invention bootcamp 2022

Using an external OLED display

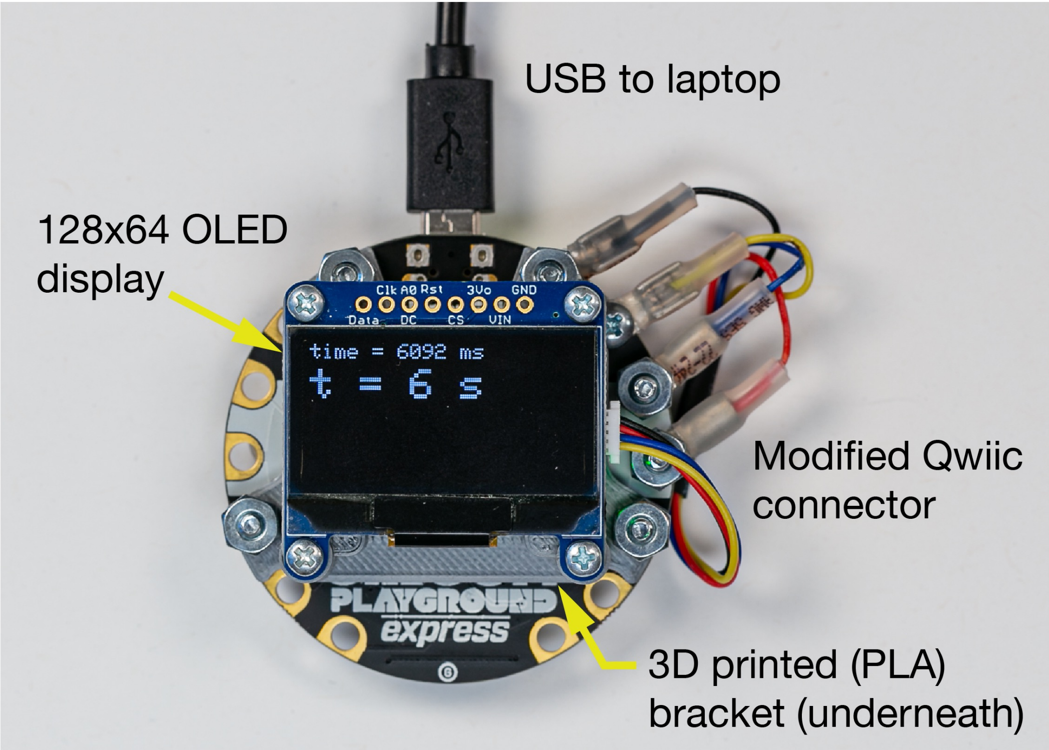
Learning Objectives

These slides should help you to

- Connect the external 128x64 OLED display to your Circuit Playground Express
- Install the Adafruit Graphics Libraries necessary to use the 128x64 OLED
- Display static text and dynamic numerical values on the 128x64 OLED
- Use functions to organize the setup and updating of the display

Hardware connections

Final configuration of hardware



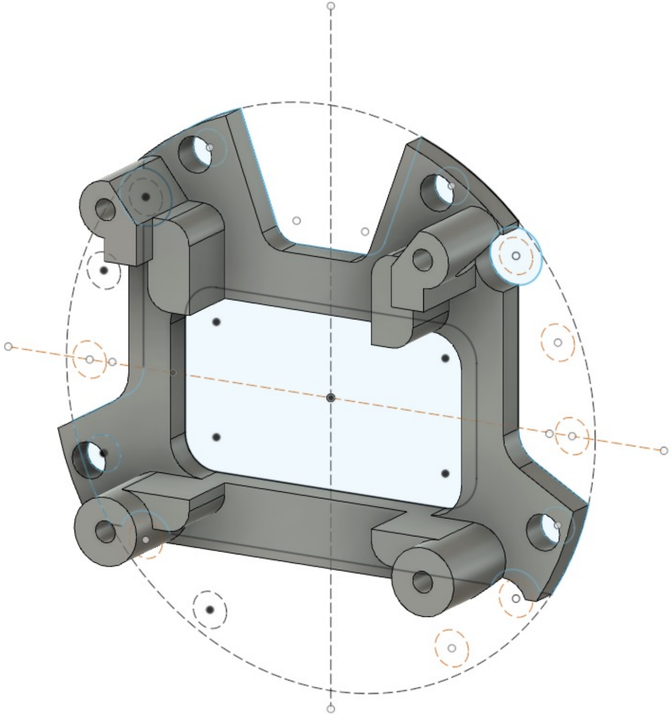
USB to laptop

128x64 OLED display

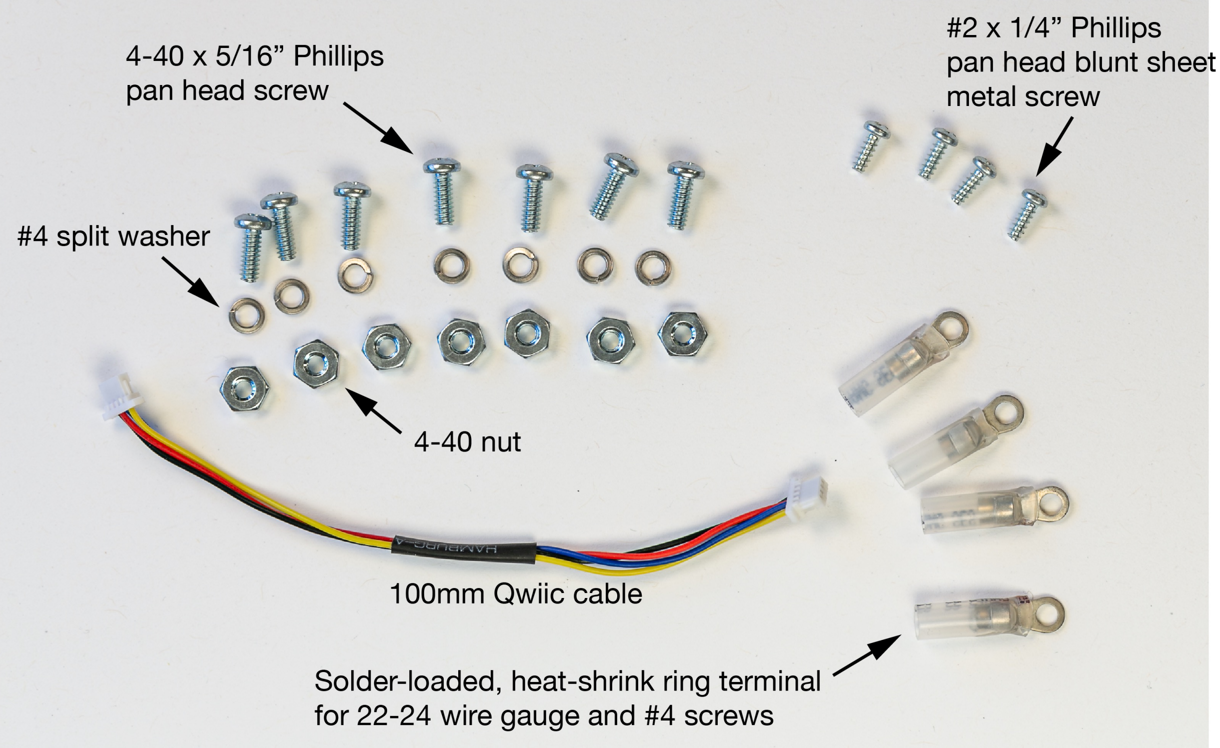
Modified Qwiic connector

3D printed (PLA) bracket (underneath)

