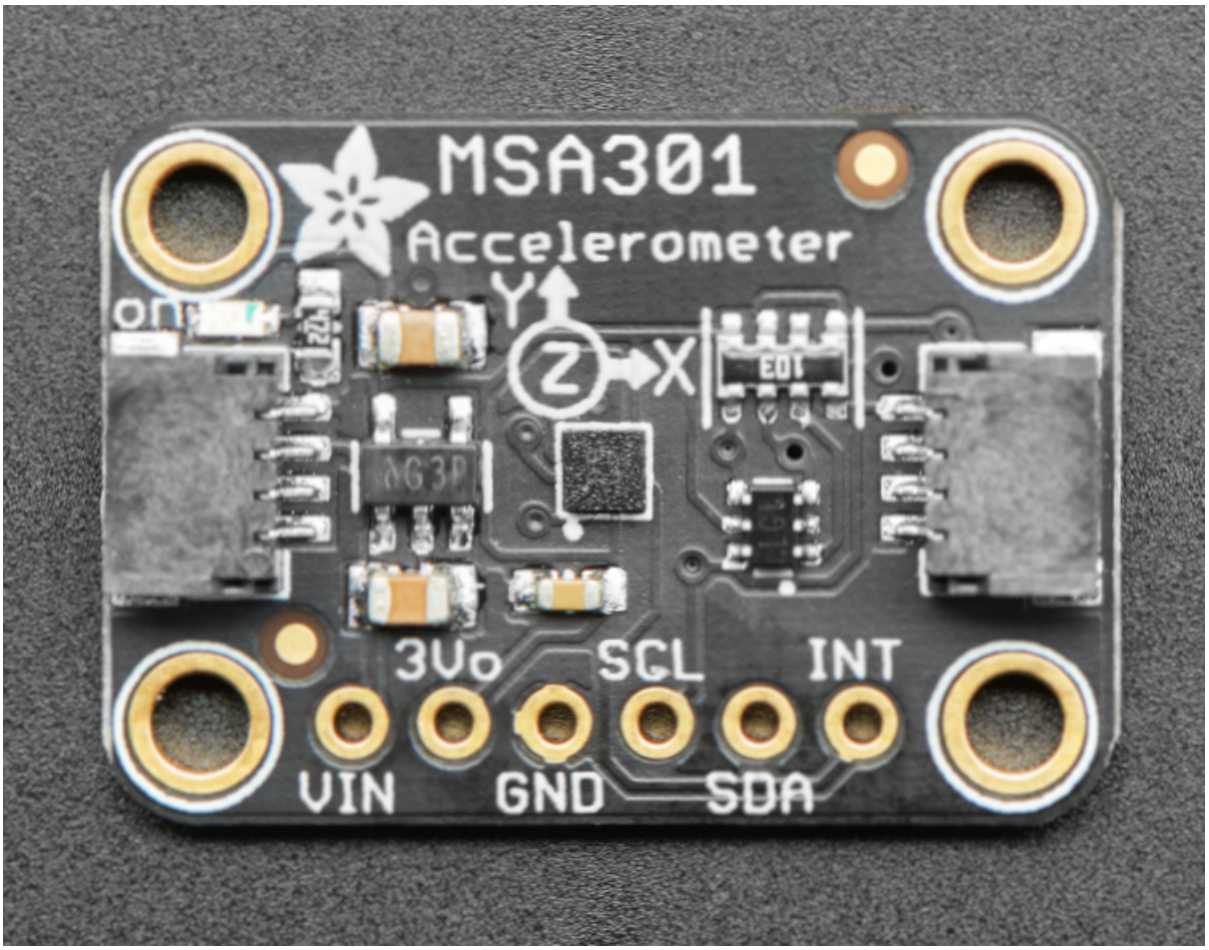
Adafruit MSA301 Triple Axis Accelerometer

