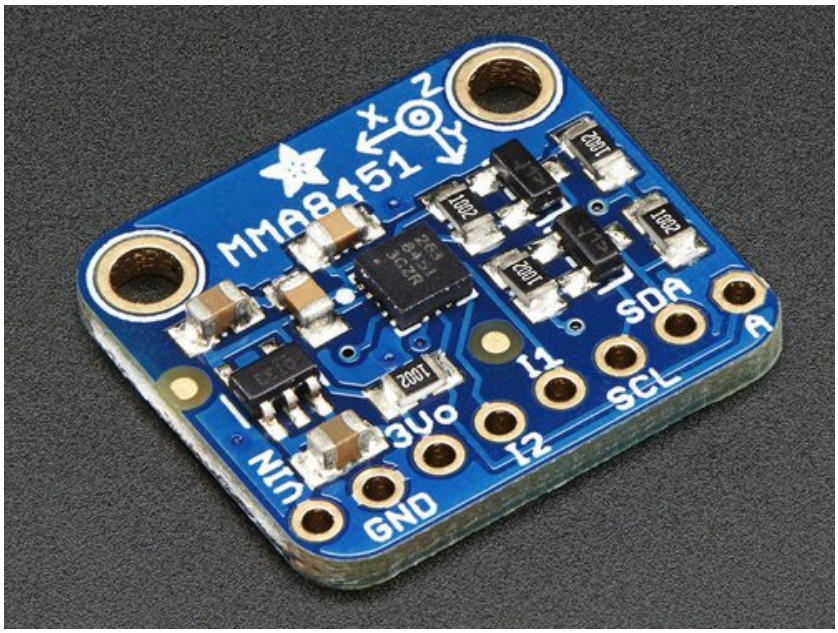
Adafruit MMA8451 Accelerometer Breakout

Created by lady ada



Last updated on 2016-08-15 08:16:17 PM UTC

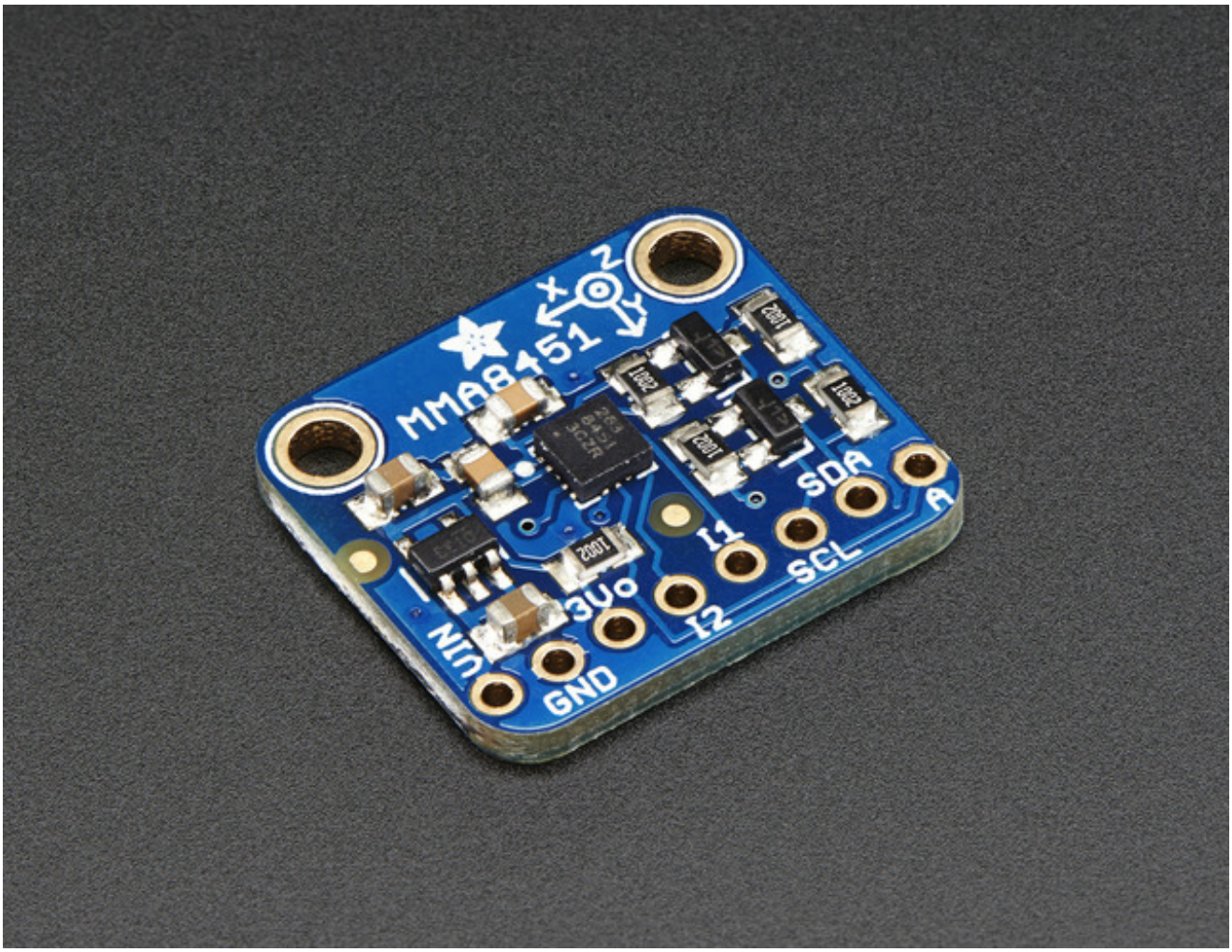
Guide Contents

Guide Contents 2
Overview 3
Pinouts 6
(http://adafru.it/dLn)Power Pins 6
I2C Pins 7
INT and ADDR Pins 7
Assembly 8
- Prepare the header strip: 8
- Add the breakout board: 9
- And Solder! 10

Wiring & Test 12
Download Adafruit_MMA8451 13
Download Adafruit_Sensor 13
Load Demo 13
Library Reference 16
- Set & Get Range 16
- Read Raw Count Data 17
- Reading Normalized Adafruit_Sensor data 17
- Read Orientation 17