3D-printed bracket



Other hardware components



4-40 x 5/16” Phillips
pan head screw

#2 x 1/4” Phillips
pan head blunt sheet
metal screw

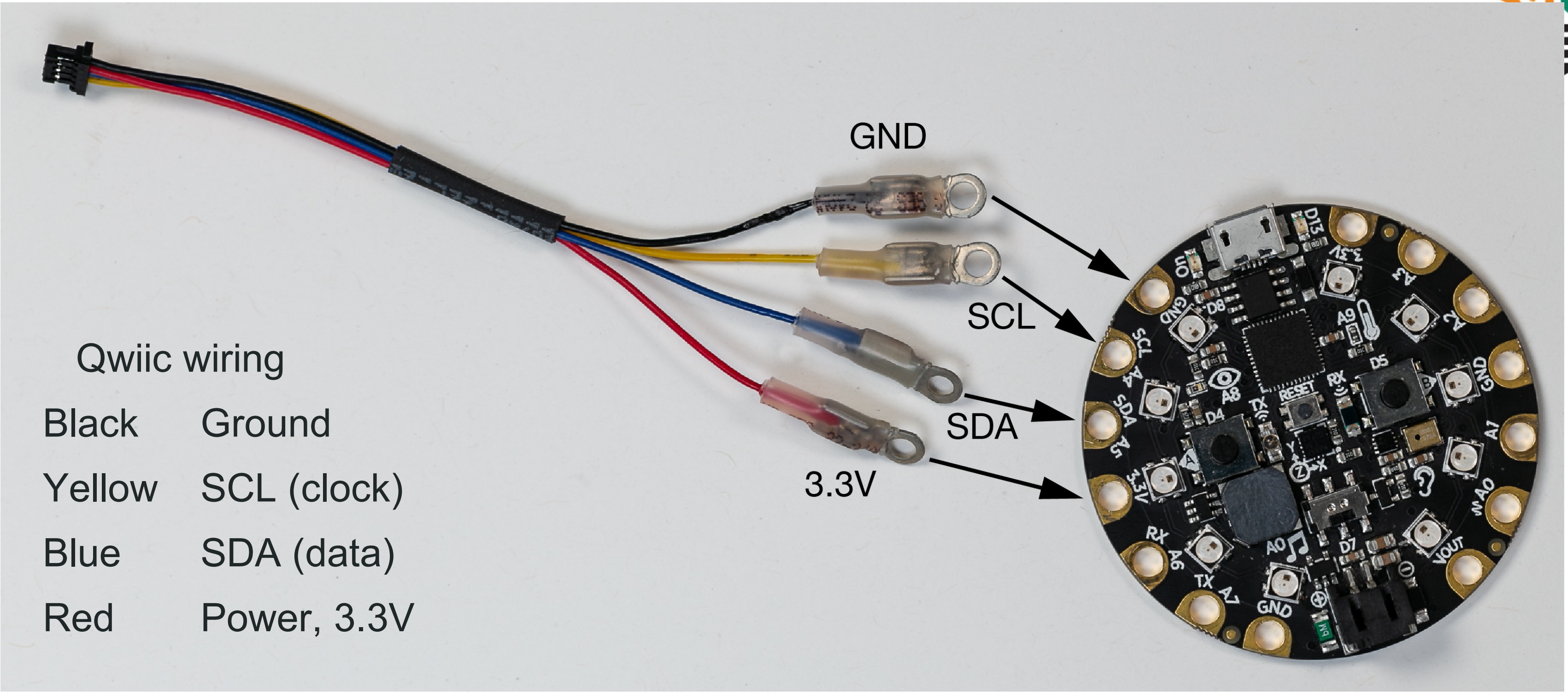
#4 split washer

4-40 nut

100mm Qwiic cable

Solder-loaded, heat-shrink ring terminal
for 22-24 wire gauge and #4 screws

Aligning the Qwiic cable to the CPX pads

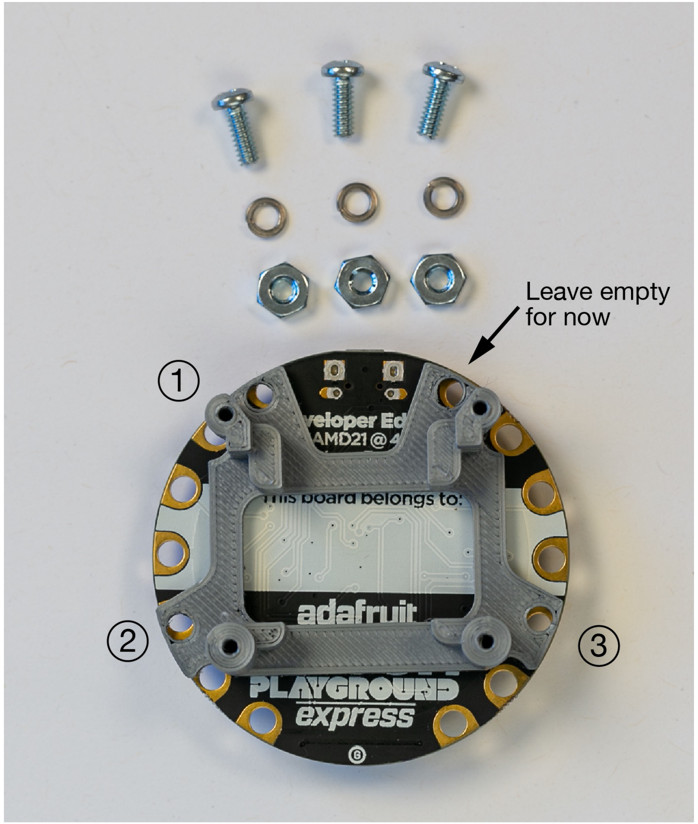


Qwiic wiring

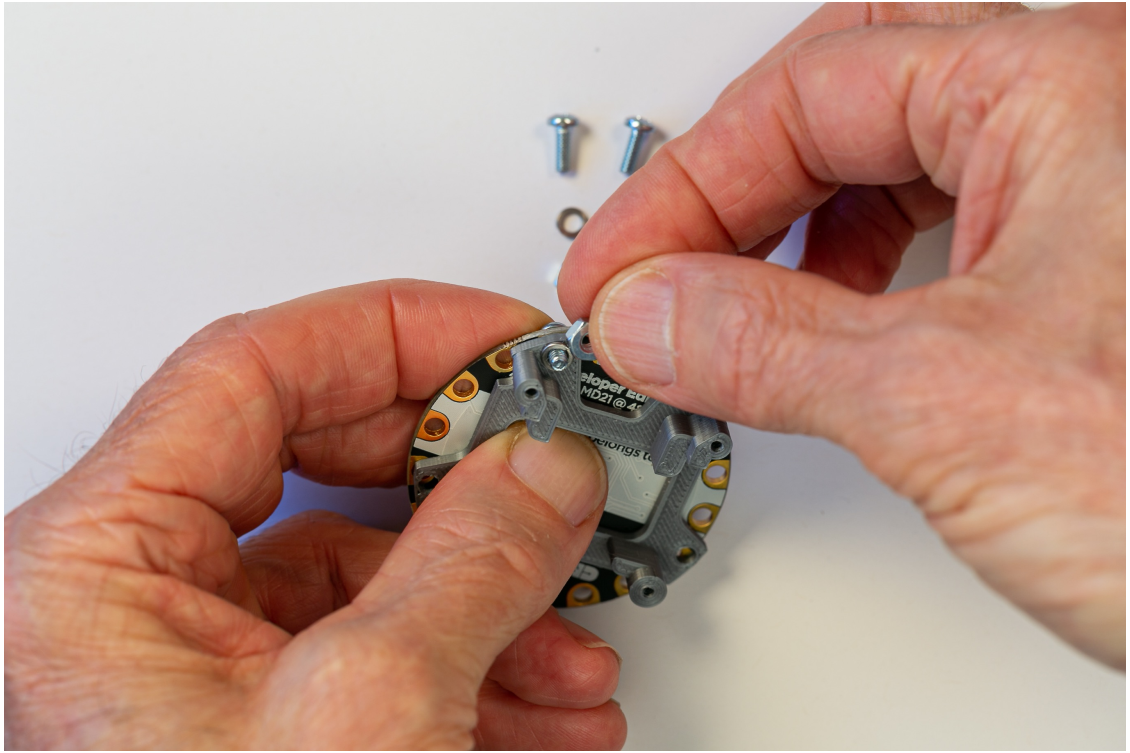
Black	Ground
Yellow	SCL (clock)
Blue	SDA (data)
Red	Power, 3.3V

Begin assembly

Use 3 screws to attach the bracket to the CPX



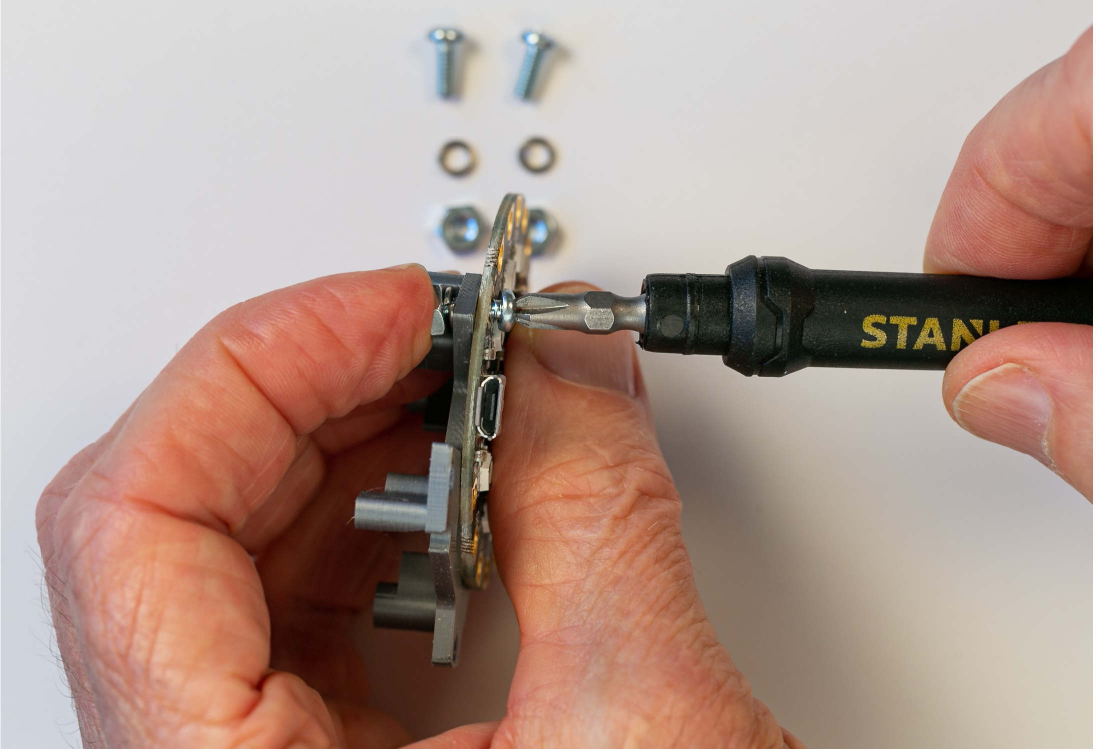
Place nut on the bracket side



Insert a 4-40 screw through the hole of the 3.3V pad next to the USB socket.

