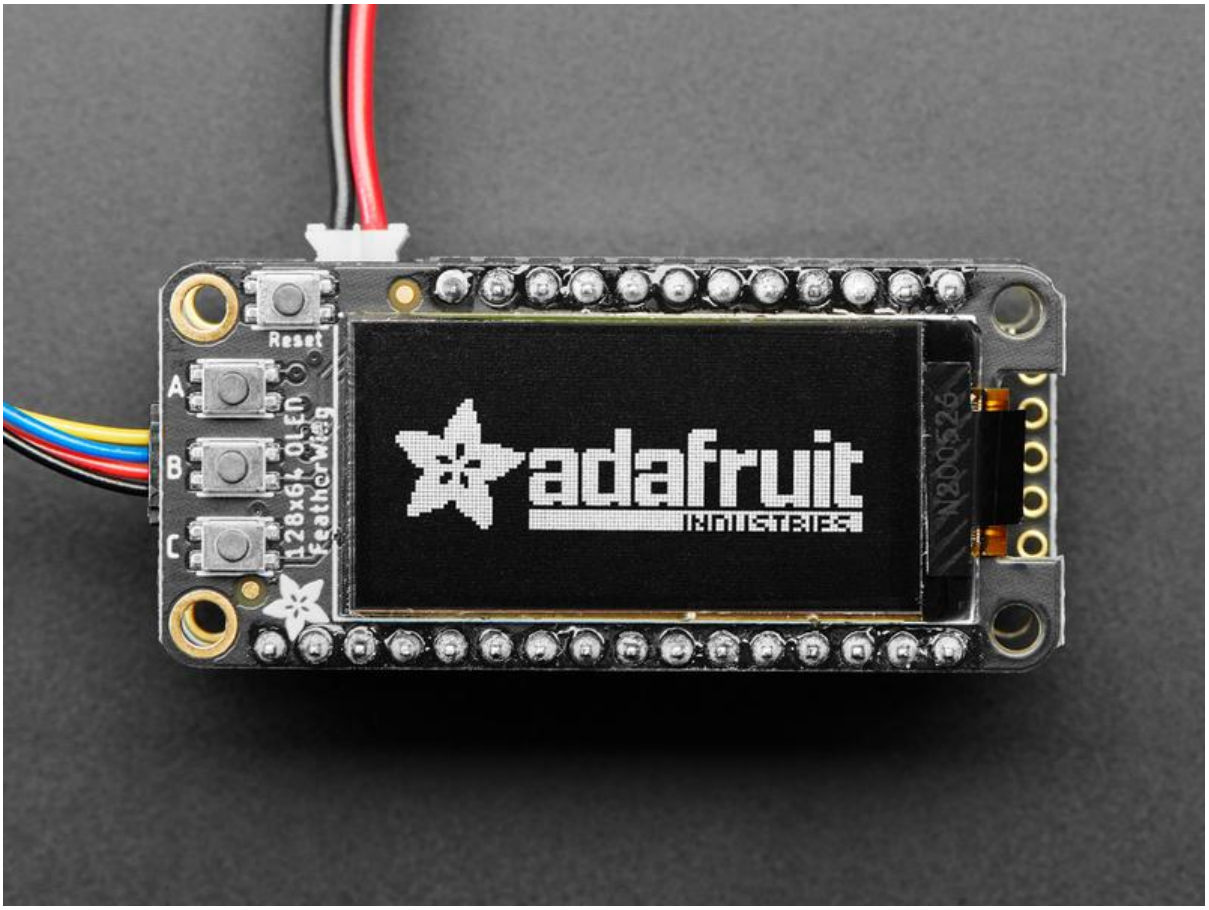
Adafruit 128x64 OLED FeatherWing

Created by Kattni Rembor

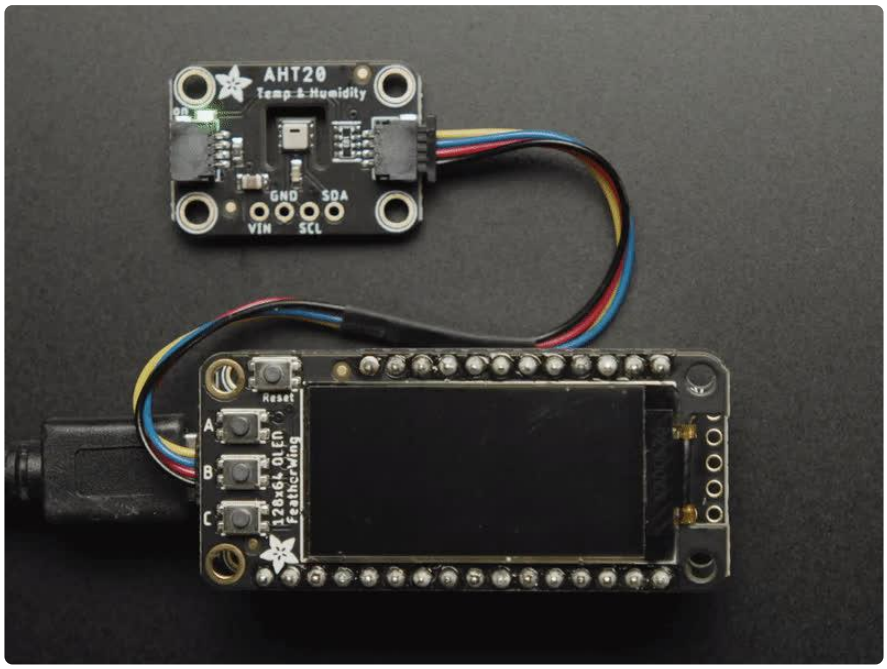


https://learn.adafruit.com/adafruit-128x64-oled-featherwing

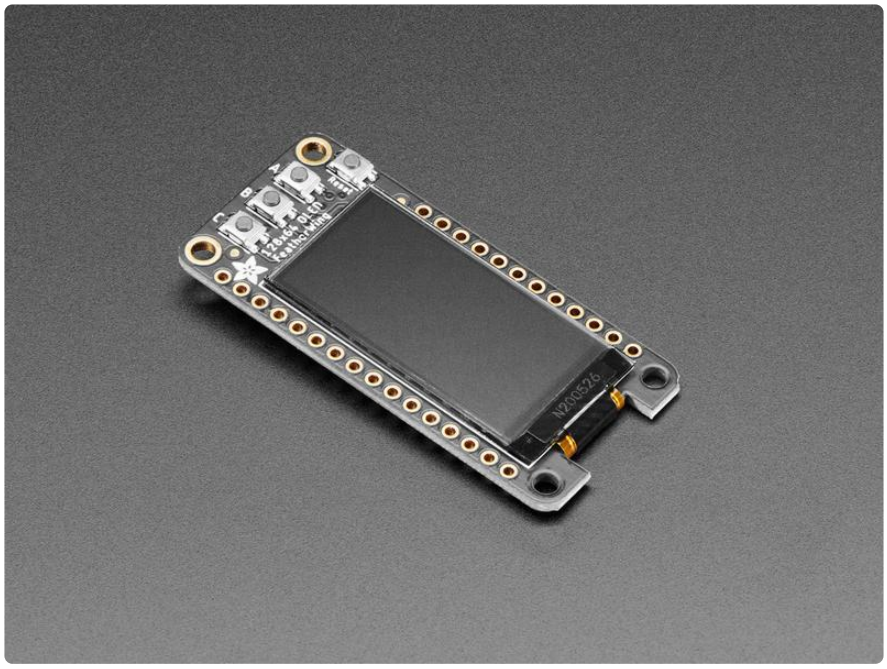
Last updated on 2021-11-15 08:08:42 PM EST

