so our example code and wiring diagram will use that.

Power Requirements

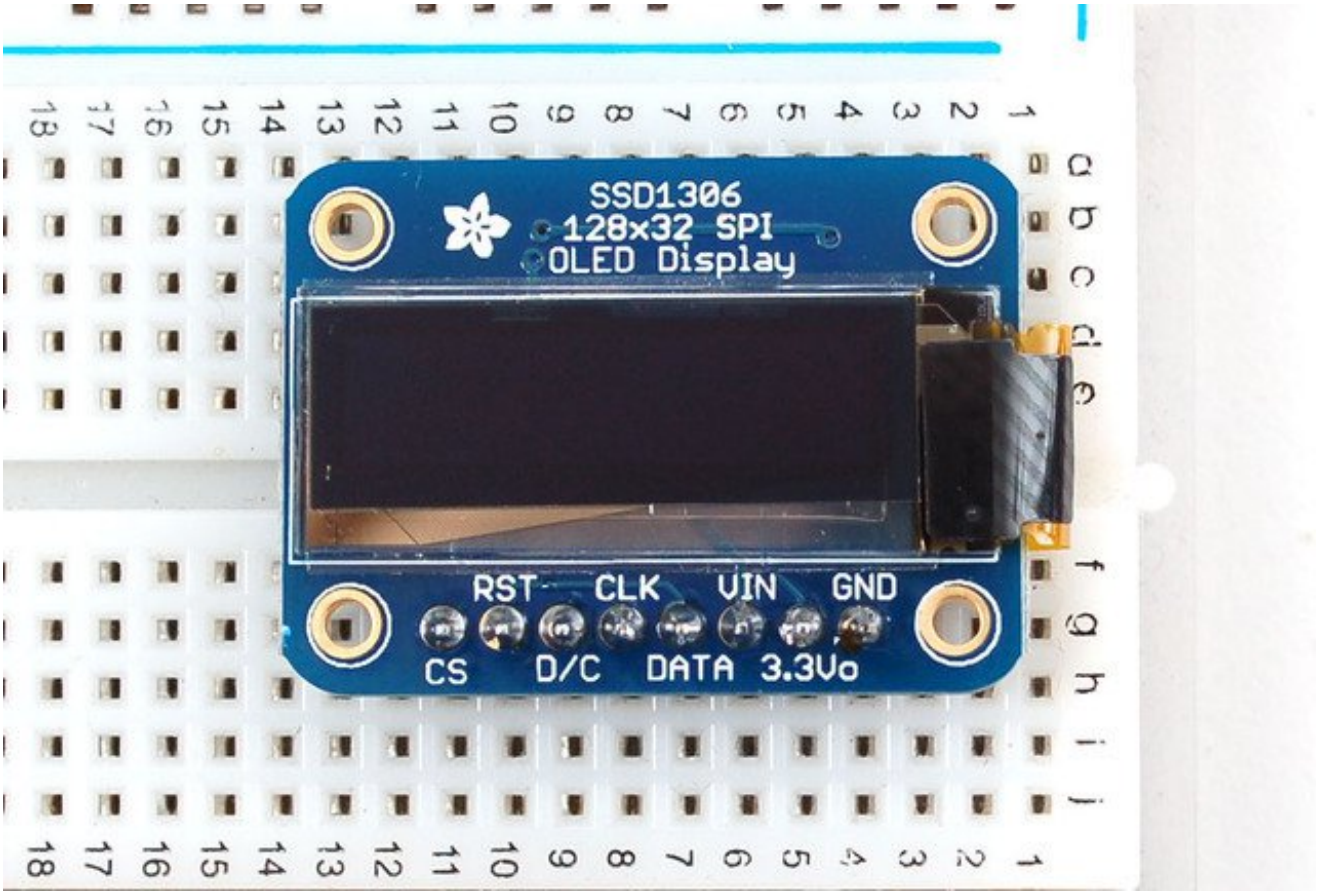
OLED Power Requirements

The OLED and driver require a 3.3V power supply and 3.3V logic levels for communication. The power requirements depend a little on how much of the display is lit but on average the display uses about 20mA from the 3.3V supply. Built into the OLED driver is a simple switch-cap charge pump that turns 3.3v-5v into a high voltage drive for the OLEDs. You can run the entire display off of one 3.3V supply or use 3.3V for the chip power and up to 4.5V for the OLED charge pump or 3.3V for the chip power and a 7-9V supply directly into the OLED high voltage pin.

5V- ready 128x64 and 128x32 OLEDs

Unless you have the older v1 128x64 OLED, you can rest assured that your OLED is 5V ready. All 1.3" 128x64 and the small 128x32 SPI and I2C are 5V ready, if you have a v2 0.96" 128x64 OLED with the 5V ready mark on the front, it's also 5V safe. If you have an older 0.96" OLED (see below) you'll need to take extra care when wiring it to a 5V micontroller. The OLED is designed to be 5V compatible so you can power it with 3-5V and the onboard regulator will take care of the rest.

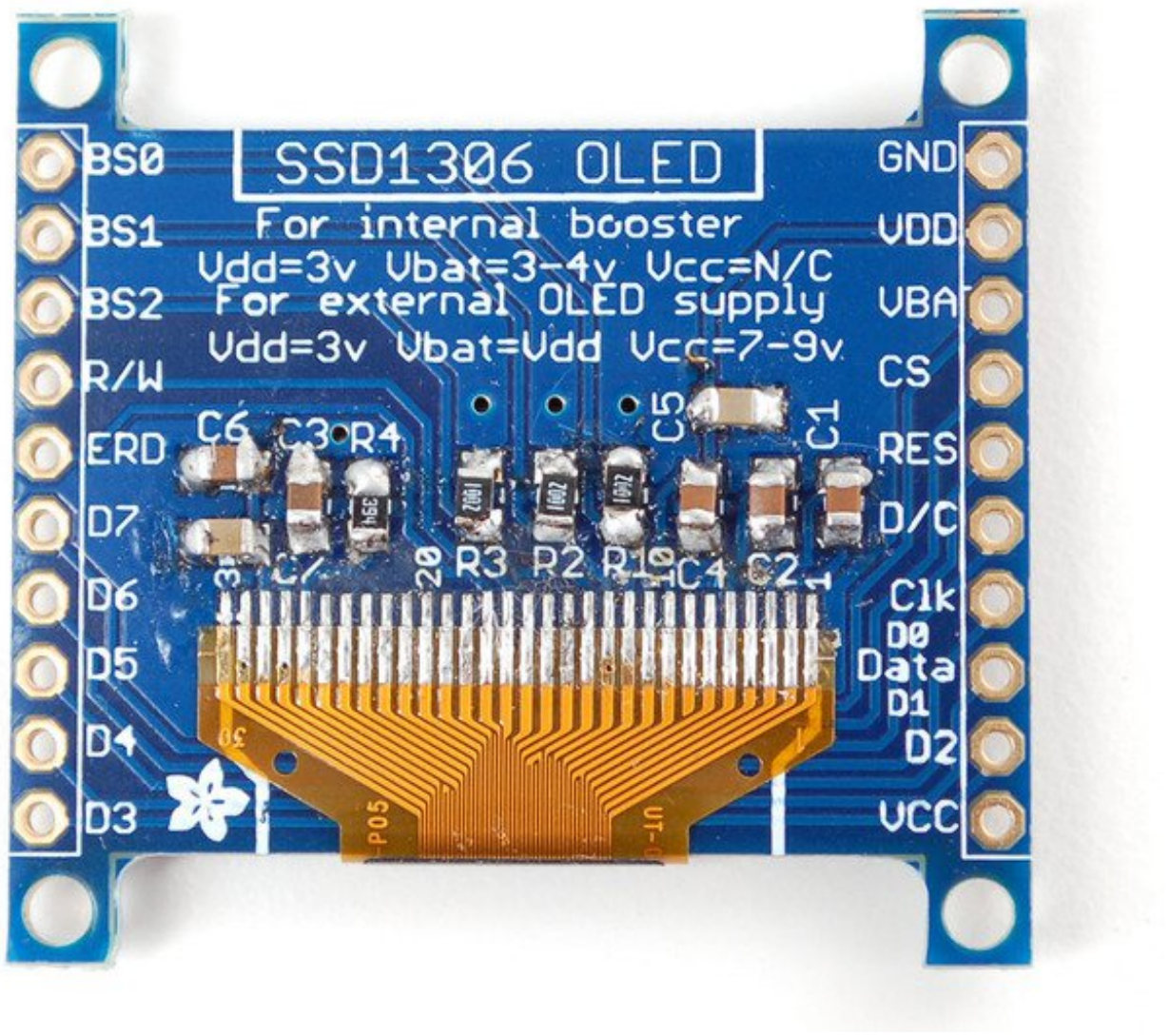
All OLEDs are safe to use with 3.3V logic and power.