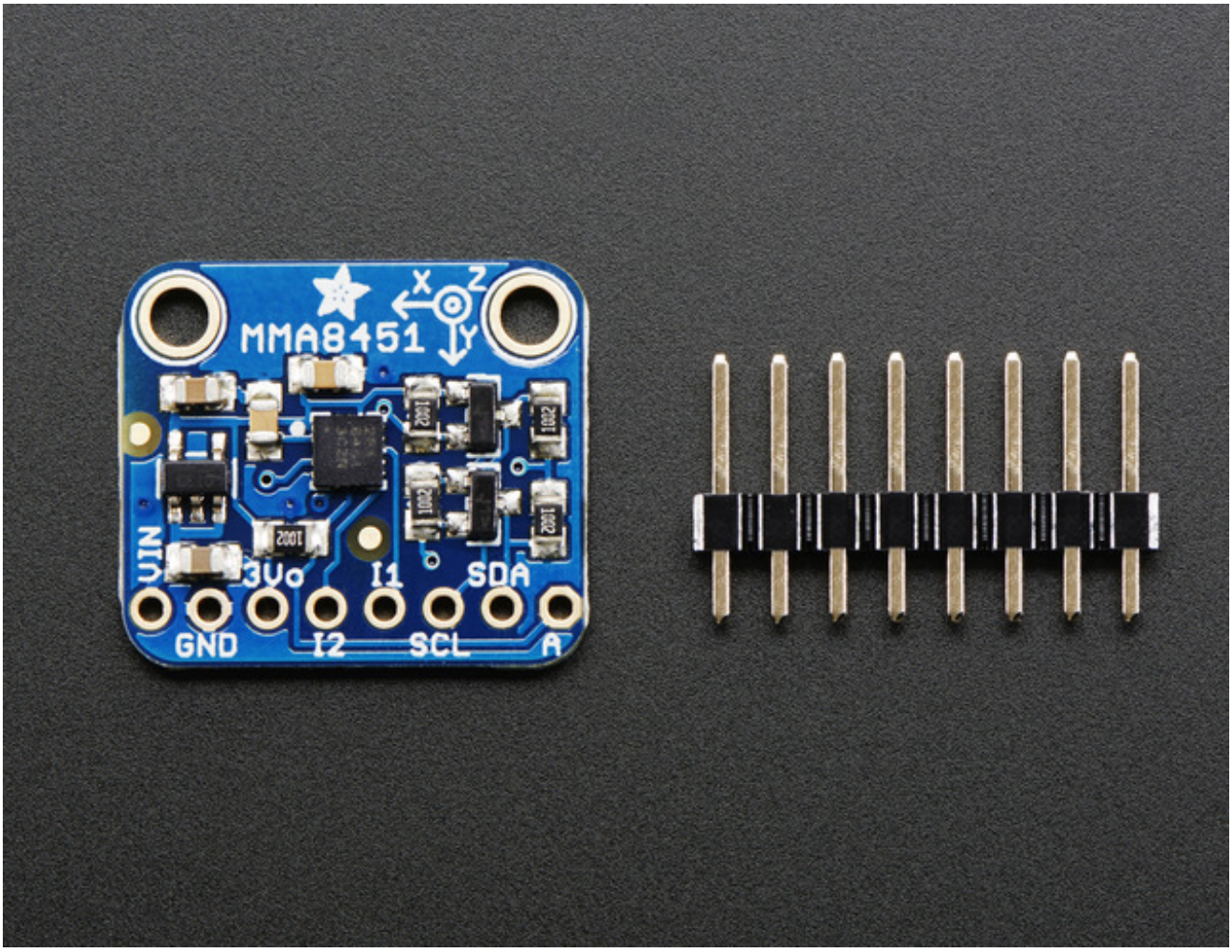
Downloads 19
Datasheet & Files 19
Schematics 19
Fabrication print 19

Overview

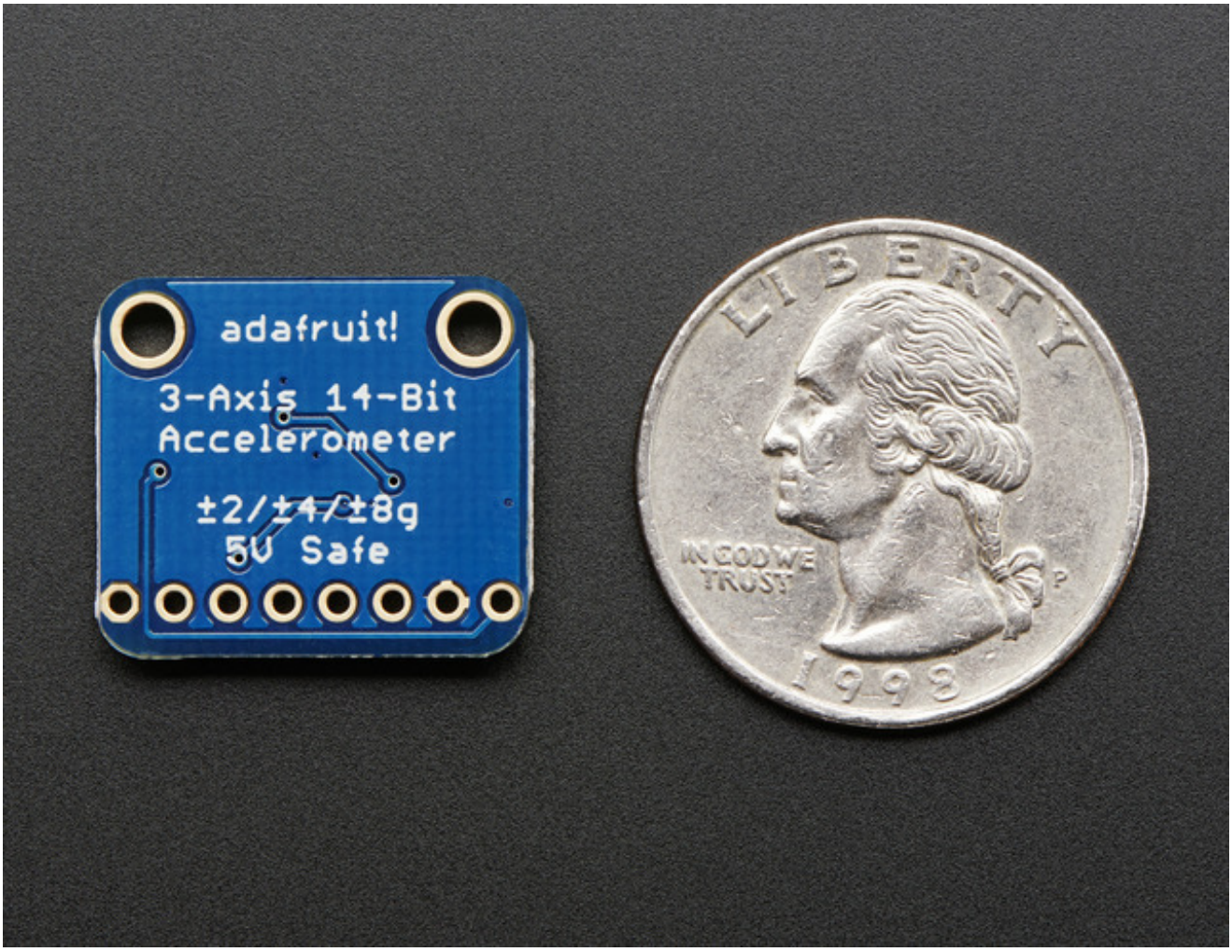


You can detect motion, tilt and basic orientation with a digital accelerometer - and the MMA8451 is a great accelerometer to start with. It's low cost, but high precision with 14-bit ADC. It has a wide usage range, from +-2g up to +-8g yet is easy to use with Arduino or another microcontroller