Table of Contents

Overview

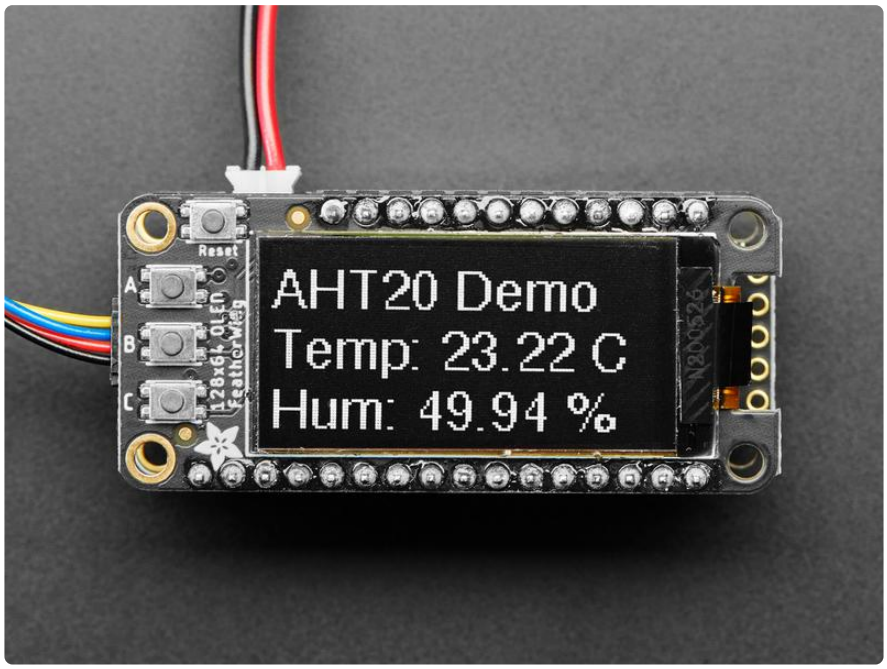


A Feather board without ambition is a Feather board without FeatherWings! This is the FeatherWing 128x64 OLED: it adds a gorgeous 128x64 monochrome OLED plus 3 user buttons to any Feather main board. Using our Feather Stacking Headers (http://adafru.it/2830) or Feather Female Headers (http://adafru.it/2886) you can connect a FeatherWing on top of your Feather board and let the board take flight!

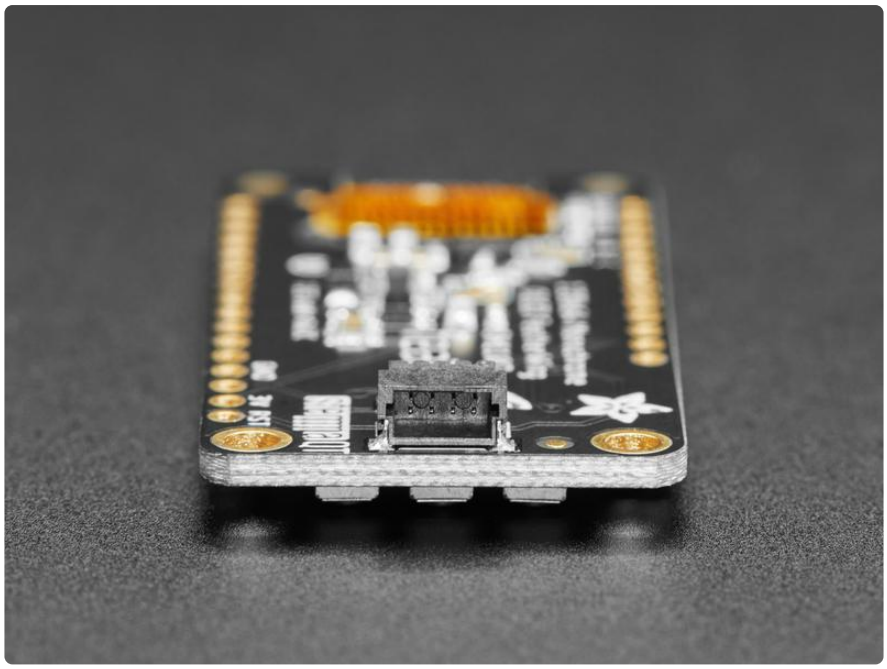


These displays are small, only about 1.3" diagonal, but very readable due to the high contrast of an OLED display. This screen is made of 128x64 individual white OLED

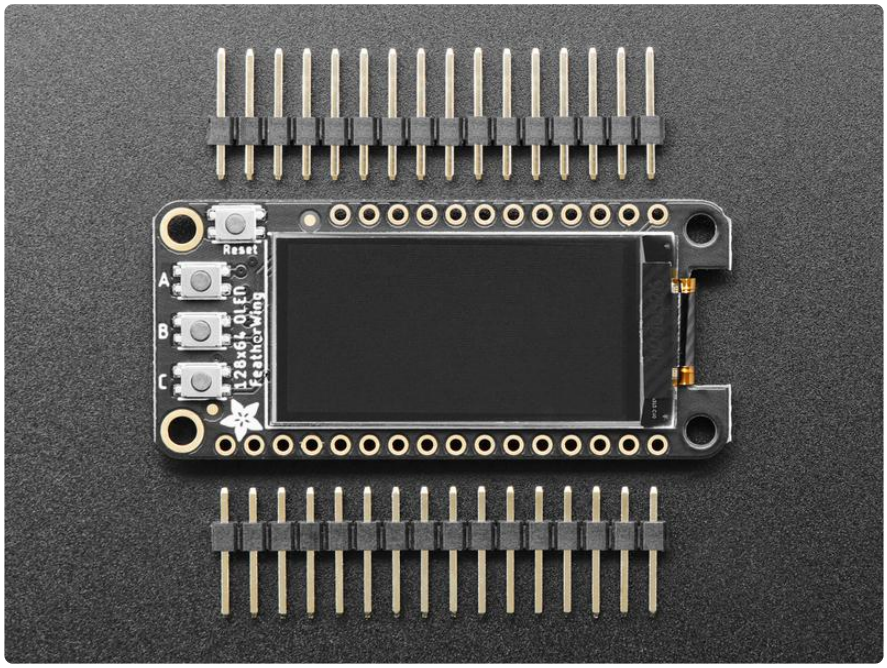
pixels and because the display makes its own light, no backlight is required. This reduces the power required to run the OLED and is why the display has such high contrast; we really like this miniature display for its crispness! We also toss on a reset button and three mini tactile buttons called A B and C, so you can add a mini user interface to your Feather. If you've used our 128x32 OLED FeatherWing (https://adafru.it/sao), you'll be happy to know that this FeatherWing is pin compatible for a quick and easy upgrade.



