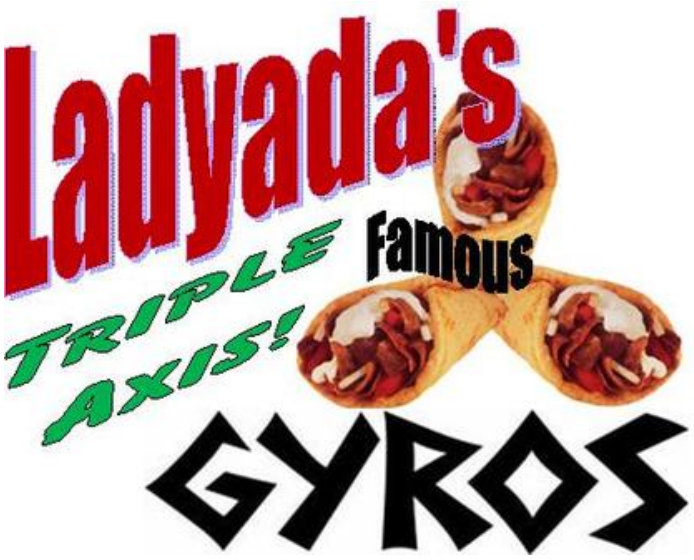
Adafruit Triple Axis Gyro Breakout

Created by Bill Earl

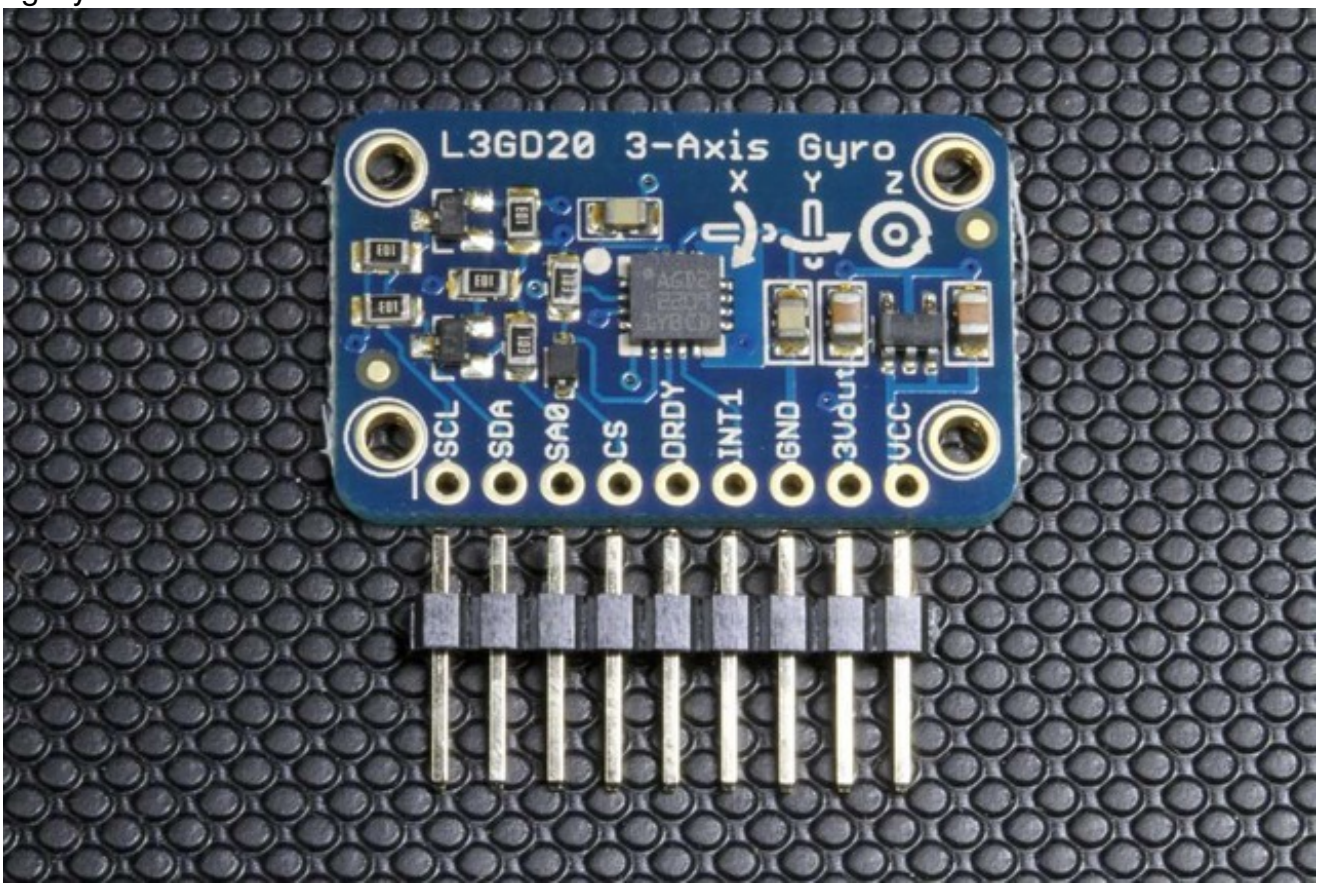


Last updated on 2016-10-17 01:39:22 PM UTC

Guide Contents

Overview

The Adafruit Triple Axis Gyro Breakout is based on the STMicro L3GD20 MEMS digital output gyroscope chip (http://adafru.it/aV1). We include a 3.3v regulator on board for compatibility with 5v controllers like the Arduino. And there are 4 holes so that it can be rigidly mounted.



How it Works:

The triple-axis gyro sensor is a MEMS (Micro Electrical Mechanical System) device consisting of 3 micro-machined 