The MMA8451 is a miniature little accelerometer from Freescale, who are (by this point) masters at the accelerometer-design game. It's designed for use in phones, tablets, smart watches, and more, but works just as well in your Arduino project. Of the MMA8451/MMA8452/MMA8453 family, the MMA8451 is the most precise with a built in 14-bit ADC. The accelerometer also has built in tilt/orientation detection so i can tell you whether your project is being held in landscape or portrait mode, and whether it is tilted forward or back

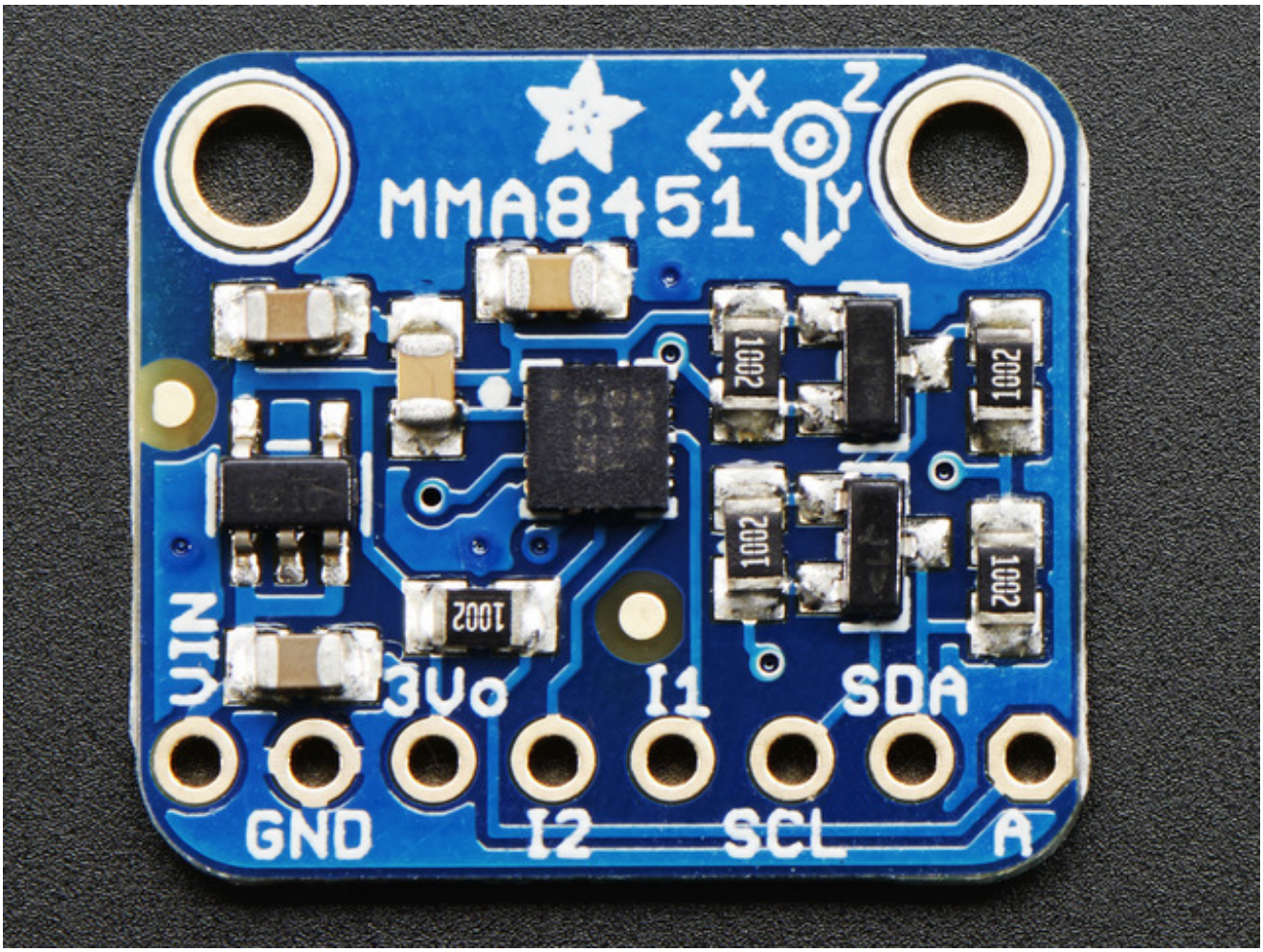


This sensor communicates over I2C so you can share it with a bunch of other sensors on the same two I2C pins. There's an address selection pin so you can have accelerometers share an I2C bus. Please note this chip requires repeated-start I2C support (in case you are looking to port this to another processor)



To get you going fast, we spun up a breakout board for this little guy. Since it's a 3V sensor, we add a low-dropout 3.3V regulator and level shifting circuitry on board. That means its perfectly safe for use with 3V or 5V power and logic.

Pinouts



The little chip in the middle of the PCB is the actual MMA8451 sensor that does all the motion sensing. We add all the extra components you need to get started, and 'break out' all the other pins you may want to connect to onto the PCB. For more details you can check out the schematics in the Downloads page.

(http://adafru.it/dLn)Power Pins

The sensor on the breakout requires 3V power. Since many customers have 5V microcontrollers like Arduino, we tossed a 3.3V regulator on the board. Its ultra-low dropout so you can power it from 3.3V-5V just fine.

- **Vin** - this is the power pin. Since the 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 micro 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