Created by Bryan Siepert

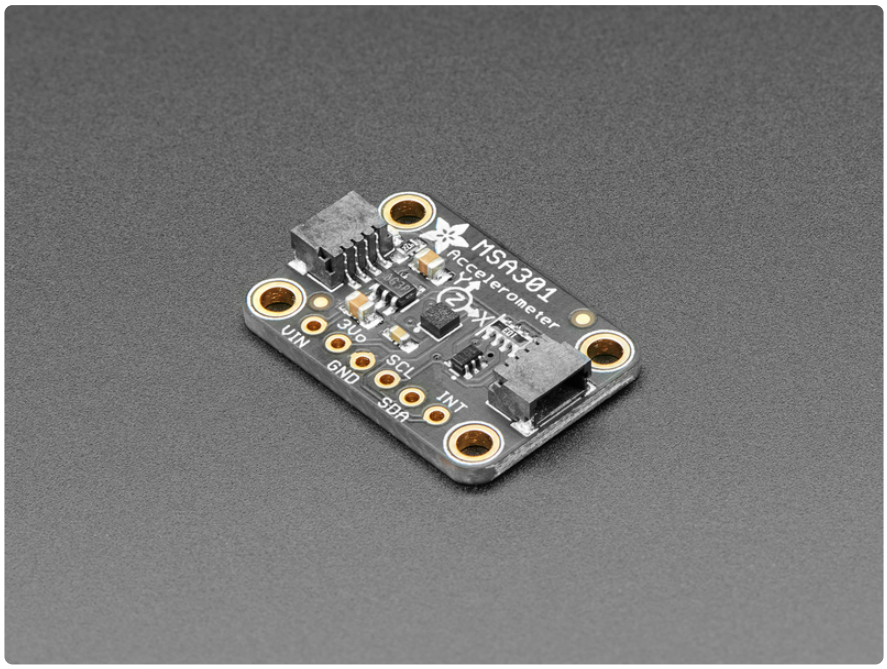


https://learn.adafruit.com/msa301-triple-axis-accelerometer

Last updated on 2024-03-08 03:33:08 PM EST

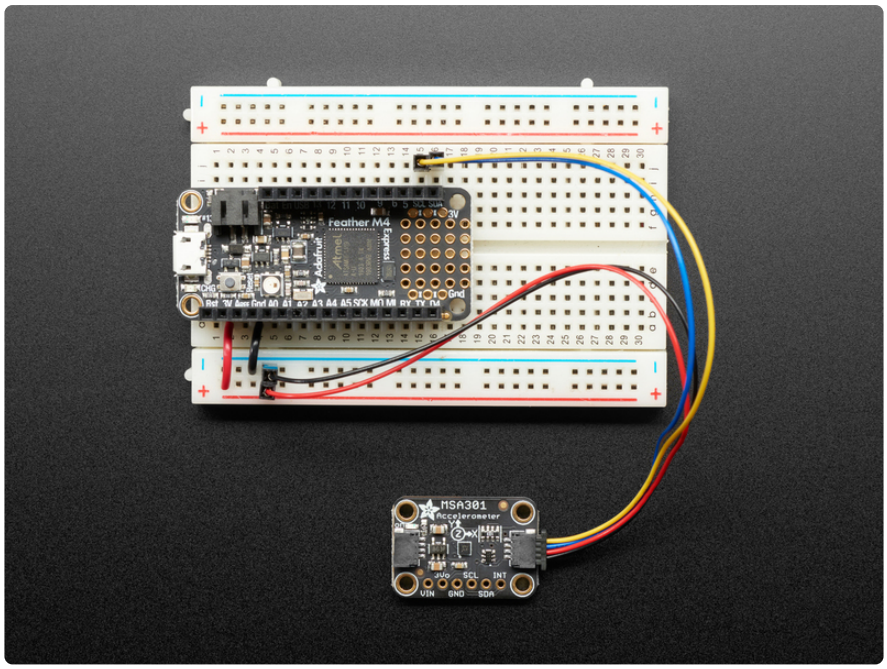
Table of Contents

Overview

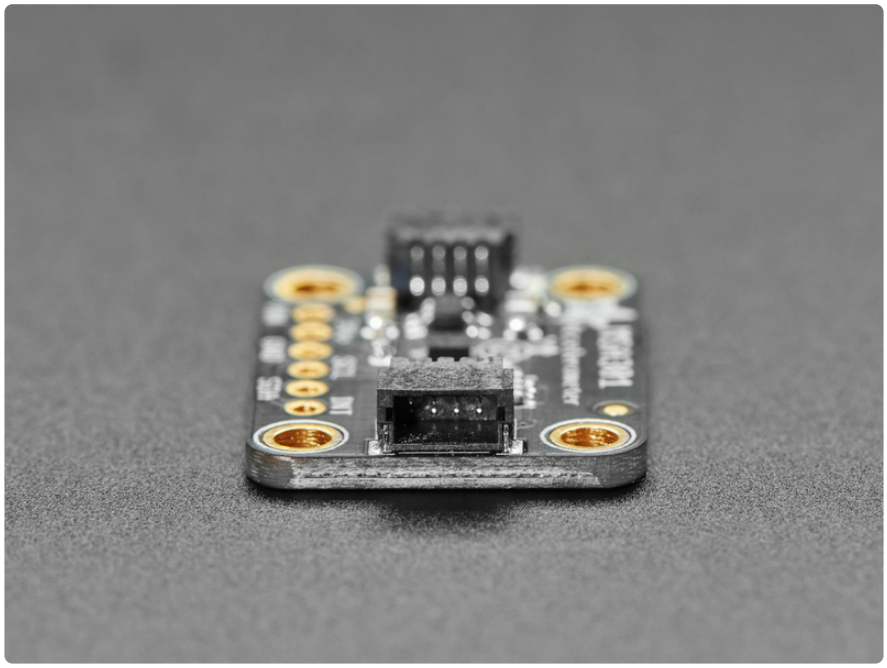


The MSA301 is a super small and low cost **triple-axis accelerometer**. It's inexpensive, but has just about every 'extra' you'd want in an accelerometer:

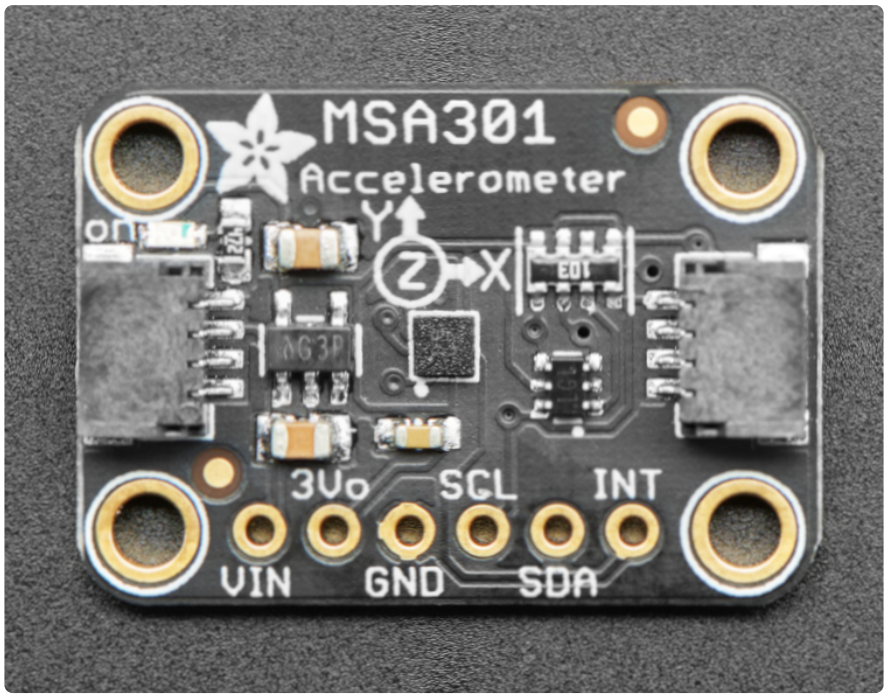
- Three axis sensing, 14-bit resolution
- ±2g/±4g/±8g/±16g selectable scaling
- I2C interface on fixed I2C address 0x26
- Interrupt output
- Multiple data rate options from 1 Hz to 500 Hz
- As low as 2uA current draw in low power mode (just the chip itself, not including any supporting circuitry)
- Tap, Double-tap, orientation & freefall detection



To make life easier, so you can focus on your important work, we've taken the MSA301 and put it onto a breakout PCB along with support circuitry to let you use this little wonder with 3.3V (Feather/Raspberry Pi) or 5V (Arduino/ Metro 328) logic levels. Additionally, since it speaks I2C, you can easily connect it up with two wires (plus power and ground!). We've even included SparkFun qwiic (https://adafru.it/Fpw) compatible STEMMA QT **(https://adafru.it/Ft4)** connectors for the I2C bus, so **you don't even need to solder!** Just wire up to your favorite microcontroller and you can use our CircuitPython/Python (https://adafru.it/FDq)or Arduino drivers to easily interface with the MSA301 (https://adafru.it/FDr) and get motion measurements ASAP.



Pinouts



Power Pins

- **Vin** - this is the power pin. Since the sensor chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

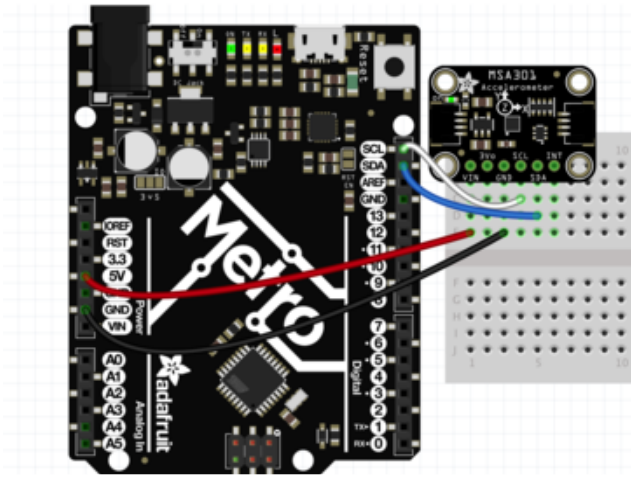
I2C Logic Pins

- **SCL** - this is the I2C clock pin, connect to your microcontroller's I2C clock line.
- **SDA** - this is the I2C data pin, connect to your microcontroller's I2C data line
- STEMMA QT **(https://adafru.it/Ft4)** - These connectors allow you to connectors to dev boards with **STEMMA QT** connectors or to other things with various associated accessories (https://adafru.it/Ft6)

Other Pins

- **INT** -This is the interrupt pin. You can setup the MSA301 to pull this low when certain conditions are met such as taps being detected.