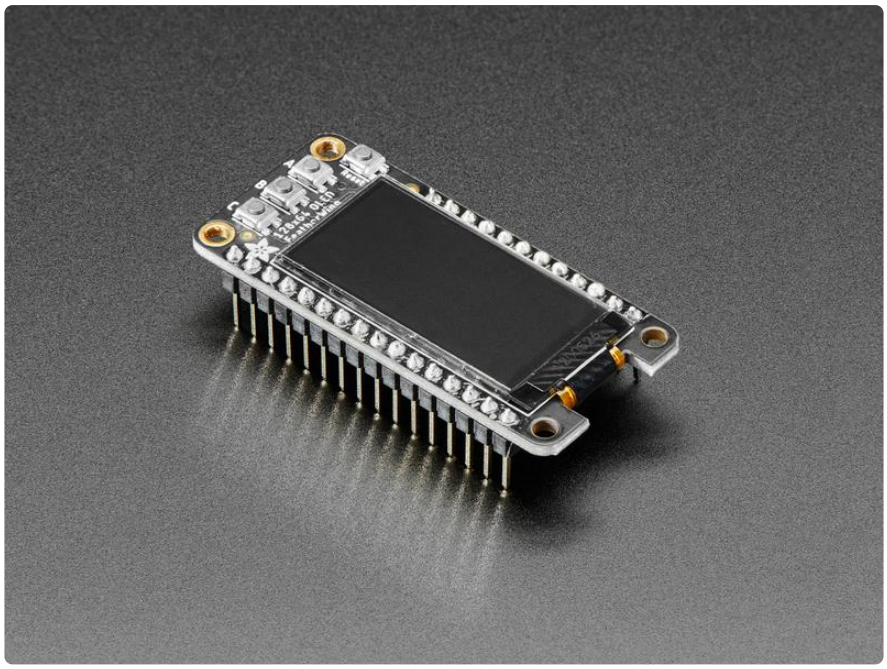
The display uses only I2C so you can easily connect it up with just two pins used (plus power and ground!). There's an auto-reset circuit and a reset button on the top. We've even included a SparkFun qwiic (https://adafru.it/Fpw)-compatible STEMMA QT (https://adafru.it/Ft4) connector for the I2C bus so you can plug and play any of our STEMMA QT, qwiic or Grove I2C sensors (https://adafru.it/Ndk) and devices!



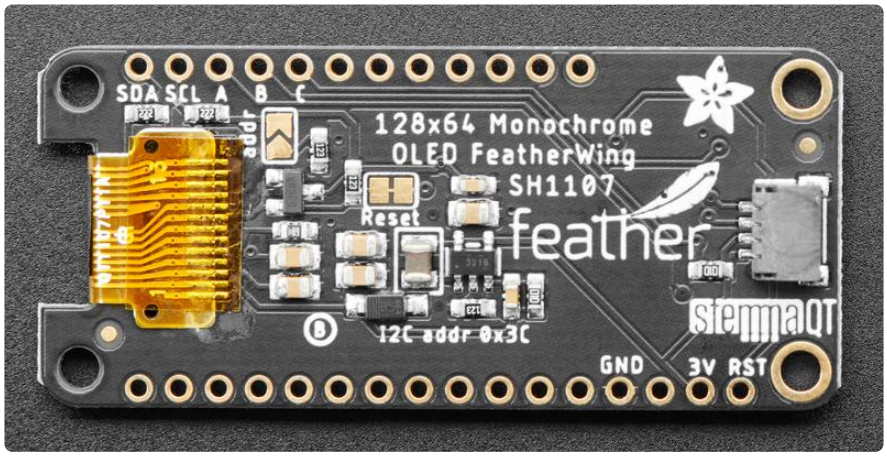
Tested working with all Feather boards. The OLED uses only the two I2C pins on the Feather, and you can pretty much stack it with any other FeatherWing, even ones that use I2C since that is a shared bus.



Comes with a set of 0.1" headers that are unattached, you'll need to solder them in to plug into your Feather board. Check out our range of Feather boards here. (https://adafru.it/l7B)

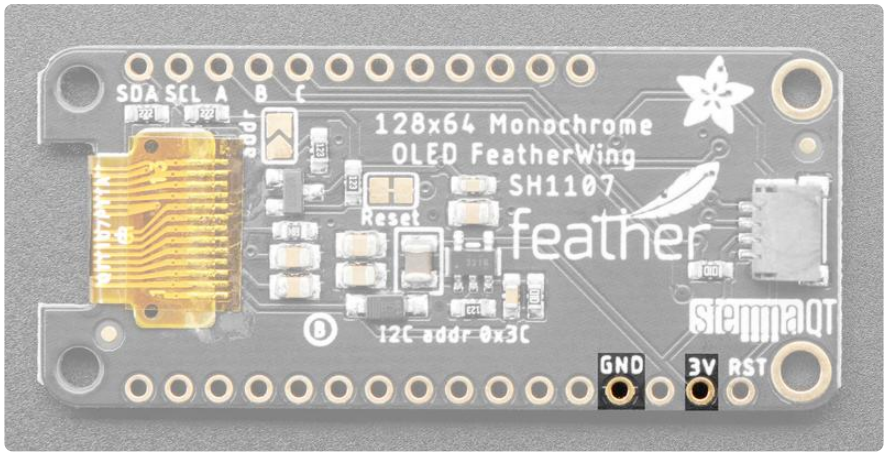


Pinouts



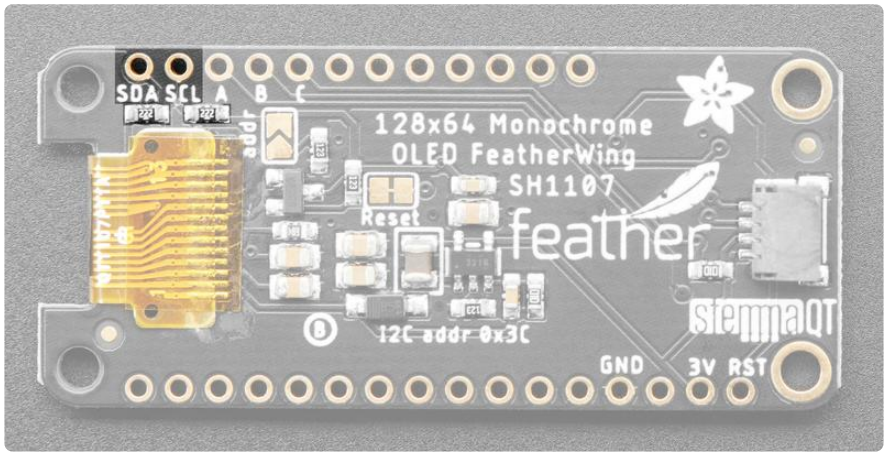
The OLED FeatherWing plugs into any Feather and adds a cute little display. To make it as cross-platform compatible as possible, we use only I2C to control the display. This is not as fast as SPI but it uses only two pins, can share the I2C bus and is fine for the small 128x62 pixel OLED.

