MicroMod SAMD51 Processor Board Hookup Guide

Introduction

With the MicroMod specification, we shrunk down the PCB size for the SAMD51 32-bit ARM Cortex-M4F MCU! This tutorial covers the basic functionality of the MicroMod SAMD51 and highlights its features.

SparkFun MicroMod SAMD51 Processor
DEV-16791

Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary.

SparkFun Mini Screwdriver

USB 3.1 Cable A to C - 3 Foot

TOL-09146

CAB-14743

SparkFun MicroMod SAMD51 Processor

DEV-16791

SparkFun MicroMod ATP Carrier Board

DEV-16885

Suggested Reading

If you aren't familiar with the MicroMod ecosystem, we recommend reading here for an overview.

MicroMod

MicroMod Ecosystem

If you aren’t familiar with the following concepts, we also recommend checking out these tutorials before continuing.

New!

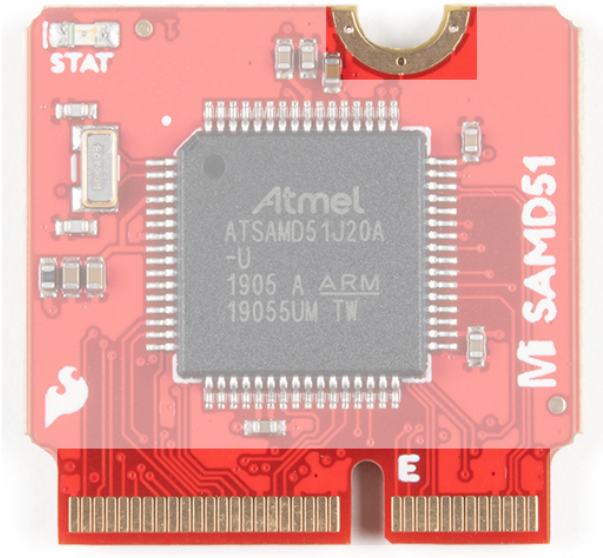
Getting Started with MicroMod

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

Hardware Overview

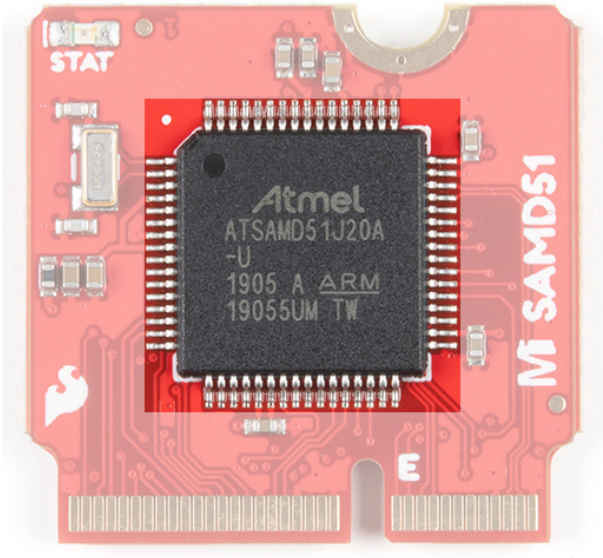
M.2 Connector

All of our MicroMod Processor Boards come equipped with the **M.2 MicroMod Connector**, which leverages the M.2 standard and specification to allow you to install your MicroMod Processor Board on your choice of carrier board. Most of the pins use a common pinout to ensure cross platform compatibility.



SAMD51 Processor

The brains of the processor board is the ATSAMD51J20 32-bit ARM Cortex M4 processor. An external **32.768kHz** crystal is used as the clock for the ATSAMD51. However, the MCU itself has a maximum CPU speed of **120MHz**.

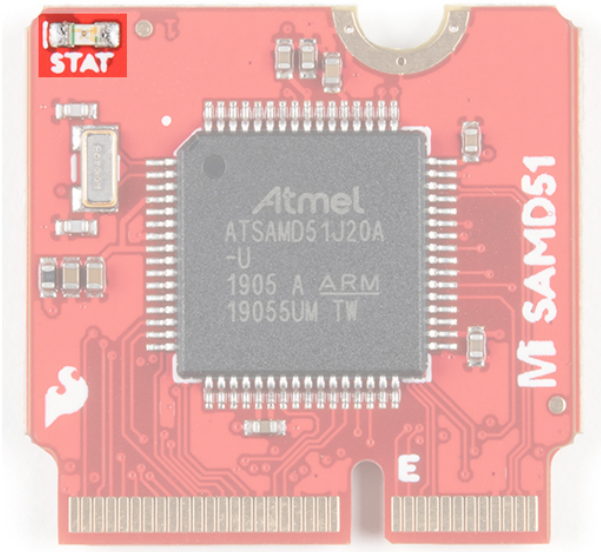


Flash Memory

On the back of the board is the W25Q128JVPIM, which adds 128Mb (16MB) of flash memory externally.