Arduino

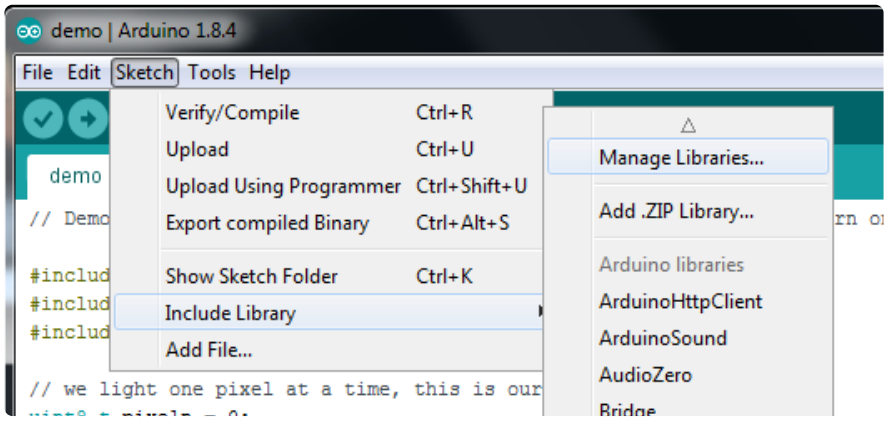


Connect **Metro or Arduino 5V** to **board VIN**
Connect **Metro or Arduino GND** to **board GND**
Connect **Metro or Arduino SCL** to **board SCL**
Connect **Metro or Arduino SDA** to **board SDA**

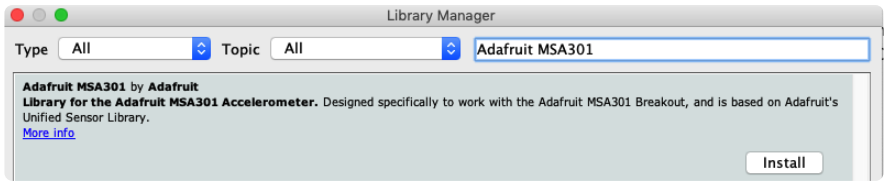
The final results should resemble the illustration above, showing an Adafruit Metro (http://adafru.it/2488) development board.

Installation

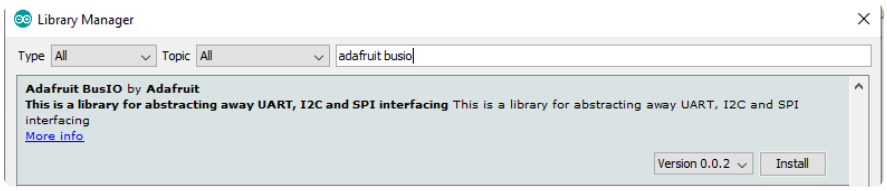
We've written a sensor library for you for the MSA301 which uses our **Adafruit_Sensor** and **BusIO** libraries. You can install both the **Adafruit_MSA301 Library** (https://adafru.it/FDr) and the **Adafruit BusIO Library** for Arduino using the Library Manager in the Arduino IDE:



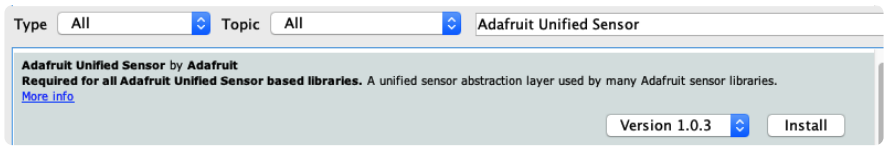
Click the **Manage Libraries ...** menu item, search for **Adafruit MSA301**, and select the **Adafruit_MSA301** library:



Then follow the same process for the **Adafruit BusIO** library:



Finally do the same for the **Adafruit Unified Sensor** library

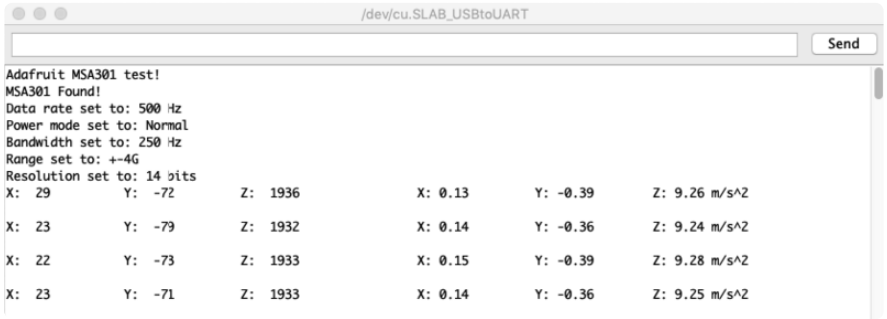


Accelerometer Demo

This first demo will show you how to get readings of what an accelerometer does best: measure acceleration!