Power Pins



OLED displays do not have a backlight, and are fairly low power, this display will draw about 10mA when in use. The display uses 3V power and logic so we just connect to the 3V and GND pins from the feather, as indicated above.

I2C Data Pins

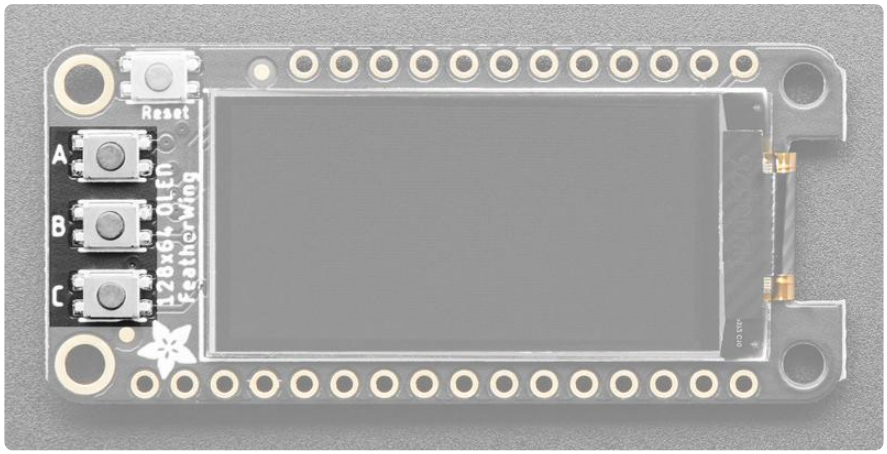


The cute little OLED does all of the data transfer over the I2C pins, highlighed above SDA and SCL. No other pins are required. There are two 2.2K pullups to 3V on each.

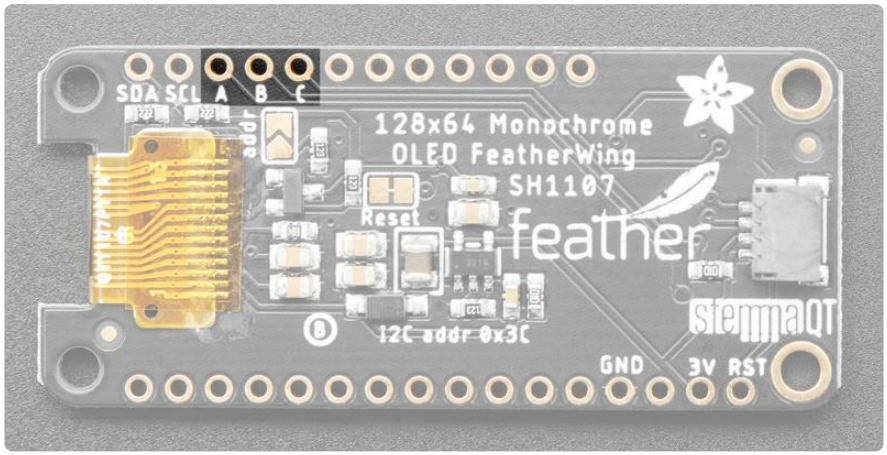
These pins can be shared with other I2C devices.

The default I2C address is 0x3C and can be changed to 0x3D by soldering closed the ADDR jumper.

Optional Buttons



The button pins ON THE BACK are labeled incorrectly in the initial revision of this FeatherWing. It should read C, B, A from left to right, where C should be next to SCL.



We had a little bit of space so we added three mini tactile buttons that you can use for user interface. We label them A B and C because each Feather has slightly different pin numbering schemes and we wanted to make it 'universal'

If you're using ATmega328P, Atmega32u4, ATSAMD51 M4 or ATSAMD21 M0 Feather

- Button A is #9 (note this is also used for the battery voltage divider so if you want to use both make sure you disable the pullup when you analog read, then turn on the pullup for button reads)
- Button B is #6
- Button C is #5

If you're using ESP8266:

- Button A is #0
- Button B is #16
- Button C is #2

If you're using WICED/STM32 Feather

- Button A is #PA15
- Button B is #PC7
- Button C is #PC5

If you're using ESP32 Feather:

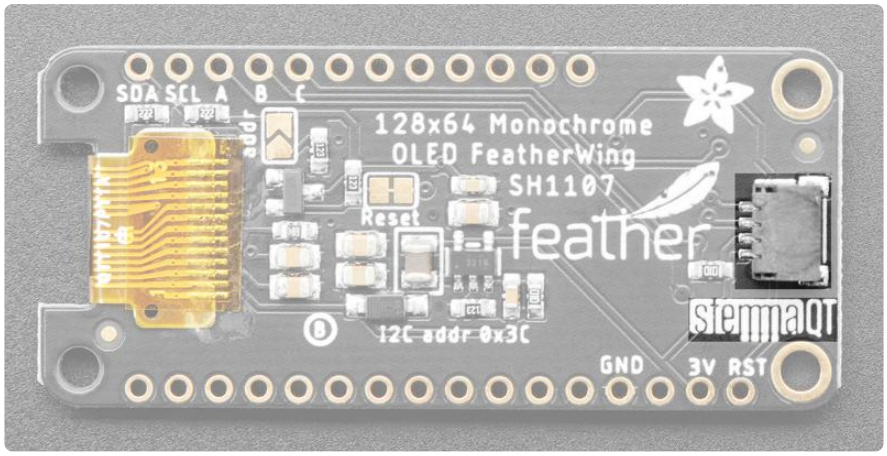
- Button A is 15
- Button B is 32
- Button C is 14

If you're using Teensyduino:

- Button A is 4
- Button B is 3
- Button C is 8

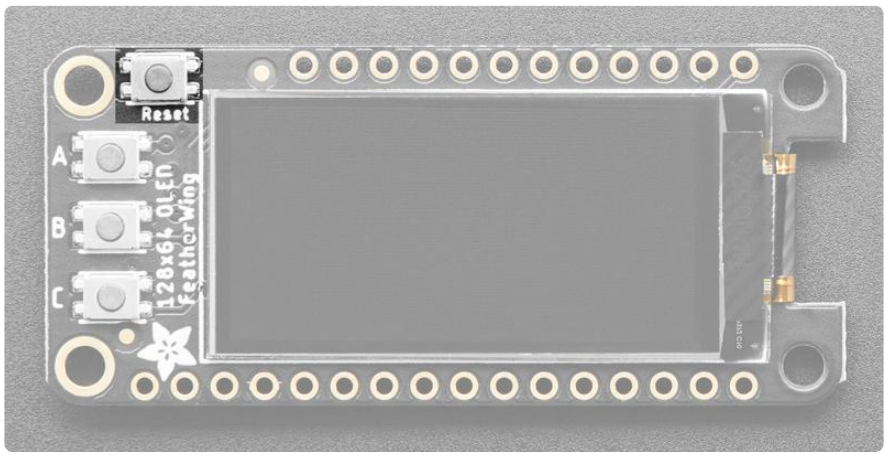
Button B has a 100K pullup on it so it will work with the ESP8266 (which does not have an internal pullup available on that pin). You will need to set up a pullup on all other pins for the buttons to work