LED

A STAT LED is added to the top side of the board. This is useful debugging or as a status indicator. Additionally, you can use this to determine if the board is in bootloader mode. Pressing the reset button twice will put the board into bootloader mode causing the LED to fade in/out. This is connected to pin 13.



MicroMod SAMD51 Processor Pin Functionality

The complete pin map can be found in the table below or you can refer to the schematic.

Heads up! The pin table below and schematic both include the SAMD51 pin associated with each MicroMod pin and this correlation can be used to identify alternate uses for pins on the SAMD51 Processor Board. For many of the General Purpose I/O pins and other pins with multiple signal options, refer to your Carrier Board's Hookup Guide for information on how those pins are configured what they are used for. Not all pins are used on every Carrier Board.

SAMD51 PROCESSOR PINOUT TABLE

MICROMOD GENERAL PINOUT TABLE

MICROMOD GENERAL PIN DESCRIPTIONS

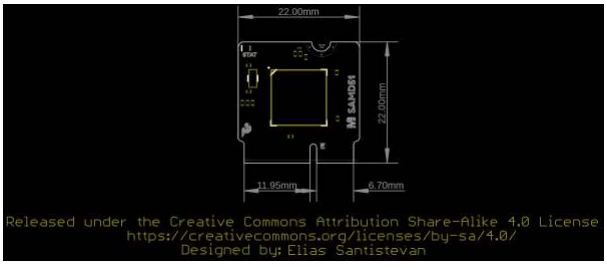
AUDIO	UART	GPIO/BUS	I²C	SDIO	SPI0	Dedicated

SAMD51 Pin	Alternate Function	Primary Function	Bottom Pin	Top Pin	Primary Function	Alternate Function	SAMD51 Pin
		(Not Connected)		**75**	GND		
		3.3V_IN	**74**	**73**	G5		D7
		3.3V	**72**	**71**	G6		D8
		-	**70**	**69**	G7		D9
		-	**68**	**67**	G8		D10
		-	**66**	**65**	G9		D11
		-	**64**	**63**	-		
		-	**62**	**61**	SPI_CIPO		47
		-	**60**	**59**	SPI_COPI		45
43		I2S_MCLK	**58**	**57**	SPI_SCK		46
40		I2S_SDO	**56**	**55**	SPI_$\overline{CS}$		48
41		I2S_SDI	**54**	**53**	I2C_SCL1	TX2	36
44		I2S_FS	**52**	**51**	I2C_SDA1	RX2	37
42		I2S_CLK	**50**	**49**	A4		22
D6		G4	**48**	**47**	PWM1	A3	21
D5		G3	**46**	**45**	GND		
D4		G2	**44**	**43**	CAN_TX		39
D3		G1	**42**	**41**	CAN_RX		38
D2		G0	**40**	**39**	GND		
18		A1	**38**	**37**	-		

		GND	**36**	**35**	-		
17	DAC	A0	**34**	**33**	GND		
19	A2	PWM0	**32**	**31**	Module Key		
		Module Key	**30**	**29**	Module Key		
		Module Key	**28**	**27**	Module Key		
		Module Key	**26**	**25**	Module Key		
		Module Key	**24**	**23**	SWDIO		
36	I2C_SCL1	UART_TX2	**22**	**21**	SWDCK		
37	I2C_SDA1	UART_RX2	**20**	**19**	UART_RX1		32
D1		D1	**18**	**17**	UART_TX1		33
D12		I2C_INT	**16**	**15**	-		
34		I2C_SCL	**14**	**13**	-		
35		I2C_SDA	**12**	**11**	-		
D0		D0	**10**	**9**	-		
29		HOST_ENABLE	**8**	**7**	GND		
		RESET# (Open Drain)	**6**	**5**	USB_D-	USBHOST_D-	30
		3.3V_EN	**4**	**3**	USB_D+	USB_HOST_D+	31
		3.3V_IN	**2**	**1**	GND		

