

Qwiic Carrier Board Hookup Guide

Introduction

The MicroMod Qwiic Carrier Board can be used to rapidly prototype with other Qwiic devices; it is available in either a single or double mounting configuration. Just like other MicroMod Carrier Boards, users are free to develop with MicroMod Processor Board of their choice.



SparkFun MicroMod Qwiic Carrier Board - Single
© DEV-17723



SparkFun MicroMod Qwiic Carrier Board - Double

DEV-17724

For a quick overview, check out our product video showcasing the MicroMod Qwiic carrier boards:

Product Showcase: SparkFun MicroMod Qwiic Carrier Board



Required Materials

To get started, users will need a few of items listed below. *(You may already have a some of these items; read through the guide and modify your cart accordingly.)*

MicroMod Processor Board

Like other MicroMod Carrier Boards, a Processor Board is required for the product to operate. Users will need a Processor Board *(of their choice)* to attach to the MicroMod M.2 connector; *since, one is not included with this product.* Below, are few options:



SparkFun MicroMod ESP32 Processor

● WRL-16781



SparkFun MicroMod nRF52840 Processor

● WRL-16984



SparkFun MicroMod RP2040 Processor

● DEV-17720



SparkFun MicroMod Update Tool

● DEV-17725

Required Hardware

A Phillips screw driver is necessary to attach the Processor board to the Carrier Board. Additionally, a USB-C cable is needed to connect the Carrier Board to a computer.



USB 3.1 Cable A to C - 3 Foot

● CAB-14743



SparkFun Mini Screwdriver

● TOL-09146

Optional Hardware

To connect Qwiic breakout boards for your MicroMod project, Qwiic cables are required. Users can also attach the Qwiic devices with **4-40** screws or stacked them with standoffs.



SparkFun Qwiic Cable Kit
○ KIT-15081



Flexible Qwiic Cable - 100mm
● PRT-17259



Screw - Phillips Head (1/4", 4-40, 10 pack)
● PRT-10453



Standoff - Metal Hex (4-40; 3/8"; 10 pack)
● PRT-10463

A single-cell Lithium-ion battery can be connected to the Qwiic Carrier Board for portability.



Lithium Ion Battery - 400mAh
○ PRT-13851



Lithium Ion Battery - 2Ah
● PRT-13855



Lithium Ion Battery - 1Ah



Lithium Ion Battery - 110mAh

○ PRT-13813

● PRT-13853

To modify the jumpers, users will need soldering equipment and/or a knife.



Solder Lead Free - 100-gram Spool

● TOL-09325



Chip Quik No-Clean Flux Pen - 10mL

● TOL-14579



Weller WLC100 Soldering Station

● TOL-14228



Hobby Knife

● TOL-09200

Suggested Reading

The MicroMod ecosystem is a unique way to allow users to customize their project to their needs. The Qwiic connect system is a simple method for interfacing with I²C devices. Click on the banners below for more information on each system.

MicroMod

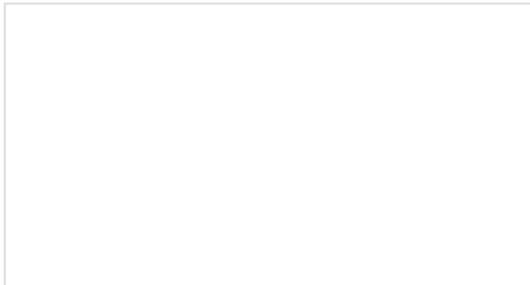
Product Showcase: SparkFun MicroMod Ecosystem



SparkFun's Qwiic Connect System

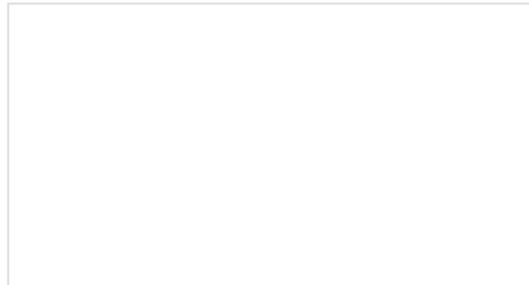


For users who aren't familiar with the following concepts, we also recommend reading the following tutorials before continuing.



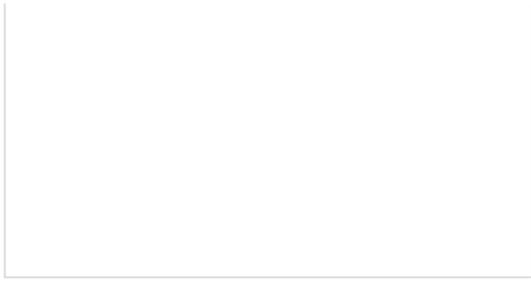
Serial Communication

Asynchronous serial communication concepts: packets, signal levels, baud rates, UARTs and more!



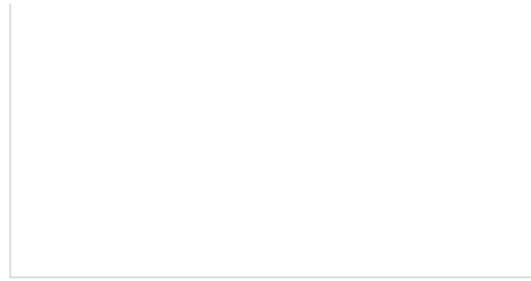
I2C

An introduction to I2C, one of the main embedded communications protocols in use today.



