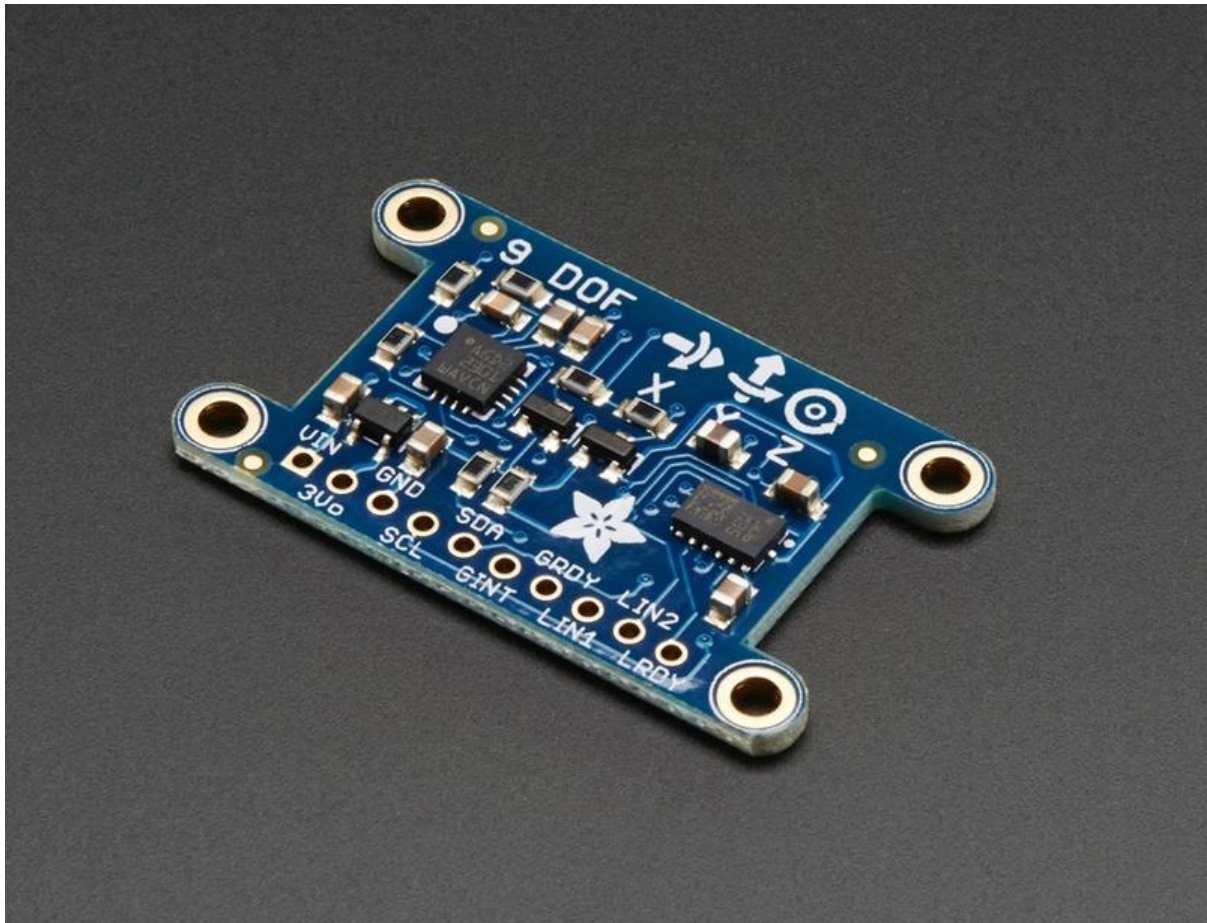




# Adafruit 9-DOF IMU Breakout

Created by Kevin Townsend



<https://learn.adafruit.com/adafruit-9-dof-imu-breakout>

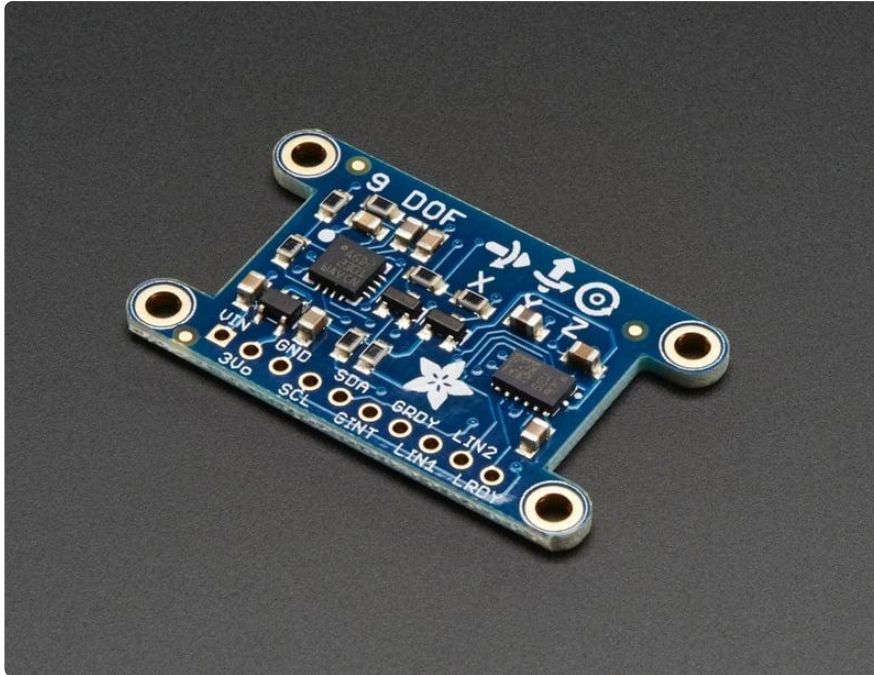
Last updated on 2022-12-01 02:08:07 PM EST

# Table of Contents

<b>Introduction</b>	<b>3</b>
<ul style="list-style-type: none"><li>• <a href="#">Related Links</a></li></ul>	
<b>Connecting It Up</b>	<b>5</b>
<ul style="list-style-type: none"><li>• <a href="#">Basic Setup (5V Logic, Arduino Uno, etc.)</a></li><li>• <a href="#">Advanced Setup</a></li><li>• <a href="#">3V3 Setup</a></li></ul>	
<b>Software</b>	<b>10</b>
<ul style="list-style-type: none"><li>• <a href="#">LSM303DLHC and L3GD20 Drivers</a></li><li>• <a href="#">Software</a></li><li>• <a href="#">Adafruit_9DOF Helper Functions</a></li><li>• <a href="#">bool accelGetOrientation ( sensors_event_t *event, sensors_vec_t *orientation )</a></li><li>• <a href="#">bool magGetOrientation ( sensors_axis_t axis, sensors_event_t *event, sensors_vec_t *mag_orientation )</a></li><li>• <a href="#">bool magTiltCompensation ( sensors_axis_t axis, sensors_event_t *mag_event, sensors_event_t *accel_event )</a></li><li>• <a href="#">Example Sketch</a></li></ul>	
<b>Design Files</b>	<b>14</b>
<ul style="list-style-type: none"><li>• <a href="#">Datasheets &amp; Files</a></li><li>• <a href="#">Dimensions (Inches)</a></li><li>• <a href="#">Schematic</a></li></ul>	
<b>F.A.Q.</b>	<b>15</b>

---

# Introduction

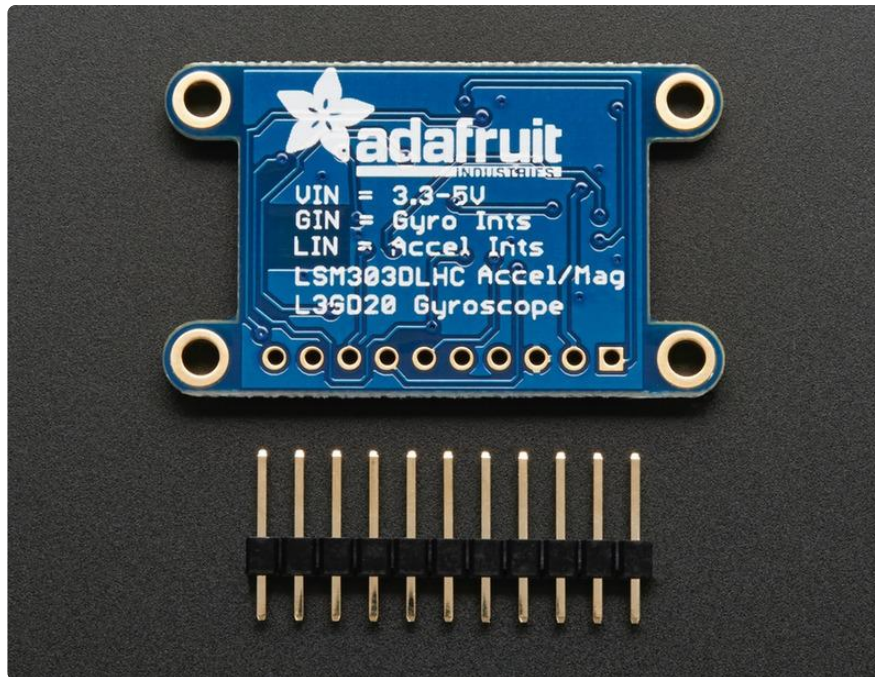


Adafruit's 9DOF (9 Degrees of Freedom) breakout board allows you to capture nine distinct types of motion or orientation related data: 3 degrees each of acceleration, magnetic orientation, and angular velocity.