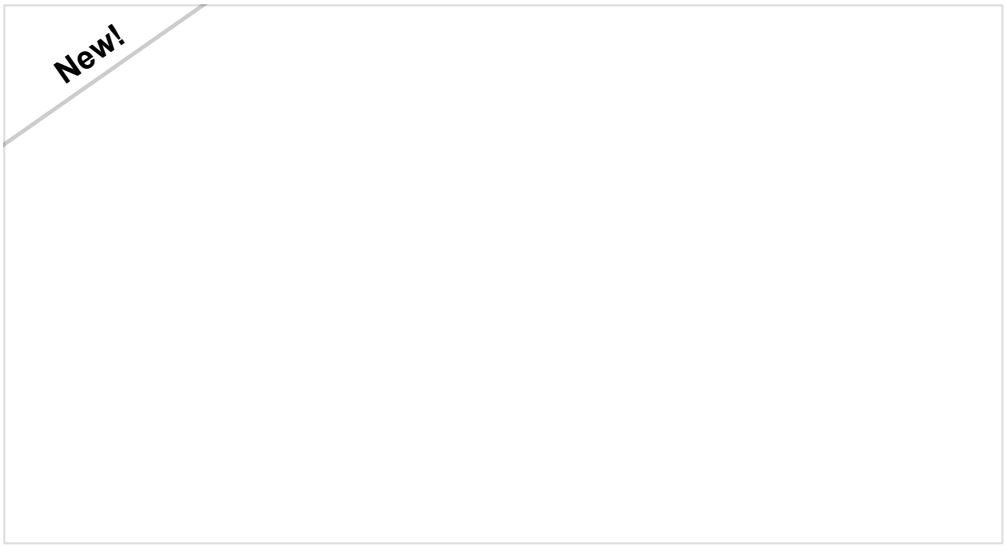
Board Dimensions

The board takes advantage of the standard MicroMod form factor.



Hardware Assembly

If you have not already, make sure to check out the Getting Started with MicroMod: Hardware Hookup for information on inserting your Processor Board into your Carrier Board.

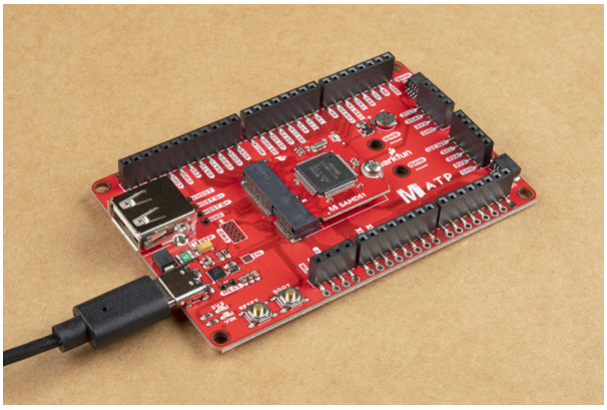


Getting Started with MicroMod

OCTOBER 21, 2020

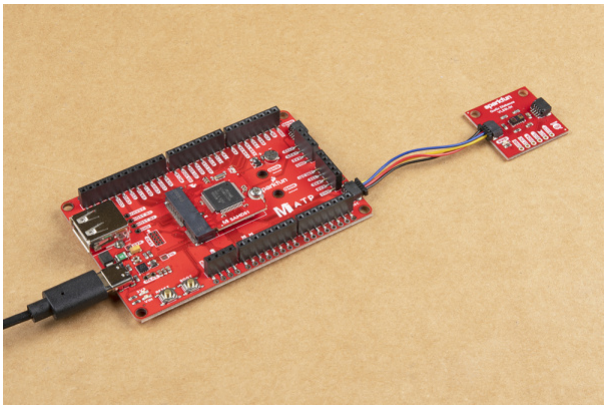
Dive into the world of MicroMod - a compact interface to connect a microcontroller to various peripherals via the M.2 Connector!

For simplicity, we'll be using the MicroMod ATP Carrier Board to program the board. At a minimum, your setup should look like the image below with the MicroMod SAMD51 inserted into the MicroMod ATP Carrier Board with a USB-C cable.



Qwiic-Enabled Device

If you decide to use a Qwiic device (because why not?!), simply insert a Qwiic cable between the two connectors.



UF2 Bootloader & Drivers

> **Heads up!** Please be aware that the SAMD51 Thing Plus is **NOT currently supported on Windows 8** due to a lack of support drivers for those specific OS's.

The SAMD51 Processor Board is easy to program thanks the UF2 bootloader. With the UF2 bootloader, the SAMD51 Thing Plus shows up on your computer as a USB storage device **without having to install drivers** for Windows 10, Mac, and Linux!

What is UF2?

UF2 stands for USB Flashing Format, which was developed by Microsoft for PXT (now known as MakeCode) for flashing microcontrollers over the Mass Storage Class (MSC), just like a removable flash drive. The file format is unique, so unfortunately, you cannot simply drag and drop a compiled binary or hex file onto the Turbo. Instead, the format of the file has extra information to tell the processor where the data goes, in addition to the data itself.

For Arduino users, the UF2 bootloader is **BOSSA compatible**, which the Arduino IDE expects on ATSAMD boards. For more information about UF2, you can read more from the MakeCode blog, as well as the UF2 file format specifiation.

Double-Tap to Launch the Bootloader

The bootloader is what allows us to load code over a simple USB interface. **To upload code, you will need to manually enter bootloader mode** by rapidly double-tapping the reset button (after hitting the reset button once, you have about half a second to hit it again). The board will remain in bootloader mode until power cycles (happens automatically after uploading code) or the reset button is hit again (once).