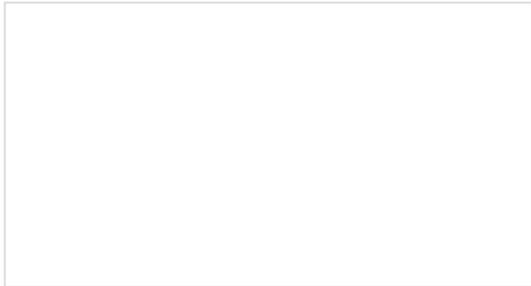
Getting Started with MicroMod

Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!



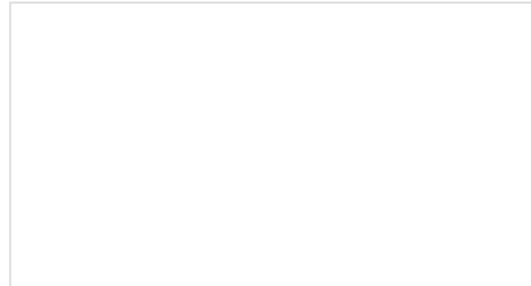
Designing with MicroMod

This tutorial will walk you through the specs of the MicroMod processor and carrier board as well as the basics of incorporating the MicroMod form factor into your own PCB designs!



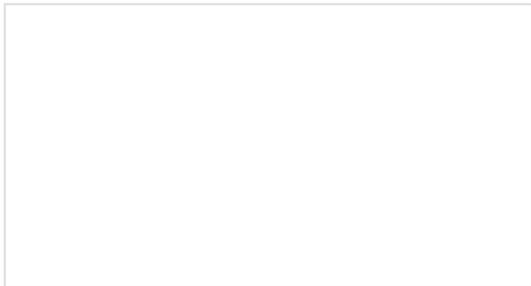
Installing an Arduino Library

How do I install a custom Arduino library? It's easy! This tutorial will go over how to install an Arduino library using the Arduino Library Manager. For libraries not linked with the Arduino IDE, we will also go over manually installing an Arduino library.



Installing Arduino IDE

A step-by-step guide to installing and testing the Arduino software on Windows, Mac, and Linux.



Installing Board Definitions in the Arduino IDE

How do I install a custom Arduino board/core? It's easy! This tutorial will go over how to install an Arduino board definition using the Arduino Board Manager. We will also go over manually installing third-party cores, such as the board definitions required for many of the SparkFun development boards.

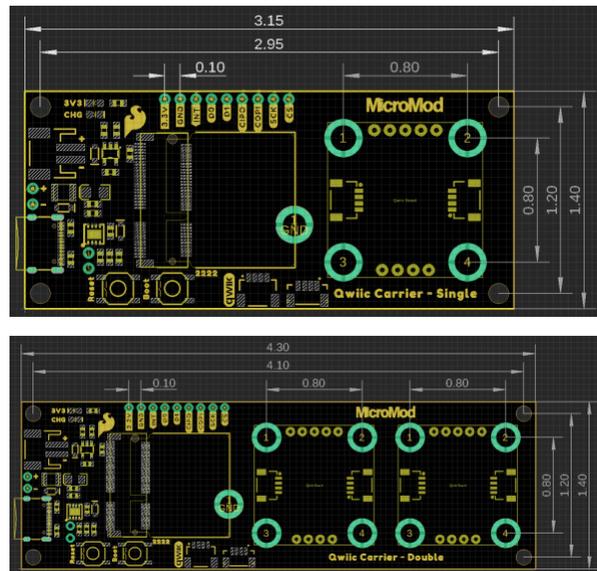
Hardware Overview

This section will cover the various hardware components and solder jumpers on the MicroMod Qwiic Carrier Board. It will also review the connections of the M.2 connector and how it interfaces with a MicroMod Processor.

Board Dimensions

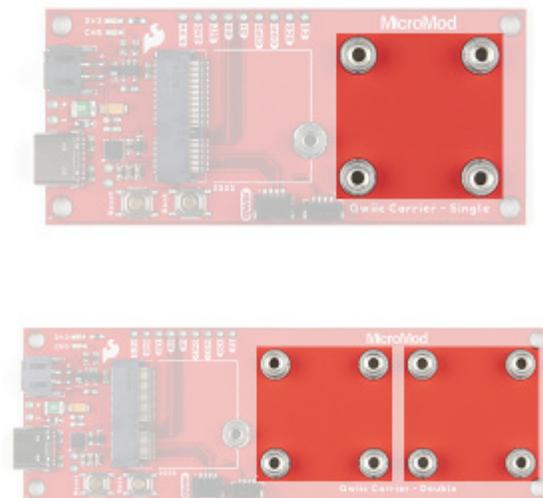
The Qwiic Carrier Board dimensions are:

- **Single:** 3.15" x 1.40" (80.00mm x 35.56mm)
- **Double:** 4.10" x 1.40" (109.22mm x 35.56mm)



Dimensions of the MicroMod Qwiic Carrier Board. (Click to enlarge)

The boards also include mounting holes and inserts for a standard 4-40 screw. The inserts are arranged to mount, attach, and/or stack compatible (1" x 1") Qwiic boards.