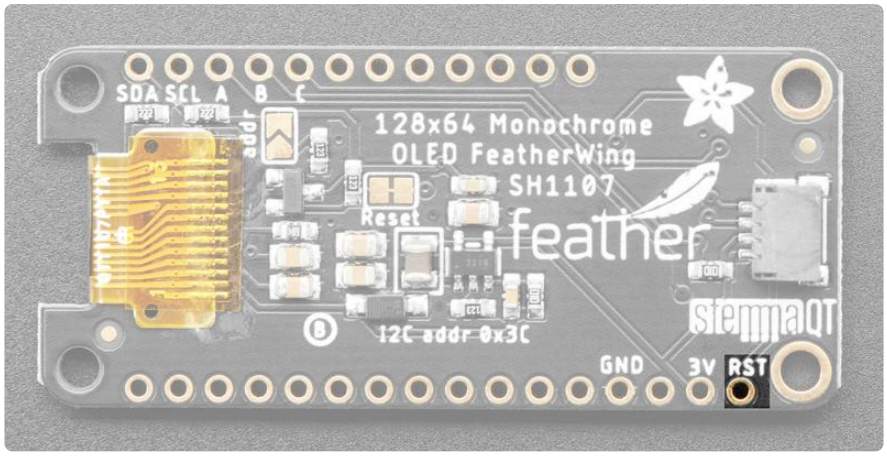
STEMMA QT



We've even included a SparkFun qwiic (https://adafru.it/Fpw)-compatible STEMMA QT (https://adafru.it/Ft4) connector for the I2C bus so you can plug and play any of our STEMMA QT, qwiic or Grove I2C (https://adafru.it/Ndk) sensors and devices! Check out the available sensors and devices on the Adafruit shop (https://adafru.it/HMF).

Reset Button



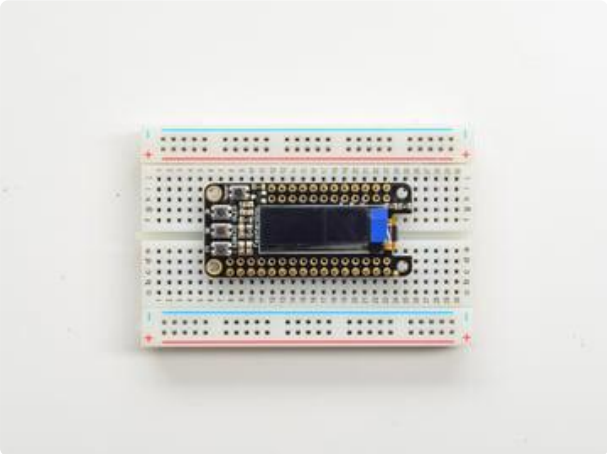


Sometimes its nice to be able to restart your program, so we also have a reset button. It is tied to the RST pin marked above.

Assembly

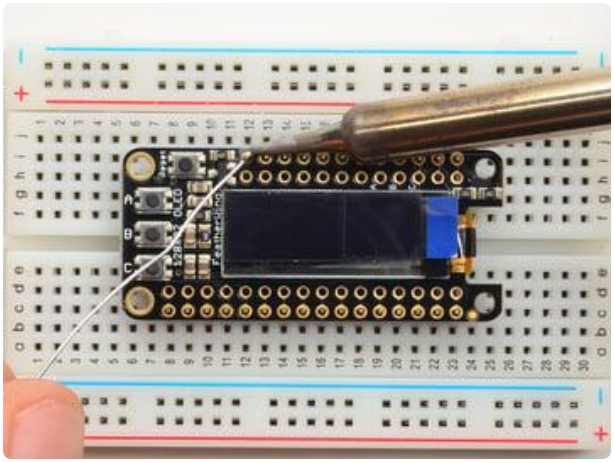
Prepare the header strip:

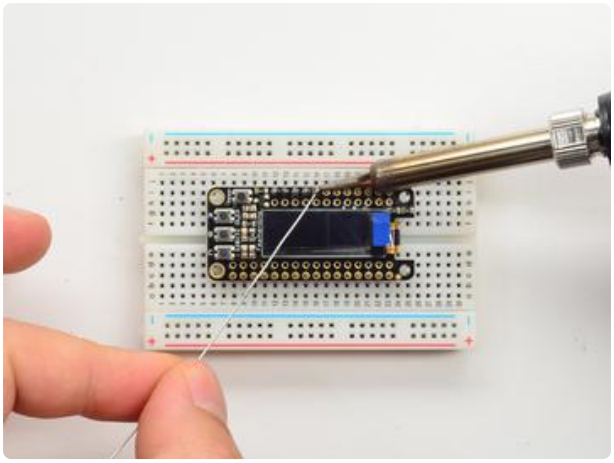
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down

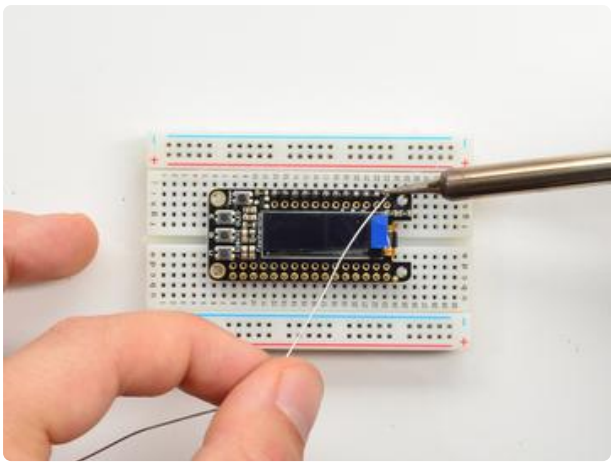


Add the FeatherWing:

Place the featherwing over the pins so that the short pins poke through the two rows of breakout pads





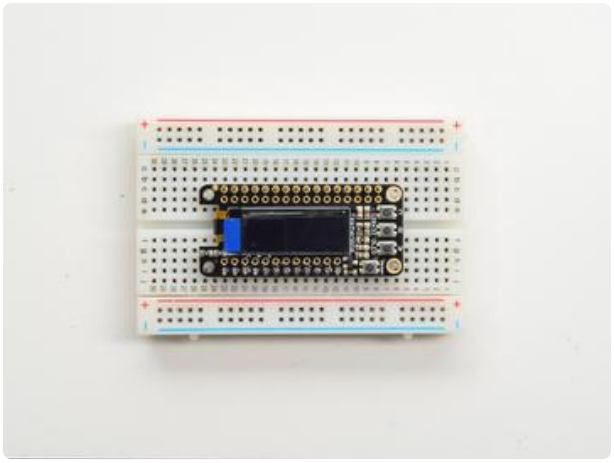


And Solder!

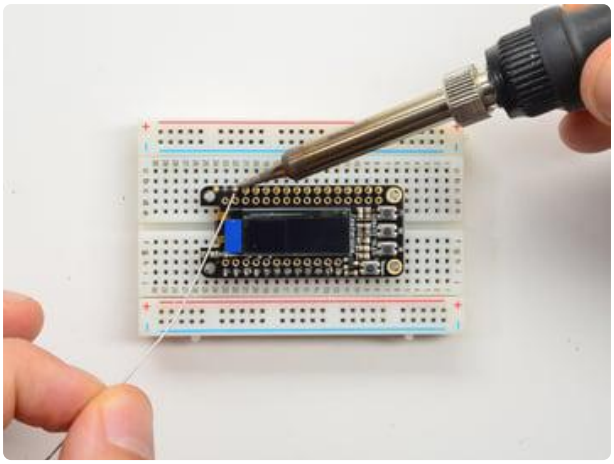
Be sure to solder all pins for reliable electrical contact.