Simply connect **GND** to ground, and **Vin** to a 3 to 5V power supply. There will be a 3.3V output on the **3Vo** pin in case you want a regulated 3.3V supply for something else.

0.96" 128x64 OLED

The older 0.96" 128x64 OLED is a little more complex to get running as it is not 5V compatible by default, so you have to provide it with 3.3V power.



- **VDD** is the 3.3V logic power. This must be 3 or 3.3V
- **VBAT** is the input to the charge pump. If you use the charge pump, this must be 3.3V to 4.2V
- **VCC** is the high voltage OLED pin. If you're using the internal charge pump, this must be left unconnected. If you're not using the charge pump, connect this to a 7-9V DC power supply.

For most users, we suggest connecting **VDD** and **VBAT** together to 3.3V and then leaving **VCC** unconnected.

Arduino Library & Examples

For all of the different kinds of small OLED monochrome displays, you'll need to install the Arduino libraries. The code we have is for any kind of Arduino, if you're using a different microcontroller, the code is pretty simple to adapt, the interface we use is basic bit-twiddling SPI or I2C

Install Adafruit SSD1306 Library

Start by installing the support library for the OLED display, you'll need it to talk to the OLED controller chip. We have the Adafruit SSD1306 library repository on GitHub (http://adafru.it/aHq) if you're interested in looking at the code.

Start by downloading the library. You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip:

Download Adafruit_SSD1306 Library
http://adafru.it/e3E

Rename the uncompressed folder **Adafruit_SSD1306** and check that the **Adafruit_SSD1306** folder contains **Adafruit_SSD1306.cpp** and **Adafruit_SSD1306.h**

Place the **Adafruit_SSD1306** library folder your ***arduinosketchfolder***/**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)

Install Adafruit GFX

You will need to do the same for the Adafurit_GFX library available here (http://adafru.it/aJa)

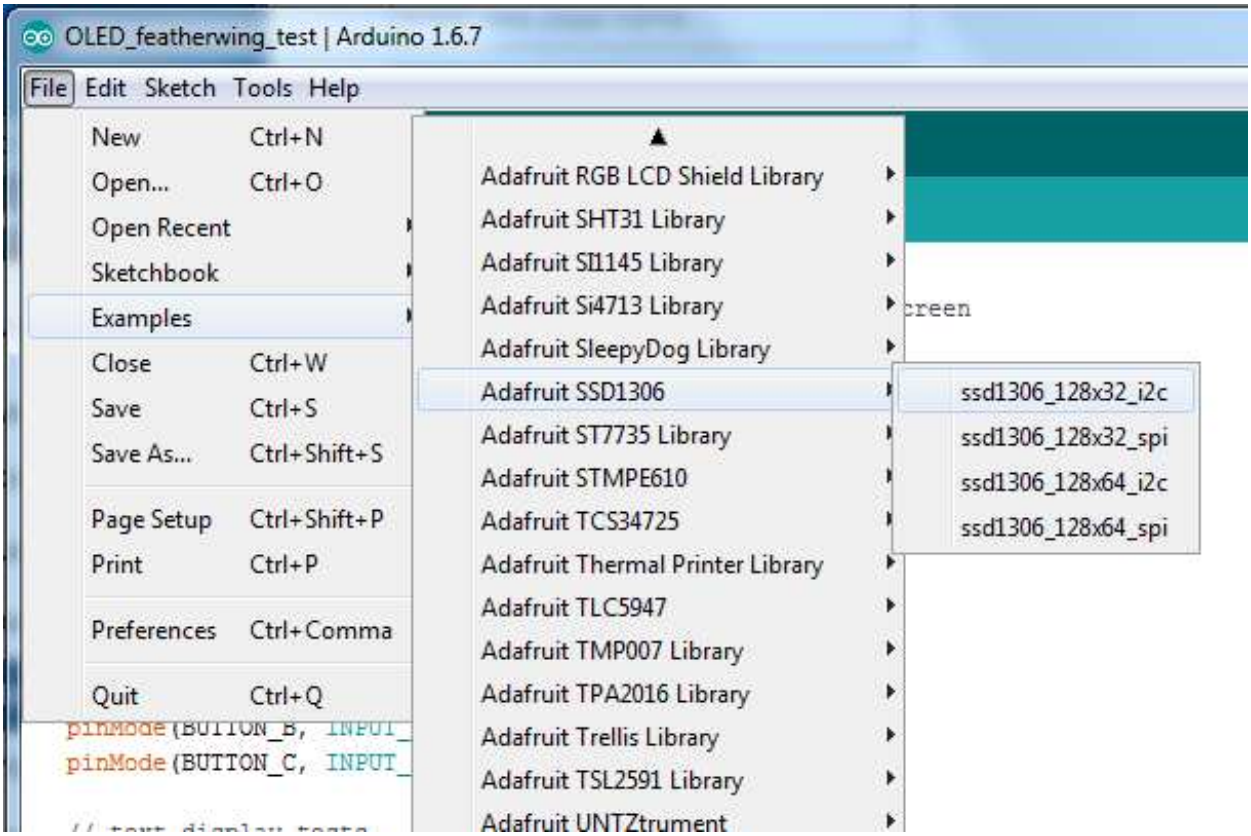
Download Adafruit GFX Library
http://adafru.it/cBB

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

Place the **Adafruit_GFX** library folder your ***arduinosketchfolder*/libraries/** folder like you did with the SSD1306 library

Run Demo!

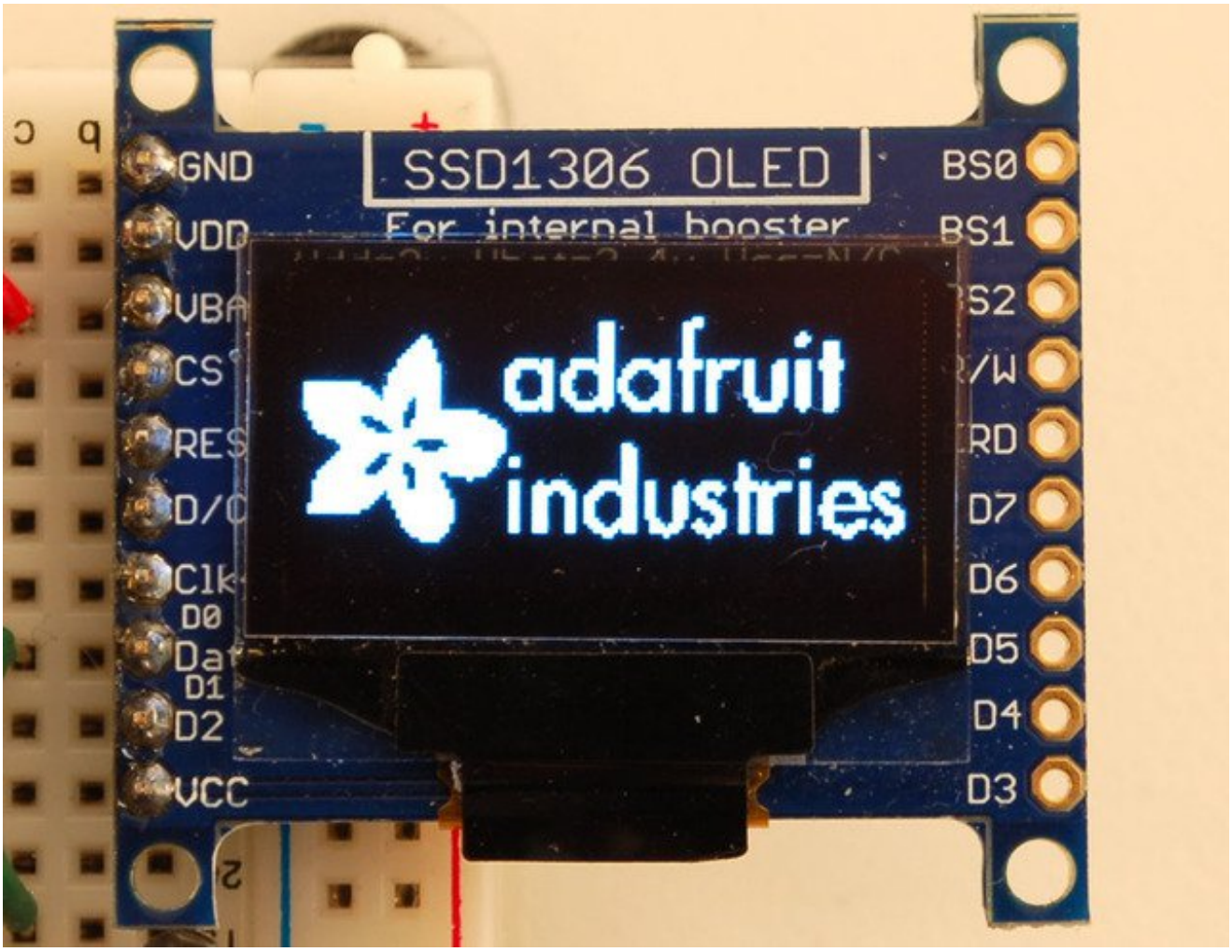
After installing the **Adafruit_SSD1306** and **Adafruit_GFX** library, restart the Arduino IDE. You should now be able to access the sample code by navigating through menus in this order: **File→Sketchbook→Libraries→Adafruit_SSD1306→SSD1306...**