If you also want barometric data, check out the [Adafruit 10DOF breakout \(http://adafru.it/1604\)](http://adafru.it/1604), which adds a BMP180 barometric pressure sensor to measure altitude and temperature.

After testing a lot of combinations of sensors, we settled on the following devices that we think offer the best results and the least amount of hassle:

- LSM303DLHC - a 3-axis accelerometer (up to +/-16g) and a 3-axis magnetometer (up to +/-8.1 gauss) on a single die
- L3GD20 - a 3-axis gyroscope (up to +/-2000 dps)



## Related Links

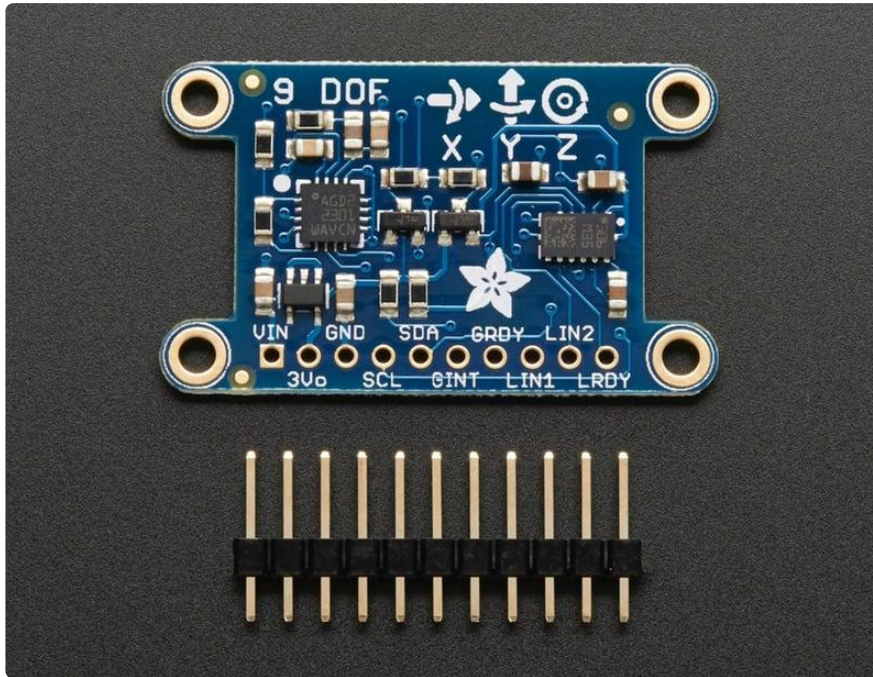
The Adafruit 9DOF board and library reuses the existing Adafruit drivers for the LSM303DLHC (accelerometer and magnetometer) and the L3GD20 (gyroscope).

For information about these particular drivers, consult the following learning guides:

- [LSM303DLHC Learning Guide \(\)](#)
- [L3GD20 Learning Guide \(\)](#)

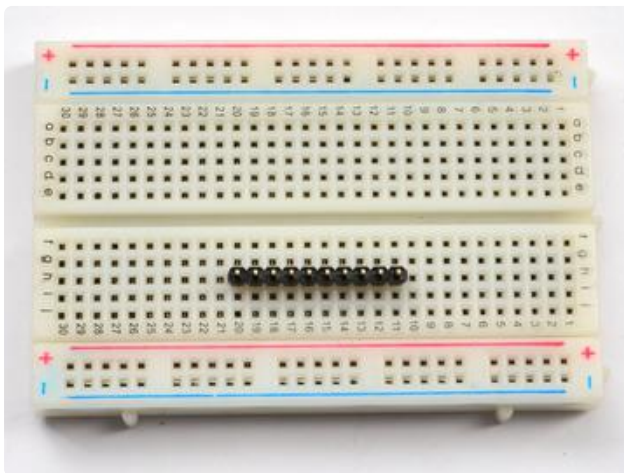
This breakout is basically just a smooshed together version of both of these so anything you can do with those libraries/guides will follow here.