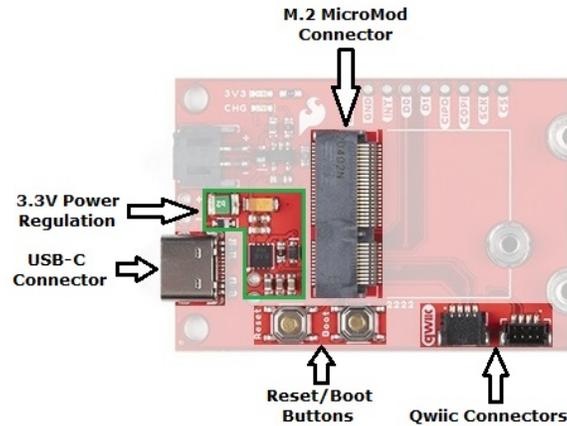


Mounting areas on the MicroMod Qwiic Carrier Board to attach Qwiic devices. (Click to enlarge)

Common Components

Most SparkFun MicroMod Carrier Boards will have some common components and all MicroMod Carrier Boards will have the keyed M.2 MicroMod connector for a Processor board. The photo and list below outline the common components between the Qwiic carrier board and other MicroMod Carrier Boards.

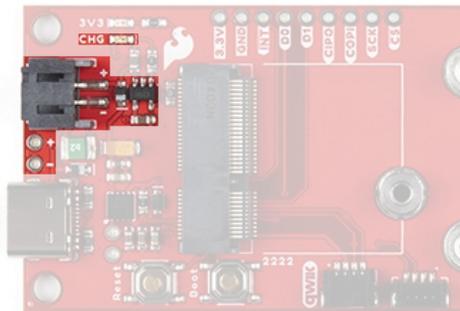
- **M.2 MicroMod Connector** - This special keyed M.2 connector lets you install your MicroMod Processor of choice to the Qwiic Carrier Board.
- **USB-C Connector** - Connect to your computer to program your processor and provide power to the board.
- **3.3V Regulator** - Provides a regulated **3.3V** and sources up to **1A**.
- **Qwiic Connectors** - The standard Qwiic connectors to connect other Qwiic devices for your MicroMod project.
- **Boot/Reset Buttons** - Push buttons to enter boot mode on processors and to reset your MicroMod circuit.



Common MicroMod components featured on the MicroMod Qwiic Carrier Board. (Click to enlarge)

Battery Charger

The board also has a MCP73831 Single-Cell Lithium-Ion/Lithium-Polymer Charge IC so you can charge an attached single-cell LiPo battery. The charge IC receives power from the USB connection and can source up to 450mA to charge an attached battery.

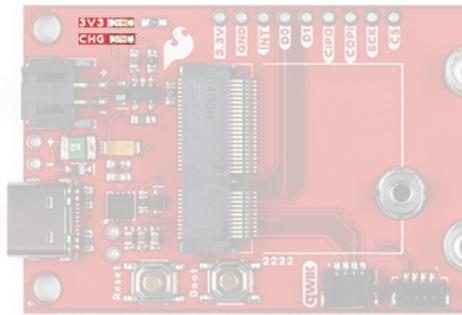


Battery charger for the MicroMod Qwiic Carrier Board. (Click to enlarge)

Status LEDs

The carrier board has two status LEDs:

- **PWR** - This LED indicates when 3.3V power is available to the board.
- **CHG** - This LED indicates the status of the charging circuit operation.



Status LEDs on the MicroMod Qwiic Carrier Board. (Click to enlarge)

Solder Jumpers

Users who have never worked with soldering jumpers and cutting PCB traces before (or for a quick refresher), check out our [How to Work with Solder Jumpers and PCB Traces](#) tutorial for detailed instructions and tips.

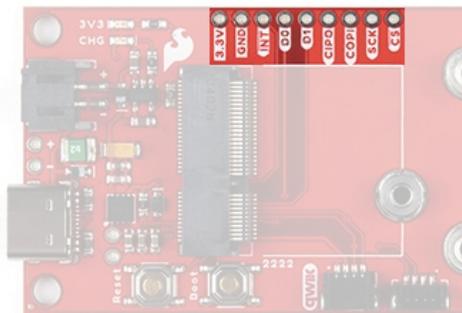
There are four adjustable solder jumpers on the MicroMod Qwiic Carrier Board labeled **MEAS**, **BYP**, **3.3V_VE** and **3.3V**. The table below briefly outlines their functionalities:

Jumper Name/Label	Description	Default State
Measure/MEAS	Open this jumper to probe the current draw at the 3.3V output of the regulator. For help measuring current, take a look at our How to Use a Multimeter tutorial.	CLOSED
Bypass/BYP	The "penny-in-the-fuse" jumper. Bypasses the 6V/2A fuse and nets VIN and V_USB together. Close only if you know what you are doing!	OPEN
Voltage Regulator Enable/VE	Voltage regulator control. Close this jumper to control the VREG in low-power applications.	OPEN
3.3V LED Power/3V3 LED	Connects the 3.3V LED to 3.3V via a 1K Ohm resistor. Open to disable the LED.	CLOSED

10	D0	D0	Digital I/O pin
11	BOOT	BOOT Button	Connected to BOOT Button. Boot is active LOW.
12	I2C_SDA	Qwiic Connector - SDA	I ² C data signal for Qwiic devices.
14	I2C_SCL	Qwiic Connector - SCL	I ² C clock signal for Qwiic devices.
16	I2C_INT	$\overline{\text{INT}}$	I ² C interrupt pin
18	D1	D1	Digital I/O pin
55	SPI_CS	$\overline{\text{CS}}$	Chip Select.
57	SPI_SCK	SCK	SPI Clock signal.
59	SPI_COPI	COPI	SPI Controller Out/Peripheral In signal.
61	SPI_CIPO	CIPO	SPI Controller In/Peripheral Out signal.