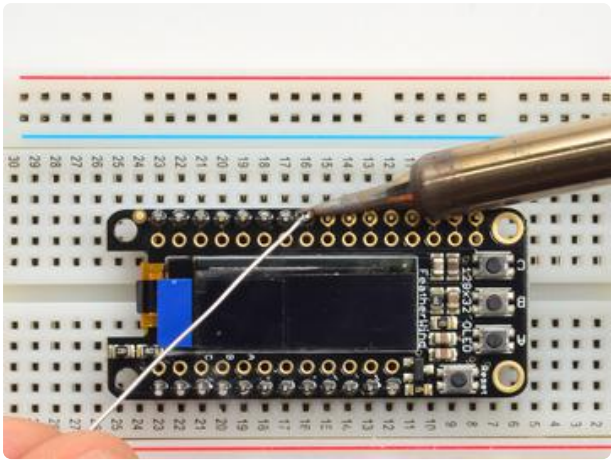
(For tips on soldering, be sure to check out our Guide to Excellent Soldering (https://adafru.it/aTk)).

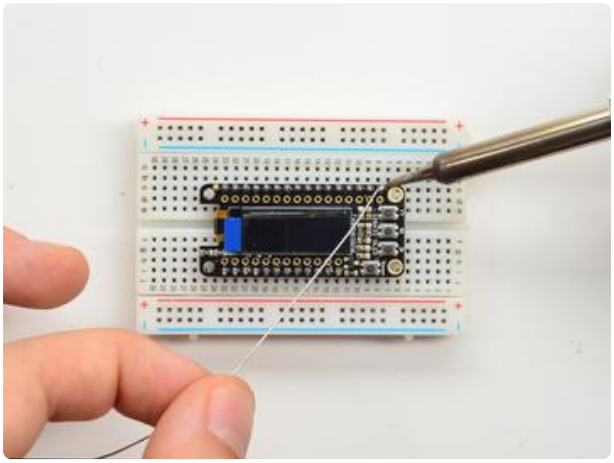
Start by soldering the first row of header

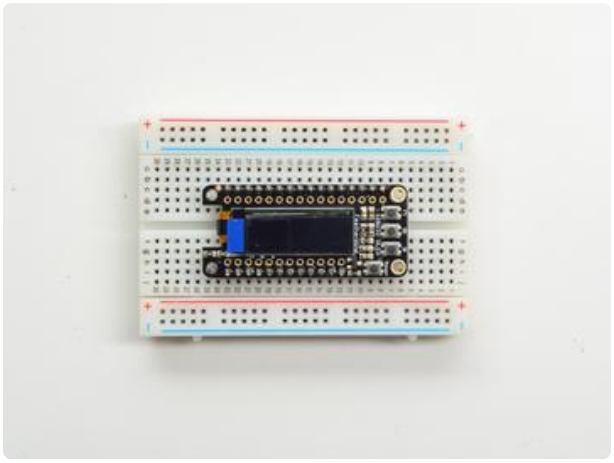


Now flip around and solder the other row completely



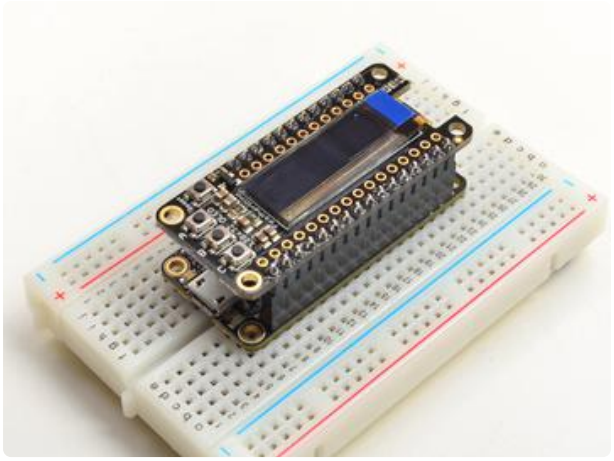






You're done with the two header strips.

Check your solder joints visually and continue onto the next steps



OK You're done! You can now plug your FeatherWing into your Feather and get your OLED on!

CircuitPython

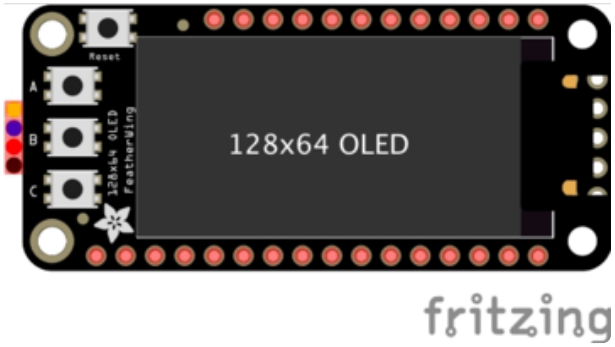
It's easy to use the Adafruit 128x64 OLED FeatherWing with CircuitPython and the Adafruit CircuitPython DisplayIO SH1107 (https://adafru.it/OaL) module. This module allows you to easily write CircuitPython code to control the display.

Not all CircuitPython builds include DisplayIO support, many SAMD21 and other 'small RAM/Flash' chips may not have support. Check the Support Matrix to verify https://circuitpython.readthedocs.io/en/latest/shared-bindings/support_matrix.html

CircuitPython Wiring

Connecting up the FeatherWing to a Feather is super simple!

- Solder the Feather with female headers on top or stacking headers.
- Attach the OLED FeatherWing using the stacking method.



CircuitPython Installation of DisplayIO SH1107 Library

To use the 128x64 OLED FeatherWing with your Adafruit CircuitPython Feather board you'll need to install the Adafruit CircuitPython DisplayIO SH1107 (https://adafru.it/OaL) module on your board.

you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (https://adafru.it/ENC). Our CircuitPython starter guide has a great page on how to install the library bundle (https://adafru.it/ABU).