## Connecting It Up



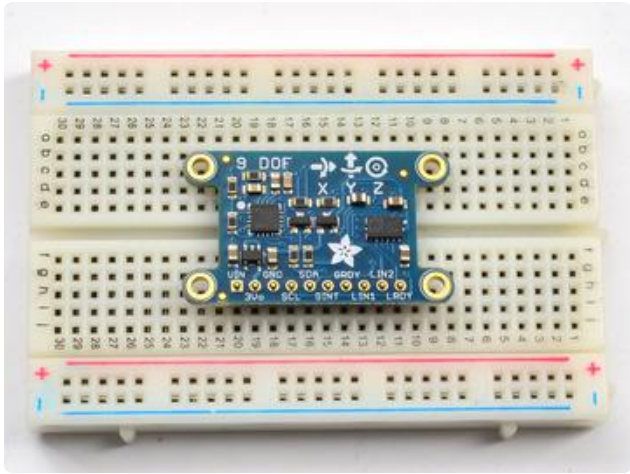
All of the sensors on the Adafruit 9DOF breakout communicate via a two-pin I2C bus, making it relatively easy to setup with a minimum number of cables:

To interface with the sensor, you will need to solder in wire or header into the breakout row at the bottom. You cannot 'press fit' or 'twist' wires in, they will not make good contact! Soldering is required

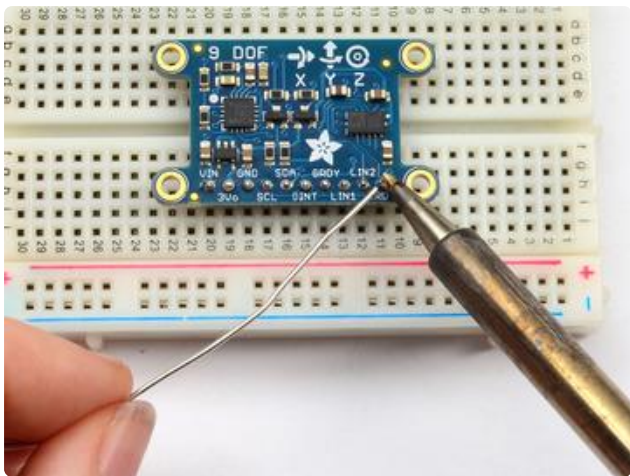
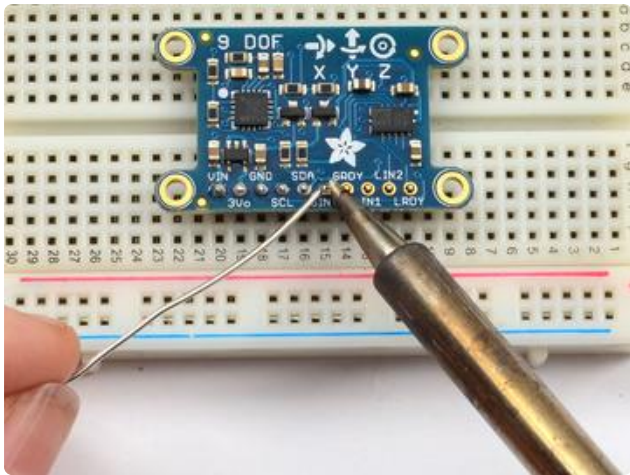
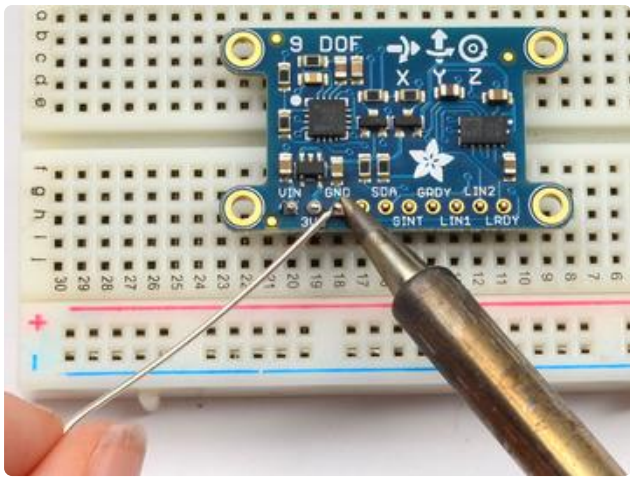


Start by breaking the header down so it is 10 pins long. Sometimes we toss in a longer strip, but its easy to break it down, just use pliers or diagonal cutters to snap it down to 10 pins.

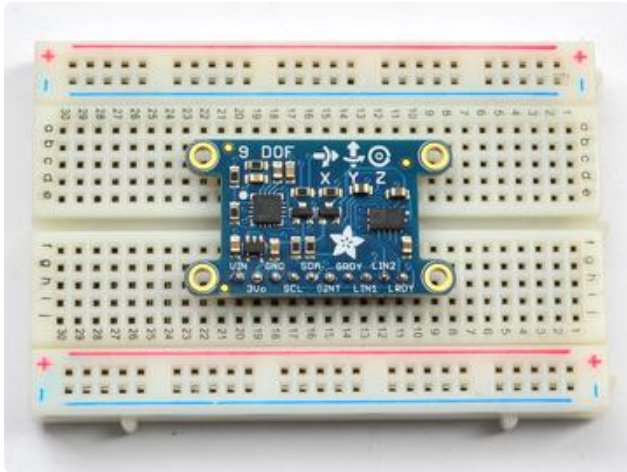
Place the long ends of the header into a solderless breadboard to keep them steady.