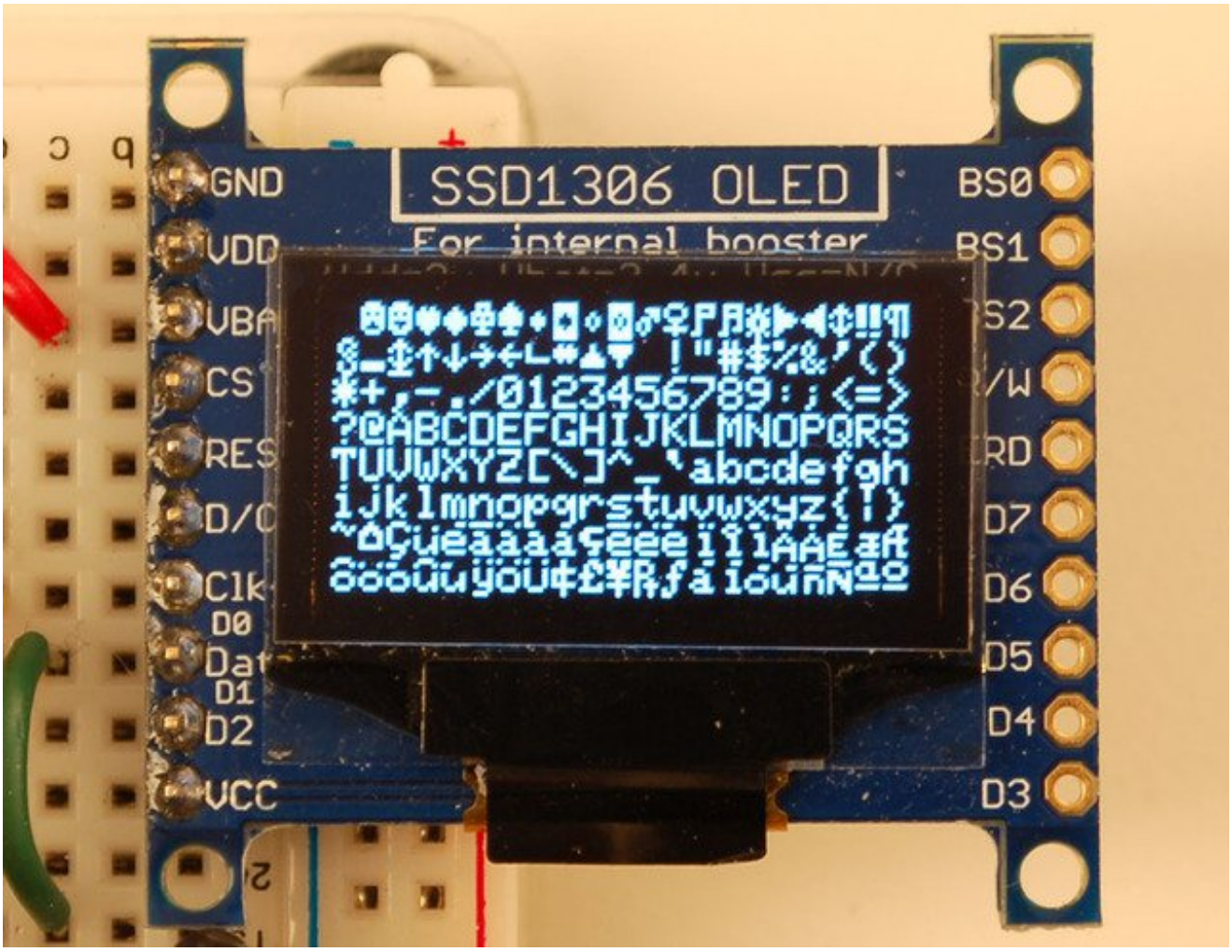


After you've finished wiring the display as indicated on the following pages, load the example sketch to demonstrate the capabilities of the library and display.

The OLED SSD1306 driver is based on the Adafruit GFX library which provides all the underlying graphics functions such as drawing pixels, lines, circles, etc. For more details about what you can do with the OLED check out the GFX library tutorial (http://adafru.it/aPx)

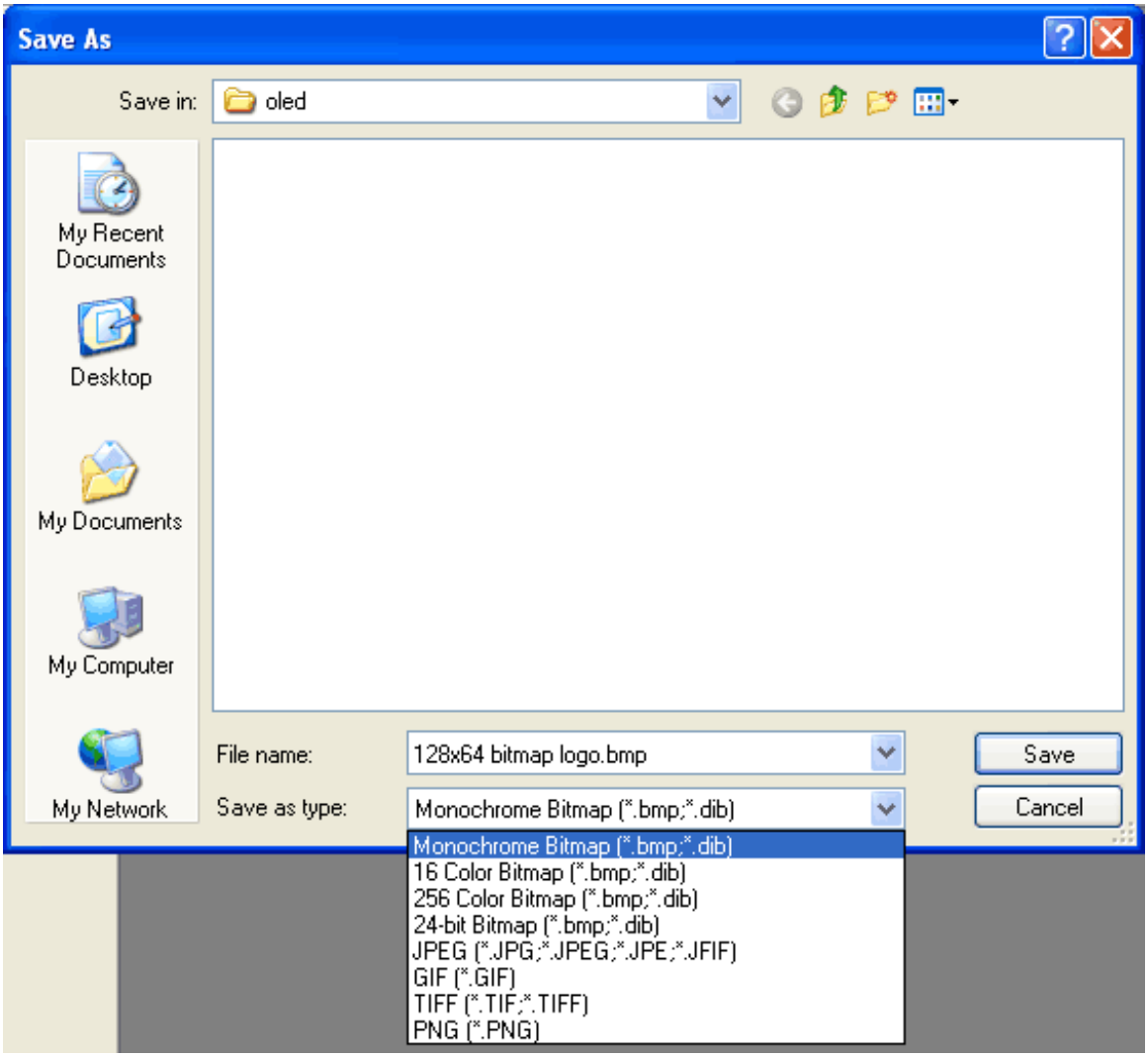
The library cannot determine the screen size via software. You have to specify it in the header file. Go into the library folder and open the file Adafruit_SSD1306.h. Toward the top you'll see a section of comments that tell how how to edit the file for different kinds of display.