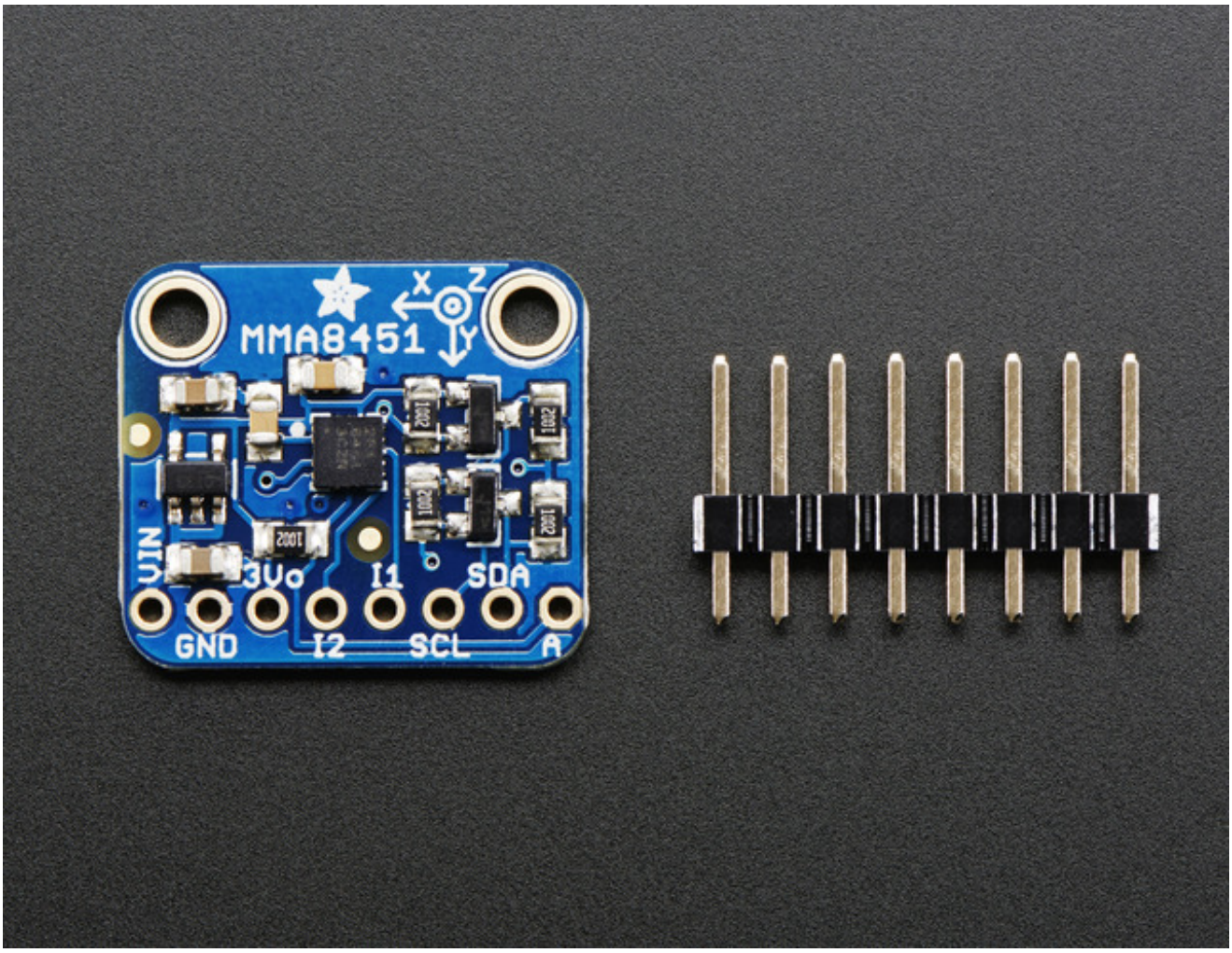
I2C Pins

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.

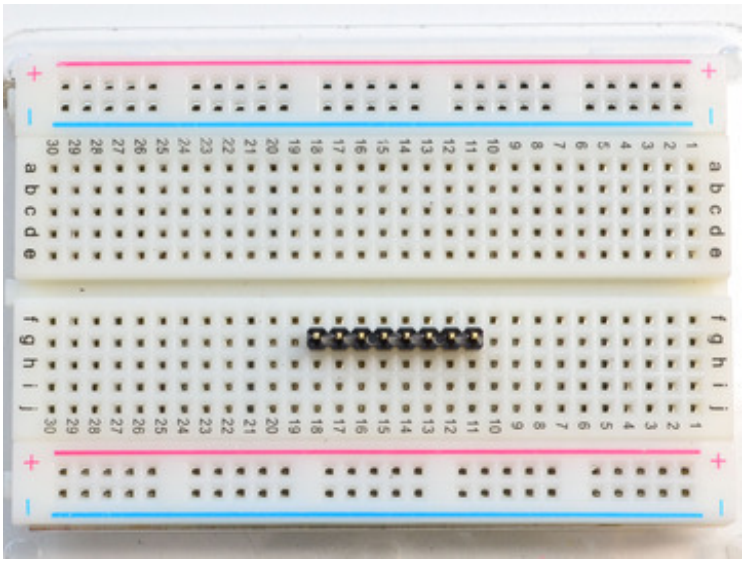
INT and ADDR Pins

- **A** is the I2C **A**ddress select pin. By default this is pulled down to ground with a 10K resistor, for an I2C address of 0x1C. You can also connect it to the 3Vo pin for an address of 0x1D
- **I1** and **I2** are the Interrupt #1 and #2 signal pins. These pins are for more advanced usage, where you want to be alerted by the chip say when data is ready to read, or if it detects a large motion. We don't have direct support in the example Arduino library for these pins, so please check the datasheet for the I2C commands

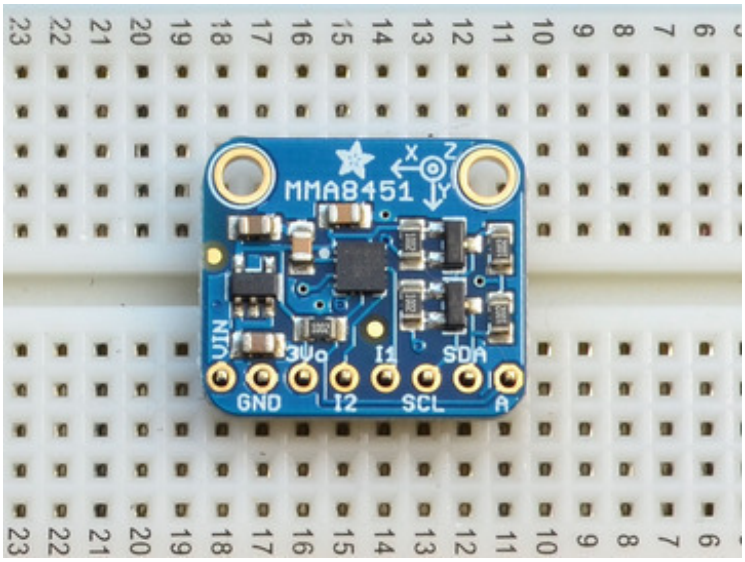
Assembly



Prepare the header strip:

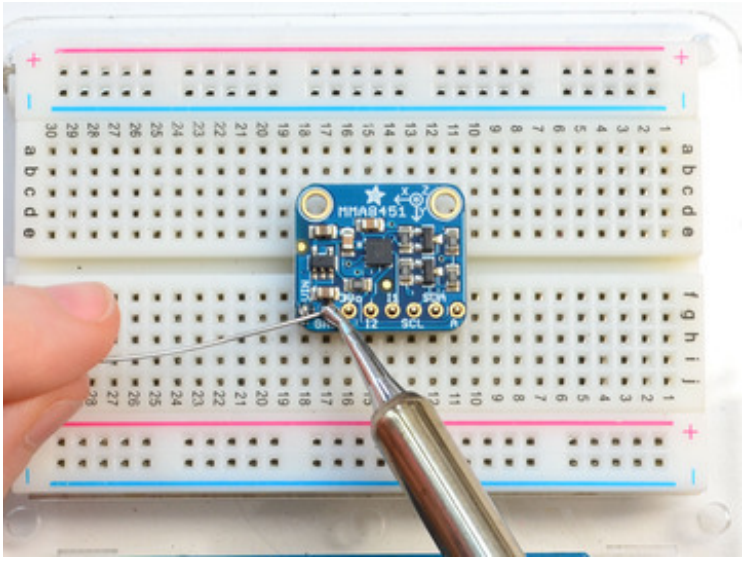
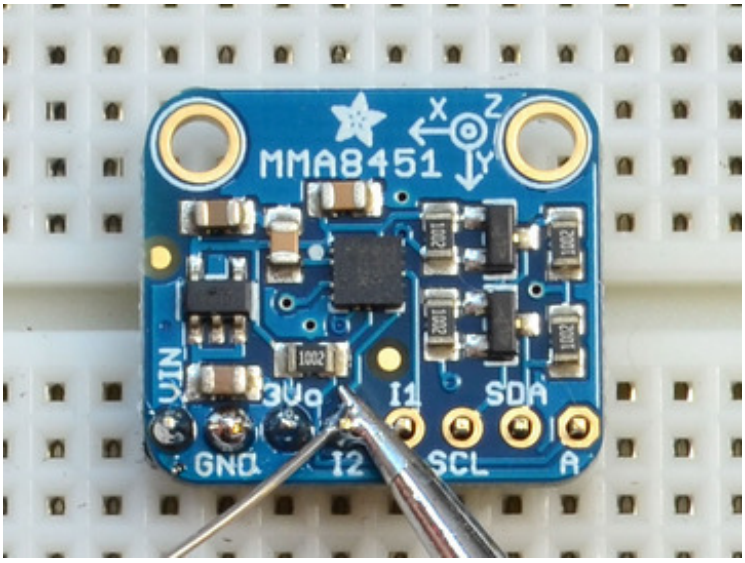
- 

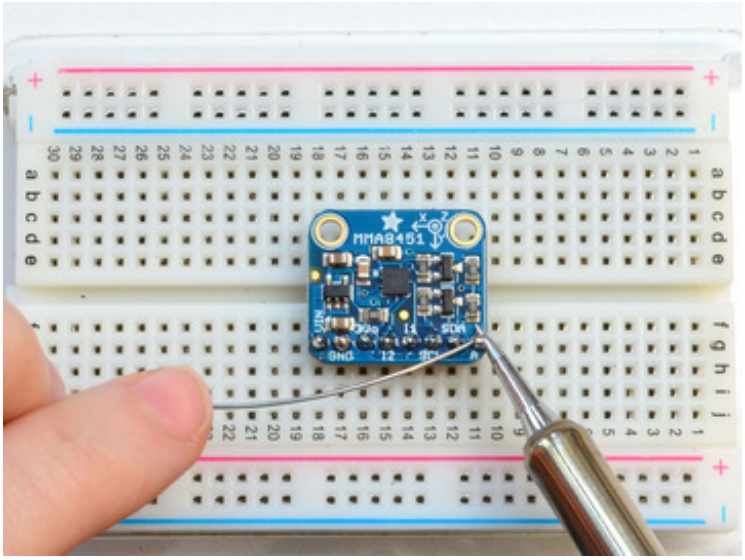
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**

- 

Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads

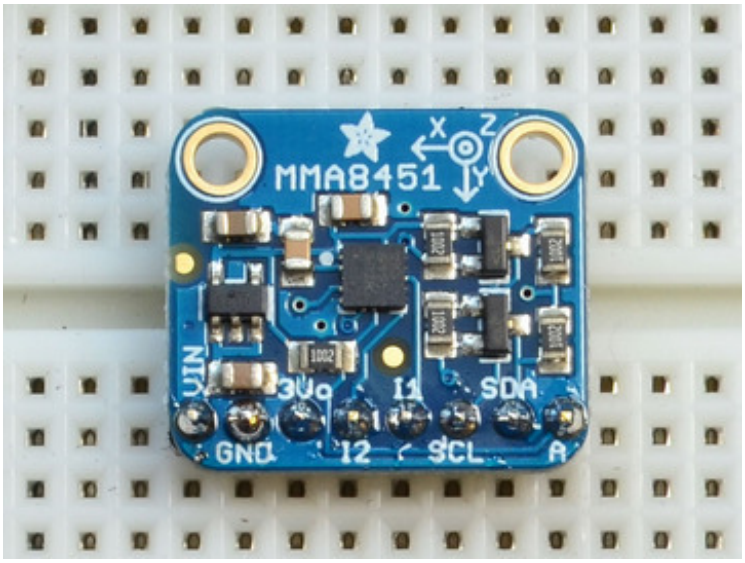
- 
- 

- 

And Solder!

Be sure to solder all pins for reliable electrical contact.