Place the 9DOF board right on top of the header so that the short pins are sticking thru the hole pads



Heat up your soldering iron and once it is ready, solder all 10 pads to the header, making sure to check that there is plenty of solder to make a mechanically strong connection and there's no solder bridging either.



That's it! you're done, continue on to the wiring step below

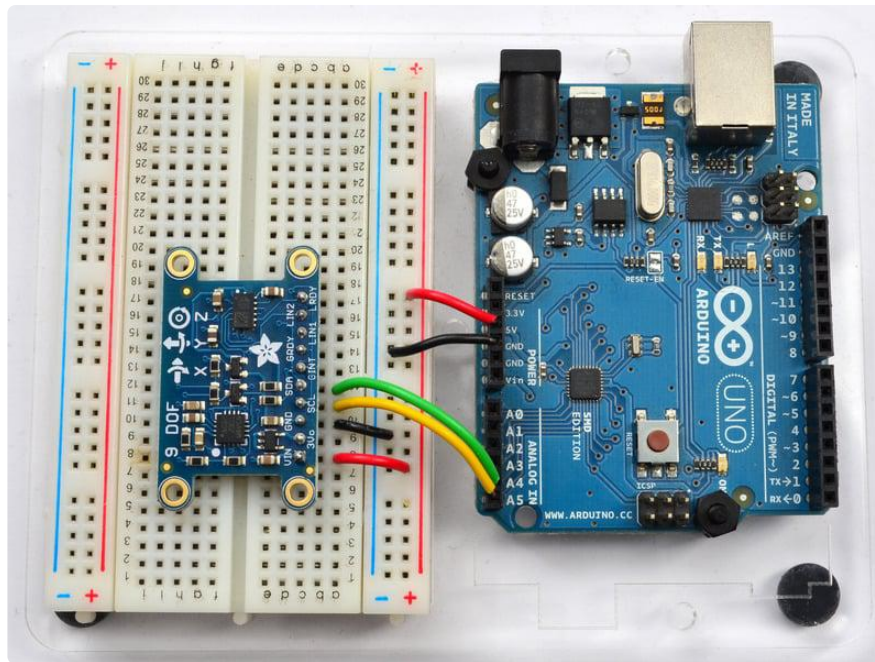
## Basic Setup (5V Logic, Arduino Uno, etc.)

We'll be using an Arduino UNO here, but the code will work on a Mega or Leonardo just fine. Most other Arduino compatibles should have no problems either but we only support official Arduinos for code.

- Connect the SCL pin on the breakout to the 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 on the breakout to the 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
- Connect the VIN pin on the breakout to 3.3V or 5V on your Uno (5V is preferred but if you have a 3V logic Arduino 3V is best)
- Connect the GND pin on the breakout to the GND pin on your Uno

That's it! With those four wires, you should be able to talk to any of the I2C chips on the board and run any of the example sketches.





## Advanced Setup

While most people probably won't need to use the pins below, we've also broken out a few extra pins for advanced users or for special use cases. If you need to use any of these pins, simply hook them up to a GPIO pin of your choice on the Uno:

- GINT - The interrupt pin on the L3GD20 gyroscope
- GRDY - The 'ready' pin on the L3GD20 gyroscope
- LIN1 - Interrupt pin 1 on the LSM303DLHC
- LIN2 - Interrupt pin 2 on the LSM303DLHC
- LRDY - The ready pin on the LSM303DLHC

These pins are all outputs from the 9-DOF breakout and are all 3.3V logic, you can use them with 5V or 3V as 3.3V registers 'high' on 5V systems.

## 3V3 Setup

If you are using an MCU or board with 3V3 logic (instead of the 5V logic used by the Arduino Uno), you can still power the 9-DOF with the VIN pin or you can use the 3Vo pin, which will bypass the on-board 3V3 regulator and level shifting:

- Connect Vin or 3Vo on the breakout to the 3.3V supply on your target MCU
- Connect GND on the breakout to GND on the target MCU

Like other breakouts on Adafruit, the 9 DOF Breakout is fully level shifted, and you can safely use it on 3V3 or 5V systems.