Double-tapping the reset button to enter bootloader mode.

On the SAMD51, the there are a clues to if it is in bootloader mode:

- The D13 LED indicator will be slowly fading (may appear to be blinking).

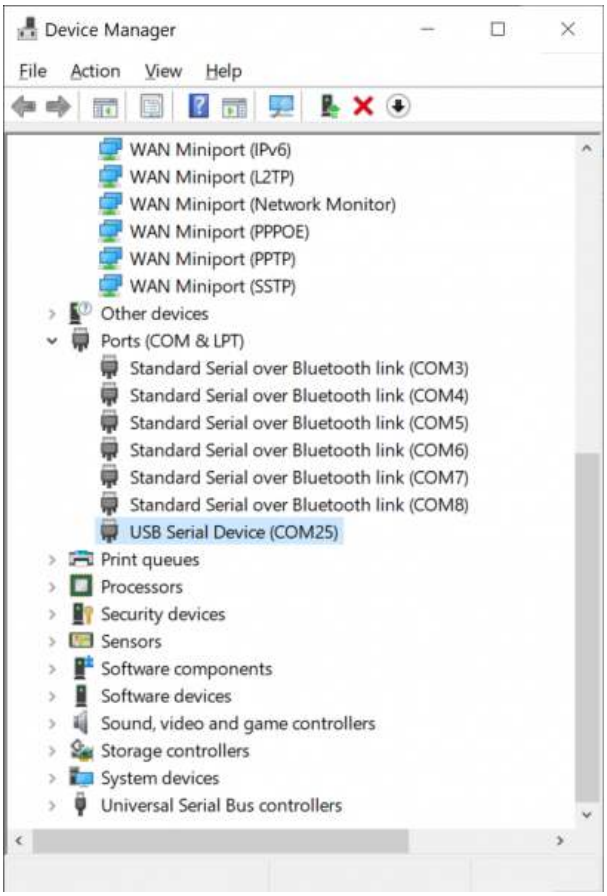
- Board will show up under a different COM port.
- The board will appear as a USB mass storage device under the name *USB Serial Device*.

Driver Verification

To verify that your driver is working, you should see the difference in the following pictures after plugging in your SparkFun SAMD51 Thing Plus. Alternatively, once you have the Arduino IDE installed, you should also see a change in the number of available Serial/COM Ports (you may need to restart the Arduino IDE for the board to populate).

Windows

Check that the board shows up in your device manager. You can click the **Start** or ⊞ (**Windows**) button and type "device" to quickly search for the application. (*On Windows 10, the quick search function is picky on the spelling of the application you are searching for. For example, you may get results using "devi" and none for "device".)



*Screenshot of Window 10 Device Manager with the SAMD51 Processor Board on **COM25**. Click to enlarge.*

Mac OSX

Open the Terminal and run the following command `ls /dev/cu.*` in the Terminal and check for the following changes (your board may show up under a different device name). To open the Terminal, open your **Applications** folder, **Utilities** folder, then double-click on **Terminal**. Otherwise, press ⌘ (**Command**) + space bar (**Space Bar**) to launch Spotlight and type "Terminal," then double-click the search result.

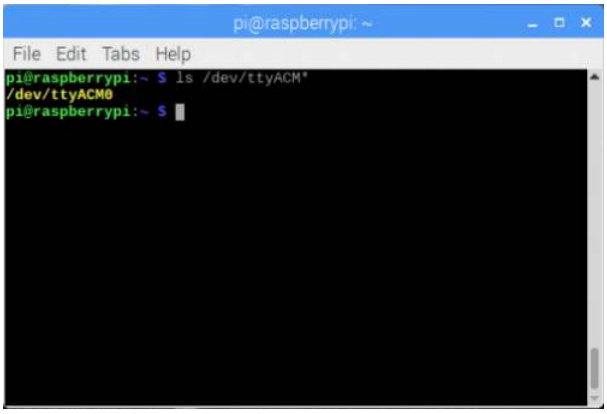
*Screenshot of Mac OSX terminal with the SAMD51 Processor Board on **cu.usbmodemFA121**. Click to enlarge.*

```
ls /dev/cu.*
```

Note: If you are still unsure of how to access the Terminal, watch this video or read this Apple support article.

Raspbian

Run the following command `ls /dev/ttyACM*` in the CLI/Terminal and check for the following changes (your board may show up under a different device name).



*Screenshot of Raspberry Pi CLI with the SAMD51 Processor Board on **ttyACM0**. Click to enlarge*

```
ls /dev/ttyACM*
```

Setting Up the Arduino IDE

Note: If this is your first time using Arduino IDE or board add-on, please review the following tutorials.