'tuning fork' structures on a silicon wafer. These structures are designed to vibrate when stimulated by an electrical signal. When rotated about the axis of the tuning fork, the tines will deflect due to the Coriolis force (http://adafru.it/aV2). This deflection is proportional to the speed of rotation.

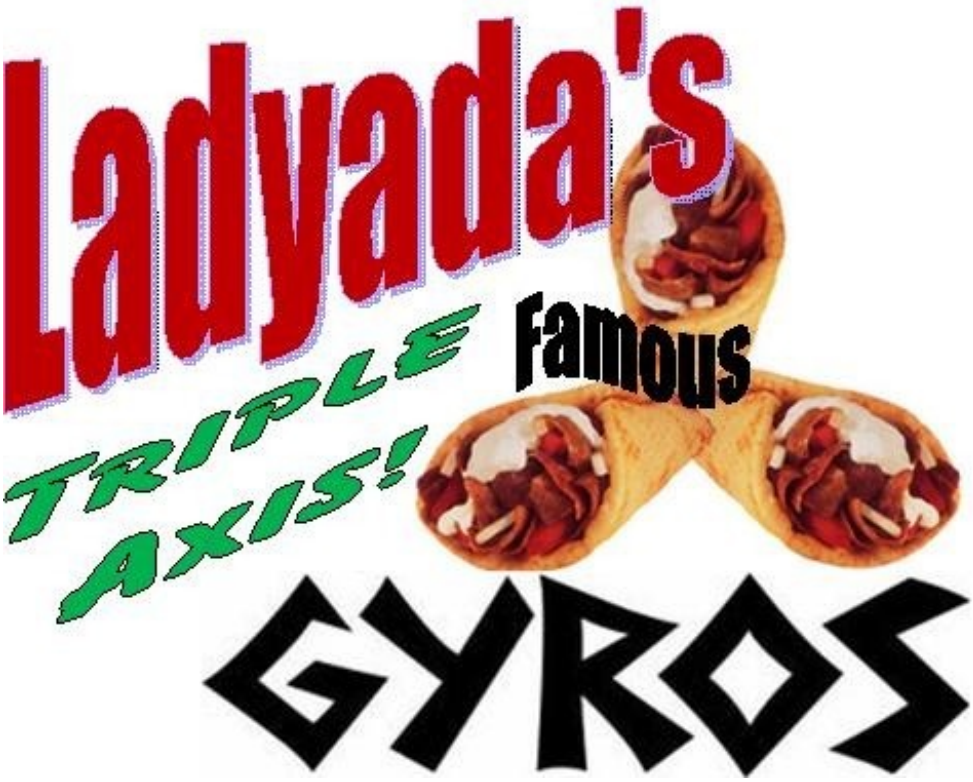
The 3 MEMS structures are arranged orthogonally, on the X, Y and Z axis. Deflection on each tuning fork is detected as a change in capacitance between sensing plates built into the MEMS structure and converted to a degrees-per-second rotation rate for each of the

three axis.