---

# Software

## LSM303DLHC and L3GD20 Drivers

You will need to have all of the following libraries available in your /libraries folder for Arduino to make use of the Adafruit 9DOF breakout:

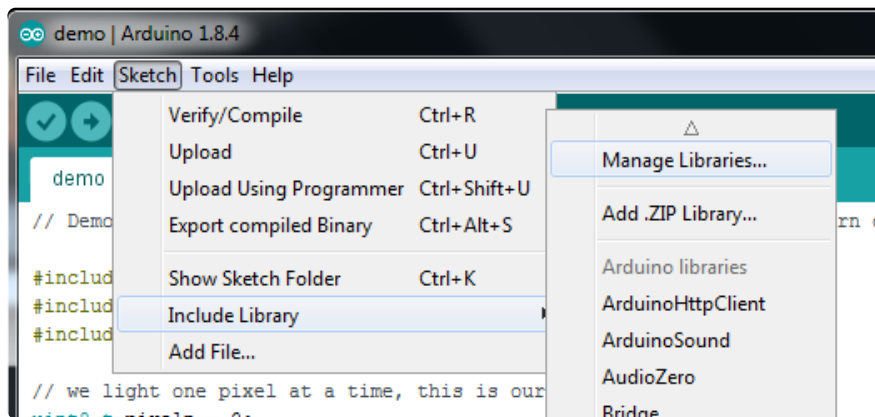
- [Adafruit Unified Sensor Library on Github \(\)](#)
- [LSM303DLHC Library on Github \(\)](#)
- [L3GD20 Library on Github \(\)](#)
- [Adafruit 9DOF Library on Github \(\)](#)

For information on the Adafruit Sensor Library and the Unified Sensor Drivers used for all of these sensors, feel free to have a look at our related learning guide: [Using the Adafruit Unified Sensor Driver \(\)](#)

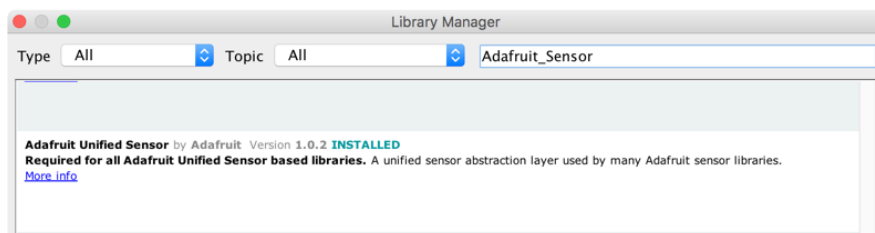
# Software

To use the Adafruit 9DOF breakout, install the following libraries.

Open up the Arduino library manager:



Search for the Adafruit Sensor library and install it



Search for the Adafruit LSM303 library and install it



Search for the Adafruit L3GD20 library and install it



Search for the Adafruit 9DOF library and install it



[If you're new to installing libraries, check out our super-awesome detailed tutorial on how to install Arduino libraries \(\)](#)

## Adafruit\_9DOF Helper Functions

The `Adafruit_9DOF.cpp` file (from [Adafruit\\_9DOF \(\)](#)) includes the following helper functions that are useful when working with typical 9DOF projects

```
bool accelGetOrientation ( sensors_event_t *event,  
sensors_vec_t *orientation )
```

This function reads the LSM303DLHC accelerometer data (supplied via the 'event' variable), and converts it into equivalent pitch (x) and roll (y) values, populating the supplied 'orientation' variables `.pitch` and `.roll` fields accordingly.

### Arguments

- `event`: The `sensors_event_t` variable containing the data from the accelerometer
- `orientation`: The `sensors_vec_t` object that will have its `.pitch` and `.roll` fields populated

## Returns

- true if the operation was successful,
- false if there was an error

## Example

See the 'pitchrollheading' example in the Adafruit\_9DOF library for an example of how to use this helper function

```
sensors_event_t accel_event;
sensors_vec_t  orientation;

/* Calculate pitch and roll from the raw accelerometer data */
accel.getEvent(&accel_event);
if (dof.accelGetOrientation(&accel_event, &orientation))
{
  /* 'orientation' should have valid .roll and .pitch fields */
  Serial.print(F("Roll: "));
  Serial.print(orientation.roll);
  Serial.print(F("; "));
  Serial.print(F("Pitch: "));
  Serial.print(orientation.pitch);
  Serial.print(F("; "));
}
}
```

**bool magGetOrientation ( sensors\_axis\_t axis,  
sensors\_event\_t \*event, sensors\_vec\_t \*mag\_orientation  
)**

This function populates the .heading field in mag\_orientation with the correct angular data (0-359°). Heading increases when rotating clockwise around the specified axis.