Open up **File -> Examples -> Adafruit MSA301 -> acceldemo** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. You should see the the values for the current configuration settings printed on startup, followed by acceleration readings for the X, Y, and Z axes similar to this:



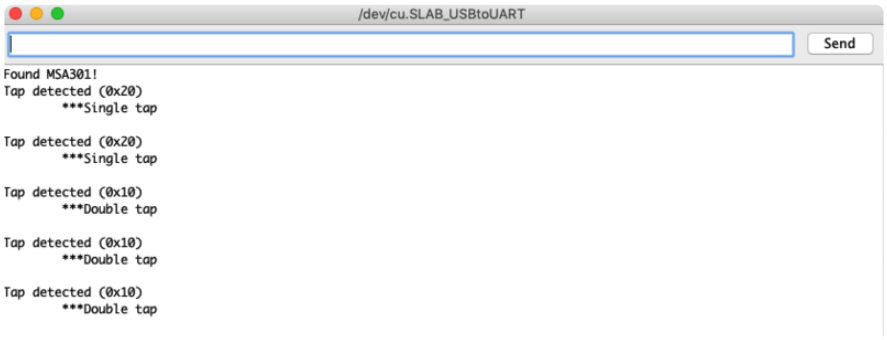
You can tilt the sensor in different directions to see the values for the different axes change.

Tap Demo

This next demo will demonstrate how to use the tap detection feature of the MSA301. By tapping the accelerometer or the PCB or something the PCB is attached to you can create a type of interface.

Open up **File -> Examples -> Adafruit MSA301 -> tapdemo** and upload to your Arduino wired up to the sensor.

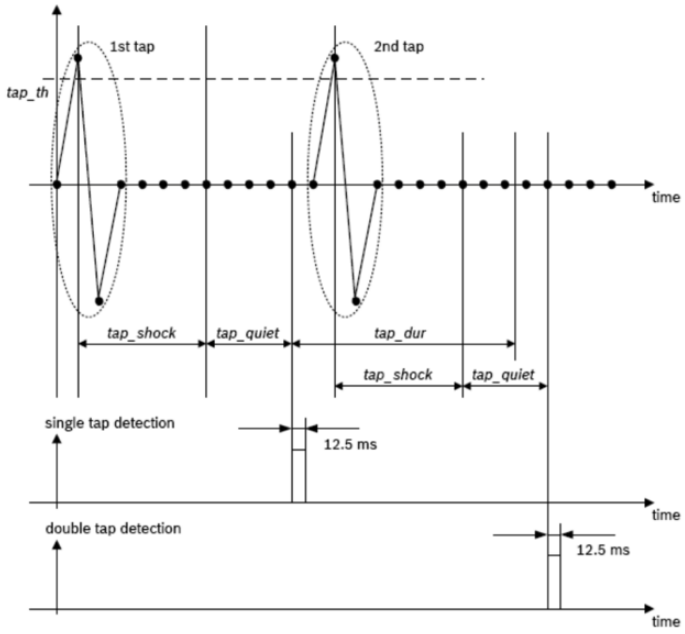
Once the sketch is loaded, open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. When you tap, you should see the sketch report that a tap was detected:



The tap detection works by looking for when one of the axes has an acceleration higher than a certain **threshold** followed by a **quiet** period where there are no more acceleration spikes above the **threshold**.

After an initial "spike" of acceleration above the **threshold,** any additional spikes within a window of time defined by the **shock** parameter are ignored.

For double tap detection, after an initial tap has been detected, another tap event within the window of time specified by the **duration** parameter will trigger a double tap interrupt if they are enabled.



The tap detection parameters are set by calling **setClick:**

```
msa.setClick(false, false, MSA301_TAPDUR_250_MS, 25);
// Method signature:
// void setClick(bool tap_quiet, bool tap_shock, msa301_tapduration_t tapduration,
uint8_t tapthresh);
```

The **threshold** is in 'raw' values so you have to adjust it based on the scale/range you've configured for the sensor. Larger numbers are less sensitive, you'll really need to just tweak as necessary for your project.

The **duration** parameter **only applies to double tap detection.** It is specified by passing an msa301_tapduration_t **(https://adafru.it/FDs).** A shorter duration will require a quicker double tap to be detected.

The **quiet** duration is either **20ms** if **true** or **30ms** if **false**. Similarly the **shock** duration is either **70ms** if **true** or **50ms** if **false**.