For a more detailed description of MEMS Gyros, seeMEMS Gyroscopes and their applications (http://adafru.it/aV3). and Everything about STMicroelectronics' 3-axis digital MEMS (http://adafru.it/aV4)gyroscopes (http://adafru.it/aV4).

What can it do?

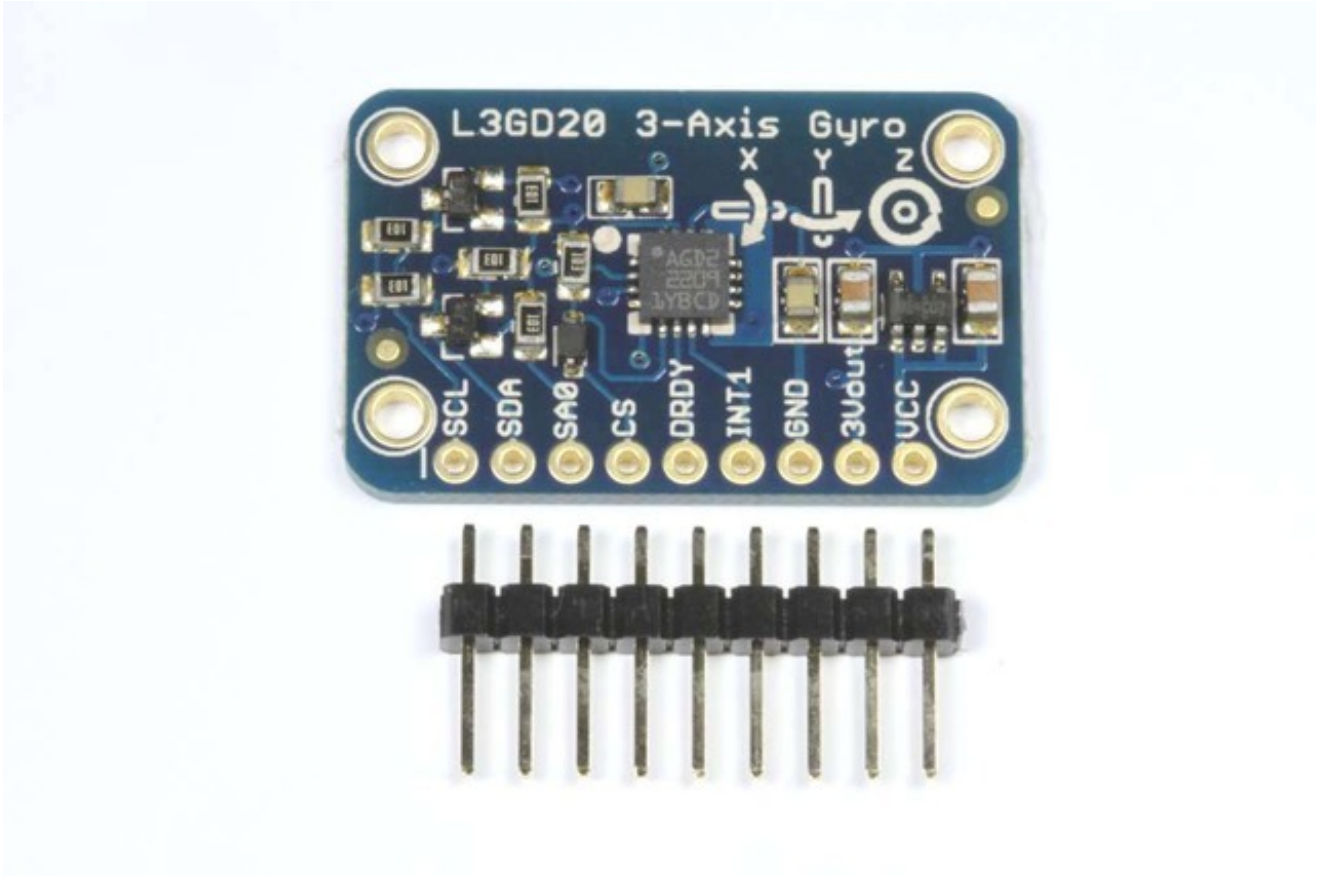
Gyroscopes are useful for many types of motion sensing applications. They are often paired with accelerometers (http://adafru.it/aV5) for inertial guidance systems, 3D motion capture and inverted pendulum (e.g. Segway) type applications. The L3GD20 is particularly versatile, with three full axes of sensing, selectable ±250, ±500, and ±2000 degree-per-second sensitivity ranges and built-in high/low pass filtering.