## Arguments

- axis: The given axis (SENSOR\_AXIS\_X, SENSOR\_AXIS\_Y, or SENSOR\_AXIS\_Z)
- event: The raw magnetometer sensor data to use when calculating out heading
- orientation: The sensors\_vec\_t object where we will assign an 'orientation.heading' value

## Returns

- true if the operation was successful,
- false if there was an error

## Example

See the 'pitchrollheading' example in the Adafruit\_9DOF library for an example of how to use this helper function

```

sensors_event_t mag_event;
sensors_vec_t orientation;

/* Calculate the heading using the magnetometer */
mag.getEvent(&mag_event);
if (dof.magGetOrientation(SENSOR_AXIS_Z, &mag_event, &orientation))
{
    /* 'orientation' should have valid .heading data now */
    Serial.print(F("Heading: "));
    Serial.print(orientation.heading);
    Serial.print(F("; "));
}

```

## bool magTiltCompensation ( sensors\_axis\_t axis, sensors\_event\_t \*mag\_event, sensors\_event\_t \*accel\_event )

This function uses the accelerometer data provided in accel\_event to compensate the magnetic sensor measurements in mag\_event to compensate for situations where the sensor is tilted (the pitch and roll angles are not equal to 0°).

### Arguments

- axis: The given axis (SENSOR\_AXIS\_X, SENSOR\_AXIS\_Y, or SENSOR\_AXIS\_Z) that is parallel to the gravity of the Earth
- mag\_event: The raw magnetometer data to adjust for tilt
- accel\_event: The accelerometer event data to use to determine the tilt when compensating the mag\_event values

### Returns

- true if the operation was successful,
- false if there was an error

### Example

```

sensors_event_t accel_event;
sensors_event_t mag_event;

...

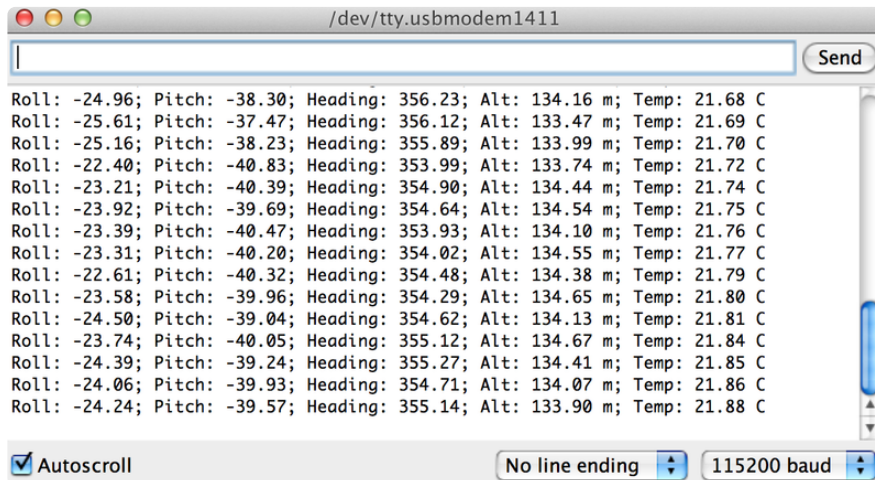
// Get a sensor event from the accelerometer and magnetometer
accel.getEvent(&accel_event);
mag.getEvent(&mag_event);

if (dof.magTiltCompensation(SENSOR_AXIS_Z, &mag_event, &accel_event))
{
    // Do something with the compensated data in mag_event!
}
else
{
    // Oops ... something went wrong (probably bad data)
}

```

# Example Sketch

If you run the pitchrollheading sketch in the examples folder, you can see a practical example using these helper functions above, which should result in output similar to the image below:



The image shows a terminal window titled "/dev/tty.usbmodem1411". The window contains a list of sensor readings, each line representing a single data point. The readings include Roll, Pitch, Heading, Altitude, and Temperature. The window also features a "Send" button, a scroll bar, and control options at the bottom: "Autoscroll" (checked), "No line ending", and "115200 baud".

```
Roll: -24.96; Pitch: -38.30; Heading: 356.23; Alt: 134.16 m; Temp: 21.68 C
Roll: -25.61; Pitch: -37.47; Heading: 356.12; Alt: 133.47 m; Temp: 21.69 C
Roll: -25.16; Pitch: -38.23; Heading: 355.89; Alt: 133.99 m; Temp: 21.70 C
Roll: -22.40; Pitch: -40.83; Heading: 353.99; Alt: 133.74 m; Temp: 21.72 C
Roll: -23.21; Pitch: -40.39; Heading: 354.90; Alt: 134.44 m; Temp: 21.74 C
Roll: -23.92; Pitch: -39.69; Heading: 354.64; Alt: 134.54 m; Temp: 21.75 C
Roll: -23.39; Pitch: -40.47; Heading: 353.93; Alt: 134.10 m; Temp: 21.76 C
Roll: -23.31; Pitch: -40.20; Heading: 354.02; Alt: 134.55 m; Temp: 21.77 C
Roll: -22.61; Pitch: -40.32; Heading: 354.48; Alt: 134.38 m; Temp: 21.79 C
Roll: -23.58; Pitch: -39.96; Heading: 354.29; Alt: 134.65 m; Temp: 21.80 C
Roll: -24.50; Pitch: -39.04; Heading: 354.62; Alt: 134.13 m; Temp: 21.81 C
Roll: -23.74; Pitch: -40.05; Heading: 355.12; Alt: 134.67 m; Temp: 21.84 C
Roll: -24.39; Pitch: -39.24; Heading: 355.27; Alt: 134.41 m; Temp: 21.85 C
Roll: -24.06; Pitch: -39.93; Heading: 354.71; Alt: 134.07 m; Temp: 21.86 C
Roll: -24.24; Pitch: -39.57; Heading: 355.14; Alt: 133.90 m; Temp: 21.88 C
```

