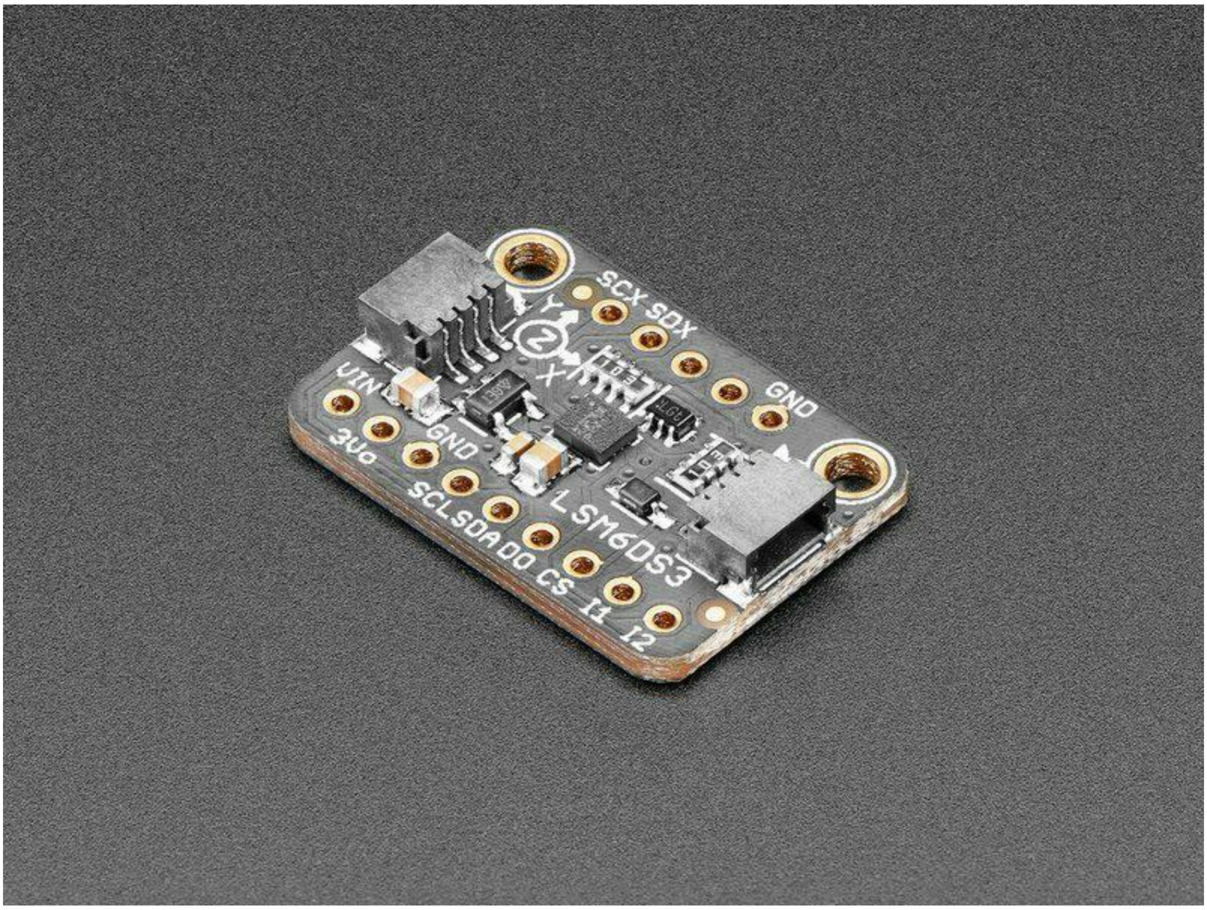
Adafruit LSM6DS3TR-C 6-DoF Accel + Gyro IMU