SSD1306 OLED
For internal booster
GND
VDD
CS
RES
D/C
Clk
D0
D1
D2
VCC
BS0
BS1
D7
D6
D5
D4
D3
adafruit
industries

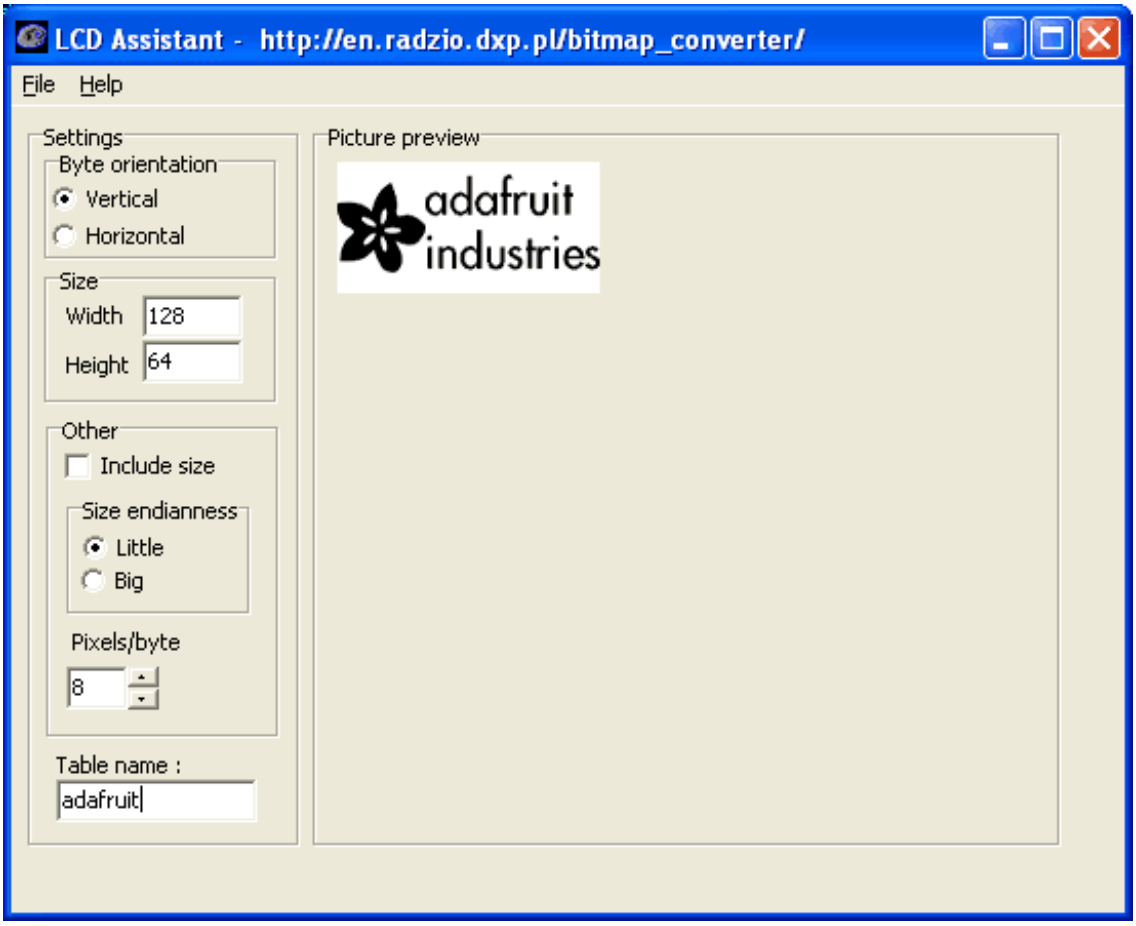


Create Bitmaps

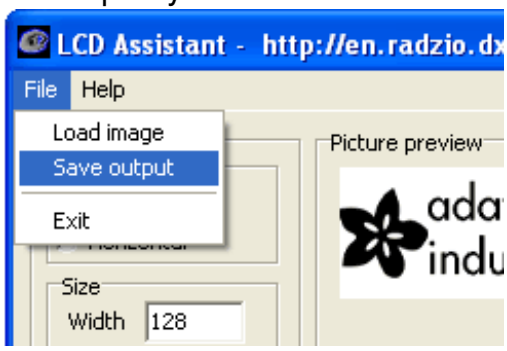
You can create bitmaps to display easily with the LCD assistant software (http://adafru.it/aPs). First make your image using any kind of graphics software such as photoshop or Paint and save as a **Monochrome Bitmap (bmp)**



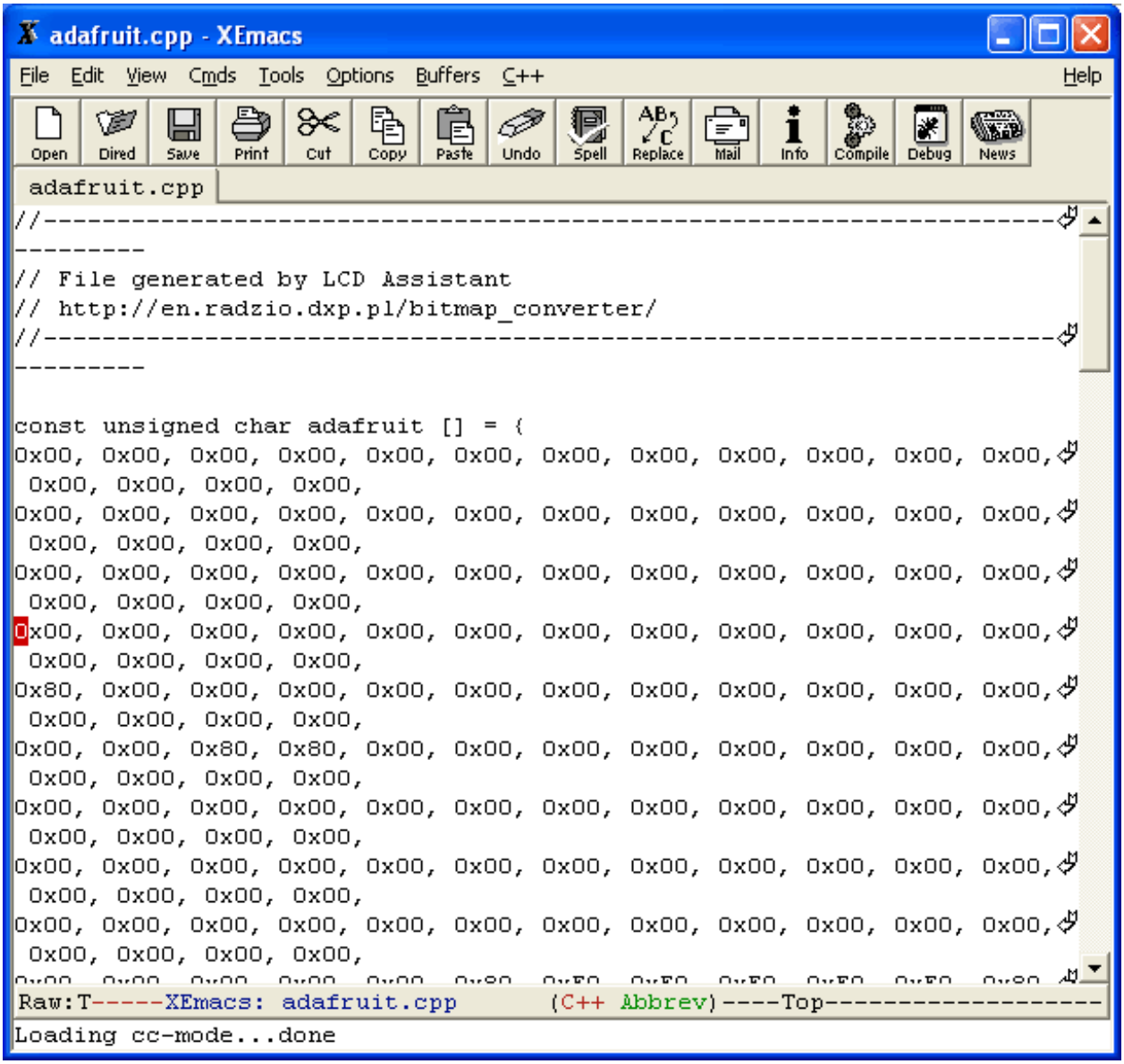
Select the following options (You might also want to try **Horizontal** if **Vertical** is not coming out right)