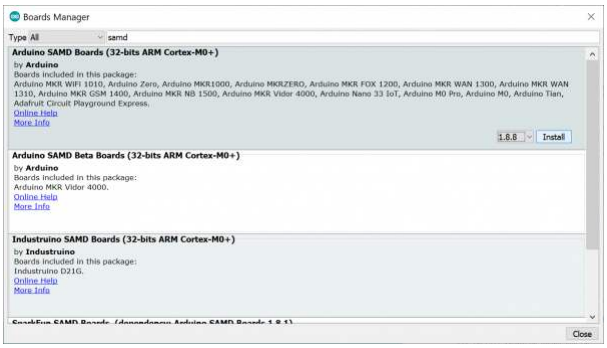
- Installing the Arduino IDE
- Installing Board Definitions in the Arduino IDE

Install Arduino SAMD Board Add-Ons

First, you'll need to install a variety of tools, including low-level ARM Cortex libraries full of generic code, arm-gcc to compile your code, and bossa to upload over the bootloader. These tools come packaged along with Arduino's SAMD board definitions for the Arduino Zero.

To install the Arduino SAMD board definitions, navigate to your board manager (**Tools** > **Board** > **Boards Manager...**), then find an entry for **Arduino SAMD Boards (32-bits ARM Cortex-M0+)** by typing **SAMD** in the search bar. Select it, and install the latest version (at the time of writing this tutorial, the board definitions work with *v1.8.9*).

> **Note:** In some cases, we noticed that you may need to hit the reset button twice to enter the bootloader mode in order to successfully upload code. This may be caused by conflicting drivers. Try uninstalling/reinstalling the drivers so that the Arduino IDE can automatically reset the board when uploading.



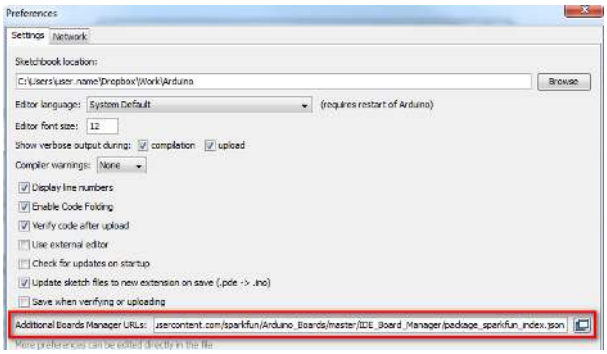
Downloading and installing the tools may take a couple minutes -- arm-gcc in particular will take the longest, it's about 250MB unpacked.

Once installed, Arduino-blue "Installed" text should appear next to the SAMD boards list entry.

Install SparkFun Board Add-On

Now that your ARM tools are installed, one last bit of setup is required to add support for the SparkFun SAMD boards. First, open your Arduino preferences (**File** > **Preferences**). Then find the **Additional Board Manager URLs** text box, and paste the below link in:

```
https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json
```

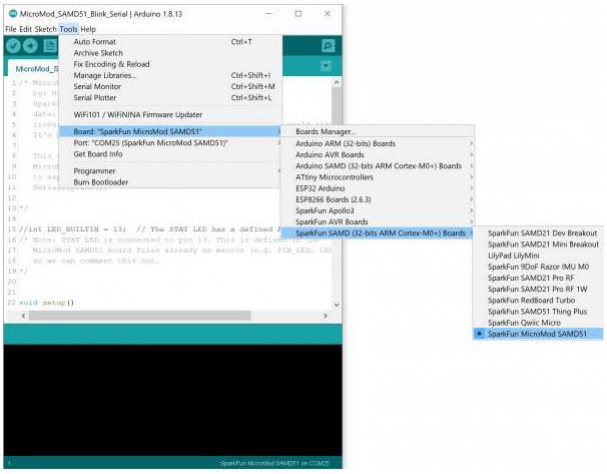


Then hit "OK", and travel back to the **Board Manager** menu. You should (but probably won't) be able to find a new entry for **SparkFun SAMD Boards**. If you don't see it, close the board manager and open it again. ¯_(ツ)_/¯.