Created by Liz Clark

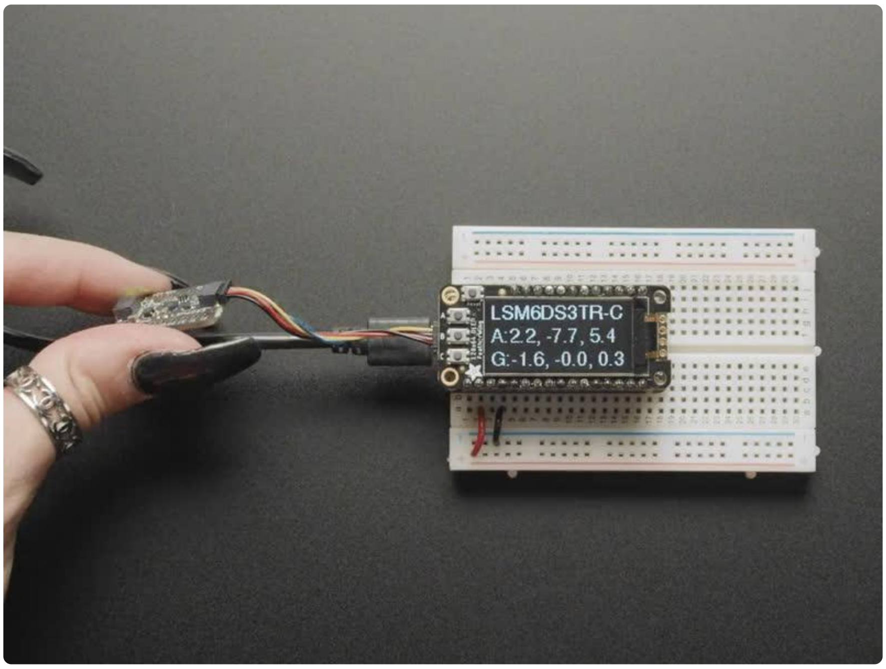


https://learn.adafruit.com/adafruit-lsm6ds3tr-c-6-dof-accel-gyro-imu

Last updated on 2022-12-01 04:12:34 PM EST

Table of Contents

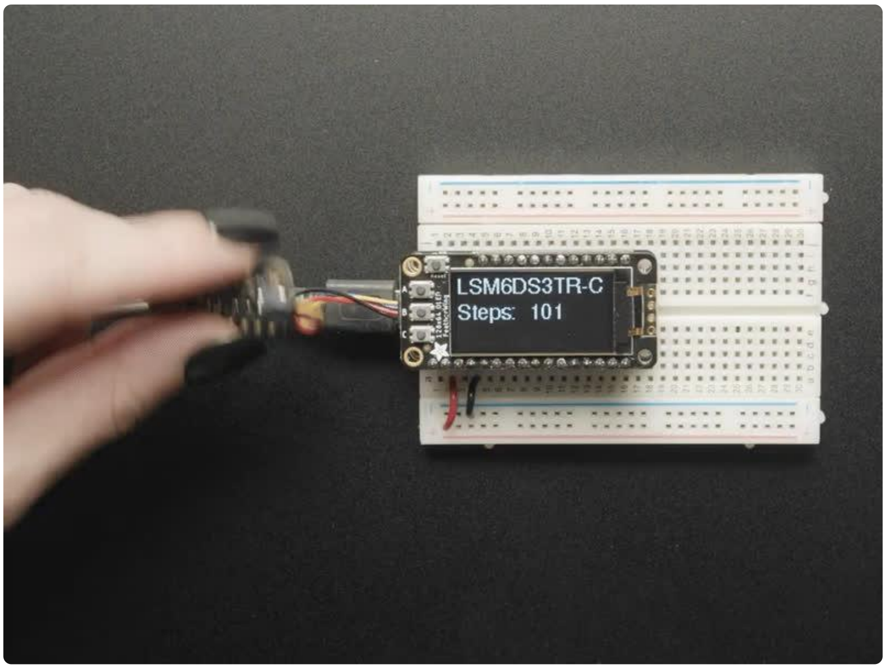
Overview



Add motion and orientation sensing to your project with this affordable 6 Degree of Freedom (6-DoF) sensor with sensors from ST. The board includes an ST LSM6DS3TR-C, a great entry-level 6-DoF IMU accelerometer + gyro. The 3-axis accelerometer can tell you which direction is down towards the Earth (by measuring gravity) or how fast the board is accelerating in 3D space. The 3-axis gyroscope can measure spin and twist.

front

This chip is very similar to the now-discontinued LSM6DS33, a great entry-level IMU. As part of the illustrious LSM6DS family, it's well-established, well-supported and this chip even has better performance! Note it is not firmware-compatible with the 'DS33, so you will need to recompile code (e.g. our Arduino and Python libraries support the whole family but you do have to indicate which exact chip you're using).



To make getting started fast and easy, we placed the sensors on a compact breakout board with voltage regulation and level-shifted inputs. That way you can use them with 3V or 5V power/logic devices without worry. Both I2C and SPI interfaces are available, so you'll be able to use them with any hardware setup. The breakout comes fully assembled and tested, with some extra header so you can use it on a breadboard. Four mounting holes make for a secure connection.