Additionally after configuring tap detection, you will need to enable the tap detection interrupts:

```
msa.enableInterrupts(true, true); // enable single, double tap
```

If you wish to have the tap detection activate the interrupt pin, you will have to enable that as well:

```
msa.mapInterruptPin(true, true); // enable the INT pin for single, double tap
```

Acceleration Example Code

The following code is part of the standard library and illustrates the basic function of measuring acceleration:

```
// Basic demo for accelerometer readings from Adafruit MSA301

#include <Wire.h>
#include <Adafruit_MSA301.h>
#include <Adafruit_Sensor.h>

Adafruit_MSA301 msa;

void setup(void) {
  Serial.begin(115200);
  while (!Serial) delay(10);     // will pause Zero, Leonardo, etc until serial
console opens

  Serial.println("Adafruit MSA301 test!");

  // Try to initialize!
  if (! msa.begin()) {
    Serial.println("Failed to find MSA301 chip");
    while (1) { delay(10); }
  }
  Serial.println("MSA301 Found!");

  //msa.setDataRate(MSA301_DATARATE_31_25_HZ);
  Serial.print("Data rate set to: ");
  switch (msa.getDataRate()) {
    case MSA301_DATARATE_1_HZ: Serial.println("1 Hz"); break;
    case MSA301_DATARATE_1_95_HZ: Serial.println("1.95 Hz"); break;
    case MSA301_DATARATE_3_9_HZ: Serial.println("3.9 Hz"); break;
    case MSA301_DATARATE_7_81_HZ: Serial.println("7.81 Hz"); break;
    case MSA301_DATARATE_15_63_HZ: Serial.println("15.63 Hz"); break;
    case MSA301_DATARATE_31_25_HZ: Serial.println("31.25 Hz"); break;
    case MSA301_DATARATE_62_5_HZ: Serial.println("62.5 Hz"); break;
    case MSA301_DATARATE_125_HZ: Serial.println("125 Hz"); break;
    case MSA301_DATARATE_250_HZ: Serial.println("250 Hz"); break;
    case MSA301_DATARATE_500_HZ: Serial.println("500 Hz"); break;
    case MSA301_DATARATE_1000_HZ: Serial.println("1000 Hz"); break;
  }

  //msa.setPowerMode(MSA301_SUSPENDMODE);
  Serial.print("Power mode set to: ");
  switch (msa.getPowerMode()) {
    case MSA301_NORMALMODE: Serial.println("Normal"); break;
    case MSA301_LOWPOWERMODE: Serial.println("Low Power"); break;
    case MSA301_SUSPENDMODE: Serial.println("Suspend"); break;
  }

  //msa.setBandwidth(MSA301_BANDWIDTH_31_25_HZ);
  Serial.print("Bandwidth set to: ");
  switch (msa.getBandwidth()) {
    case MSA301_BANDWIDTH_1_95_HZ: Serial.println("1.95 Hz"); break;
    case MSA301_BANDWIDTH_3_9_HZ: Serial.println("3.9 Hz"); break;
    case MSA301_BANDWIDTH_7_81_HZ: Serial.println("7.81 Hz"); break;
    case MSA301_BANDWIDTH_15_63_HZ: Serial.println("15.63 Hz"); break;
    case MSA301_BANDWIDTH_31_25_HZ: Serial.println("31.25 Hz"); break;
    case MSA301_BANDWIDTH_62_5_HZ: Serial.println("62.5 Hz"); break;
    case MSA301_BANDWIDTH_125_HZ: Serial.println("125 Hz"); break;
    case MSA301_BANDWIDTH_250_HZ: Serial.println("250 Hz"); break;
    case MSA301_BANDWIDTH_500_HZ: Serial.println("500 Hz"); break;
  }
```

```
  //msa.setRange(MSA301_RANGE_2_G);
  Serial.print("Range set to: ");
  switch (msa.getRange()) {
    case MSA301_RANGE_2_G: Serial.println("+-2G"); break;
    case MSA301_RANGE_4_G: Serial.println("+-4G"); break;
    case MSA301_RANGE_8_G: Serial.println("+-8G"); break;
    case MSA301_RANGE_16_G: Serial.println("+-16G"); break;
  }

  //msa.setResolution(MSA301_RESOLUTION_14 );
  Serial.print("Resolution set to: ");
  switch (msa.getResolution()) {
    case MSA301_RESOLUTION_14: Serial.println("14 bits"); break;
    case MSA301_RESOLUTION_12: Serial.println("12 bits"); break;
    case MSA301_RESOLUTION_10: Serial.println("10 bits"); break;
    case MSA301_RESOLUTION_8: Serial.println("8 bits"); break;
  }
}

void loop() {
  msa.read();      // get X Y and Z data at once
  // Then print out the raw data
  Serial.print("X:  "); Serial.print(msa.x);
  Serial.print("  \tY:  "); Serial.print(msa.y);
  Serial.print("  \tZ:  "); Serial.print(msa.z);
  delay(100);

  /* Or....get a new sensor event, normalized */
  sensors_event_t event;
  msa.getEvent(&event);

  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("\t\tX: "); Serial.print(event.acceleration.x);
  Serial.print(" \tY: "); Serial.print(event.acceleration.y);
  Serial.print(" \tZ: "); Serial.print(event.acceleration.z);
  Serial.println(" m/s^2 ");

  Serial.println();

  delay(100);
}
```