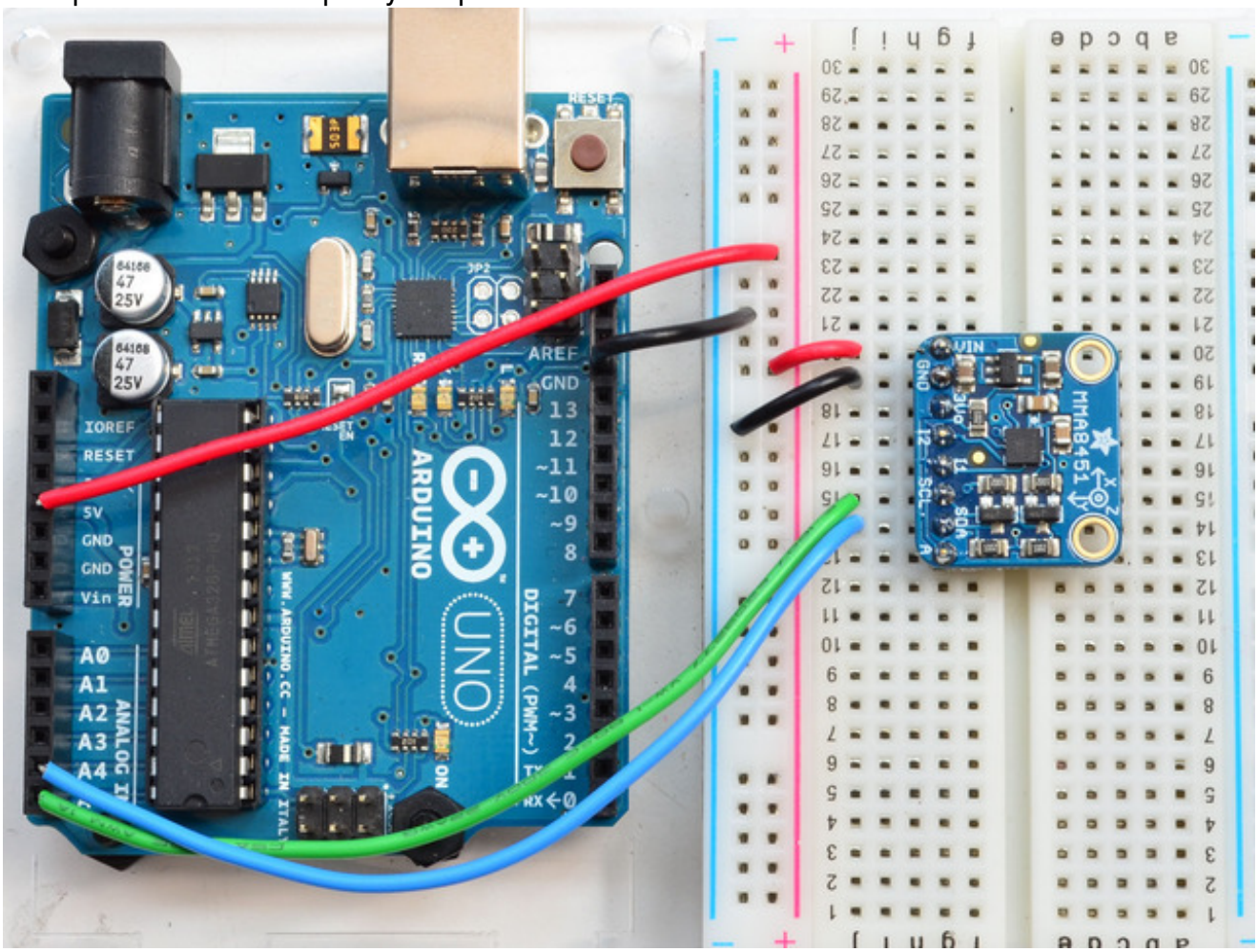
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](http://adafru.it/aTk) (http://adafru.it/aTk).

- 

You're done! Check your solder joints visually and continue onto the next steps

Wiring & Test

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C **with repeated-start support**, then port the code - its pretty simple stuff!



- Connect **Vin** to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock**SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data**SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

The MMA8451 has a default I2C address of **0x1D** and can be changed to 0x1C by tying the **A** pin to GND

Download Adafruit_MMA8451

To begin reading sensor data, you will need to download the Adafruit_MMA8451_Library from our github repository (http://adafru.it/dLJ). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip
Download the Adafruit MMA8451 Library
http://adafru.it/dLK
Rename the uncompressed folder **Adafruit_MMA8451** and check that the **Adafruit_MMA8451** folder contains **Adafruit_MMA8451.cpp** and **Adafruit_MMA8451.h**

Place the **Adafruit_MMA8451** library folder your **arduinosketchfolder/libraries/** folder. You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

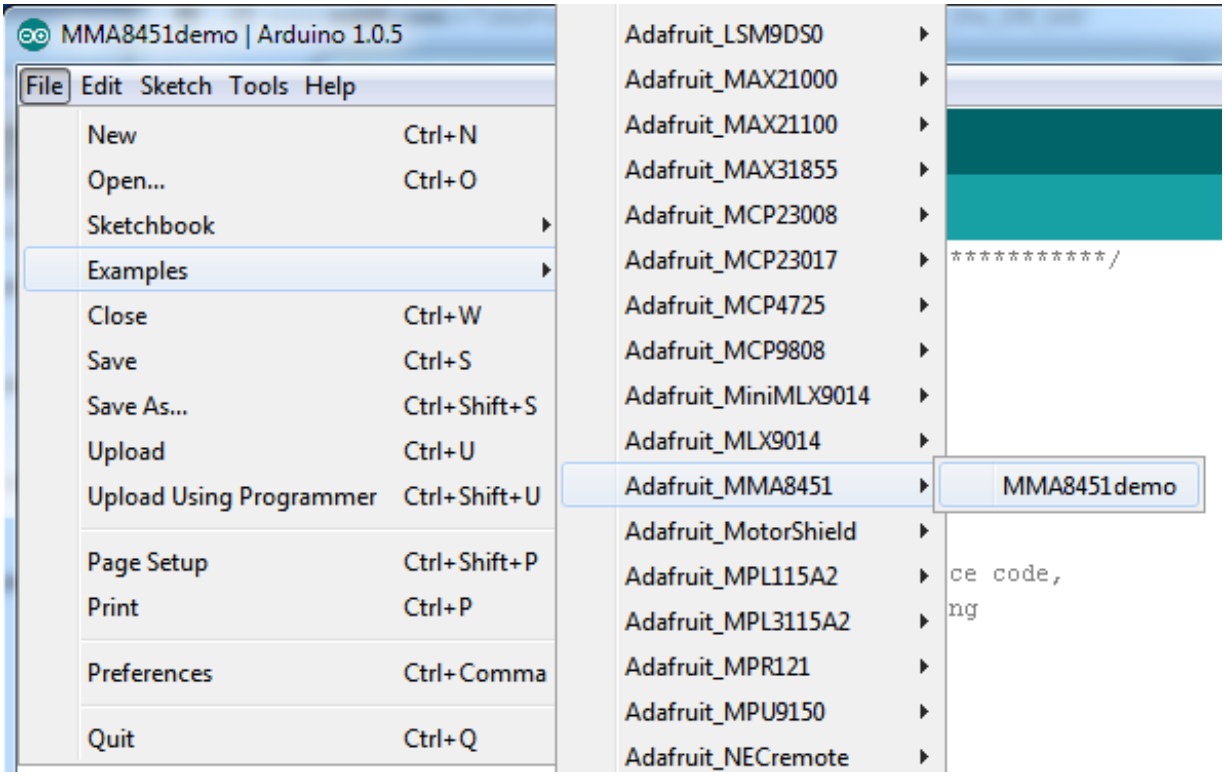
We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (http://adafru.it/aYM)

Download Adafruit_Sensor

The MMA8451 library uses the Adafruit_Sensor support backend so that readings can be normalized between sensors. You can grab Adafruit_Sensor from the github repo (http://adafru.it/aZm) or just click the button below.
Download Adafruit_Sensor Library
http://adafru.it/cMO
Install like you did with Adafruit_MMA8451