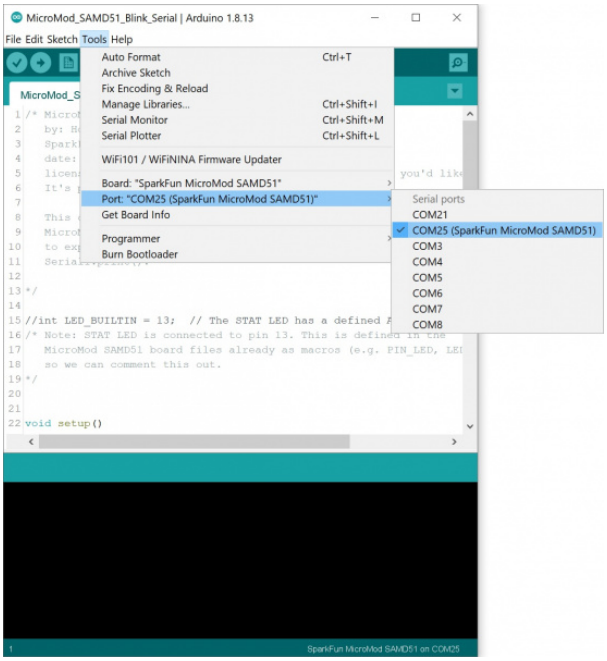
This installation should be much faster; you've already done the heavy lifting in the previous section. When writing this tutorial, the board version used in this tutorial should be **v1.7.6**. You may have a higher version as the board if there are any updates.

Select the Board and Serial Port

Once the board is installed, you should see a new entry in your **Tools** > **Board** list. **Select your SparkFun MicroMod SAMD51**.



Finally, select the SparkFun MicroMod SAMD51's **port** when the board is connected to your computer. Navigate back up to the **Tool** > **Port** menu. The port menu will have the SparkFun MicroMod SAMD51's port, labeled as such here. On a Windows machine, the serial port should come in the form of "**COM#**". On a Mac or Linux machine, the port will look like "**/dev/cu.usbmodem####**".



In the picture above my MicroMod SAMD51 is listed under **COM25**; this is because I've plugged in about as many microcontrollers into my computer, but the one in your port menu should be smaller.

Arduino Examples

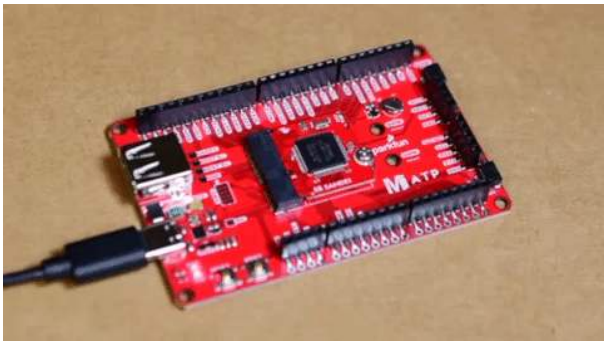
The SAMD51 is a powerful microcontroller. Here are a few examples that highlight some of its functionality.

Example 1: Blink and Hello World!

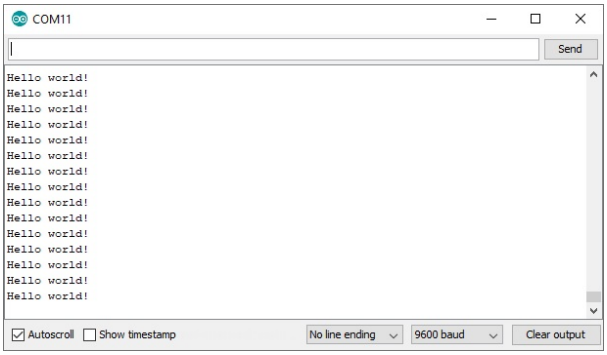
Now that you have a Processor Board secure in the Carrier Board, let's upload a combined sketch to the board. We will combine the blink and Hello World! sketches into one. Copy and paste the following code in the Arduino IDE. Head to **Tools** > **Board** to select the correct board definition (in this case, **SparkFun MicroMod SAMD51**). Select the correct COM port that the board enumerated to. Hit upload.

```
/* MicroMod SAMD51 Test Code
   by: Ho Yun "Bobby" Chan
   SparkFun Electronics
   date: October 7, 2020
   license: Public Domain - please use this code however you'd like.
   It's provided as a learning tool.

   This code is provided to show how to control the SparkFun
   MicroMod SAMD51's STAT LED within a sketch. It also serves
   to explain the difference between Serial.print() and
   Serial1.print().

*/

//int LED_BUILTIN = 13;  // The STAT LED has a defined Arduino pin
/* Note: STAT LED is connected to pin 13. This is defined in the
   MicroMod SAMD51 board files already as macros (e.g. PIN_LED, LED_BUILTIN)
   so we can comment this out.
*/


void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);  // Set RX LED as an output
  // TX LED is set as an output behind the scenes

  Serial.begin(9600); //This pipes to the serial monitor
  Serial.println("Initialize Serial Monitor");

  Serial1.begin(9600); //This is the UART, pipes to sensors attached to board
  Serial1.println("Initialize Serial Hardware UART Pins");
}

void loop()
{
  Serial.println("Hello world!");  // Print "Hello World" to the Serial Monitor
  Serial1.println("Hello! Can anybody hear me?");  // Print "Hello!" over hardware UART

  digitalWrite(LED_BUILTIN, LOW);   // set the LED_BUILTIN ON
  delay(1000);              // wait for a second

  digitalWrite(LED_BUILTIN, HIGH);    // set the LED_BUILTIN OFF
  delay(1000);              // wait for a second
}
```