Ladyada's Triple Gyro Special! (http://adafru.it/1032)

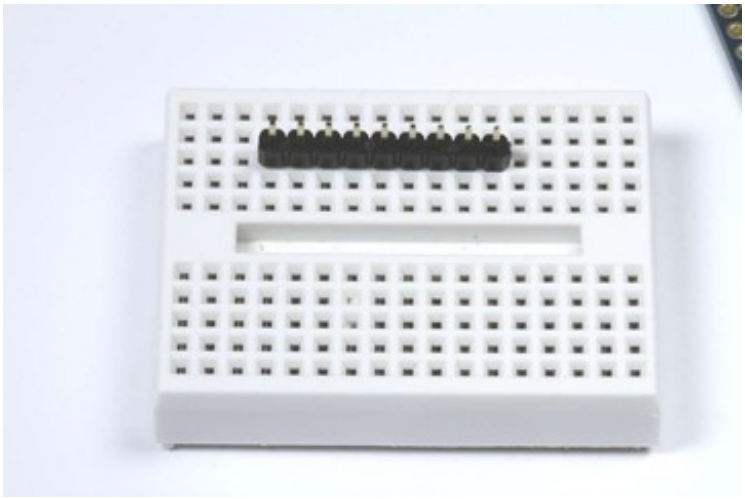
3 micro-machined gyros with I2C, SPI and a 3v regulator on a bite-sized breakout board. *(Tomato, onion and tzatziki sauce extra.)*

Assembly and Wiring



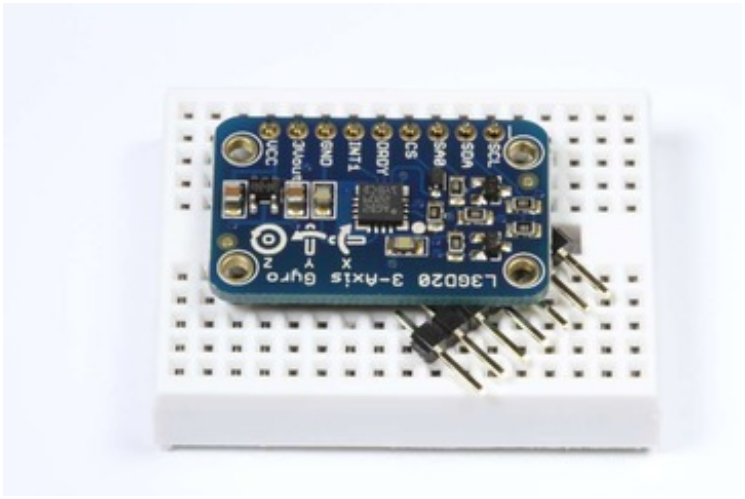
Assembly:

The board comes with all surface-mount components pre-soldered. The included header strip can be soldered on for convenient use on a breadboard or with 0.1" connectors. However, for applications subject to extreme accelerations, shock or vibration, locking connectors or direct soldering is advised.



Prepare the header strip:

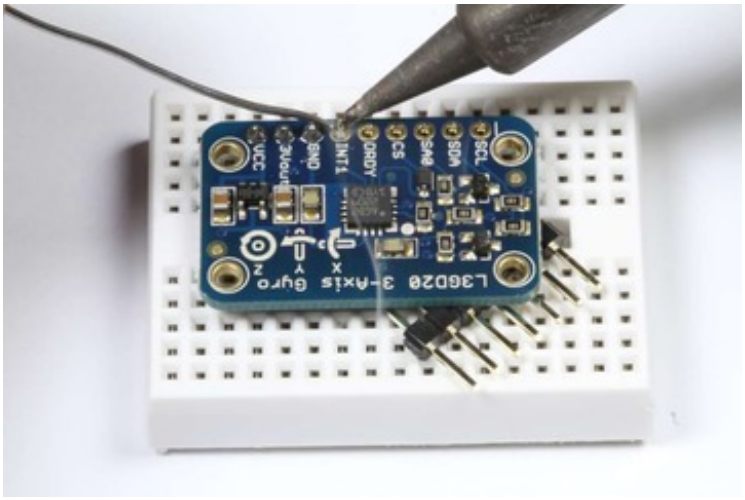
Cut the strip[to length if necessary. It will be easier to solder if you insert it into a breadboard - long pins down.