Breakout Pins

The Qwiic Carrier Board features a 3.3V, a ground, seven I/O breakout pins. The functionality of these pins are detailed in the table above.



Breakout pins on the MicroMod Qwiic Carrier Board. (Click to enlarge)

Hardware Assembly

For those unfamiliar with the MicroMod ecosystem, be sure to review the Getting Started with MicroMod guide.



Getting Started with MicroMod

OCTOBER 21, 2020

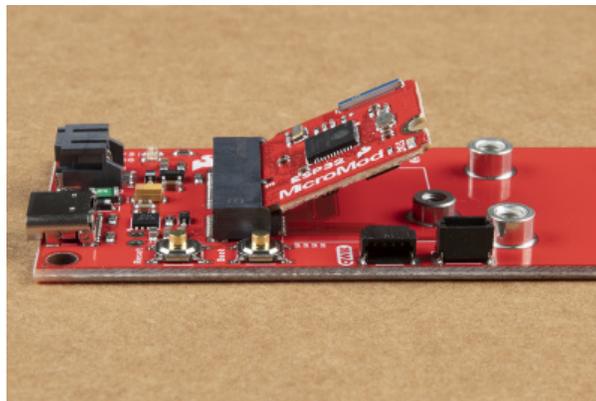
Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!

Processor Board

To get started with the Qwiic Carrier Board, users will need a compatible Processor. Insert the MicroMod Processor board into the M.2 socket at an angle, with its edge connector aligned to the matching slots.

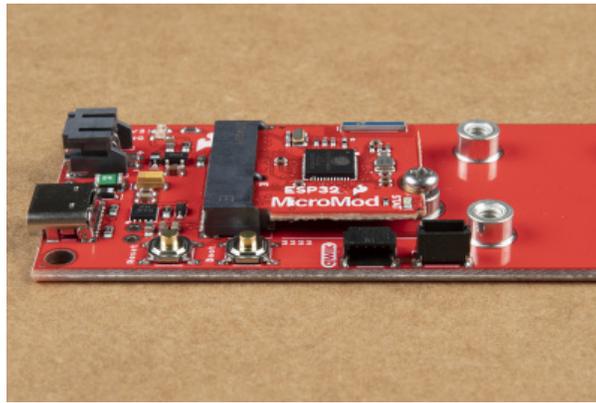
Note: The dimensions of the Processor board's edge connector prevents it from mating with the slots of the M.2 socket in reverse. As an extra safeguard, the screw insert is spaced to only match the screw key of MicroMod Processor boards.

When inserted properly, the Processor board will rest at an angle:



Inserting a processor board into the M.2 socket on the MicroMod Qwiic Carrier Board. (Click to enlarge)

To secure the processor board, gently hold down on the board and attach the M.2 screw with a Phillip's head (PH0 or PH1) screw driver. Below, is an example of an assembled MicroMod system:

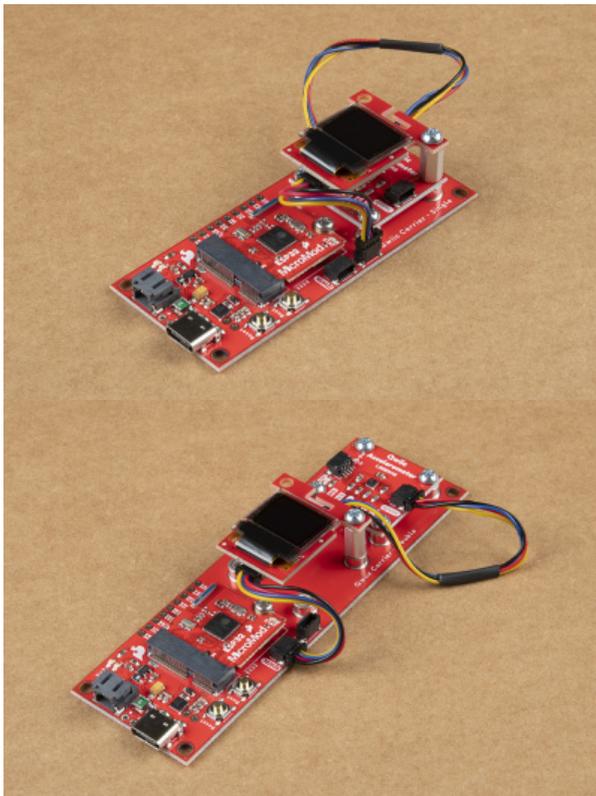


A processor board attached to the MicroMod Qwiic Carrier Board. (Click to enlarge)

Qwiic Devices

Screw inserts are available in either a single or double configuration to attach/mount 1" x 1" Qwiic devices to the Qwiic carrier board. The inserts are compatible with 4-40 screws; additionally, 4-40 3/8" standoffs can also be used to stack boards vertically.

To electronically connect the Qwiic devices to the Carrier Board, users will need Qwiic cables. For more information, check out our Qwiic ecosystem page.



Mounting and connecting Qwiic devices to the MicroMod Qwiic Carrier Board. (Click to enlarge)

Programming

To program the processor board utilized on the Qwiic Carrier Board; connect the board to a computer with a USB-C cable. Depending on the Processor board, user may need to install drivers (*if they have not done so already*).