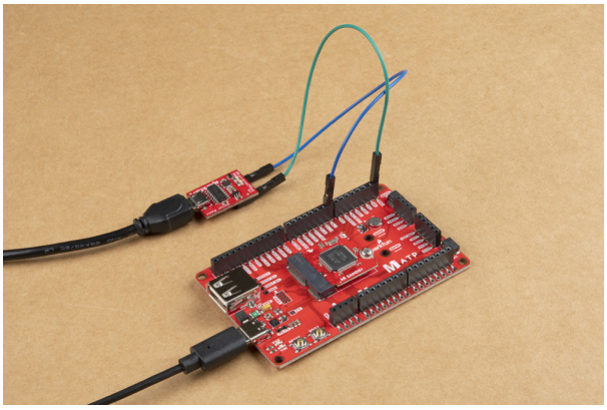
Check out the MicroMod SAMD51's built-in LED. The LED should blink every second.



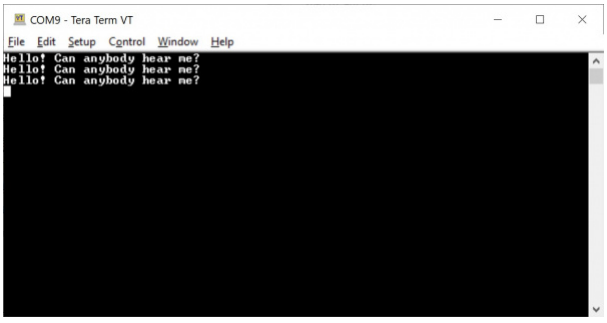
Open the Arduino IDE's serial monitor set at **9600** baud. You should see every programmer's favorite two-word phrase.



Want to go the extra mile? Grab a 3.3V USB-to-serial converter (in this case, we are using the Serial Basic Breakout CH340). Then connect GND to GND using a M/M jumper wire. Add another M/M jumper wire between TX from the MicroMod ATP Carrier board and the RXI pin of the USB-to-serial converter. Connect the appropriate USB cable to the USB-to-serial converter.



Open another serial terminal (in this case, we'll use Tera Term on a Windows) and connect to the USB-to-serial converter at **9600** baud. You should see a different message being sent from the hardware serial pin.



Example 2: Qwiic-Enabled Device Qwiic Distance Sensor (VL53L1X)

Note: If you have not previously installed an Arduino library, please check out our installation guide.

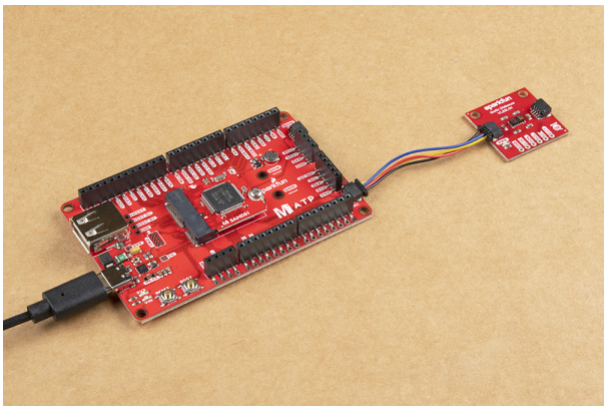
There's a plethora of Qwiic-enabled devices available to connect. In this example, we will be using a Qwiic distance sensor to test. If you have not already, head over to the Qwiic Distance Sensor's (VL53L1X) tutorial to install the library.

Qwiic Distance Sensor (VL53L1X) Hookup Guide

JUNE 18, 2018

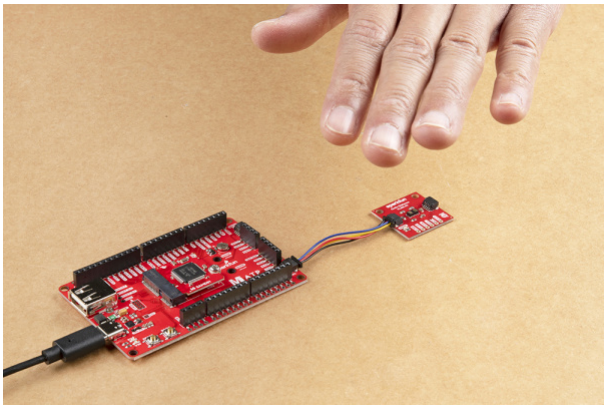
The Qwiic VL53L1X time of flight sensor is capable of several modes, as well as having a range of 4M. Let's hook it up and find out just how far away that thing over there is.

Connect the distance sensor to the Qwiic connector. Make sure to connect to the one labeled **I2C**. There is a second port labeled as **I2C1**. You would need to adjust the code in order to use the second I^2C port.