Place the washer on the screw and the nut on the screw – the split washer should be located under the nut

Place nut on the bracket side

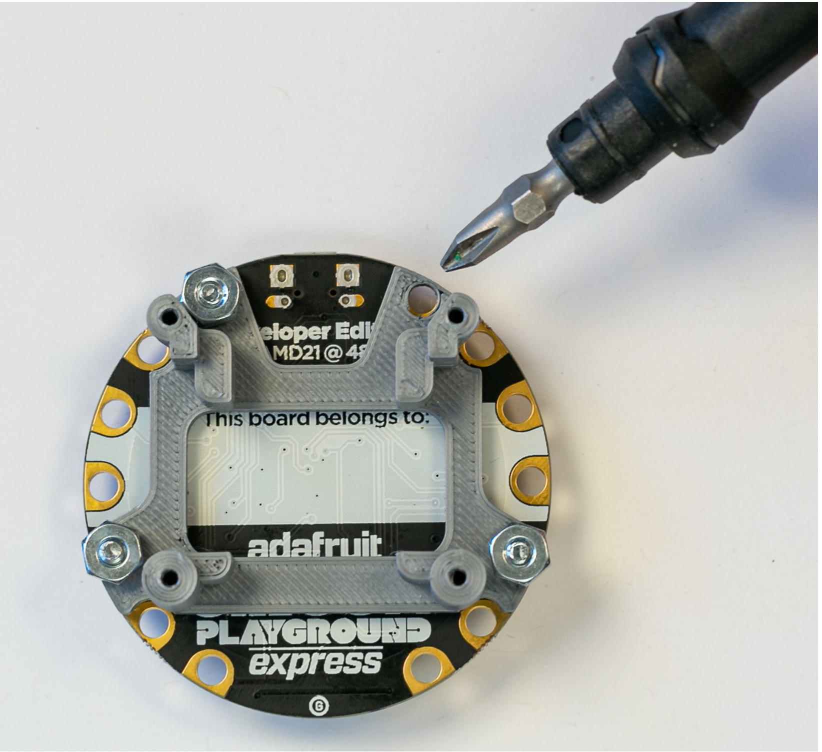


Use a Phillips head screwdriver to tighten the screw.

Tighten just enough to keep the nut from coming loose.

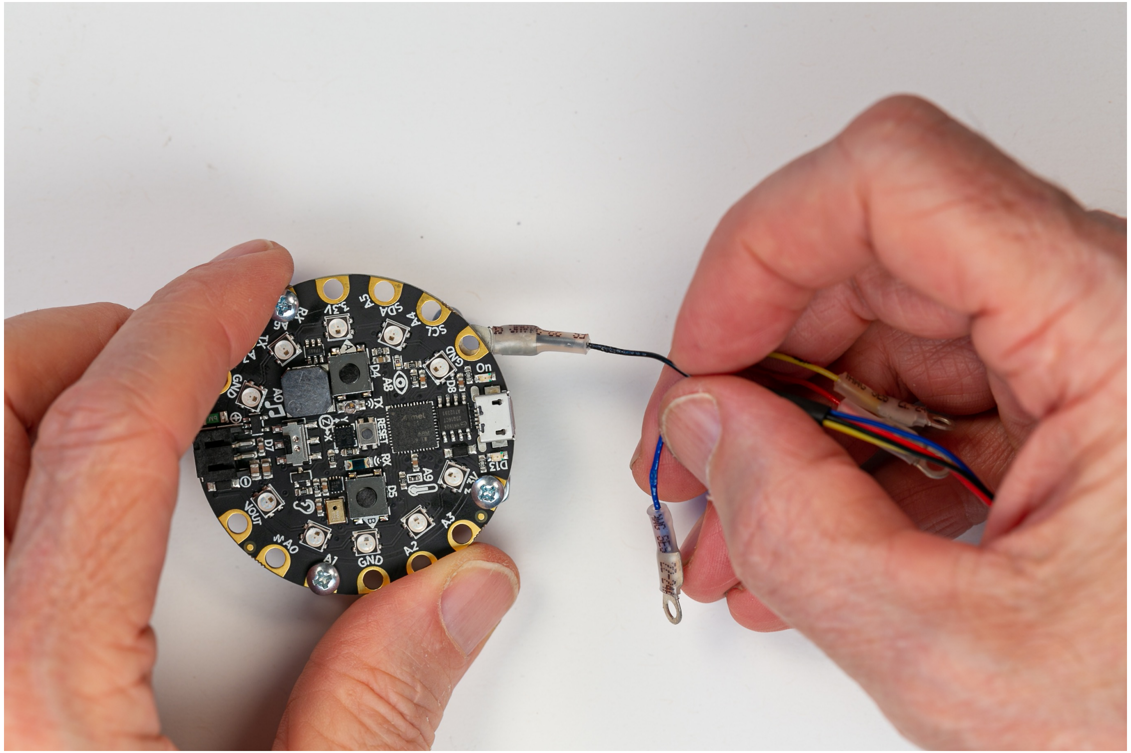
Initially you want to allow the bracket to move so that other screws can be inserted

Three of the four screws should be installed



The hole for the GND pad next to the USB socket should be empty

Attach the black cable

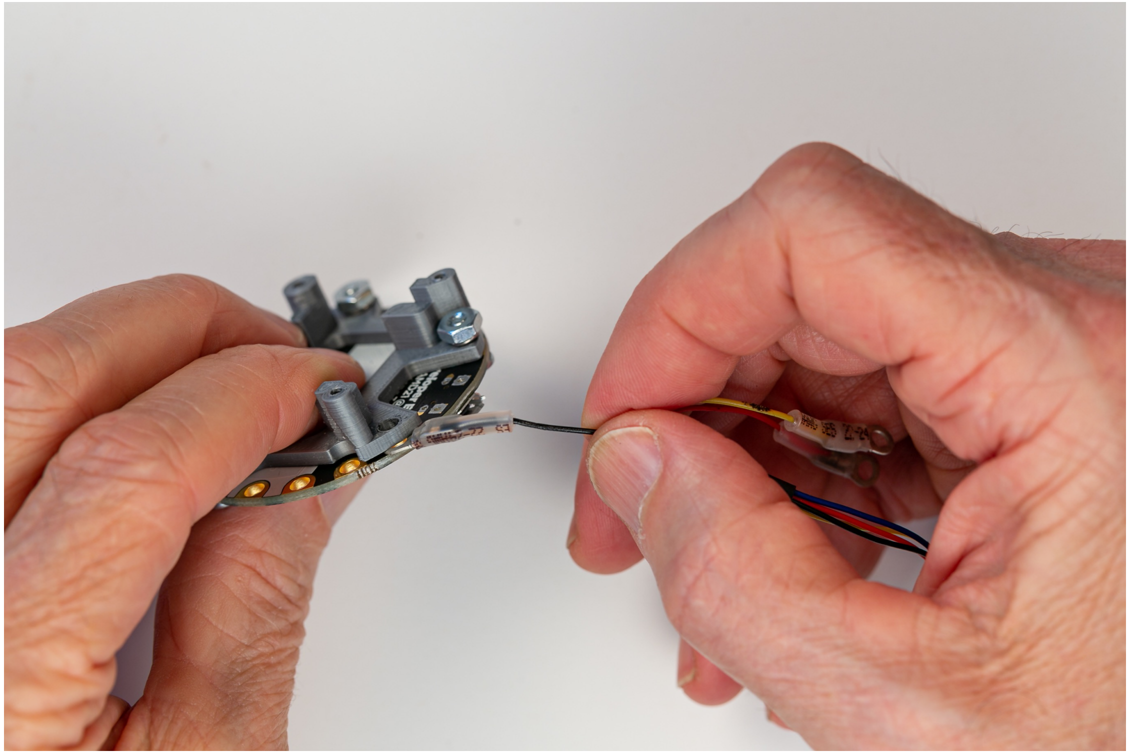


Isolate the terminal of the black wire on the Qwiic cable.

The flat end of that terminal will be inserted between the bracket and the CPX

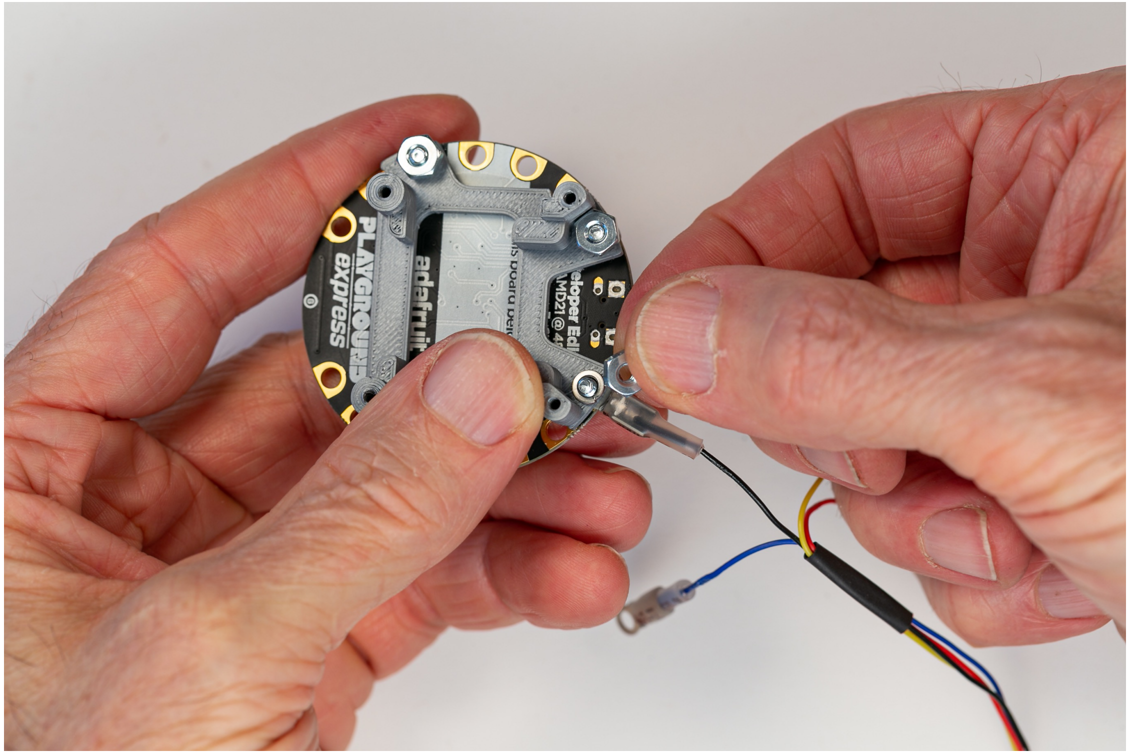
You may need to slightly loosen the three screws that you have already installed