and import your monochrome bitmap image. Save the output to a **cpp** file



You can use the output directly with our example code

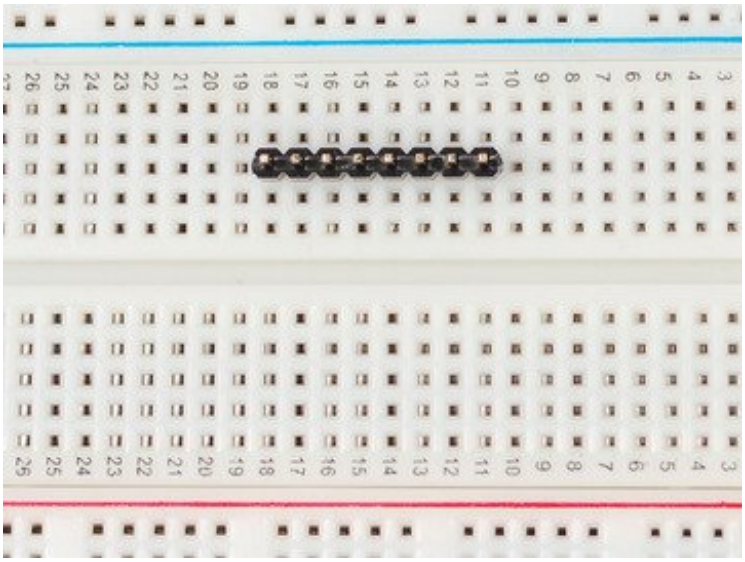

adafruit.cpp - XEmacs
File Edit View Cmds Tools Options Buffers C++ Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
adafruit.cpp
//--

// File generated by LCD Assistant
// http://en.radzio.dxp.pl/bitmap_converter/
//--
----
const unsigned char adafruit [] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
Raw:T-----XEmacs: adafruit.cpp (C++ Abbrev)----Top----------
Loading cc-mode...done

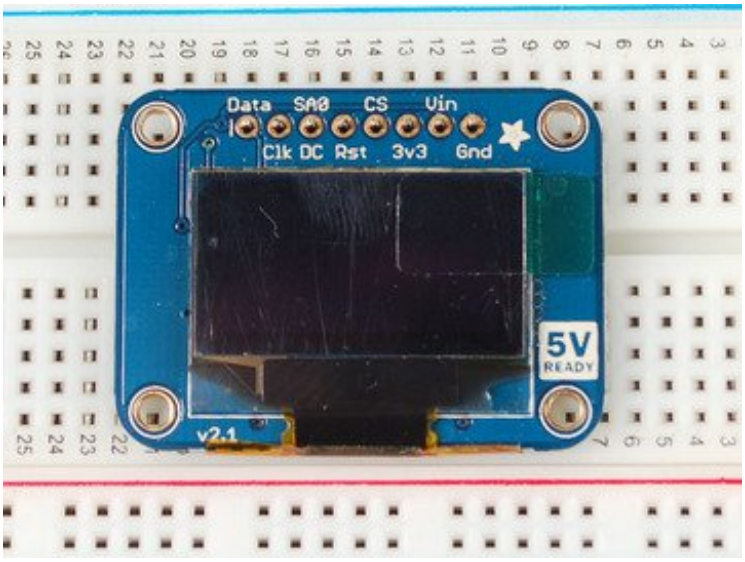
Wiring 128x64 OLEDs

Solder Header

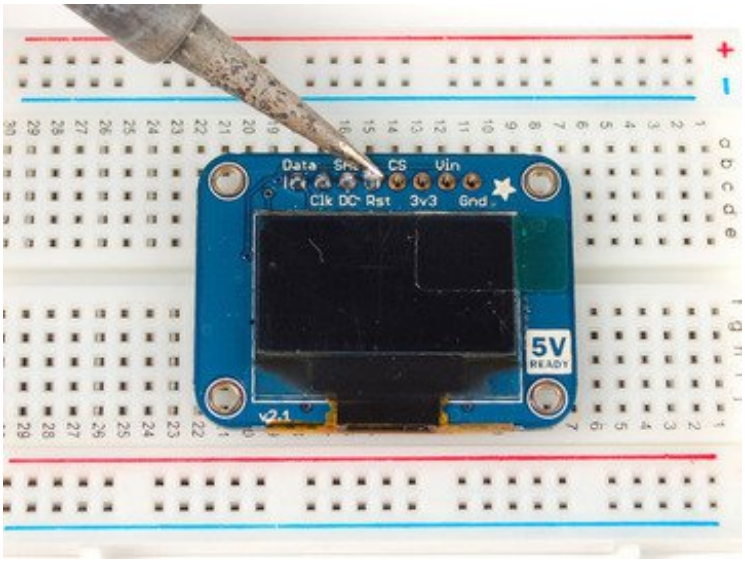
Before you start wiring, a strip of header must be soldered onto the OLED. It is not possible to "press-fit" the header, it must be attached!

- 

  Start by placing an 8-pin piece of header with the **long** ends down into a breadboard for stability

- 

  Place the OLED on top so all the short ends of the header stick thru the header pads

- 

Finish by soldering each of the 8 pins to the 8 pads!