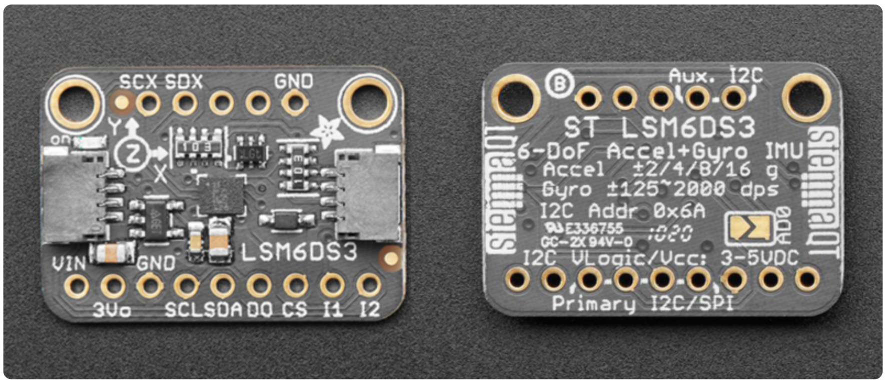
stemma

Additionally, since it speaks I2C, you can easily connect it up with two wires (plus power and ground!). We've even included SparkFun qwiic () compatible STEMMA QT () connectors for the I2C bus so you don't even need to solder! Just wire up to your favorite micro like the STM32F405 Feather () with a plug-and-play cable to get 6 DoF data ASAP. You can change the I2C address on the back using the solder jumper to have two of these sensor boards on one bus.

back

We also wrote libraries to help you get these sensors integrated with your Arduino/C++. This library contains an Arduino driver for the accel/gyro (). For advanced Arduino usage, ST has their own fully-featured library that includes extras such as FIFO management and tap detection () for the LSM6DS3TR-C. We also have a Python/CircuitPython library that will work on microcontroller or single board Linux computers ().

Pinouts



The default I2C address is 0x6A.

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 - I2C clock pin, connect to your microcontroller I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a 10K pullup on this pin.
- SDA -I2C data pin, connect to your microcontroller I2C data line. This pin is level shifted so you can use 3-5V logic, and there's a 10K pullup on this pin.
- STEMMA QT () - These connectors allow you to connectors to dev boards with S TEMMA QT connectors or to other things with various associated accessories ().

Address Jumper

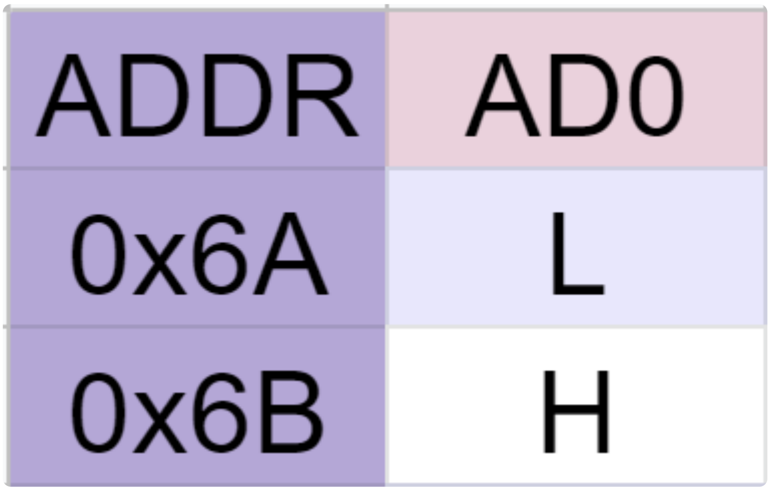
On the back of the board is one address jumper, labeled AD0, to the right of the I2C Addr label on the board silk. This jumper allows you to chain up to 2 of these boards on the same pair of I2C clock and data pins. To do so, you solder the jumper "closed" by connecting the two pads.

The default I2C address is 0x6A. The other address options can be calculated by “adding” the AD0 to the base of 0x6A.

AD0 sets the lowest bit with a value of 1. The final address is 0x6A + AD0 which would be 0x6B.

If AD0 is soldered closed, the address is 0x6A + 1 = 0x6B

The table below shows all possible addresses, and whether the pin(s) should be high (closed) or low (open).