Attach the black cable



Slide the flat part of the terminal between the CPX and the bracket

Attach the black cable

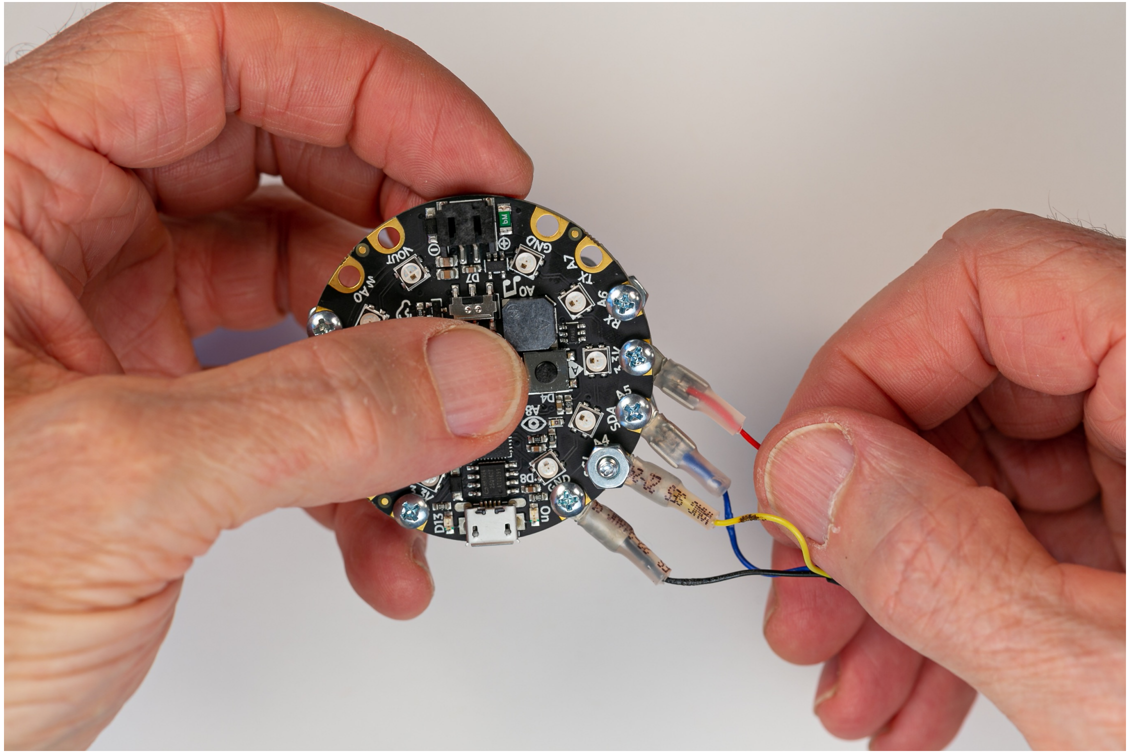


Insert a 4-40 screw through the hole aligned with the pad, the round terminal and the bracket.

Add a split washer and nut. Tighten the screw.

Now tighten all four screws that hold the bracket to the CPX

Attach the yellow, blue and red cables

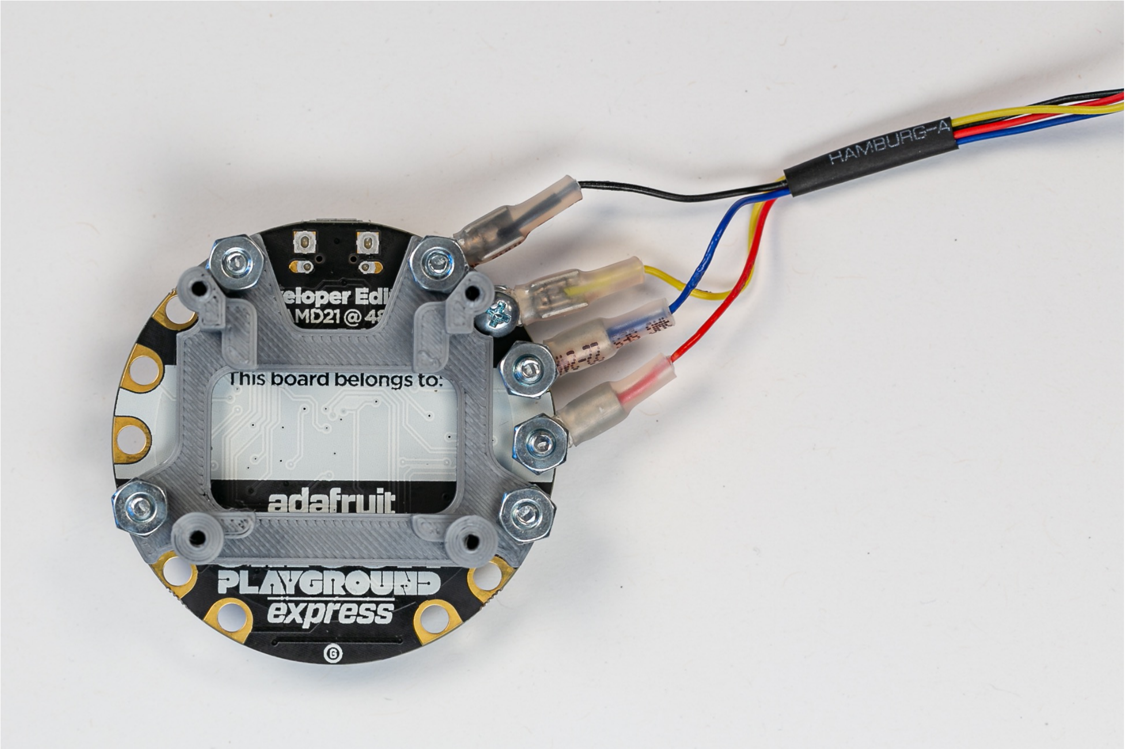


Qwiic wiring

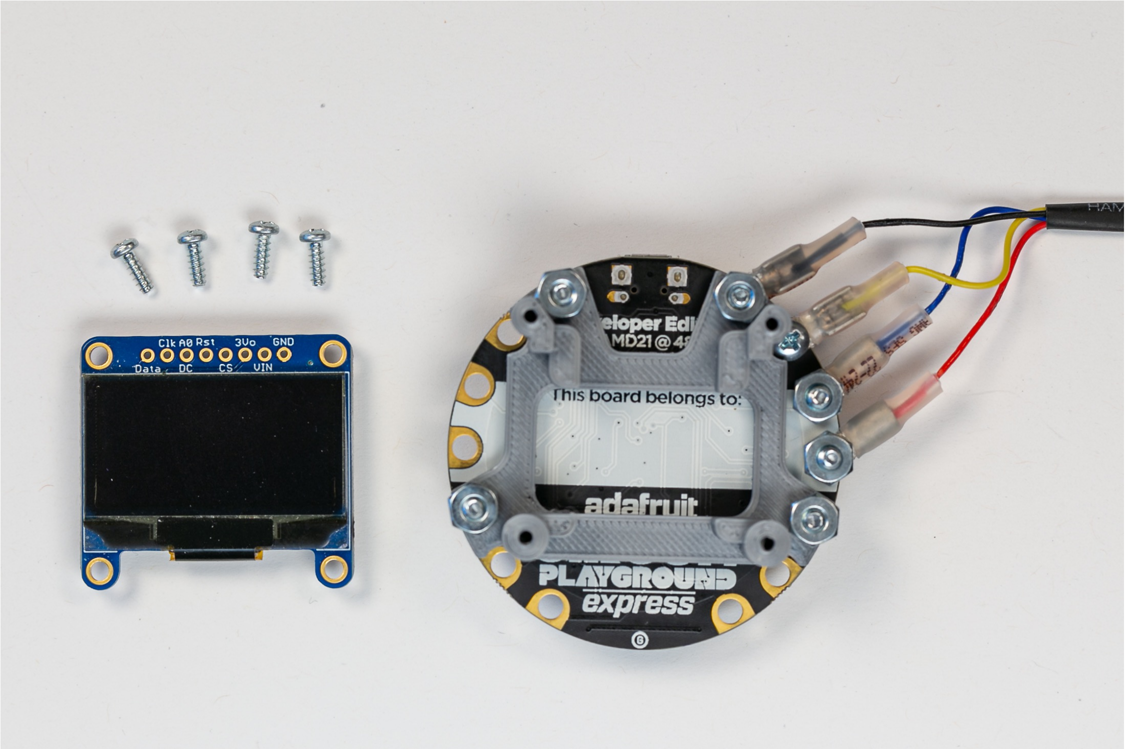
Black	Ground
Yellow	SCL (clock)
Blue	SDA (data)
Red	Power, 3.3V

Note that the screw for the yellow wire will need to be oriented in the opposite direction to the screws holding the other terminals