Install the following libraries individually on your board:

- adafruit_displayio_sh1107
- adafruit_bus_device

To work through the code example below, you'll also need the following library:

- adafruit_display_text

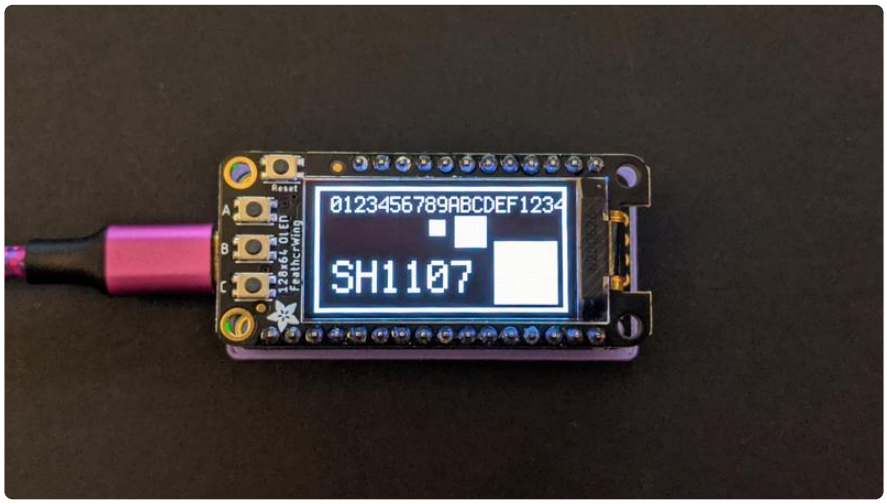
Before continuing make sure your board's lib folder or root filesystem has the adafruit_displayio_sh1107.mpy, adafruit_bus_device and adafruit_display_text files and folders copied over.

CircuitPython Usage

Save the following to your Feather as code.py:

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
"""
Author: Mark Roberts (mdroberts1243) from Adafruit code
This test will initialize the display using displayio and draw a solid white
background, a smaller black rectangle, miscellaneous stuff and some white text.

"""


import board
import displayio
import terminalio

# can try import bitmap_label below for alternative
from adafruit_display_text import label
import adafruit_displayio_sh1107

displayio.release_displays()
# oled_reset = board.D9

# Use for I2C
i2c = board.I2C()
display_bus = displayio.I2CDisplay(i2c, device_address=0x3C)

# SH1107 is vertically oriented 64x128
WIDTH = 128
HEIGHT = 64
BORDER = 2

display = adafruit_displayio_sh1107.SH1107(
    display_bus, width=WIDTH, height=HEIGHT, rotation=0
)
```

```
# Make the display context
splash = displayio.Group()
display.show(splash)

color_bitmap = displayio.Bitmap(WIDTH, HEIGHT, 1)
color_palette = displayio.Palette(1)
color_palette[0] = 0xFFFFFF  # White

bg_sprite = displayio.TileGrid(color_bitmap, pixel_shader=color_palette, x=0, y=0)
splash.append(bg_sprite)

# Draw a smaller inner rectangle in black
inner_bitmap = displayio.Bitmap(WIDTH - BORDER * 2, HEIGHT - BORDER * 2, 1)
inner_palette = displayio.Palette(1)
inner_palette[0] = 0x000000  # Black
inner_sprite = displayio.TileGrid(
    inner_bitmap, pixel_shader=inner_palette, x=BORDER, y=BORDER
)
splash.append(inner_sprite)

# Draw some white squares
sm_bitmap = displayio.Bitmap(8, 8, 1)
sm_square = displayio.TileGrid(sm_bitmap, pixel_shader=color_palette, x=58, y=17)
splash.append(sm_square)

med_bitmap = displayio.Bitmap(16, 16, 1)
med_square = displayio.TileGrid(med_bitmap, pixel_shader=color_palette, x=71, y=15)
splash.append(med_square)

lrg_bitmap = displayio.Bitmap(32, 32, 1)
lrg_square = displayio.TileGrid(lrg_bitmap, pixel_shader=color_palette, x=91, y=28)
splash.append(lrg_square)

# Draw some label text
text1 = "0123456789ABCDEF123456789AB"  # overly long to see where it clips
text_area = label.Label(terminalio.FONT, text=text1, color=0xFFFFFF, x=8, y=8)
splash.append(text_area)
text2 = "SH1107"
text_area2 = label.Label(
    terminalio.FONT, text=text2, scale=2, color=0xFFFFFF, x=9, y=44
)
splash.append(text_area2)

while True:
    pass
```



That's all there is to using CircuitPython with your 128x64 OLED FeatherWing!

Python Docs

Python Docs (https://adafru.it/NFU)

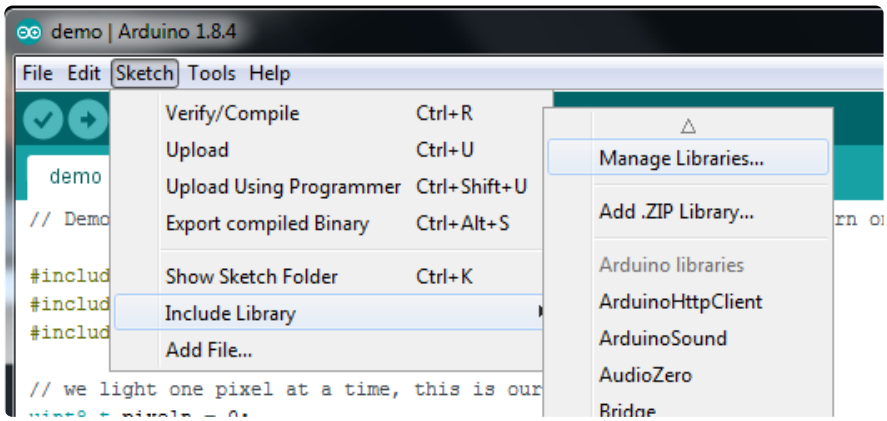
Arduino Code

The OLED display we use is well supported and works for all Feathers, all you need is a little library support and you will be drawing in no time!

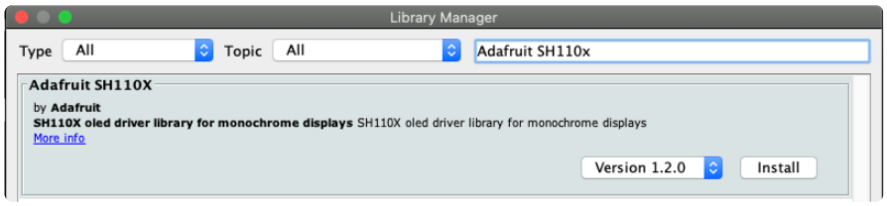
Install Arduino Libraries

Using the OLED FeatherWing with Arduino sketches requires that two libraries be installed: Adafruit_SH110x, which handles the low-level communication with the hardware, and Adafruit_GFX, which builds atop this to add graphics functions like lines, circles and text.

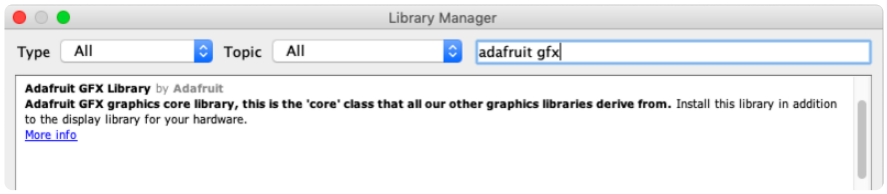
Open up the library manager:



Search for the Adafruit SH110x library and install it



Search for the Adafruit GFX library and install it



If using an earlier version of the Arduino IDE (prior to 1.8.10), also locate and install Adafruit_BusIO (newer versions will install this dependency automatically).

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

Run Example Code

We have a basic demo that works with all Feathers, so compile/upload this sketch:

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>

Adafruit_SH1107 display = Adafruit_SH1107(64, 128, &Wire);

// OLED FeatherWing buttons map to different pins depending on board:
#if defined(ESP8266)
  #define BUTTON_A  0
  #define BUTTON_B 16
  #define BUTTON_C  2
#elif defined(ESP32) && !defined(ARDUINO_ADAFRUIT_FEATHER_ESP32S2)
  #define BUTTON_A 15
  #define BUTTON_B 32
  #define BUTTON_C 14
#elif defined(ARDUINO_STM32_FEATHER)
  #define BUTTON_A PA15
  #define BUTTON_B PC7
  #define BUTTON_C PC5
#elif defined(TEENSYDUINO)
  #define BUTTON_A  4
  #define BUTTON_B  3
  #define BUTTON_C  8
#elif defined(ARDUINO_NRF52832_FEATHER)
  #define BUTTON_A 31
  #define BUTTON_B 30
  #define BUTTON_C 27
#else // 32u4, M0, M4, nrf52840, esp32-s2 and 328p
```

```
  #define BUTTON_A  9
  #define BUTTON_B  6
  #define BUTTON_C  5
#endif

void setup() {
  Serial.begin(115200);

  Serial.println("128x64 OLED FeatherWing test");
  display.begin(0x3C, true); // Address 0x3C default

  Serial.println("OLED begun");

  // Show image buffer on the display hardware.
  // Since the buffer is intialized with an Adafruit splashscreen
  // internally, this will display the splashscreen.
  display.display();
  delay(1000);

  // Clear the buffer.
  display.clearDisplay();
  display.display();

  display.setRotation(1);
  Serial.println("Button test");

  pinMode(BUTTON_A, INPUT_PULLUP);
  pinMode(BUTTON_B, INPUT_PULLUP);
  pinMode(BUTTON_C, INPUT_PULLUP);

  // text display tests
  display.setTextSize(1);
  display.setTextColor(SH110X_WHITE);
  display.setCursor(0,0);
  display.print("Connecting to SSID\n'adafruit':");
  display.print("connected!");
  display.println("IP: 10.0.1.23");
  display.println("Sending val #0");
  display.display(); // actually display all of the above
}

void loop() {
  if(!digitalRead(BUTTON_A)) display.print("A");
  if(!digitalRead(BUTTON_B)) display.print("B");
  if(!digitalRead(BUTTON_C)) display.print("C");
  delay(10);
  yield();
  display.display();
}
```