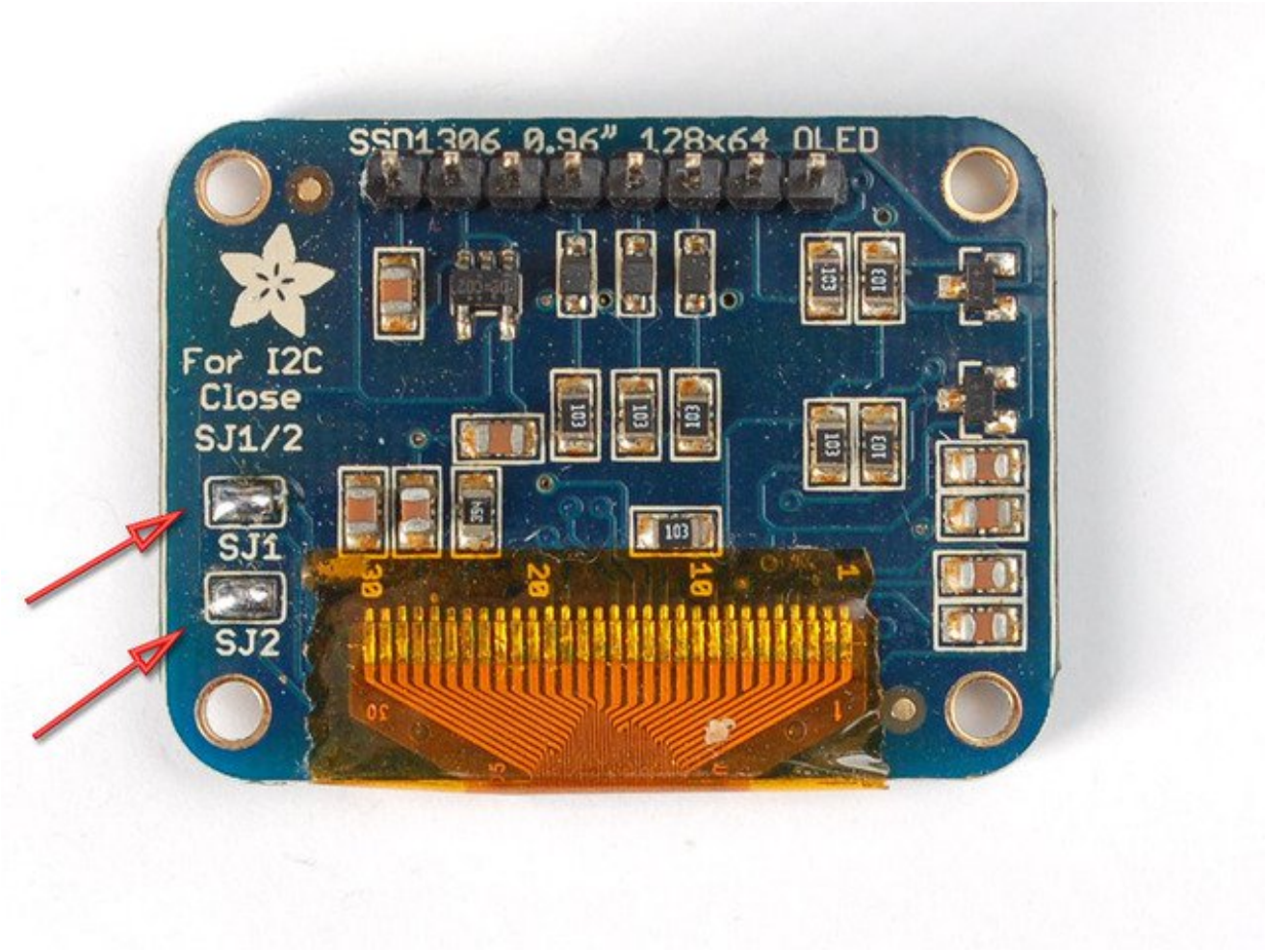
I2C or SPI

The nice thing about the 128x64 OLEDs is that they can be used with I2C (+ a reset line) or SPI. SPI is generally faster than I2C but uses more pins. It's also easier for some microcontrollers to use SPI. Anyways, you can use either one with this display

Using with I2C

The display can be used with any I2C microcontroller. Because the I2C interface is for 'writing' to the display only, you'll still have to buffer the entire 512 byte frame in the microcontroller RAM - you can't read data from the OLED (even though I2C is a bidirectional protocol).

To start, you'll need to solder the two jumpers on the back of the OLED.**Both** must be soldered 'closed' for I2C to work!



Finally, connect the pins to your Arduino

- **GND** goes to ground
- **Vin** goes to 5V
- **Data** to I2C SDA (on the Uno, this is **A4** on the Mega it is **20** and on the Leonardo digital **2**)
- **Clk** to I2C SCL (on the Uno, this is **A5** on the Mega it is **21** and on the Leonardo digital **3**)
- **RST** to digital **4** (you can change this pin in the code, later)

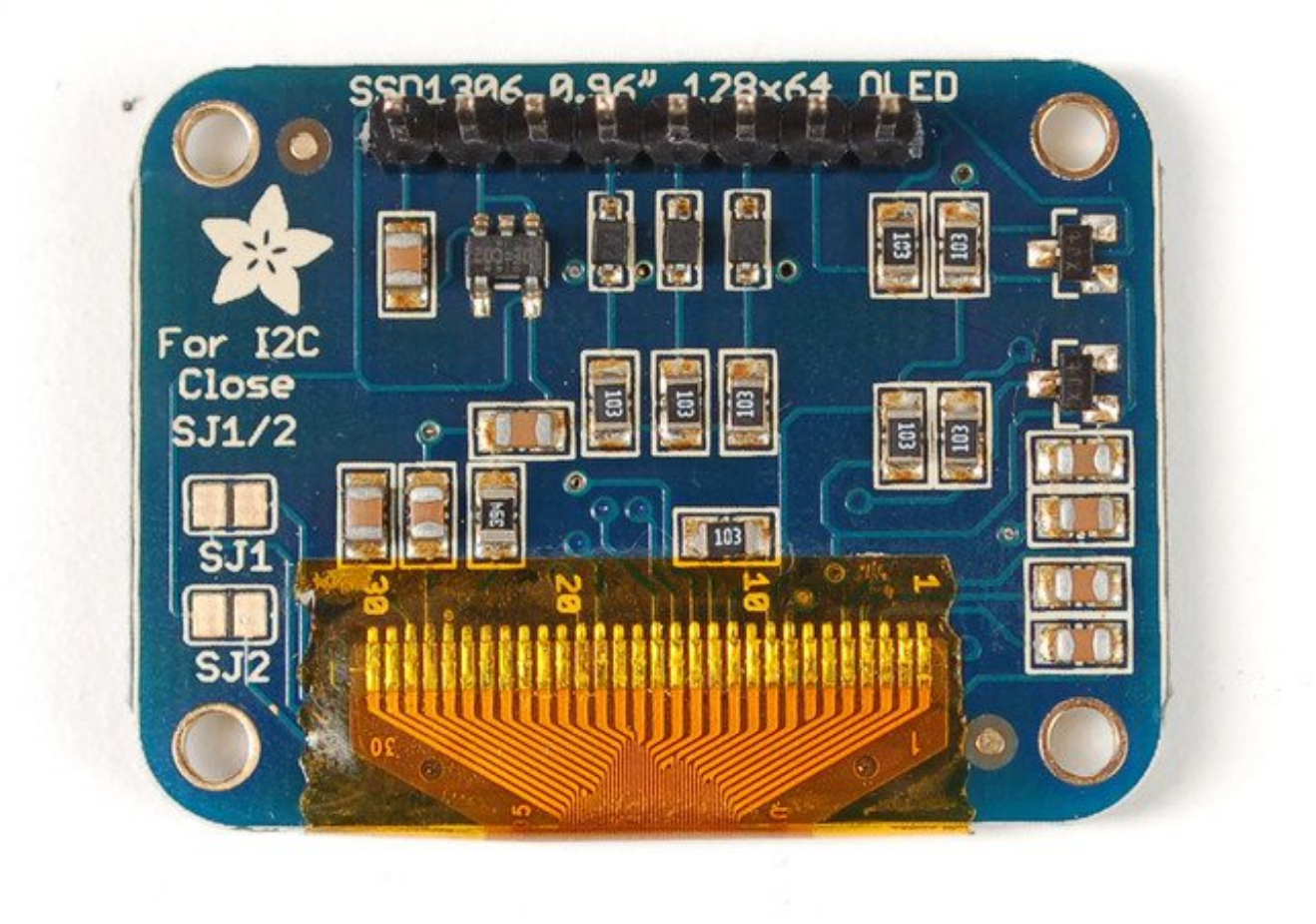
This matches the example code we have written. Once you get this working, you can try a different Reset pin (you can't change the SCA and SCL pins).

Finally you can run the
File→Sketchbook→Libraries→Adafruit_SSD1306→SSD1306_128x64_i2c example

Using with SPI

The breakouts are ready for SPI by default, but if you used them for I2C at some point,

you'll need to remove the solder jumpers. Use wick or a solder sucker to make sure both are clear!



Finally, connect the pins to your Arduino -

- **GND** goes to ground
- **Vin** goes to 5V
- **DATA** to digital **9**
- **CLK** to digital **10**
- **D/C** to digital **11**
- **RST** to digital **13**
- **CS** to digital **12**

*(**Note**: If using the display with other SPI devices, D/C, CLK and DAT may be shared, but CS must be unique for each device.)*