Completed attachment of Qwiic cable



Locate the OLED and 4 #2 sheet metal screws

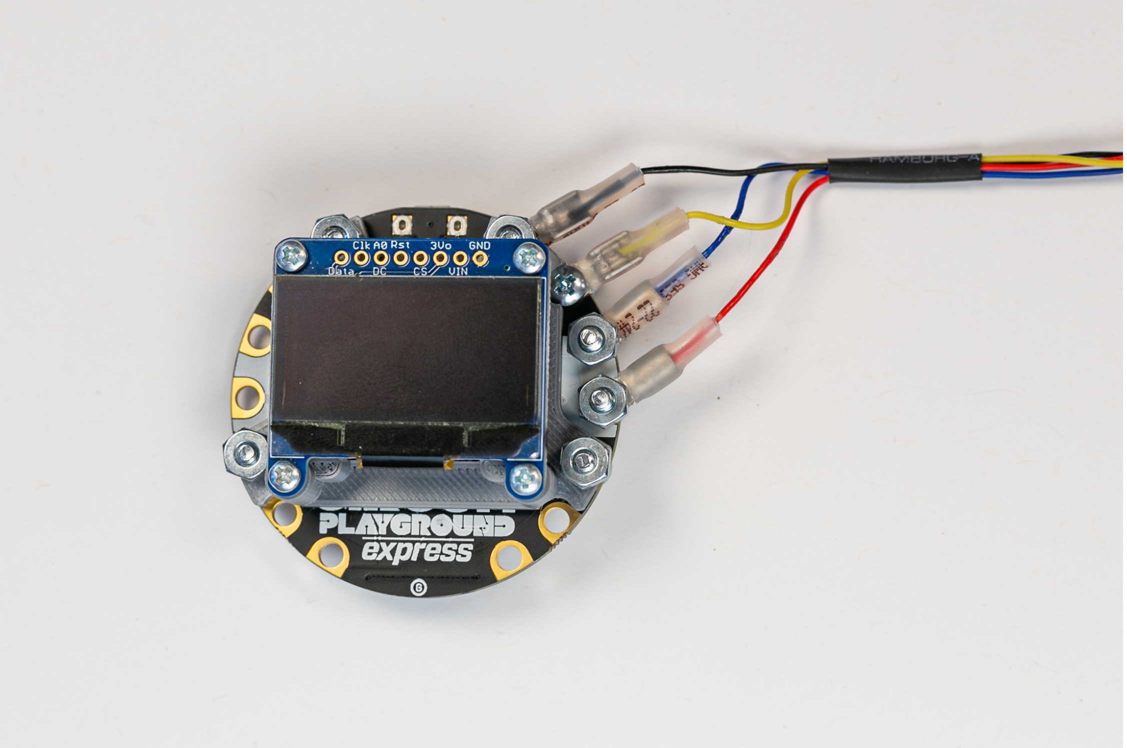


Locate the OLED and the four #2 sheet metal screws in your kit.

Attach the OLED to the bracket with the screws.

Be careful not to over-tighten the screws. The screws should be snug, and the OLED should now wobble relative to the bracket.

Completed attachment of the OLED



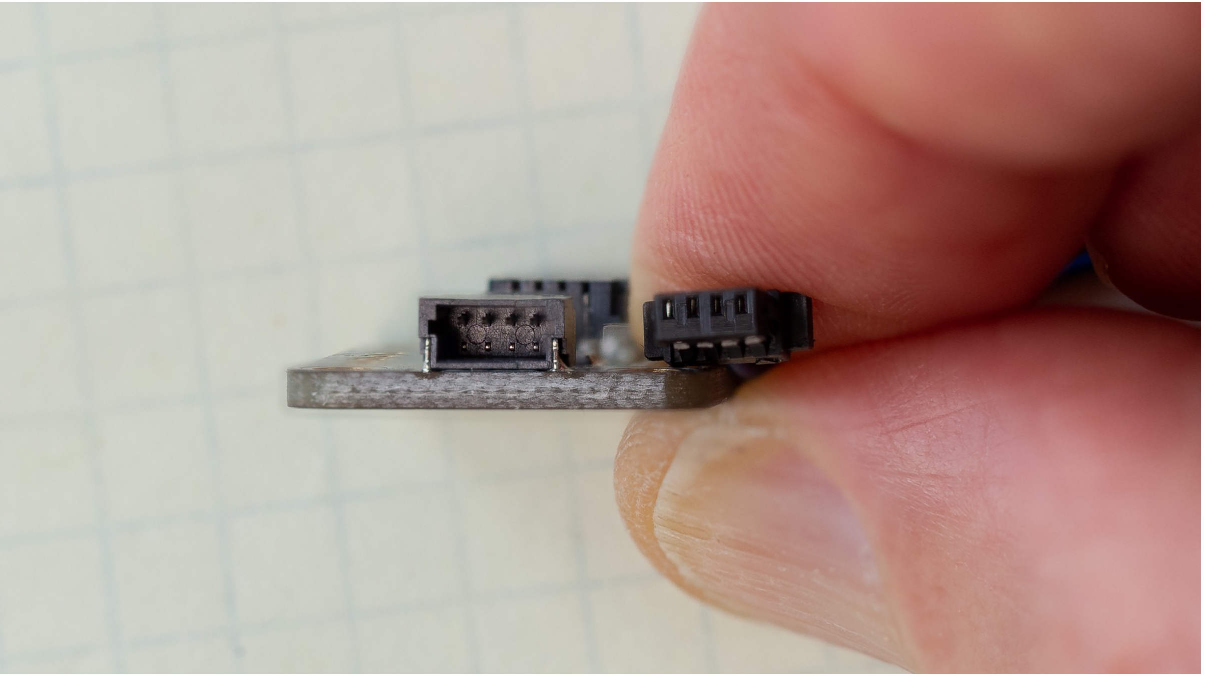
Next, we'll attach the Qwiic cable to the OLED.

Please be careful with the next step.

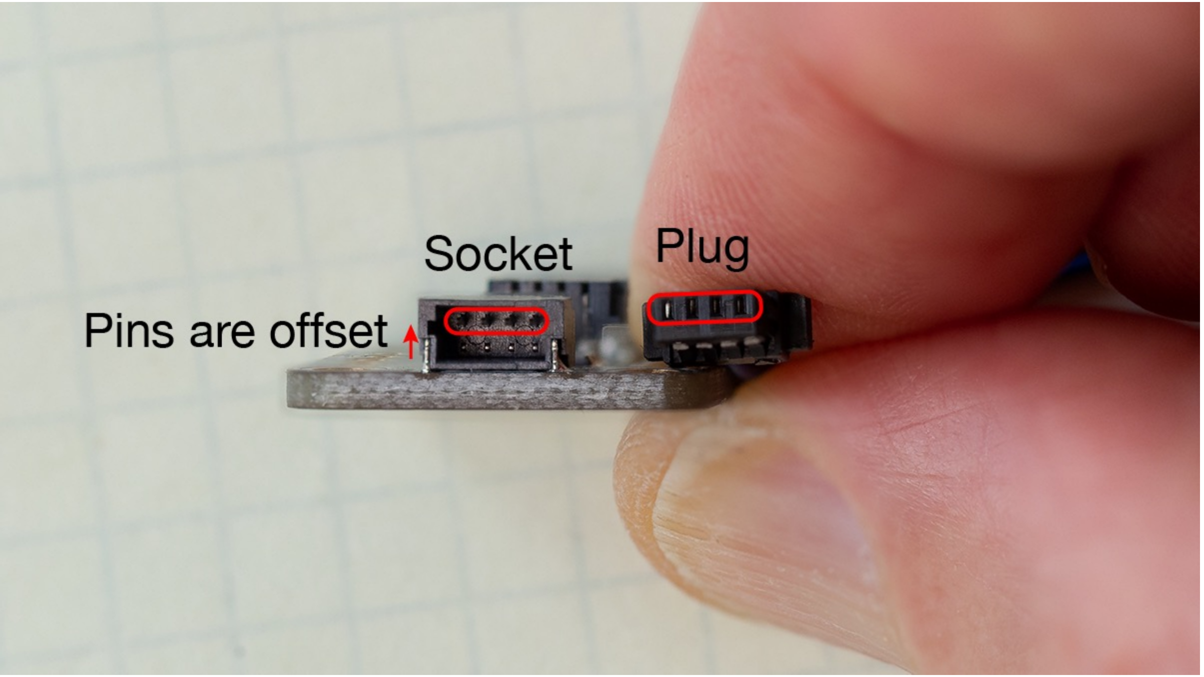
You may need to rotate the terminals to allow the free end of the Qwiic cable to reach the socket on the OLED.

Be careful when inserting the Qwiic cable into the socket on the OLED

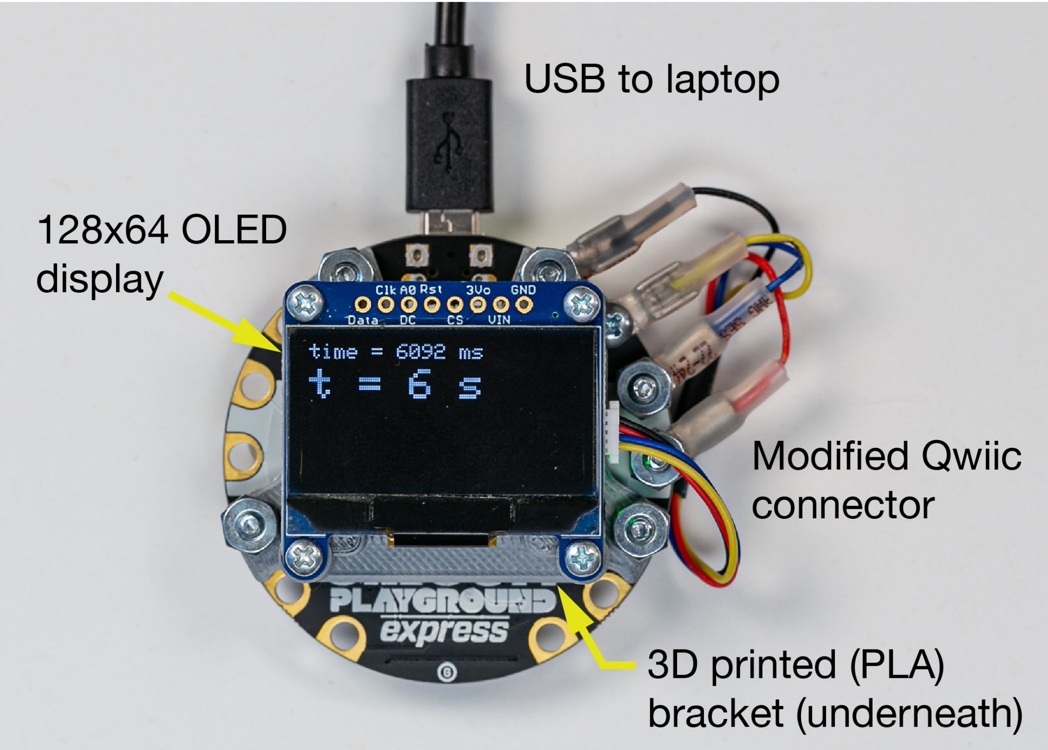
Pins and sockets of Qwiic connectors have an offset alignment