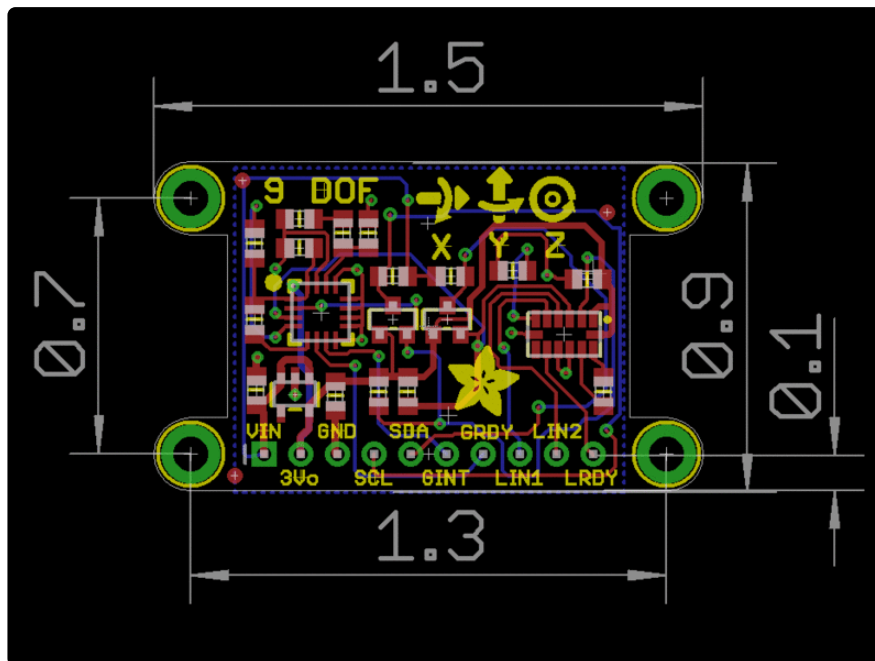
---

## Design Files

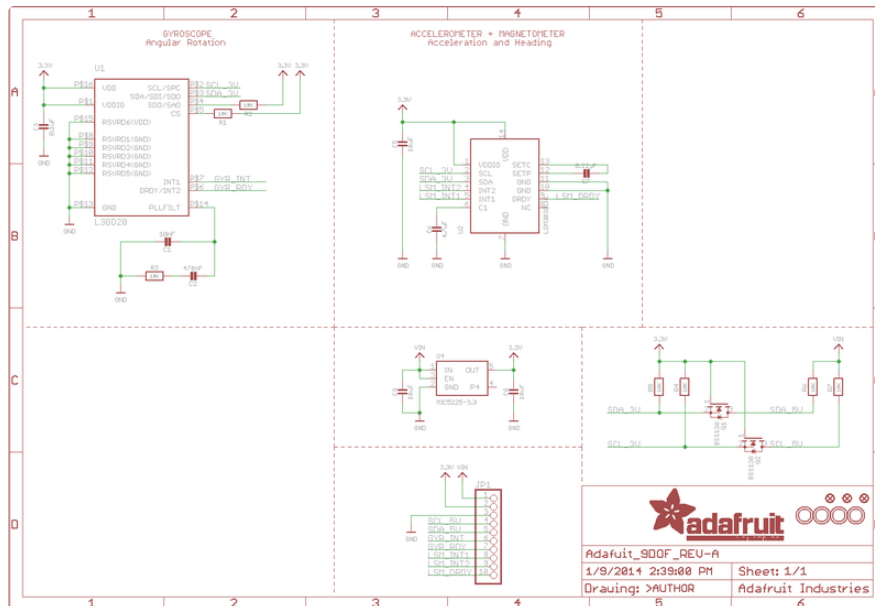
## Datasheets & Files

- [LSM303 Datasheet \(\)](#)
- [L3GD20 Datasheet \(\)](#)
- [EagleCAD PCB files on GitHub \(\)](#)
- [Fritzing object in Adafruit Fritzing library \(\)](#)

# Dimensions (Inches)



# Schematic



# F.A.Q.