Add the breakout board:

Place the breakout board over the pins.

(You can prop up the free edge of the board with some extra pins to better align it for soldering.)



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)).

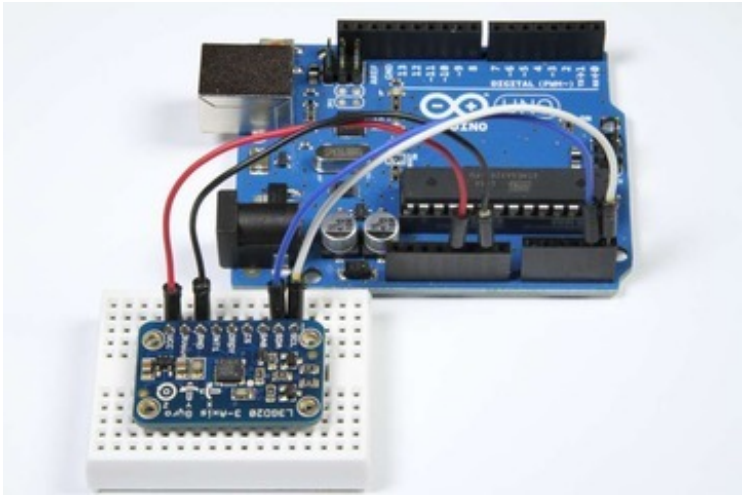
Wiring:

You'll need to power the breakout, you can power it from 3V-5VDC, connect ground to GND and VCC to our power supply (3-5V)

The L3GD20 breakout board supports both I2C and SPI communication. I2C requires the fewest connections, so we will start with that:

Wiring for I2C:

I2C requires only 2 pins (in addition to VCC and ground).

- 

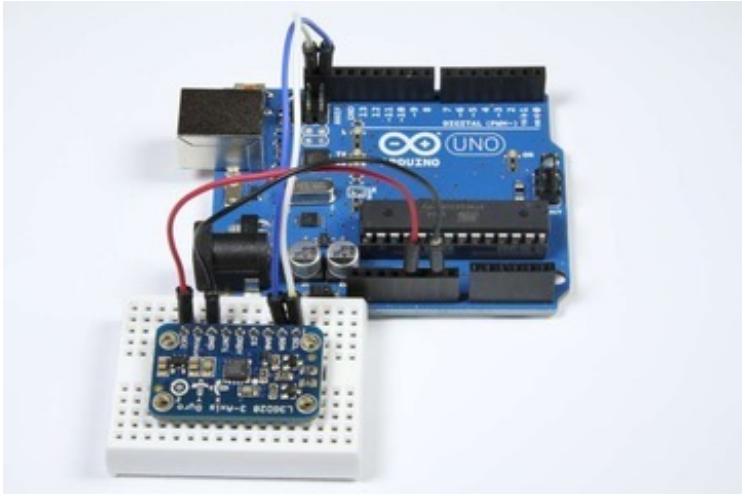
'Classic' Arduino Wiring:

On pre-R3 Arduinos, the I2C pins are:

- SDA = Analog 4
- SCL = Analog 5

For the Mega

- SDA = Digital 20
- SCL = Digital 21

- 

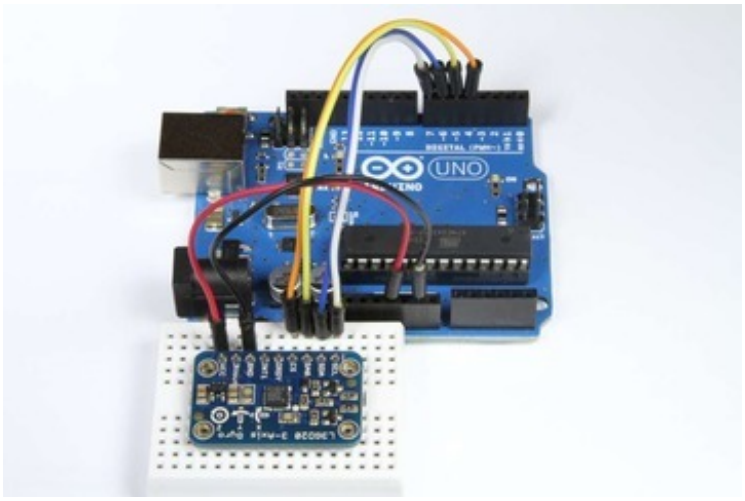
R3 and Later Arduino Wiring:

Although the 'classic' wiring will still work, All R3 and later Arduinos (including Mega, Due and Leonardo) have SDA and SCL pins on the extended header next to AREF for compatibility.

Wiring for SPI:

Wiring for SPI:

The SPI interface requires 4 wires (in addition to VCC and Ground).

- 

SPI Wiring:

The library uses "software SPI" so the choice of pins is a little more flexible. What we show here is compatible with the example code included with the library:

- CS = Digital 4
- SAO = Digital 5
- SDA = Digital 6
- SCL = Digital 7

Programming

The Adafruit L3GD20 Library for the Arduino implements a convenient device class to handle the the low-level device communication with the Gyro module. The programming interface is described below:

Construction:

To use the L3GD20 in your sketch, you must first call a constructor to create a device object. There are two forms of the constructor:

- **Adafruit_L3GD20(void);**
- **Adafruit_L3GD20(int8_t cs, int8_t mosi, int8_t miso, int8_t clk);**

The first version takes no parameters and is used for I2C communication. The second version is for SPI communication and requires that you specify the pins to be used.

I2C Example: (use with I2C wiring (http://adafru.it/aV6))

```
// No need to specify pins for I2C
Adafruit_L3GD20 gyro();
```

SPI Example: (use with SPI wiring (http://adafru.it/aV6))

```
// Define the pins for SPI
#define GYRO_CS 4 // labeled CS
#define GYRO_DO 5 // labeled SA0
#define GYRO_DI 6  // labeled SDA
#define GYRO_CLK 7 // labeled SCL

Adafruit_L3GD20 gyro(GYRO_CS, GYRO_DO, GYRO_DI, GYRO_CLK);
```

Initialization:

Before using the device object you constructed, you must initialize it with the sensitivity range you want to use:

- **bool begin(gyroRange_t rng);**

where "**rng**" can be one of:

- **L3DS20_RANGE_250DPS** - for 250 degrees-per-second range (default)
- **L3DS20_RANGE_500DPS** - for 500 degrees-per-second range
- **L3DS20_RANGE_2000DPS** - for 2000 degrees-per-second range

Example:

```
void setup()
{
  Serial.begin(9600);

  // Try to initialise and warn if we couldn't detect the chip
  if (!gyro.begin(gyro.L3DS20_RANGE_250DPS))
  {
    Serial.println("Oops ... unable to initialize the L3GD20. Check your wiring!");
    while (1);
  }
}
```

Sensing Rotation:

To sense rotation, you must first call the "read()" function to take a reading:

- **void read(void);**

This function takes no parameters. After calling "read()". The raw x, y and z readings can be retrieved from the device object's "data" member.

- **data.x** - x-axis rotation rate in degrees-per-second
- **data.y** - y-axis rotation rate in degrees-per-second
- **data.z** - z-axis rotation rate in degrees-per-second

Example:

```
void loop()
{
  gyro.read();
  Serial.print("X: "); Serial.print((int)gyro.data.x);   Serial.print(" ");
  Serial.print("Y: "); Serial.print((int)gyro.data.y);   Serial.print(" ");
  Serial.print("Z: "); Serial.println((int)gyro.data.z); Serial.print(" ");
  delay(100);
}
```

Alternate Units:

The values reported by the read() function are in degrees-per-second (dps) For some calculations, it may be more convenient to work in radians. To convert dps to radians-per-second (rad/s), simply multiply by 0.017453293 as in the following code:

```
#define SENSORS_DPS_TO_RADS             (0.017453293F)        /**< Degrees/s to rad/s multiplier */

void loop()
{
  gyro.read();
  Serial.print("X: "); Serial.print((int)gyro.data.x * SENSORS_DPS_TO_RADS);   Serial.print(" ");
  Serial.print("Y: "); Serial.print((int)gyro.data.y * SENSORS_DPS_TO_RADS);   Serial.print(" ");
  Serial.print("Z: "); Serial.println((int)gyro.data.z * SENSORS_DPS_TO_RADS); Serial.print(" ");
  delay(100);
}
```

Calibration:

The L3GD20 is calibrated at the factory to close tolerances and will provide sufficient accuracy for most applications.