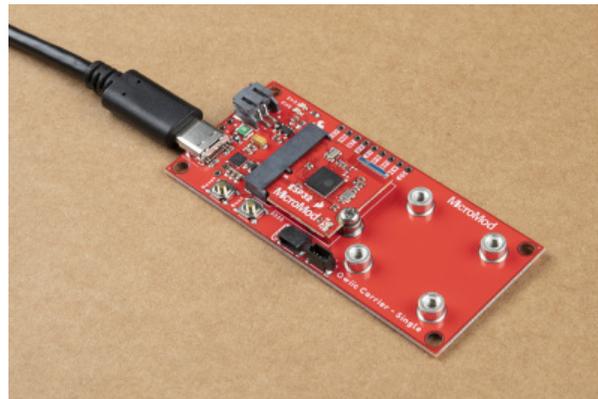
Note: Make sure that the correct board definitions are installed in the Arduino IDE, for the selected Processor board. For help installing board definitions, use the MicroMod processor boards landing page and review the associated hookup guide for that hardware.



Installing Board Definitions in the Arduino IDE

SEPTEMBER 9, 2020

How do I install a custom Arduino board/core? It's easy! This tutorial will go over how to install an Arduino board definition using the Arduino Board Manager. We will also go over manually installing third-party cores, such as the board definitions required for many of the SparkFun development boards.



USB connection to the MicroMod Qwiic carrier board for programming the attached processor board. (Click to enlarge)

Example

Below is a simple demonstration of using the MicroMod Qwiic Carrier Board to create a bubble level, using an accelerometer and OLED display. Users will also need other parts and accessories; all parts besides the Qwiic carrier board are listed below:



SparkFun Qwiic Cable Kit
○ KIT-15081



SparkFun Micro OLED Breakout (Qwiic)
● LCD-14532



SparkFun MicroMod ESP32 Processor
● WRL-16781



USB 3.1 Cable A to C - 3 Foot
● CAB-14743



SparkFun Mini Screwdriver
● TOL-09146



Screw - Phillips Head (1/4", 4-40, 10 pack)
● PRT-10453



Triple Axis Accelerometer Breakout - LIS2DH12 (Qwiic)



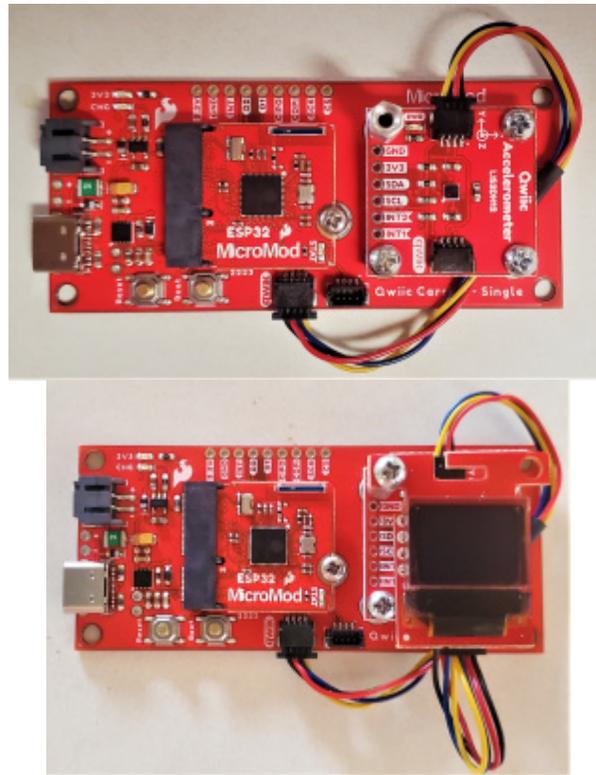
Standoff - Metal Hex (4-40; 3/8"; 10 pack)
● PRT-10463

Note: We have chosen the MicroMod ESP32 for the processor board for this demonstration; however, users are free to utilize the processor board of their choice.

Please note, that the sample code below is intended to program processor boards that are compatible with the Arduino IDE. Therefore, processor boards like the RP2040 are going to need extra effort to adapt the code (technical assistance for which, is **not available**).

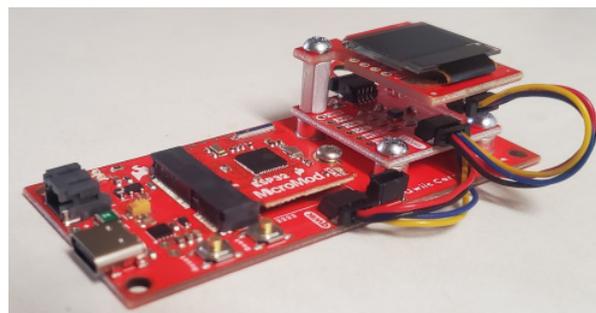
Assembly

The assemble process is relatively straight forward. The primary thing that users need to **pay attention** is the orientation of the OLED display relative to the axes of the accelerometer. *The x-axis should point to the right of the OLED display and the y-axis should point to the top of the display (see images below).*



Mounting and connecting the Qwiic devices to the MicroMod Qwiic Carrier Board. (Click to enlarge)

Some users may find it the movement of the bubble in the example code to be more intuitive if the OLED display is stacked above the accelerometer.