Pins and sockets of Qwiic connectors have an offset alignment



Final configuration of hardware



USB to laptop

128x64 OLED display

Modified Qwiic connector

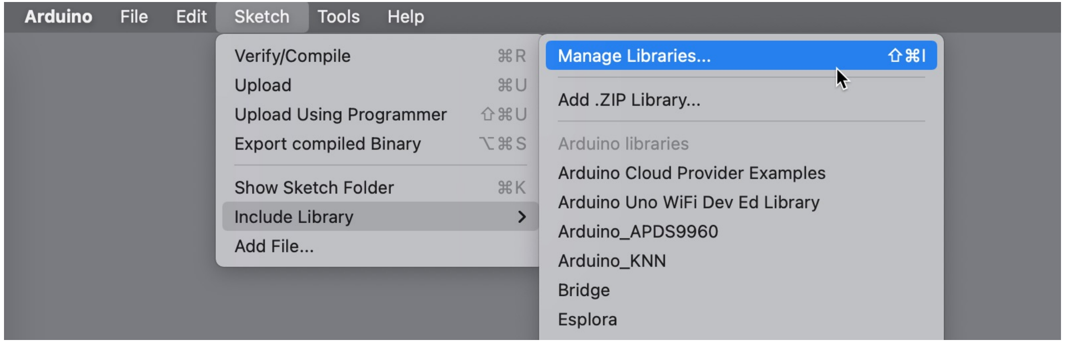
3D printed (PLA) bracket (underneath)

Install graphics libraries

Install the Adafruit SSD1306 Library

From the IDE menus …

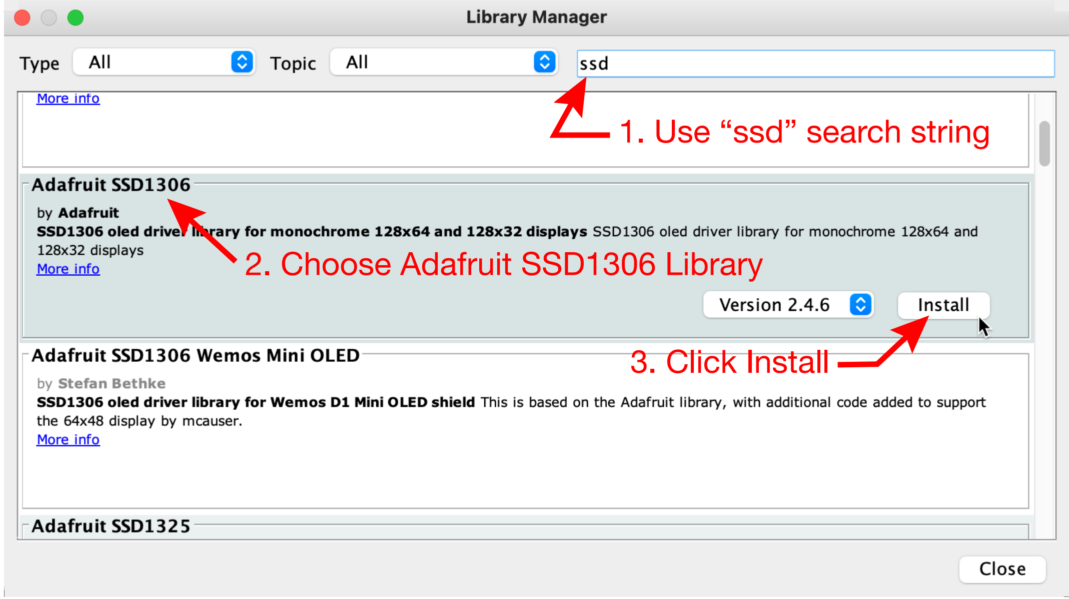
- Select “Sketch” → “Include Library” → “Manage Libraries …”



Install the Adafruit SSD1306 Library

From the IDE menus …

- Select "Sketch" → "Include Library" → "Manage Libraries …"
- Enter "ssd" in the search box
- Choose Adafruit SSD1306
- Click "Install"



ssd

1. Use "ssd" search string

More info

Adafruit SSD1306

by Adafruit

SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled driver library for monochrome 128x64 and 128x32 displays

More info

2. Choose Adafruit SSD1306 Library

3. Click Install

Adafruit SSD1306 Wemos Mini OLED

by **Stefan Bethke**

SSD1306 oled driver library for Wemos D1 Mini OLED shield This is based on the Adafruit library, with additional code added to support the 64x48 display by mcauser.

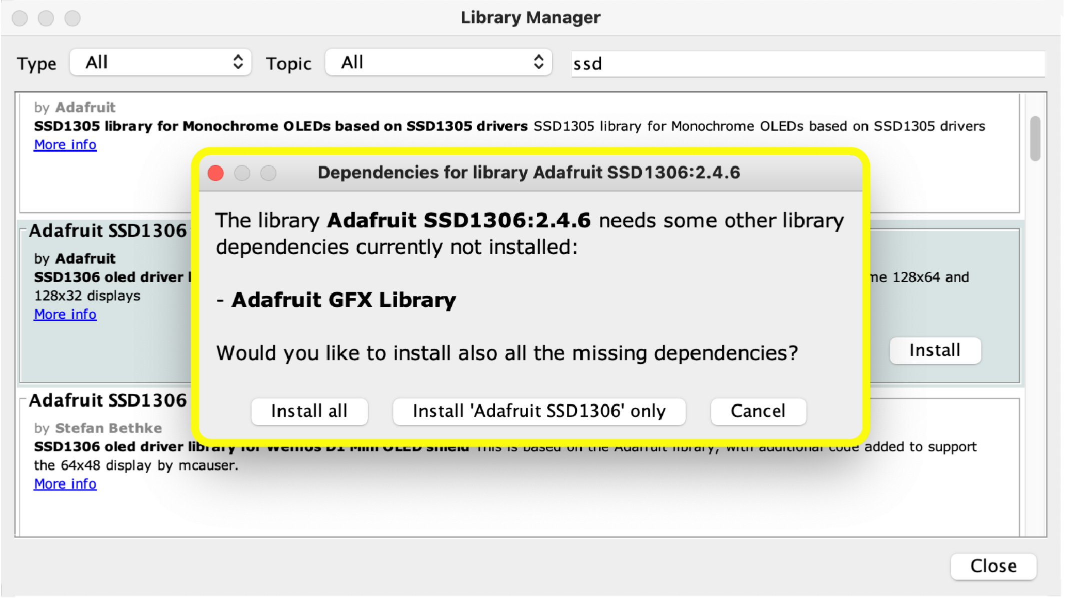
More info

Adafruit SSD1325

Install the Adafruit SSD1306 Library

From the IDE menus …

- Select “Sketch” → “Include Library” → “Manage Libraries …”
- Enter “ssd” in the search box
- Choose Adafruit SSD1306
- Click “Install”
- Click “Install all” to add GFX



Type All Topic All ssd

by **Adafruit**
SSD1305 library for Monochrome OLEDs based on SSD1305 drivers SSD1305 library for Monochrome OLEDs based on SSD1305 drivers

Adafruit SSD1306
by **Adafruit**
SSD1306 oled driver l …me 128x64 and
128x32 displays

Adafruit SSD1306
by **Stefan Bethke**
SSD1306 oled driver library for Wemos D1 Mini OLED shield This is based on the Adafruit library, with additional code added to support the 64x48 display by mcauser.

Dependencies for library Adafruit SSD1306:2.4.6

The library **Adafruit SSD1306:2.4.6** needs some other library dependencies currently not installed:

- **Adafruit GFX Library**

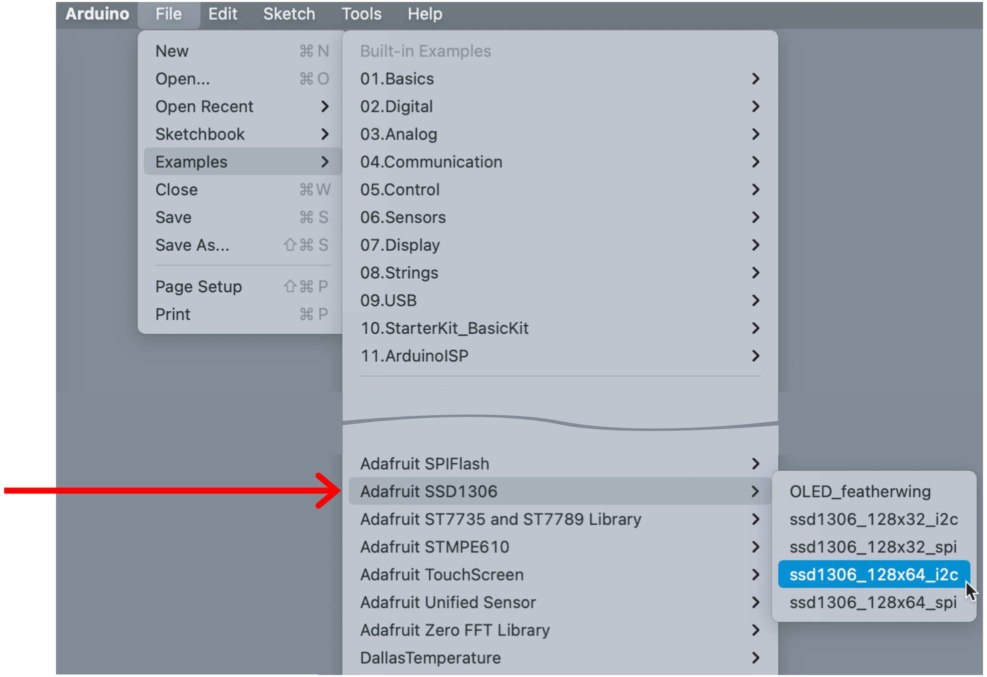
Would you like to install also all the missing dependencies?