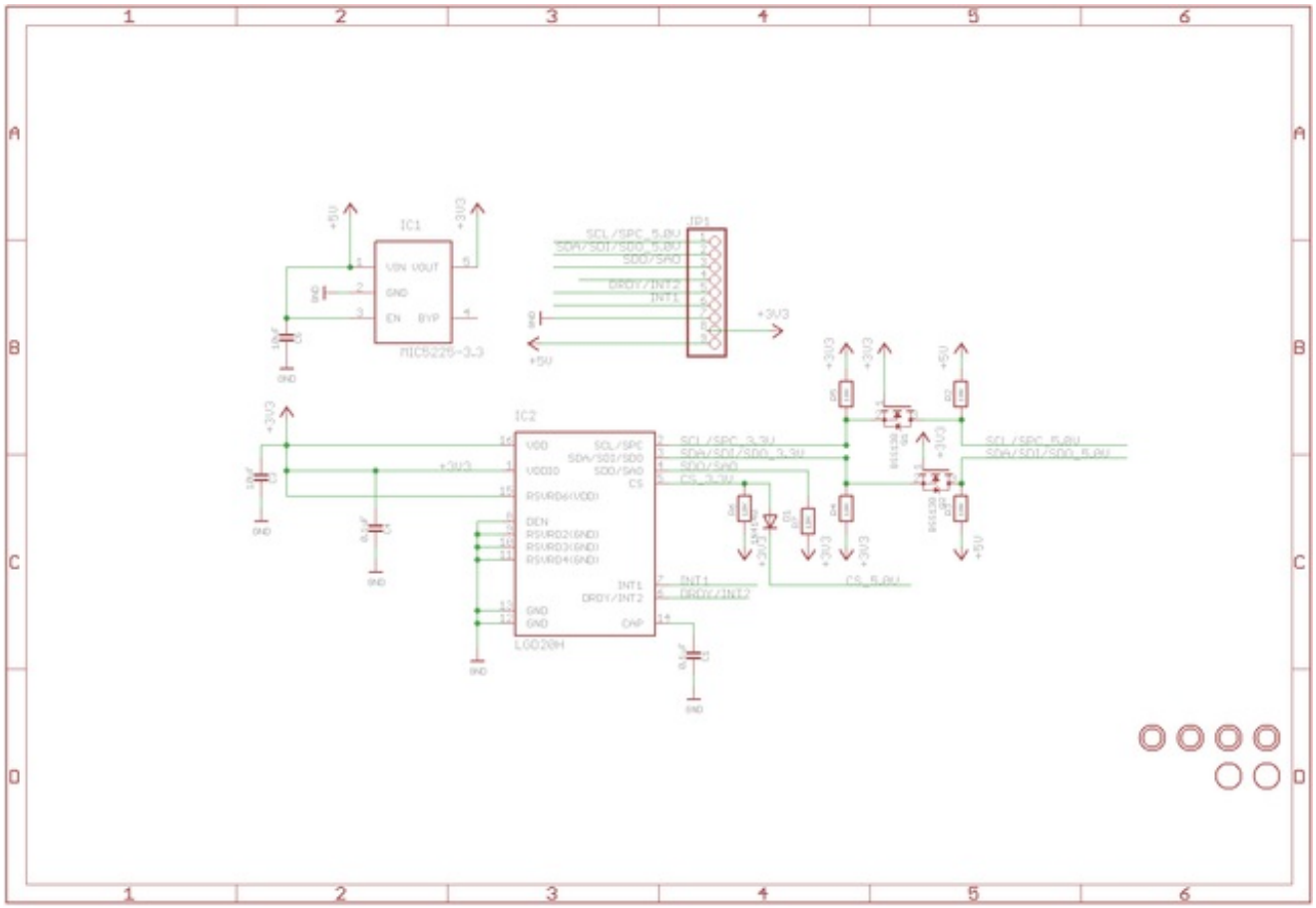
For critical applications where maximum accuracy is required, the gyro should be calibrated for zero-rate and sensitivity. For detailed information on how to calibrate a MEMS gyro, please refer to section 5.3 of this technical article (http://adafru.it/c7L).
The L3GD20 includes an on-chip temperature sensor, but this sensor isn't intended to be used to measure ambient temperature. The register seems to be used to provide a temperature offset for the die temperature to account for variation across multiple gyros or over time, but the value this offset is relative to unfortunately isn't specified in the datasheet, so there is no obvious way to use it in a useful manner as-is.