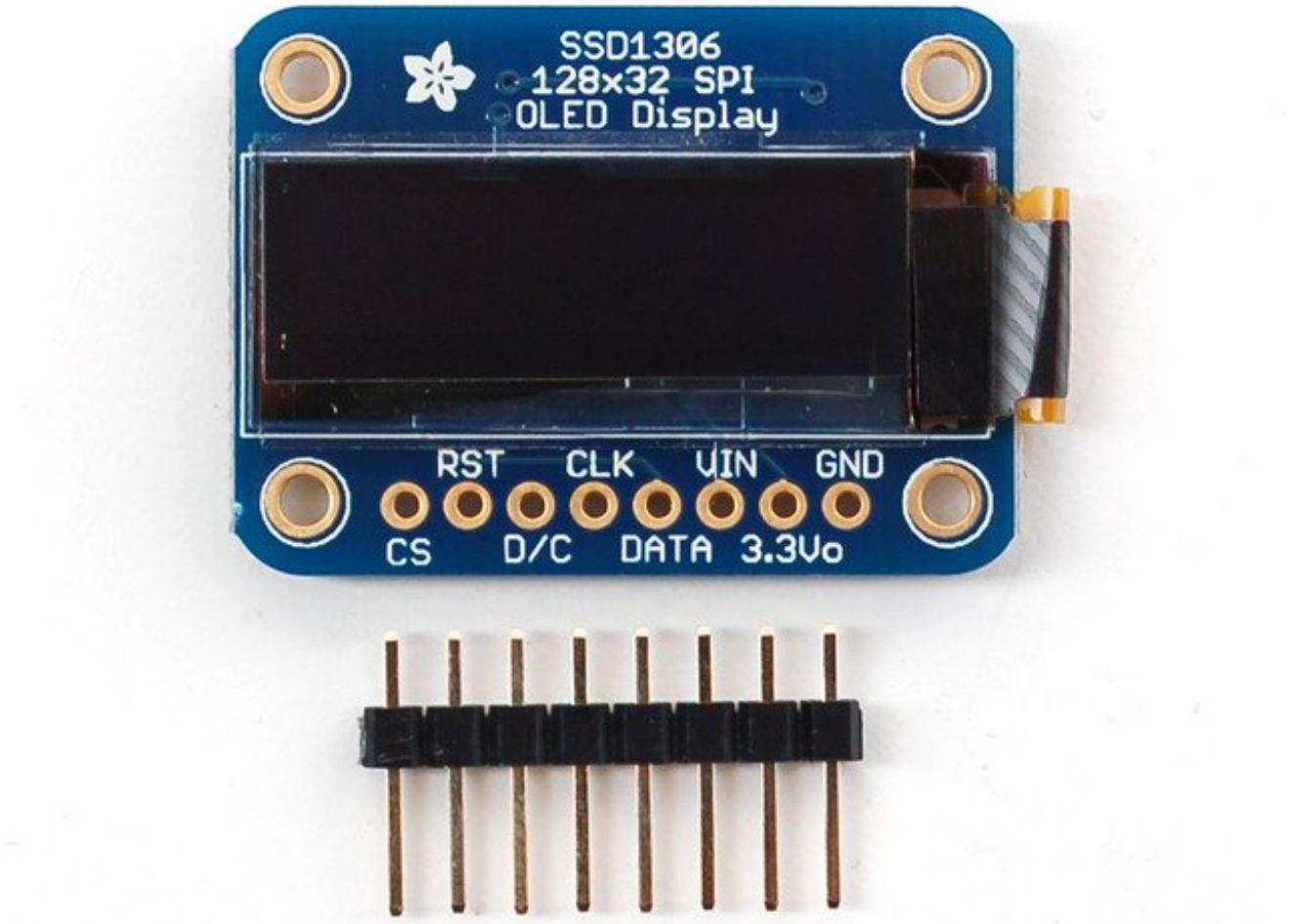
This matches the example code we have written. Once you get this working, you can try another set of pins.

Finally you can run the
File→Sketchbook→Libraries→Adafruit_SSD1306→SSD1306_128x64_spi example

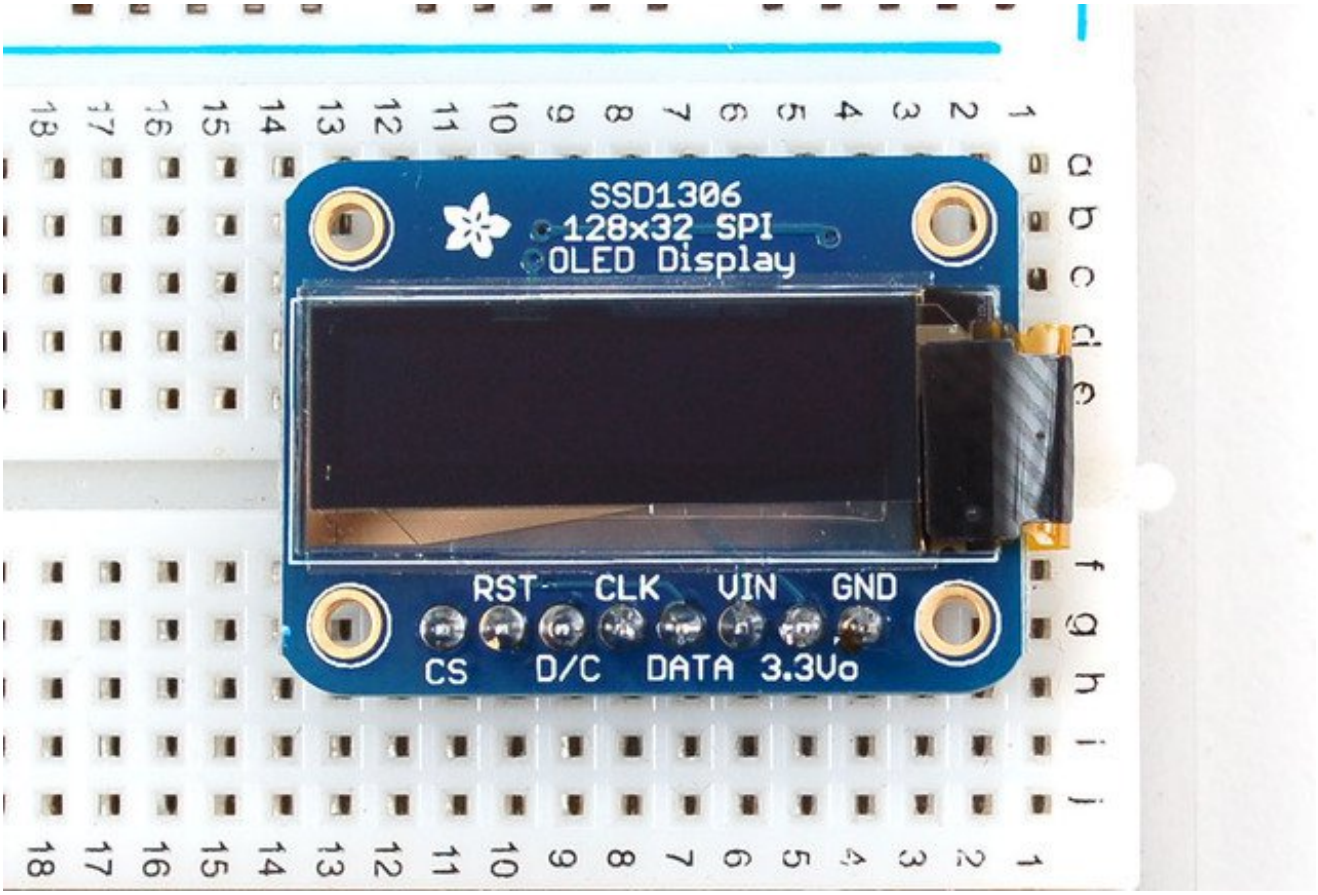
Wiring 128x32 SPI OLED display

128x32 SPI OLED

The 128x32 SPI OLED is very easy to get up and running because it has built in level shifting. First up, take a piece of 0.1" header 8 pins long.



Plug the header long end down into a breadboard and place the OLED on top. Solder the short pins into the OLED PCB.



Finally, connect the pins to your Arduino -**GND** goes to ground, **Vin** goes to 5V, **DATA** to digital **9**, **CLK** to digital **10**, **D/C** to digital **11**, **RST** to digital **13** and finally **CS** to digital **12.**

*(**Note**: If using the display with other SPI devices, D/C, CLK and DAT may be shared, but CS must be unique for each device.)*

This matches the example code we have written. Once you get this working, you can try another set of pins.