Don't forget to daisy-chain the boards together with Qwiic cables. (Click to enlarge)

Example Code

Users can download the example code here or by clicking the button below. Besides the RP2040 processor board, users only need to select the proper port and board from the board manager in the Arduino IDE before uploading the code.

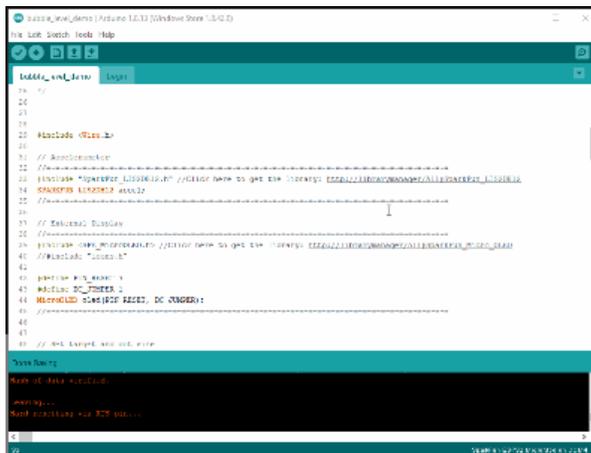
[DOWNLOAD EXAMPLE CODE](#)

The following sections break down the example code execution for users who might want to modify the code. *Please note, as defined in our terms of service, that we do not provide technical assistance for any code modifications.*

Dependencies

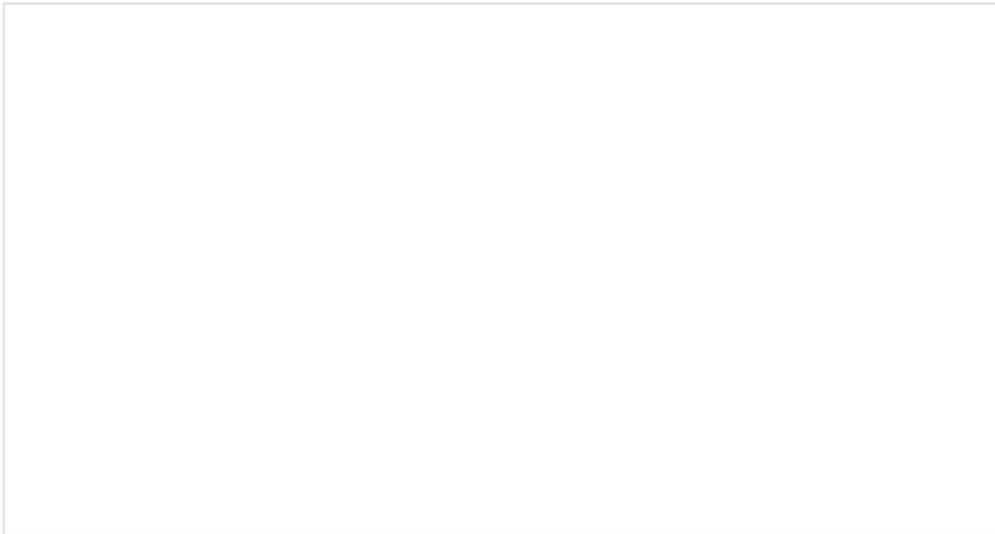
Below are the dependencies for the code including the libraries, pin definitions, and variable instantiations. The example code requires both the Micro OLED and LIS2DH12 Arduino libraries to be installed in the Arduino IDE.

Tip: While in the Arduino IDE, users can click the links in the code (i.e. `//Click here to get the library: http://librarymanager/All#SparkFun_LIS2DH12) to pull up the required library in the library manager.`



```
16 //
17 //
18 //
19 #include <Wire.h>
20 // Accelerometer
21 //=====
22 #include "SparkFun_LIS2DH12.h" //Click here to get the library: http://librarymanager/All#SparkFun_LIS2DH12
23 #define LIS2DH12_ADDR 0x18
24 //=====
25 //
26 // External Display
27 //=====
28 #include "SSD1306.h" //Click here to get the library: http://librarymanager/All#SparkFun_SSD1306
29 #include "Adafruit_GFX.h"
30 #include "Adafruit_SSD1306.h"
31 //=====
32 //
33 #define PIN_A0 1
34 #define PIN_A1 2
35 #define PIN_A2 3
36 //=====
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
```

Clicking the link to pull up the required library in the Arduino IDE. (Click to enlarge)



Installing an Arduino Library
JANUARY 11, 2013

How do I install a custom Arduino library? It's easy! This tutorial will go over how to install an Arduino library using the Arduino Library Manager. For libraries not linked with the Arduino IDE, we will also go over manually installing an Arduino library.

Libraries and Pin Definitions

This section of the code contains the required libraries and pin definitions for the code.

```
#include <Wire.h>

// Accelerometer
//=====
#include "SparkFun_LIS2DH12.h" //Click here to get the library: http://librarymanager/All#SparkFun\_LIS2DH12
SPARKFUN_LIS2DH12 accel;
//=====

// External Display
//=====
#include <SFE_MicroOLED.h> //Click here to get the library: http://librarymanager/All#SparkFun\_MicroOLED
#include "icons.h"

#define PIN_RESET 7
#define DC_JUMPER 1
MicroOLED oled(PIN_RESET, DC_JUMPER);
//=====
```

Variables

This section of the code instantiates most of the variables used in the code.

```
// Set target and dot size
int radiusLarge = 10; // Target area 2
int radiusSmall = 4; // Target area 1
int radiusBubble = 2; // Bubble size

// Set initial roll and pitch measurement
double averagedRoll = 0.0;
double averagedPitch = 0.0;
```