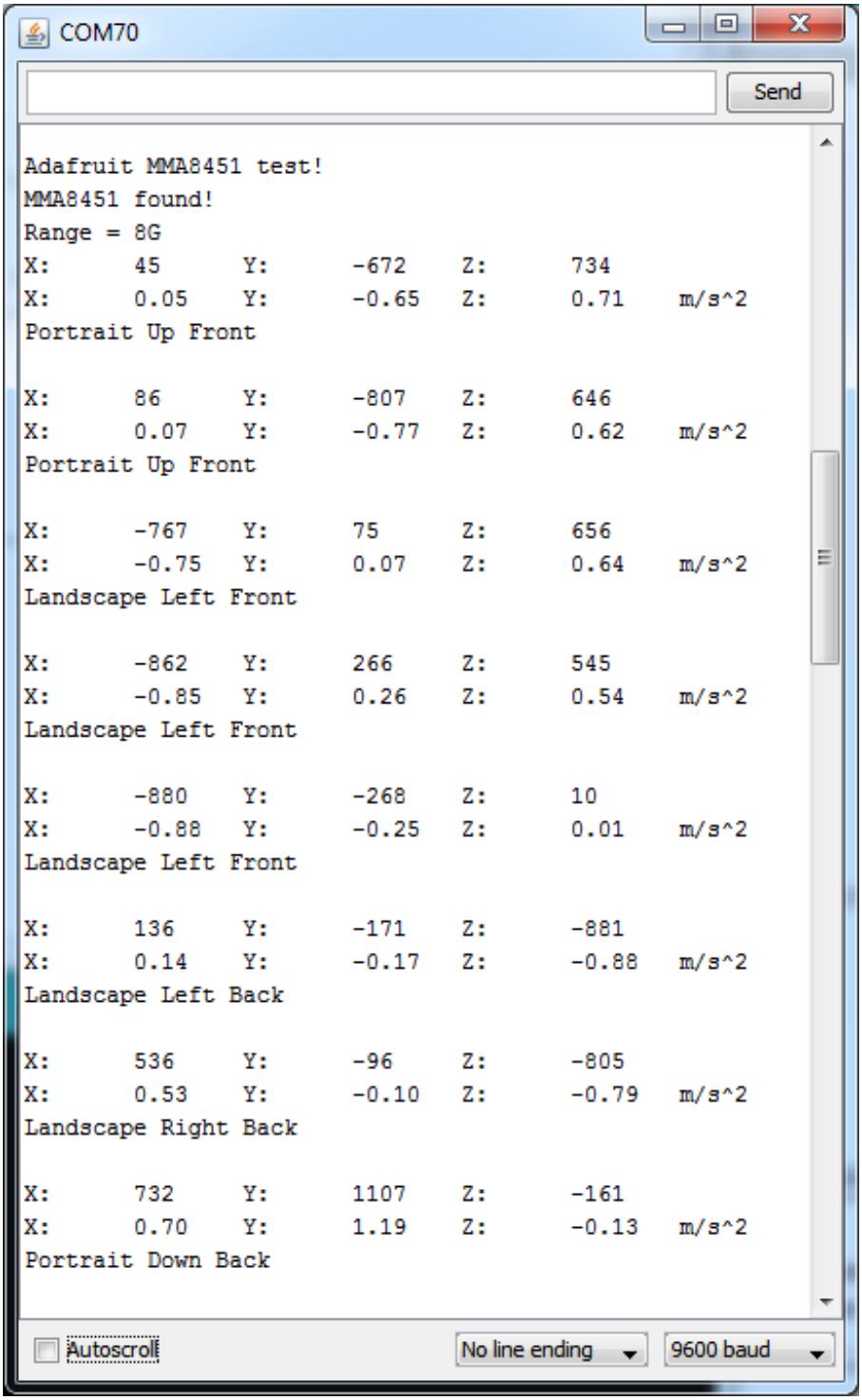
Load Demo

Open up **File->Examples->Adafruit_MMA8451->MMA8451demo** and upload to your Arduino wired up to the sensor



(http://adafru.it/dLL)

Thats it! Now open up the serial terminal window at 9600 speed to begin the test.



There's three lines of output from the sensor.

Example for line 1:

X: 45 Y: -672 Z: 734

This is the "raw count" data from the sensor, its a number from -8192 to 8191 (14 bits) that measures over the set range. The range can be set to 2G, 4G or 8G

Example for line 2:

X: -0.07 Y: 0.09 Z: 0.98 m/s^2

This is the Adafruit_Sensor'ified nice output which is in m/s*s, the SI units for measuring acceleration. No matter what the range is set to, it will give you the same units, so its nice to use this instead of mucking with the raw counts

Example for line 3:

Portrait Up Front

This is the output of the orientaiton detection inside the chip. Since inexpensive accelerometers are often used to detect orientation and tilt, this sensor has it built in. The orientation can be Portrait or Landscape, then Up/Down or Left/Right and finally tilted forward or tilted back. Note that if the sensor is tilted less than 30 degrees it cannot determine the forward/back orientation. If you play with twisting the board around you'll get the hang of it.

Library Reference

The library we have is simple and easy to use

You can create the **Adafruit_MMA8451** object with:

```
Adafruit_MMA8451 mma = Adafruit_MMA8451();
```

There are no pins to set since you must use the I2C bus!

Then initialize the sensor with:

```
mma.begin()
```

this function returns **True** if the sensor was found and responded correctly and **False** if it was not found. We suggest something like this:

```
if (! mma.begin()) {
  Serial.println("Couldnt start")
  while (1);
}
Serial.println("MMA8451 found!");
```

Set & Get Range

You can set the accelerometer max range to ±2g, ±4g or ±8g with

```
mma.setRange(MMA8451_RANGE_2_G);
mma.setRange(MMA8451_RANGE_4_G);
mma.setRange(MMA8451_RANGE_8_G);
```

And read what the current range is with

```
mma.getRange()
```

Which returns 1 for ±2g, 2 for ±4g and 3 for ±8g

Read Raw Count Data

You can read the raw counts data with

```
mma.read();
```

The x, y and z data is then available in **mma.x**, **mma.y** and **mma.z**
All three are read in one transaction.

Reading Normalized Adafruit_Sensor data

We recommend using the Adafruit_Sensor interface which allows reading into an event structure. First create a new event structure

```
sensors_event_t event;
```

Then read the event whenever you want

```
mma.getEvent(&event);
```

The normalized SI unit data is available in **event.acceleration.x, event.acceleration.y** and **event.acceleration.z**

Read Orientation

The sensor has built in tilt/orientation detection. You can read the current orientation with

```
mma.getOrientation();
```