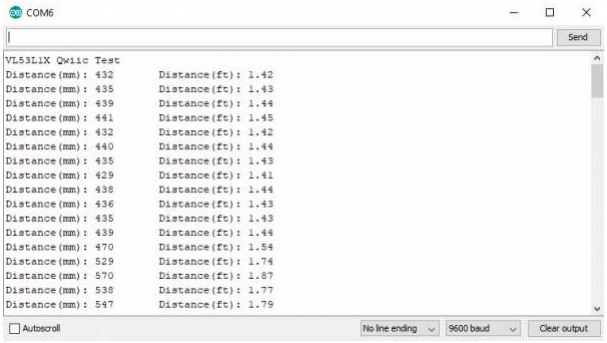


For simplicity, we'll use the first example just like the tutorial. Open up the first example by heading to **File** > **Examples** > **SparkFun VL53L1x 4M Laser Distance Sensor** > **Example1_ReadDistance**. If you have not already, head to **Tools** > **Board** to select the correct board definition (in this case, **SparkFun MicroMod SAMD51**). Select the correct COM port that the board enumerated to. Hit upload.

Open a the Arduino Serial Monitor at **9600** baud and start moving your hand over the sensor.



You should see an output similar to the image below. The values will vary depending on the distance away from the sensor.



What's next? Try combining the distance sensor example with the HID keyboard/mouse library to wake up a Raspberry Pi from it's screensaver whenever an object is in front of the sensor. Better yet, try adding some buttons and writing code to make a game controller for a computer.

Resources and Going Further

Now that you've successfully got your MicroMod SAMD51 Processor Board up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- Board Dimensions (PNG)
- Datasheet (PDF)
- GitHub

Need some inspiration for your next project? Check out some of these related tutorials using HID mouse/keyboard or adding more SERCOM Ports for your SAMD51:

Wireless Joystick Hookup Guide

A hookup guide for the SparkFun Wireless Joystick Kit.

Adding More SERCOM Ports for SAMD Boards

How to setup extra SPI, UART, and I2C serial ports on a SAMD-based boards.

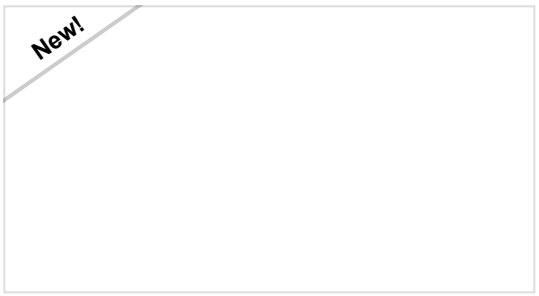
Keyboard Shortcut, Qwiic Keypad

A simple project using the Qwiic Keypad and the RedBoard Turbo to create your own custom hotkey-pad.

Qwiic Pro Micro USB-C (ATmega32U4) Hookup Guide

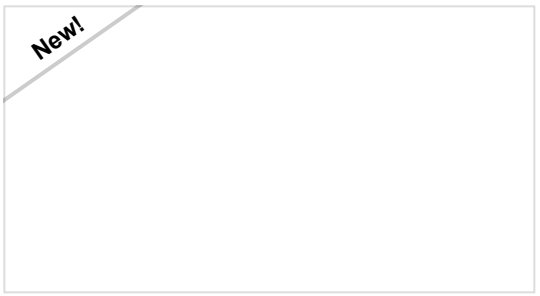
An overview of the ATmega32U4-based Qwiic Pro Micro USB-C, how to install it, and how to use it with Arduino.

Or check out other tutorials with MicroMod:



MicroMod Artemis Processor Board Hookup Guide

Get started with the Artemis MicroMod Processor Board in this tutorial!

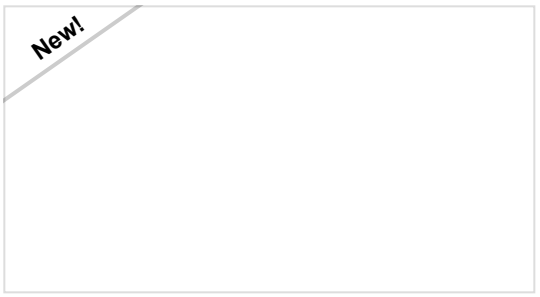


MicroMod ESP32 Processor Board Hookup Guide

A short Hookup Guide to get started with the SparkFun MicroMod ESP32 Processor Board

SparkFun MicroMod Input and Display Carrier Board Hookup Guide

A short Hookup Guide to get started with the SparkFun MicroMod Input and Display Carrier Board



MicroMod Machine Learning Carrier Board Hookup Guide

Get hacking with this tutorial on our Machine Learning Carrier Board!