Setup Loop

This section of the code runs the setup *loop*, which initializes the serial output, I²C connection, the OLED display operation, and the accelerometer operation and configuration.

```

void setup()
{
  // Initialize Output for Print Statements
  Serial.begin(115200);
  Serial.println("SparkFun Accel Example");

  // Initialize I2C Connection
  Wire.begin();
  //Wire.setClock(400000);

  beginDisplay(); //Check if an external Qwiic OLED is attached and display splash screen

  // Check for accelerometer
  if (accel.begin() == false)
  {
    Serial.println("Accelerometer not detected. Check address jumper and wiring. Freezing...");
    while (1)
      ;
  }

  // Set sample/data rate for accelerometer
  // The larger the avgAmount the faster we should read the sensor
  //accel.setDataRate(LIS2DH12_ODR_100Hz); //6 measurements a second
  accel.setDataRate(LIS2DH12_ODR_400Hz); //25 measurements a second
}

```

Setup Functions

These functions are executed in the setup *loop*, to initialize the OLED display I²C connection and display the splash screen.

```

// Ping an I2C address and see if it responds
bool isConnected(uint8_t deviceAddress)
{
  Wire.beginTransmission(deviceAddress);
  if (Wire.endTransmission() == 0)
    return true;
  return false;
}

void beginDisplay()
{
  // Connect to address display is on by pinging addresses
  // 0x3D is default on Qwiic board
  if (isConnected(0x3D) == true || isConnected(0x3C) == true)
  {
    //Init and display splash
    oled.begin(); // Initialize the OLED
    oled.display(); // Display splash screen
    delay(1200);
    oled.clear(PAGE); // Clear the display's internal memory

    oled.setCursor(15, 7); //x, y
    oled.setFontType(0); //Set font to smallest
    oled.print(F("Bubble"));
    oled.setCursor(19, 20); //x, y
    oled.print(F("Level"));

    oled.display();
    delay(1200);
  }
}

```

Main loop

This section of the code runs the main loop. The code retrieves roll and pitch calculation, clears the display buffer, displays the bubble and target circles or areas. If the position of the bubble is within the target are, the roll and pitch angles are displayed as x and y coordinates on the screen.

```

void loop()
{
  getAngles(); // Calculate roll and pitch angles based on the accelerometer readings
  oled.clear(PAGE); // Clear the display's internal memory

  // Set dot position
  double bubbleX = LCDWIDTH / 2 - averagedPitch; // Bubble location on x-axis
  double bubbleY = LCDHEIGHT / 2 - averagedRoll; // Bubble location on y-axis

  // Limit bubble position to edge of screen
  if (bubbleX < radiusBubble) { bubbleX = radiusBubble; }
  else if (bubbleX > LCDWIDTH - radiusBubble) { bubbleX = LCDWIDTH - radiusBubble - 1; }
  if (bubbleY < radiusBubble) { bubbleY = radiusBubble; }
  else if (bubbleY > LCDHEIGHT - radiusBubble) { bubbleY = LCDHEIGHT - radiusBubble - 1; }

  // Draw circle relative to dot
  oled.circle(LCDWIDTH / 2, LCDHEIGHT / 2, radiusLarge);
  oled.circle(LCDWIDTH / 2, LCDHEIGHT / 2, radiusSmall);
  oled.circleFill(bubbleX, bubbleY, radiusBubble);

  // Display angle/position once bubble is inside larger target area
  if ( sqrt(averagedPitch * averagedPitch + averagedRoll * averagedRoll) < (radiusLarge - radius
Bubble))
  {
    oled.setFontType(0); //Set font to smallest

    oled.setCursor(LCDWIDTH/2 - 21, 0); //x, y
    oled.print("X:");
    oled.print(-averagedPitch);
    oled.setCursor(LCDWIDTH/2 - 21, LCDHEIGHT - 8); //x, y
    oled.print("Y:");
    oled.print(averagedRoll);

    // oled.setCursor(LCDWIDTH/2 - 11, 0); //x, y
    // oled.print(averagedRoll);
    // if (-averagedPitch < 0) { oled.setCursor(LCDWIDTH - 29, LCDHEIGHT/2 - 3); }
    // else { oled.setCursor(LCDWIDTH - 23, LCDHEIGHT/2 - 3); }
    // oled.print(-averagedPitch);
  }

  oled.display();
}