Test wiring and library with the demo code from the Adafruit library

Load ssd1306_128x64_i2c from examples

Select “File” → “Examples” → “Adafruit SSD1306” → “ssd1306_128x64_i2c”

Upload the sketch



Adafruit SSD1306 appears after you add the library

128x64 is resolution

i2c is communication protocol

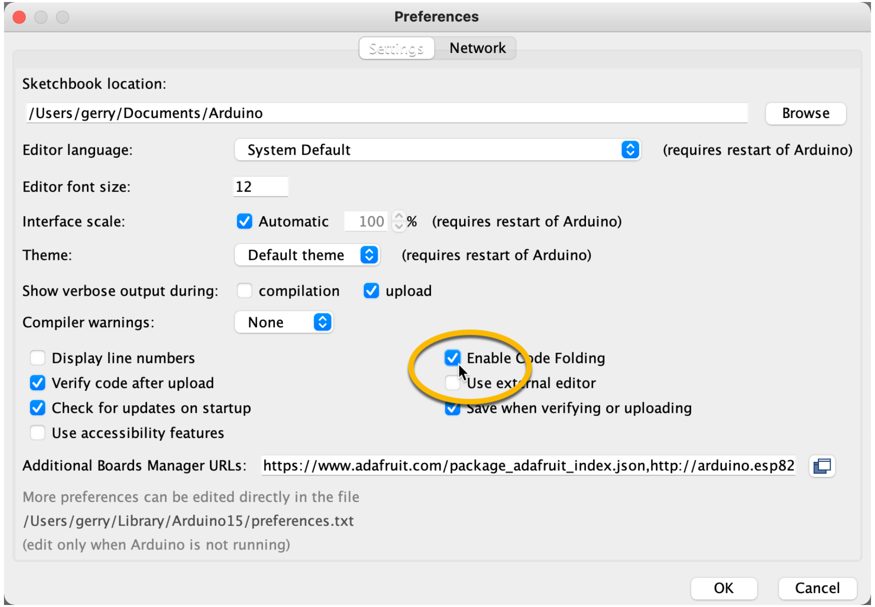
Demonstrate static text and dynamic values with the OLEDdisplayFunctions.ino sketch

Download and run OLEDdisplayFunctions.ino

- Download OLEDdisplayFunctions.ino
- Upload the sketch
- Inspect the code with code-folding

Download and run OLEDdisplayFunctions.ino

- Turn on Code Folding in Preferences for Arduino IDE
- Open ODEdisplay functions



Sketchbook location:
/Users/gerry/Documents/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation ☑ upload

Compiler warnings: None

□ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features

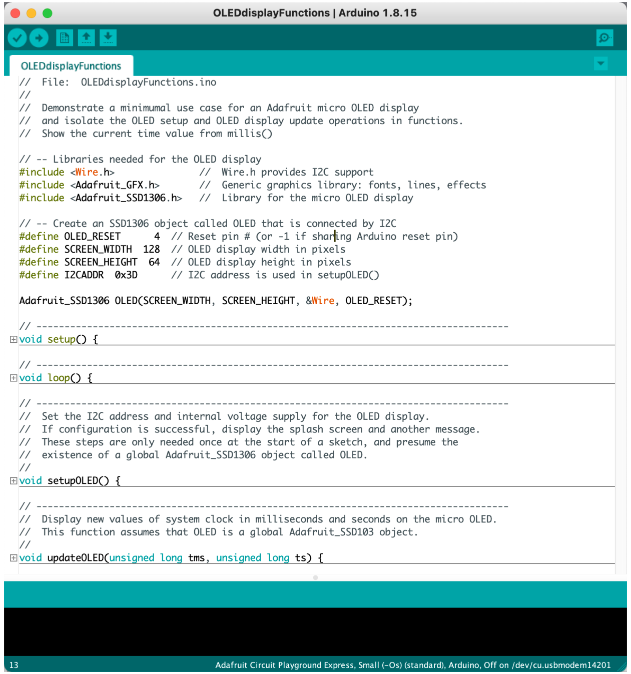
☑ Enable Code Folding
□ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://www.adafruit.com/package_adafruit_index.json,http://arduino.esp82

More preferences can be edited directly in the file
/Users/gerry/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

Download and run OLEDdisplayFunctions.ino

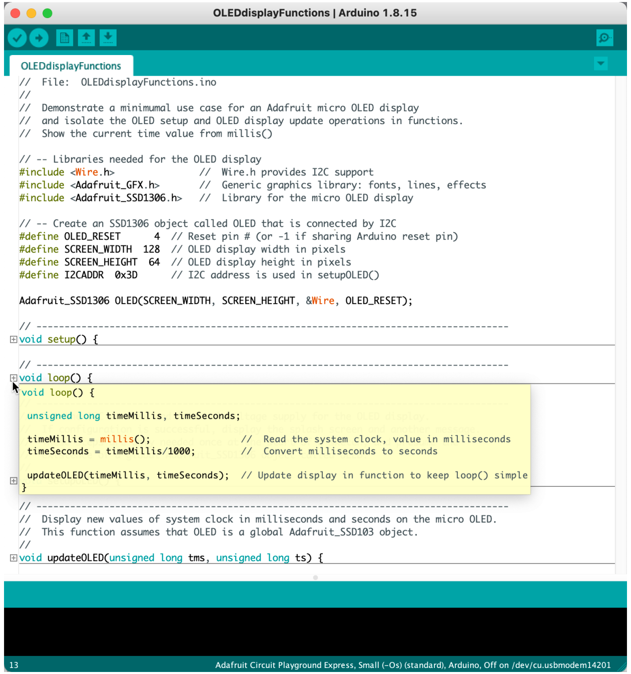
- Turn on Code Folding in Preferences for Arduino IDE
- Open ODEdisplay functions
- Notice + and – signs near function declarations


OLEDdisplayFunctions | Arduino 1.8.15
OLEDdisplayFunctions
13
Adafruit Circuit Playground Express, Small (-Os) (standard), Arduino, Off on /dev/cu.usbmodem14201

```
//  File:  OLEDdisplayFunctions.ino
//
//  Demonstrate a minimumal use case for an Adafruit micro OLED display
//  and isolate the OLED setup and OLED display update operations in functions.
//  Show the current time value from millis()

// -- Libraries needed for the OLED display
#include <Wire.h>                //  Wire.h provides I2C support
#include <Adafruit_GFX.h>        //  Generic graphics library: fonts, lines, effects
#include <Adafruit_SSD1306.h>    //  Library for the micro OLED display

// -- Create an SSD1306 object called OLED that is connected by I2C
#define OLED_RESET      4  // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_WIDTH  128  // OLED display width in pixels
#define SCREEN_HEIGHT  64  // OLED display height in pixels
#define I2CADDR  0x3D      // I2C address is used in setupOLED()

Adafruit_SSD1306 OLED(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// -------------------------------------------------------------------------------------
⊞void setup() {

// -------------------------------------------------------------------------------------
⊞void loop() {

// -------------------------------------------------------------------------------------
//  Set the I2C address and internal voltage supply for the OLED display.
//  If configuration is successful, display the splash screen and another message.
//  These steps are only needed once at the start of a sketch, and presume the
//  existence of a global Adafruit_SSD1306 object called OLED.
//
⊞void setupOLED() {

// -------------------------------------------------------------------------------------
//  Display new values of system clock in milliseconds and seconds on the micro OLED.
//  This function assumes that OLED is a global Adafruit_SSD103 object.
//
⊞void updateOLED(unsigned long tms, unsigned long ts) {
```

Download and run OLEDdisplayFunctions.ino

- Turn on Code Folding in Preferences for Arduino IDE
- Open ODEdisplay functions
- Notice + and – signs near function declarations
- Hover over + sign (do not click) to reveal code



```
//  File:  OLEDdisplayFunctions.ino
//
//  Demonstrate a minimumal use case for an Adafruit micro OLED display
//  and isolate the OLED setup and OLED display update operations in functions.
//  Show the current time value from millis()

// -- Libraries needed for the OLED display
#include <Wire.h>               //  Wire.h provides I2C support
#include <Adafruit_GFX.h>       //  Generic graphics library: fonts, lines, effects
#include <Adafruit_SSD1306.h>   //  Library for the micro OLED display

// -- Create an SSD1306 object called OLED that is connected by I2C
#define OLED_RESET      4  // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_WIDTH  128  // OLED display width in pixels
#define SCREEN_HEIGHT  64  // OLED display height in pixels
#define I2CADDR  0x3D      // I2C address is used in setupOLED()

Adafruit_SSD1306 OLED(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// -------------------------------------------------------------------------------
⊞void setup() {

// -------------------------------------------------------------------------------
⊞void loop() {
```

```
void loop() {

 unsigned long timeMillis, timeSeconds;

 timeMillis = millis();               //  Read the system clock, value in milliseconds
 timeSeconds = timeMillis/1000;       //  Convert milliseconds to seconds

 updateOLED(timeMillis, timeSeconds);  // Update display in function to keep loop() simple
}
```

```
// -------------------------------------------------------------------------------
//  Display new values of system clock in milliseconds and seconds on the micro OLED.
//  This function assumes that OLED is a global Adafruit_SSD103 object.
//
⊞void updateOLED(unsigned long tms, unsigned long ts) {
```