The return value ranges from 0 to 7

- 0: Portrait Up Front
- 1: Portrait Up Back

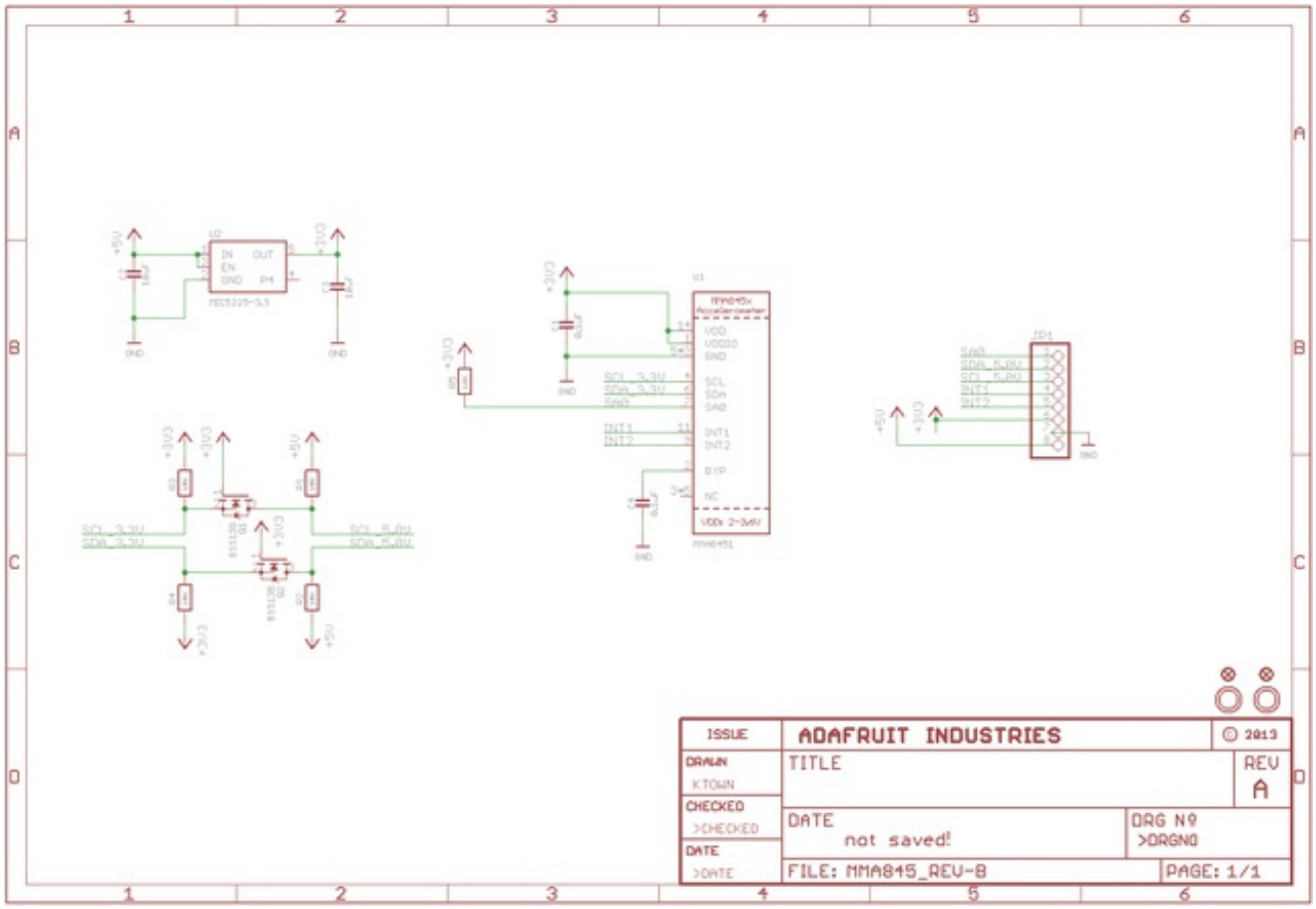
- 2: Portrait Down Front
- 3: Portrait Down Back
- 4: Landscape Right Front
- 5: Landscape Right Back
- 6: Landscape Left Front
- 7: Landscape Left Back

Downloads

Datasheet & Files

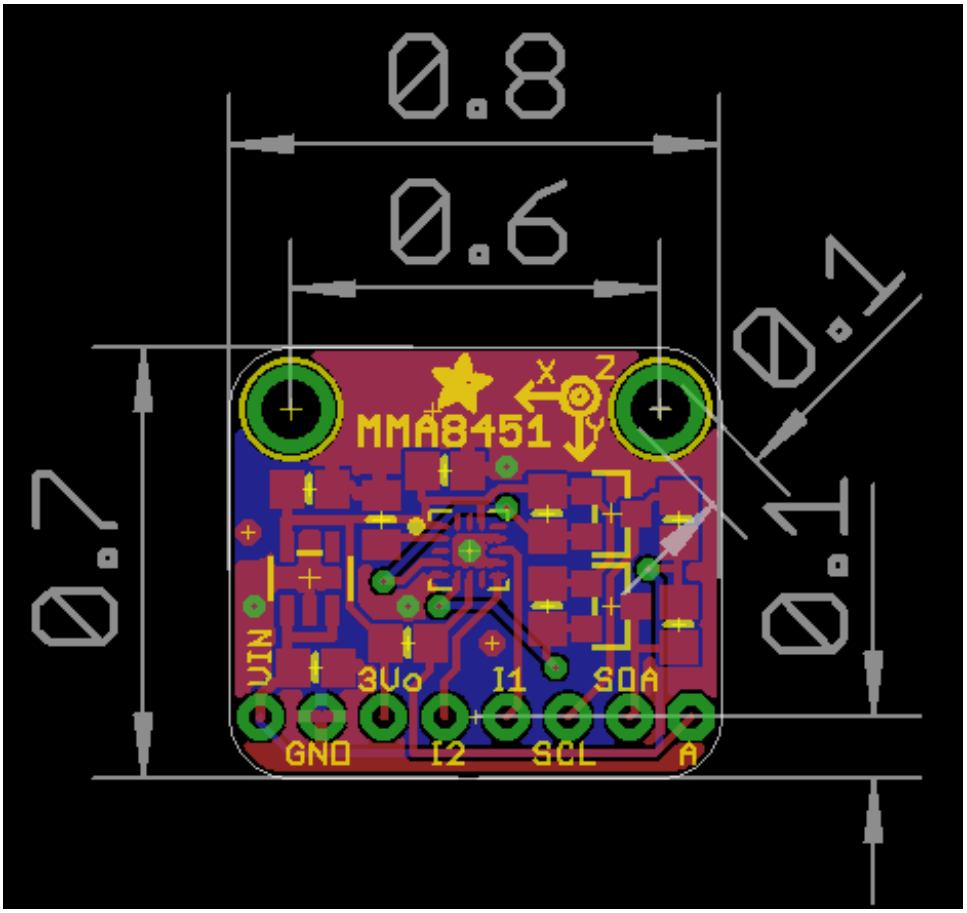
- MMA8451-Q Datasheet (http://adafru.it/dLO)
- Fritzing object in Adafruit Fritzing library (http://adafru.it/aP3)
- EagleCAD PCB files on GitHub (http://adafru.it/pIF)

Schematics



Fabrication print

Dimensions are in Inches


0.8
0.6
0.7
0.1
0.1
MMA8451
X
Z
Y
VIN
3Vo
I1
SDA
GND
I2
SCL
A