```

getAngles() Function

This function is executed in the main loop. The code retrieves accelerometer readings, translates the readings into roll and pitch angles, and calculates an average based on a sample set size (avgAmount = 16).

```

void getAngles()
{
  averagedRoll = 0.0;
  averagedPitch = 0.0;
  const int avgAmount = 16;

  // Average readings after 'avgAmount' samples
  for (int reading = 0 ; reading < avgAmount ; reading++)
  {
    while (accel.available() == false) delay(1); // Wait for accelerometer connection

    // Retrieve data from accelerometer
    float accelX = accel.getX();
    float accelY = accel.getY();
    float accelZ = accel.getZ();

    // Optional modification: https://www.nxp.com/docs/en/application-note/AN3461.pdf
    //int signZ = constrain(accelZ, -1, 1);
    //double roll = atan2(accelY , signZ * sqrt( accelZ * accelZ + .001 * abs(accelX) ) ) * 57.
3;

    // Calculate roll and pitch angles
    double roll = atan2(accelY , accelZ) * 57.3;
    double pitch = atan2((-accelX) , sqrt(accelY * accelY + accelZ * accelZ)) * 57.3;
    if (constrain(accelZ, -1, 1) == -1) { roll = atan2(accelY, -accelZ) * 57.3; } // Invert if upside down

    averagedRoll += roll;
    averagedPitch += pitch;

    // Debug Print Statements
    //Serial.print(roll, 6);
    //Serial.print(", ");
    //Serial.print(pitch, 6);
    //Serial.print(", ");
    //
    //Serial.print(accelX);
    //Serial.print(", ");
    //Serial.print(accelY);
    //Serial.print(", ");
    //Serial.print(accelZ);
    //Serial.println("");
  }

  averagedRoll /= (float)avgAmount;
  averagedPitch /= (float)avgAmount;

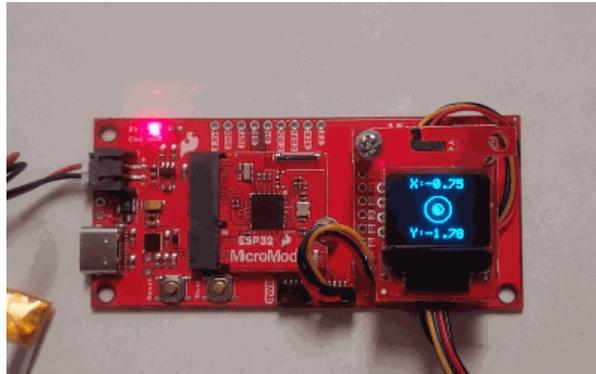
  // Debug Print Statements
  //Serial.print(averagedRoll, 6);
  //Serial.print(", ");
  //Serial.print(averagedPitch, 6);
  //Serial.println(", ");
}

```

```
}
```

Demo Operation

Below is a demonstration of the example code in action. Enjoy!



Demonstration of the bubble level code operating. (Click to enlarge)

Troubleshooting

🔗 Not working as expected and need help?

For technical assistance and more information on a product that is not working as you expected, please review the SparkFun Technical Assistance page for some initial troubleshooting.

[SPARKFUN TECHNICAL ASSISTANCE PAGE](#)

For additional assistance, the SparkFun Forums: MicroMod are a great place to find and ask for help. First-time users will need to create a Forum Account to search product forums and post questions.

[SPARKFUN FORUMS: MICROMOD](#)

Resources and Going Further

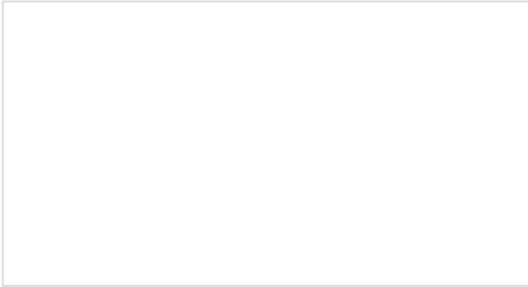
Want more information on the SparkFun Qwiic Carrier Board? Check out these links!

- Schematic
 - Single (PDF)
 - Double (PDF)
- Board Dimensions
 - Single (PDF)
 - Double (PDF)
- Eagle Files
 - Single (ZIP)
 - Double (ZIP)
- GitHub Hardware Repo
- Demo Code
- Product Showcase Video

MicroMod Documentation:

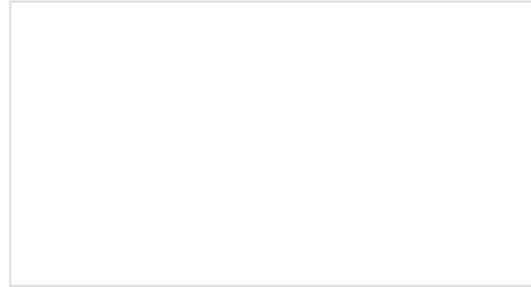
- [Getting Started with MicroMod](#)
- [Designing with MicroMod](#)
- [MicroMod Info Page](#)
- [MicroMod Forums](#)

Looking for some inspiration? Check out these other MicroMod tutorials:



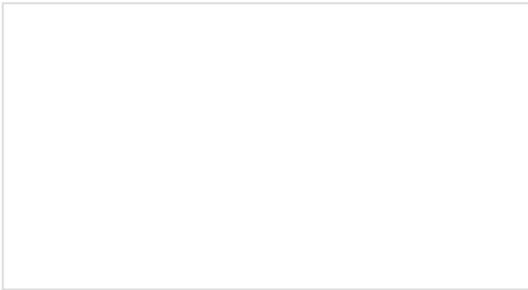
Getting Started with MicroMod

Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!



Designing with MicroMod

This tutorial will walk you through the specs of the MicroMod processor and carrier board as well as the basics of incorporating the MicroMod form factor into your own PCB designs!



MicroMod RP2040 Processor Board Hookup Guide

This tutorial covers the basic functionality of the MicroMod RP2040 Processor Board and highlights the features of the dual-core ARM Cortex-M0+ processors development board. Get started with the first microcontroller from the Raspberry Pi Foundation!



MicroMod Asset Tracker Carrier Board Hookup Guide

Get started with the SparkFun MicroMod Asset Tracker Carrier Board following this Hookup Guide. The Asset Tracker uses the u-blox SARA-R510M8S LTE-M / NB-IoT module to provide a host of data communication options.