13 Adafruit Circuit Playground Express, Small (-Os) (standard), Arduino, Off on /dev/cu.usbmodem14201

OLEDdisplayFunctions.ino (header)

```
//  File:  OLEDdisplayFunctions.ino
//
//  Demonstrate a minimumal use case for an Adafruit micro OLED display
//  and isolate the OLED setup and OLED display update operations in functions.
//  Display the time value returned by millis() in both milliseconds and seconds.

// -- Libraries needed for the OLED display
#include <Wire.h>               //  Wire.h provides I2C support
#include <Adafruit_GFX.h>       //  Generic graphics library: fonts, lines, effects
#include <Adafruit_SSD1306.h>   //  Library for the micro OLED display

// -- Create an SSD1306 object called OLED that is connected by I2C
#define OLED_RESET      4    // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_WIDTH   128    // OLED display width in pixels
#define SCREEN_HEIGHT   64    // OLED display height in pixels
#define I2CADDR       0x3D    // I2C address is used in setupOLED()

Adafruit_SSD1306 OLED(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

OLEDdisplayFunctions.ino   (setup)

```
void setup() {

  // -- Serial monitor is only used for debugging
  Serial.begin(115200);
  delay(2000);                               //  Wait for Serial object to start
  Serial.println("Serial monitor started");  //  Look for this messagae in Serial Monitor

  setupOLED();         // Do standard set-up work in a reusable function
}
```

OLEDdisplayFunctions.ino   (loop)

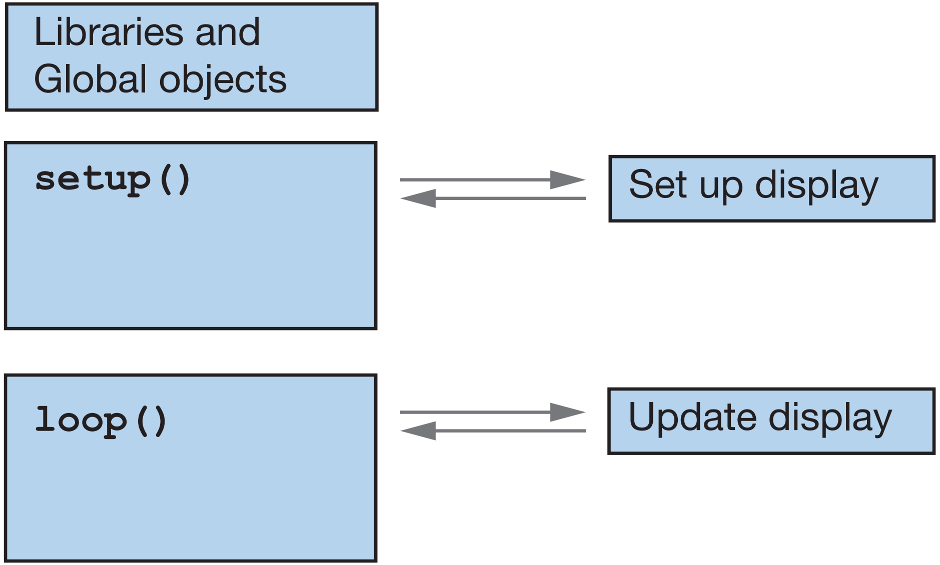
```
void loop() {

  unsigned long timeMillis, timeSeconds;

  timeMillis = millis();                //  Read the system clock, value in milliseconds
  timeSeconds = timeMillis/1000;        //  Convert milliseconds to seconds

  updateOLED(timeMillis, timeSeconds);  // Update display in function to keep loop() simple
}
```

Organize code into separate functions

Keep setup and loop compact

Using functions for discrete tasks is a big advantage when working with more complex codes

setupDisplay and updateDisplay can be reused or used as templates in other sketches



Displaying text and numbers on the OLED

1. Code in setupOLED initializes the OLED
   - Start the “oled” object
   - Display message that OLED is starting
2. Code in updateOLED
   - Display value in milliseconds, and append “ms”
   - Display “t = ”, the value of time in seconds, and “ s”

You will adapt updateOLED to other sketches

Steps to display fixed text

1. Set text size – optional if current size is OK
2. Move cursor to a starting position
3. Add text to the display buffer with .print method
4. Update the display with .display method when butter is finished

```
OLED.setTextSize(1);
OLED.setCursor(0,0);
OLED.print(F("Message"));
OLED.display();
```

NOTE: Fixed strings are enclosed in the F(…) macro

Steps to display dynamic numbers

1. Set text size – optional if current size is OK
2. Move cursor to a starting position
3. Add numerical to the display buffer with .print method
4. Update the display with .display method when butter is finished

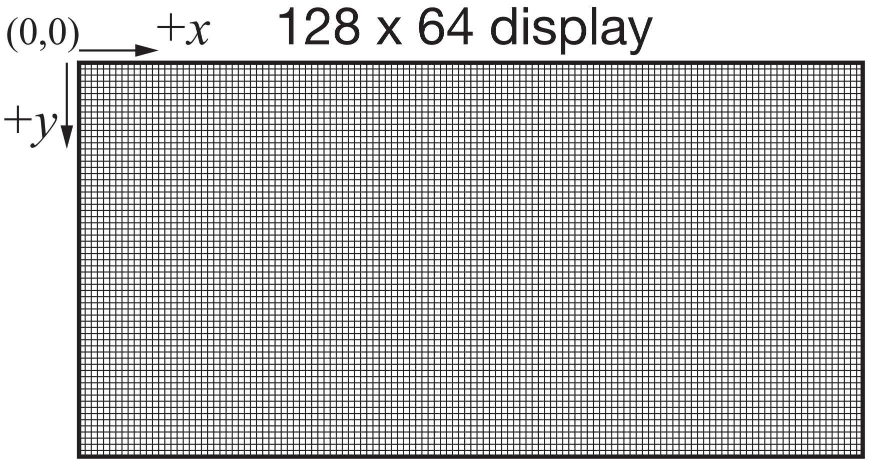
```
OLED.setTextSize(1);
OLED.setCursor(20,0);
OLED.print(t);
OLED.display();
```

NOTE: Numerical values are **not** enclosed in the F(…) macro

OLED display is a 2D grid of pixels

Origin of the display coordinates is the upper left corner

- x is the horizontal position (long axis), increasing to the right
- y is the vertical position, increasing downward



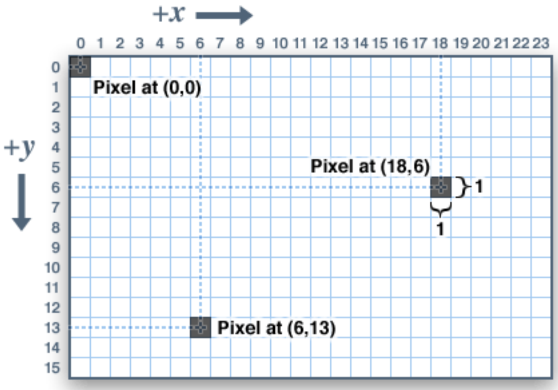


Image from: https://learn.adafruit.com/adafruit-gfx-graphics-library

See: https://learn.adafruit.com/monochrome-oled-breakouts/arduino-library-and-examples

Characters are drawn as bit-maps

Each character is a predefined bit-map, ie. the pixel pattern

- Locate text from upper left corner
- GFX library hands the details

```
OLED.setCursor(3,4);
OLED.print(F("A"));
OLED.display();
```

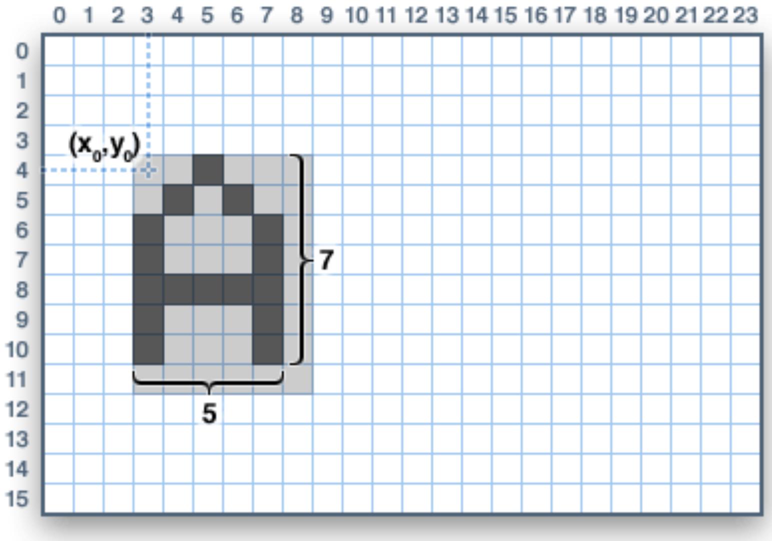


Image from
https://learn.adafruit.com/adafruit-gfx-graphics-library

Practice

Modify OLEDdisplayFunctions.ino

- Add a new variable, timeMinutes to loop, convert seconds to minutes
- Add timeMinutes as a 3rd argument in the call to updateDisplay
- Modify updateDisplay
  - Add input argument for timeMinutes
  - Add steps to display time in minutes
- Add a display of output from the on-board light sensor
- Add a display of output from the on-board temperature sensor