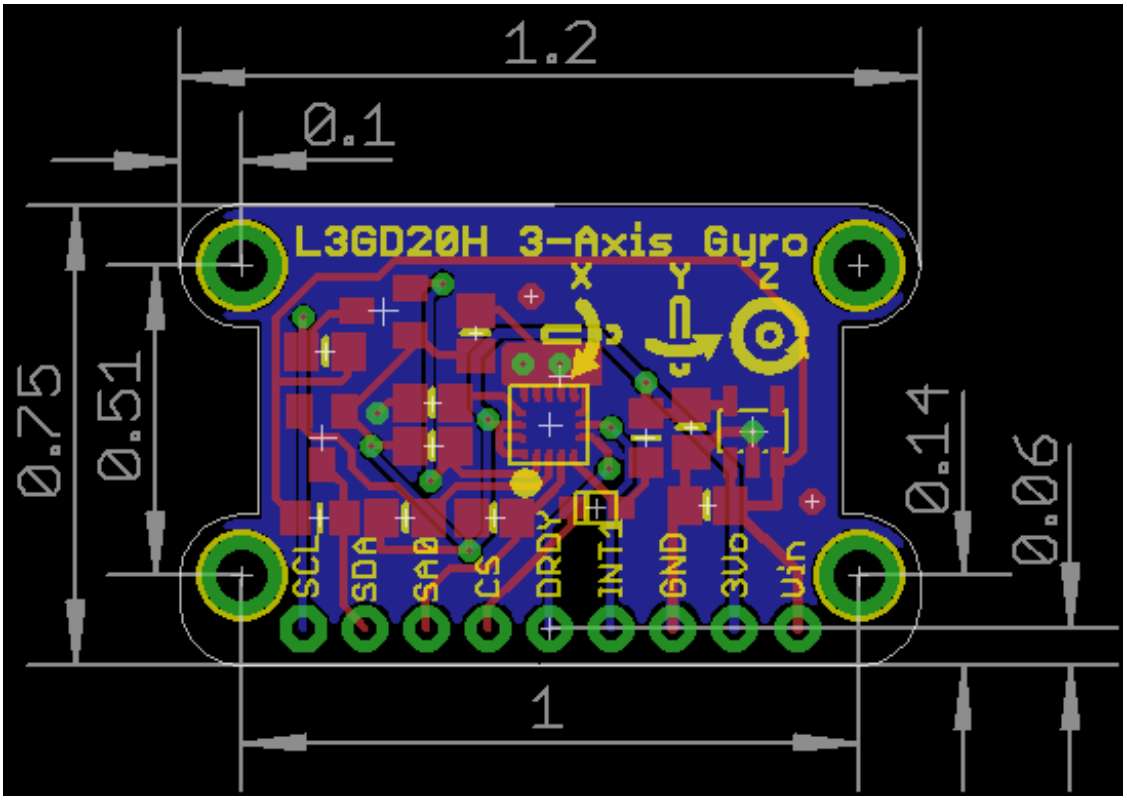
Downloads

Files

- L3GD20 Data Sheet (http://adafru.it/aV1)
- Technical Article: Everything about STMicroelectronics' 3-axis digital MEMS (http://adafru.it/aV4)gyroscopes (http://adafru.it/aV4)
- Adafruit L3GD20 Library driver for Arduino (http://adafru.it/cZ4)
- EagleCAD PCB files on GitHub (http://adafru.it/rga)
- Fritzing object in Fritzing library (http://adafru.it/aP3)

Schematic & Fabrication Print




1.2
0.1
L3GD20H 3-Axis Gyro
X
Y
Z
0.75
0.51
0.14
0.06
SCL
SDA
SA0
CS
DRDY
INT
GND
3Vo
Vin
1

F.A.Q.