You should see the OLED display a splash screen then spit out some text (it's a make-believe WiFi connection status screen...this doesn't actually do anything, just showing how typical project might look). If you press the A B or C buttons it will also print those out.

Do more!

You can use any of the Adafruit GFX library commands to draw onto your OLED, that means that you get all sorts of shapes, fonts, lines, etc available. Check out GFX for all the underlying graphics support functions and how they work (https://adafru.it/doL)

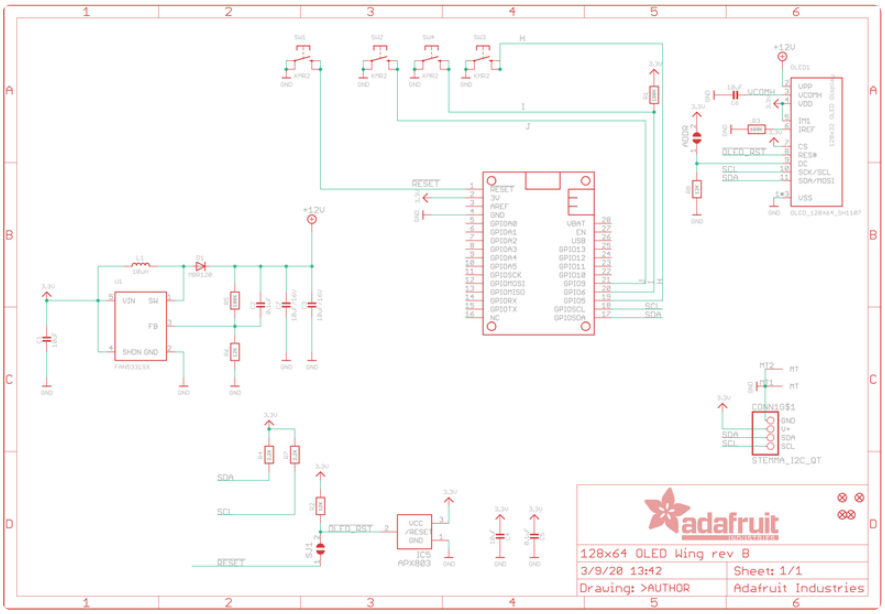
Remember you need to call display() after drawing to refresh the screen!

Downloads

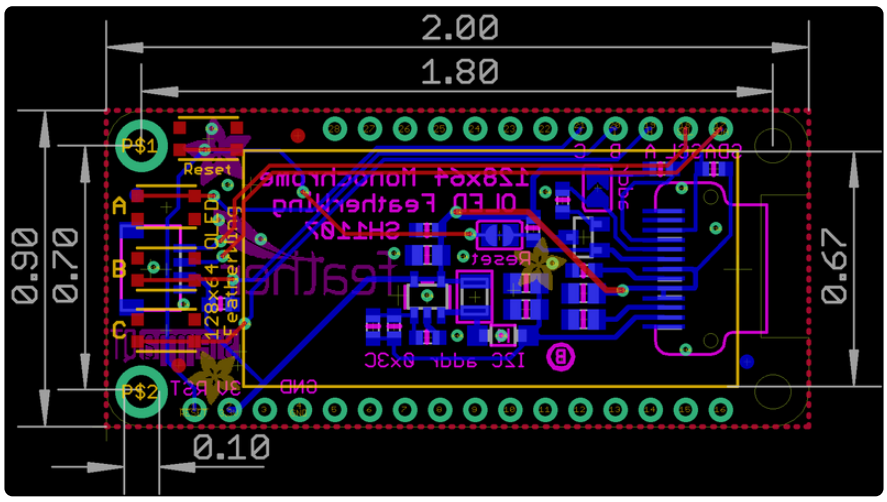
Files

- SH1107 Datasheet (https://adafru.it/Ndp)
- PCB Files in EagleCAD format (https://adafru.it/nbc)
- Fritzing object available in the Adafruit Fritzing Library (https://adafru.it/Ndq)

Schematic



Fab Print



X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Display Development Tools ***category:***

Click to view products by Adafruit ***manufacturer:***

Other Similar products are found below :

KIT 60121-3 S5U13U11P00C100 MAX14521EEVKIT KIT 60145-3 S5U13748P00C100 DFR0413 3248 DLPLCR90EVM MAX20069EVKIT# KIT95000-3 LCD-16396 PIM370 1109 MCIMX-LVDS1 MIKROE-2449 MIKROE-2453 131 DEV-13628 1590 MIKROE-2269 1601 1770 1947 1983 1987 2050 2218 2219 2260 2345 2418 2423 2454 2455 2478 2674 SK-220RD-PI FIT0477 333 1774 334 TE-M321-SDK DFR0428 cs-epapersk-03 338 DEV-14442 FIT0478 cs-paperino-01 OM-E-OLE ALTHSMCMIPILCD