Tap Detection Example Code

Here is the full code for the tap detection example:

```
// Basic demo for tap/doubletap readings from Adafruit MSA301

#include <Adafruit_MSA301.h>

Adafruit_MSA301 msa;

void setup() {
  Serial.begin(115200);
  while (!Serial) { delay(10); }

  // Try to initialize!
  if (! msa.begin()) {
    Serial.println("Failed to find MSA301 chip");
    while (1) { delay(10); }
  }
  Serial.println("Found MSA301!");

  msa.setPowerMode(MSA301_NORMALMODE);
```

```
  msa.setDataRate(MSA301_DATARATE_1000_HZ);
  msa.setBandwidth(MSA301_BANDWIDTH_500_HZ);
  msa.setRange(MSA301_RANGE_2_G);
  msa.setResolution(MSA301_RESOLUTION_14 );

  msa.setClick(false, false, MSA301_TAPDUR_250_MS, 25);
  msa.enableInterrupts(true, true);  // enable single/double tap
}

void loop() {

  uint8_t motionstat = msa.getMotionInterruptStatus();
  if (motionstat) {
    Serial.print("Tap detected (0x"); Serial.print(motionstat, HEX);
Serial.println(")");
    if (motionstat & (1<<5)) {
      Serial.println("\t***Single tap");
    }
    if (motionstat & (1<<4)) {
      Serial.println("\t***Double tap");
    }
    Serial.println("");
  }
  delay(10);

}
```

Arduino Docs

Arduino Docs (https://adafru.it/FBr)

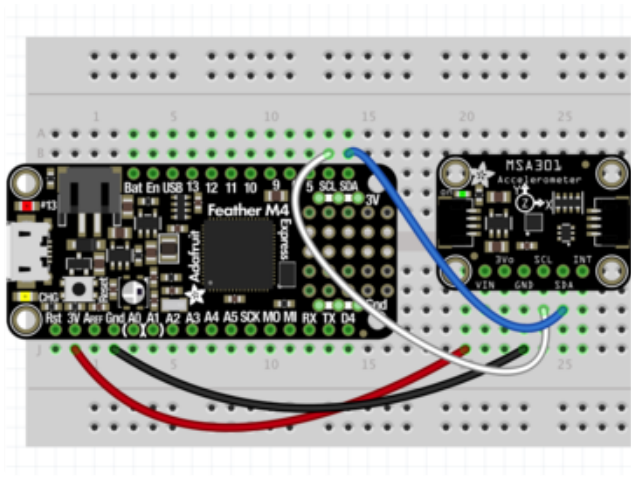
Python & CircuitPython

It's easy to use the MSA301 sensor with CircuitPython and the Adafruit CircuitPython MSA301 (https://adafru.it/FDq) module. This module allows you to easily write Python code that reads the acceleration and adjust the measurement settings.

You can use this sensor with any CircuitPython microcontroller board or with a Linux single board computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

First wire up a MSA301 to your board exactly as follows. Here is an example of the MSA301 wired to a Feather (http://adafru.it/3857) using I2C:



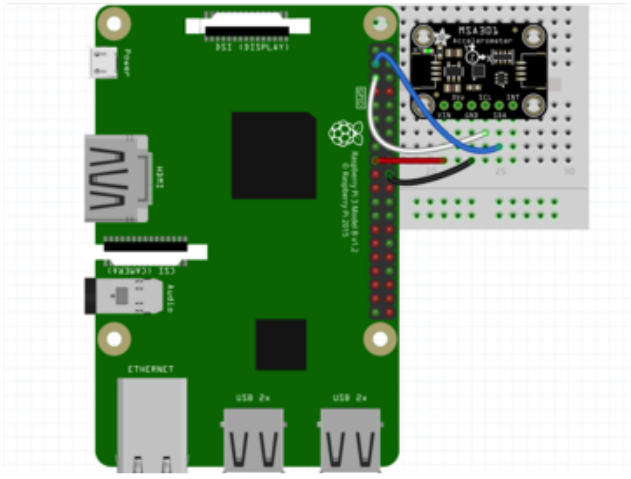
Board 3V to **sensor VIN**
Board GND to **sensor GND**
Board SCL to **sensor SCL**
Board SDA to **sensor SDA**

Note: This breakout includes pullup resistors on the I2C lines, no external pullups are required.

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi (http://adafru.it/3055). For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C:



Pi 3V3 to **sensor VIN**
Pi GND to **sensor GND**
Pi SCL to **sensor SCL**
Pi SDA to **sensor SDA**

CircuitPython Installation of MSA301 Library

You'll need to install the Adafruit CircuitPython MSA301 (https://adafru.it/FDq) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Amd) for your board.

Next 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/uap). Our CircuitPython starter guide has a great page on how to install the library bundle (https://adafru.it/ABU).

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- **adafruit_msa301.mpy**
- **adafruit_bus_device**
- **adafruit_register**

Before continuing make sure your board's **lib** folder or root filesystem has the **adafruit_msa301.mpy, adafruit_bus_device**, and **adafruit_register** files and folders copied over.