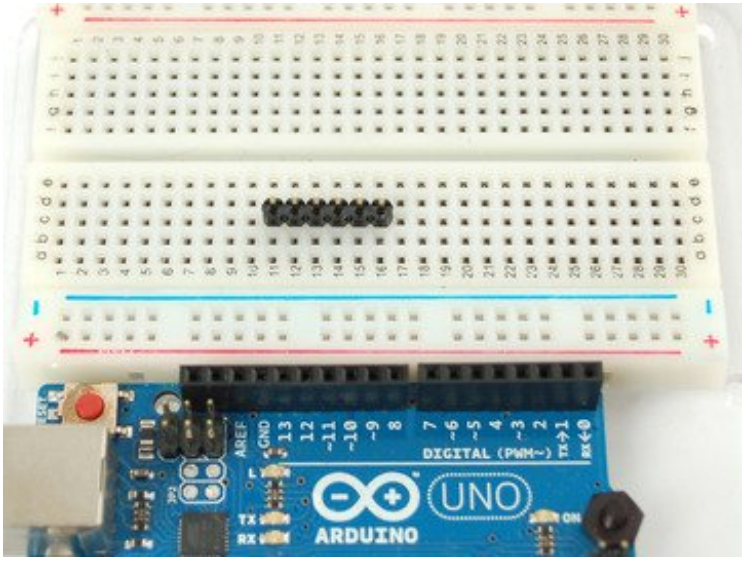
Finally you can run
the **File→Sketchbook→Libraries→Adafruit_SSD1306→SSD1306_128x32_SPI example**

If you're using the 128x32 OLED, be sure to uncomment the "#define SSD1306_128_32" in the top of Adafruit_SSD1306.h to change the buffer size

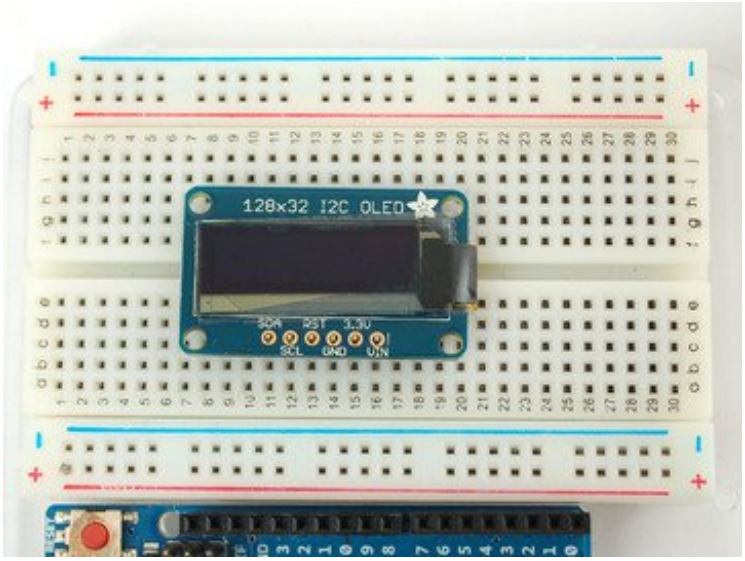
Wiring 128x32 I2C Display

128x32 I2C OLED

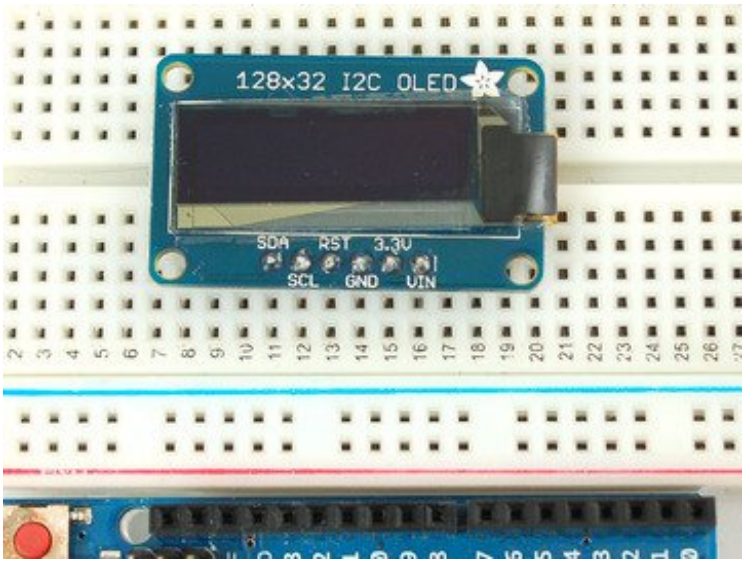
The 128x32 I2C OLED is very easy to get up and running because it has built in level shifting and regulator. First up, take a piece of 0.1" header 6 pins long.

- 

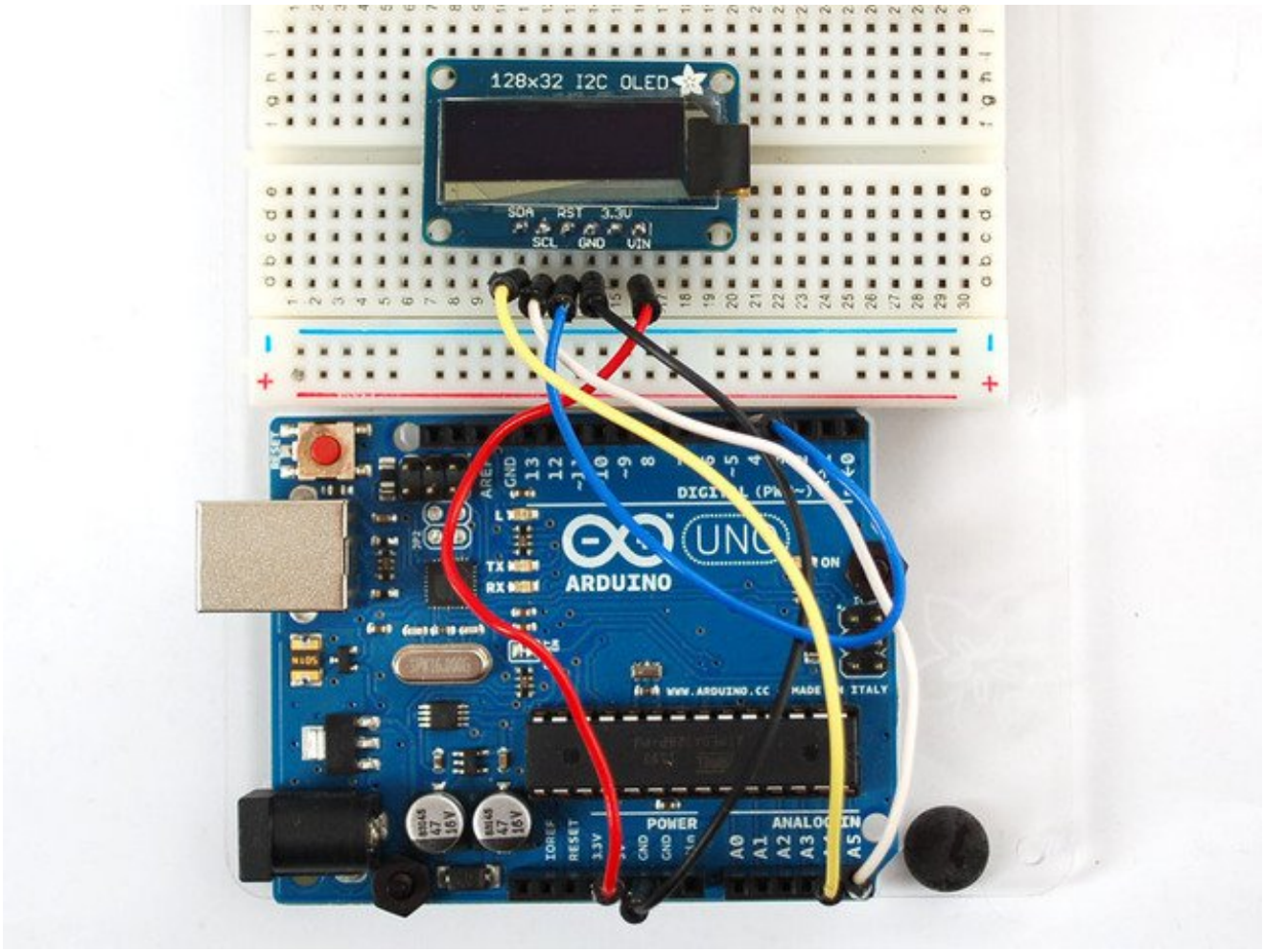
Plug the header long end down into a breadboard

- 

Place the OLED on top

- 

Solder the short pins into the OLED PCB.



Finally, connect the pins to your Arduino

- **GND** goes to ground
- **Vin** goes to 5V
- **SDA** to I2C Data (on the Uno, this is **A4** on the Mega it is **20** and on the Leonardo

digital **2**)

- **SCL** to I2C Clock(on the Uno, this is **A5** on the Mega it is **21** and on the Leonardo digital **3**)
- **RST** to digital **4** (you can change this pin in the code, later)

This matches the example code we have written. Once you get this working, you can change the RST pin. You cannot change the I2C pins, those are 'fixed' in hardware

Finally you can run
the **File→Sketchbook→Libraries→Adafruit_SSD1306→SSD1306_128x32_i2c** example