Next connect to the board's serial REPL (https://adafru.it/Awz)so you are at the CircuitPython `>>>` prompt.

Python Installation of MSA301 Library

You'll need to install the **Adafruit_Blinka** library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-msa301`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the acceleration measurements from the board's Python REPL.

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import time
import board
import adafruit_msa301

i2c = board.I2C()

msa = adafruit_msa301.MSA301(i2c)
```

```
>>> import time
>>> import board
>>> import adafruit_msa301
>>> i2c = board.I2C()
>>> msa = adafruit_msa301.MSA301(i2c)
```

Now you're ready to read the **acceleration** values from the sensor using the property provided by the library.

Here is an example showing how to print the acceleration values:

```
>>> msa.acceleration
(0.177159, -0.790034, 9.78685)
>>>
```

You can find more details about what the library allows by reading the library documentation (https://adafru.it/FBs).

That's it! Using the MSA301 accelerometer with CircuitPython makes it easy to get started.

Tap Detection

Next, let's take a look at another neat feature of the MSA301: Tap detection!

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
Press any key to enter the REPL. Use CTRL-D to reload.
Adafruit CircuitPython 4.1.0 on 2019-08-03; Adafruit Metro M4 Express with samd51j19
>>> import time
>>> import board
>>> import busio
>>> from adafruit_msa301 import MSA301, TapDuration
>>> i2c = busio.I2C(board.SCL, board.SDA)
>>> msa = MSA301(i2c)
```

Next we'll configure tap detection by calling enable_tap_detection (https://adafru.it/FDt). Here we're giving an example of the different parameters you can tune, but calling it without any arguments will set it to detect single taps with reasonable settings defaults:

```
>>> msa.enable_tap_detection(tap_count=2, threshold=30, long_initial_window=True,
... long_quiet_window=False, double_tap_window=TapDuration.DURATION_250_MS)
```

Finally we set up a loop, calling the **tapped** property to check and print if a double tap has been detected, since we passed **2** to the **enable_tap_detection** (https://adafru.it/FDt)which specifies double tap detection:

```
>>>while True:
...     if msa.tapped:
...         print("Double Tap!")
...     time.sleep(0.001)
...
Double Tap!
Double Tap!
Double Tap!
Double Tap!
Double Tap!
Double Tap!
Double Tap!
Double Tap!
```

Acceleration Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
from adafruit_msa3xx import MSA301

i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
msa = MSA301(i2c)

while True:
    print("%f %f %f" % msa.acceleration)
    time.sleep(0.5)
```

Tap Detection Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
from adafruit_msa3xx import MSA301
```

```
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
msa = MSA301(i2c)

msa.enable_tap_detection()

while True:
    if msa.tapped:
        print("Single Tap!")
    time.sleep(0.01)
```

Python Docs

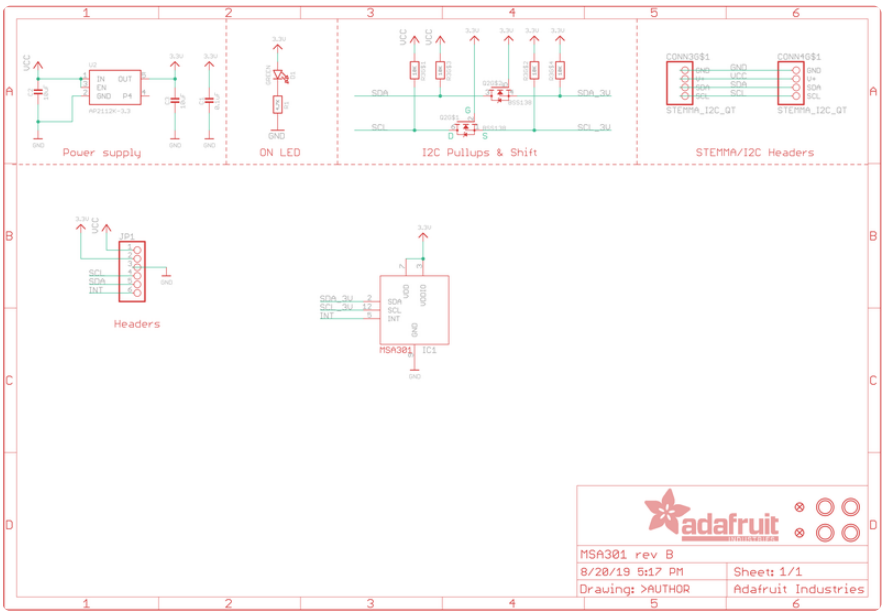
Python Docs (https://adafru.it/FBs)

Downloads

Files

MSA301 Datasheet (https://adafru.it/FDu)
EagleCAD files on GitHub (https://adafru.it/FDv)
3D CAD files on GitHub (https://adafru.it/FGM)
Fritzing object in Adafruit Fritzing Library (https://adafru.it/FDw)

Schematic



Fab Print