ADDR	AD0
0x6A	L
0x6B	H

SPI Logic Pins

All pins going into the breakout have level shifting circuitry to make them 3-5V logic level safe. Use whatever logic level is on VIN!

- SCL - This is also the SPI Clock pin, it's an input to the chip
- DO - this is the serial Data Out / Microcontroller In Sensor Out pin, for data sent from the LSM6DS3TR-C to your processor.
- SDA - this is also the Serial Data In / Microcontroller Out Sensor In pin, for data sent from your processor to the LSM6DS3TR-C
- CS - this is the Chip Select pin, drop it low to start an SPI transaction. It's an input to the chip

If you want to connect multiple LSM6DS3TR-C to one microcontroller, have them share the SDI, DO and SCK pins. Then assign each one a unique CS pin.

Other Pins

- INT1 -This is the primary interrupt pin. You can setup the LSM6DS3TR-C to pull this low when certain conditions are met such as new measurement data being available. Consult the datasheet () for usage.
- INT2 -This is the primary interrupt pin. You can setup the LSM6DS3TR-C to pull this low when certain conditions are met such as new measurement data being available. Consult the datasheet () for usage.
- D0 - I2C Address pin. Pulling this pin high or bridging the solder jumper on the back will change the I2C address from 0x6A to 0x6B.
- SCX, SDX, Aux. I2C - Pins for advanced users to connect the LSM6DS3TR-C to another sensor. Consult the datasheet () for usage.

Power LED

- Power LED - In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled on. It is the green LED.

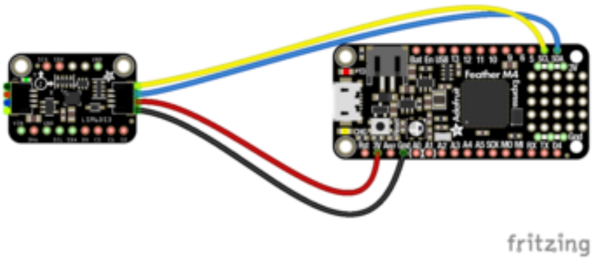
Python & CircuitPython

It's easy to use the LSM6DS3TR-C with Python or CircuitPython, and the Adafruit_CircuitPython_LSM6DS () module. This module allows you to easily write Python code that reads the values from the LSM6DS3TR-C's accelerometer and gyroscope.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library ().

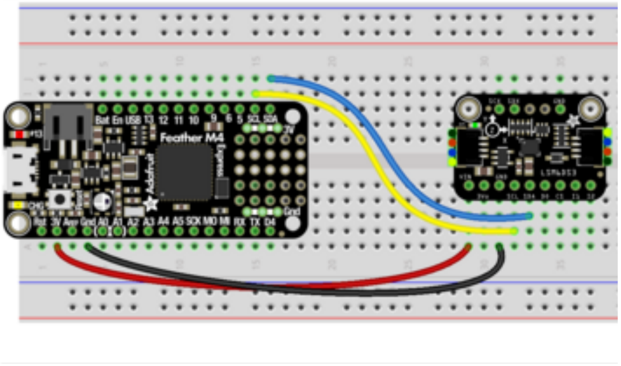
CircuitPython Microcontroller Wiring

First, wire up a LSM6DS3TR-C to your board exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C using one of the handy STEMMA QT () connectors:



Board 3V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

You can also use standard 0.100" pitch headers to wire it up on a breadboard:

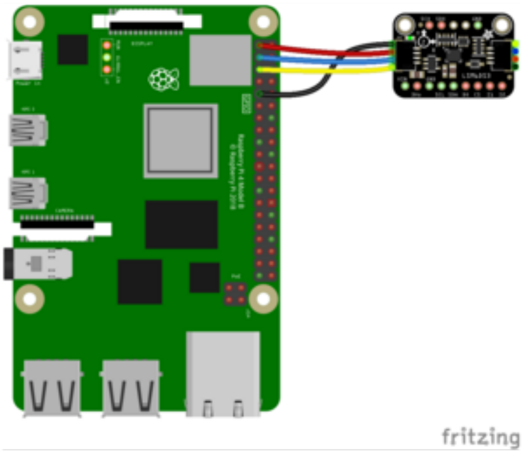


Board 3V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

Python Computer Wiring

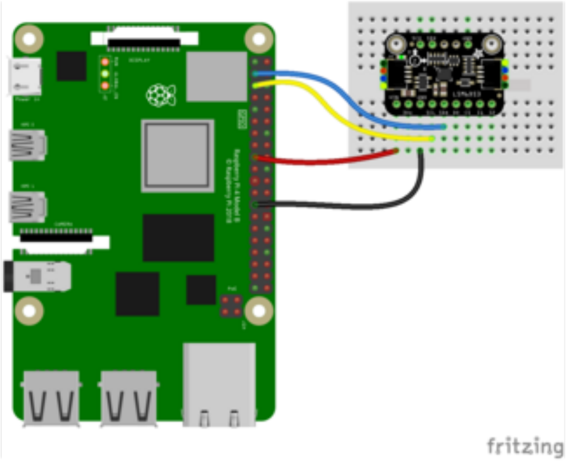
Since there's dozens of Linux computers/boards you can use, below shows wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported ().

Here's the Raspberry Pi wired to the sensor using I2C and a STEMMA QT () connector:



Pi 3V to sensor VIN (red wire)
Pi GND to sensor GND (black wire)
Pi SCL to sensor SCL (yellow wire)
Pi SDA to sensor SDA (blue wire)

Finally here is an example of how to wire up a Raspberry Pi to the sensor using a solderless breadboard:



Pi 3V to sensor VIN (red wire)
Pi GND to sensor GND (black wire)
Pi SCL to sensor SCL (yellow wire)
Pi SDA to sensor SDA (blue wire)

Python Installation of LSM6DS 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 ()!

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

- `pip3 install adafruit-circuitpython-lsm6ds`

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

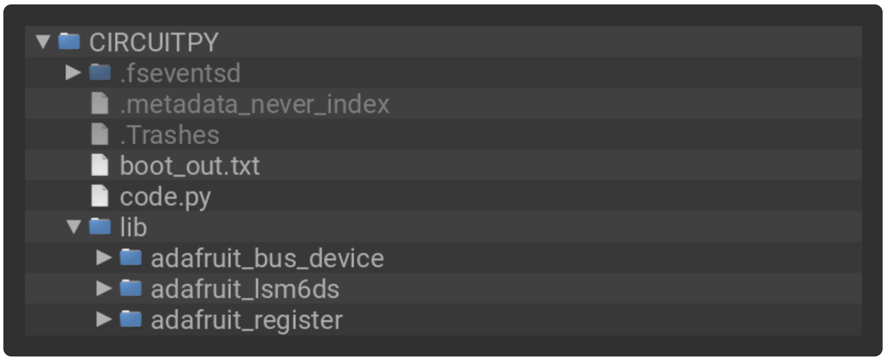
CircuitPython Usage

To use with CircuitPython, you need to first install the LSM6DS library, and its dependencies, into the lib folder on your CIRCUITPY drive. Then you need to update code.py with the example script.

Thankfully, we can do this in one go. In the example below, click the Download Project Bundle button below to download the necessary libraries and the code.py file in a zip file. Extract the contents of the zip file, and copy the entire lib folder and the c ode.py file to your CIRCUITPY drive.

Your CIRCUITPY/lib folder should contain the following folders:

- adafruit_bus_device/
- adafruit_lsm6ds/
- adafruit_register/



```
▼ CIRCUITPY
    ▶ .fseventsd
      .metadata_never_index
      .Trashes
      boot_out.txt
      code.py
    ▼ lib
        ▶ adafruit_bus_device
        ▶ adafruit_lsm6ds
        ▶ adafruit_register
```

Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing code.py with whatever you named the file:

`python3 code.py`

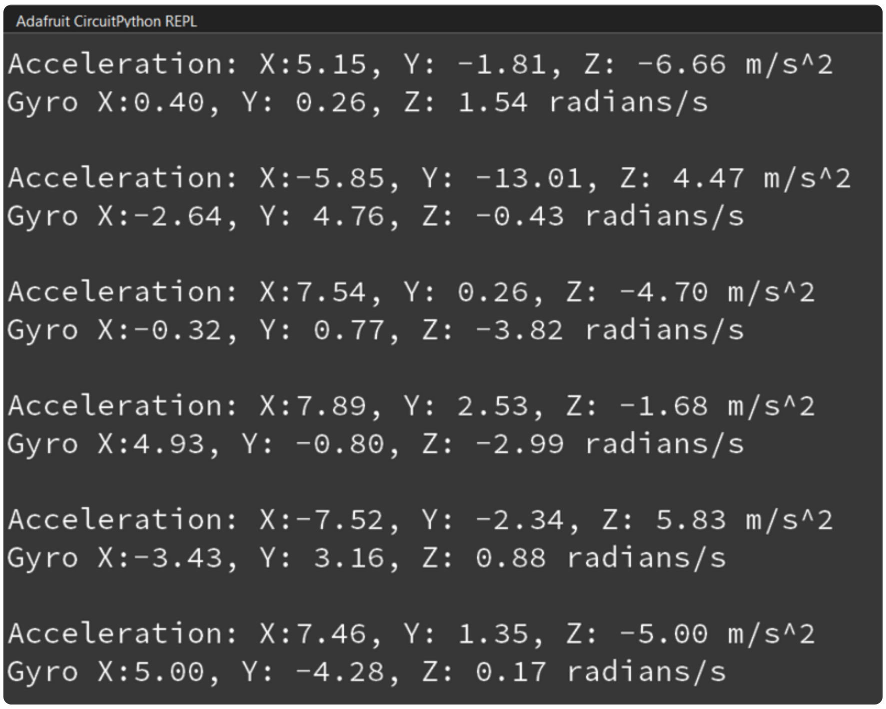
Example Code

```
# SPDX-FileCopyrightText: Copyright (c) 2022 Edrig
#
# SPDX-License-Identifier: MIT
import time
import board
from adafruit_lsm6ds.lsm6ds3 import LSM6DS3
```

```
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
sensor = LSM6DS3(i2c)

while True:
    print("Acceleration: X:%.2f, Y: %.2f, Z: %.2f m/s^2" % (sensor.acceleration))
    print("Gyro X:%.2f, Y: %.2f, Z: %.2f radians/s" % (sensor.gyro))
    print("")
    time.sleep(0.5)
```

If running CircuitPython: Once everything is saved to the CIRCUITPY drive, connect to the serial console () to see the data printed out!

If running Python: The console output will appear wherever you are running Python.



```
Acceleration: X:5.15, Y: -1.81, Z: -6.66 m/s^2
Gyro X:0.40, Y: 0.26, Z: 1.54 radians/s

Acceleration: X:-5.85, Y: -13.01, Z: 4.47 m/s^2
Gyro X:-2.64, Y: 4.76, Z: -0.43 radians/s

Acceleration: X:7.54, Y: 0.26, Z: -4.70 m/s^2
Gyro X:-0.32, Y: 0.77, Z: -3.82 radians/s

Acceleration: X:7.89, Y: 2.53, Z: -1.68 m/s^2
Gyro X:4.93, Y: -0.80, Z: -2.99 radians/s

Acceleration: X:-7.52, Y: -2.34, Z: 5.83 m/s^2
Gyro X:-3.43, Y: 3.16, Z: 0.88 radians/s

Acceleration: X:7.46, Y: 1.35, Z: -5.00 m/s^2
Gyro X:5.00, Y: -4.28, Z: 0.17 radians/s
```

Twist and turn your LSM6DS3TR-C and see the values from the accelerometer and gyroscope print out to the REPL!

First you import the necessary modules and libraries. Then you instantiate the sensor on I2C.

Then you're ready to read data from the sensor's accelerometer and gyroscope.

Finally, inside the loop, you see the x, y and z values from the accelerometer and gyroscope.

The accelerometer measures the x, y and z movement values and the gyroscope measures the x, y and z values in radians.

That's all there is to using the LSM6DS3TR-C with CircuitPython!

Python Docs

Python Docs ()

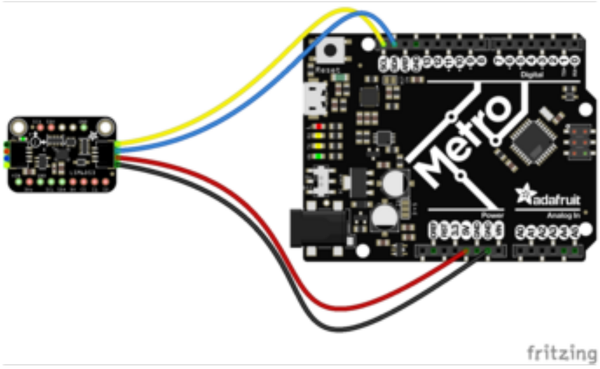
Arduino

Using the LSM6DS3TR-C with Arduino involves wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit_LSM6DS () library and running the provided example code.

Wiring

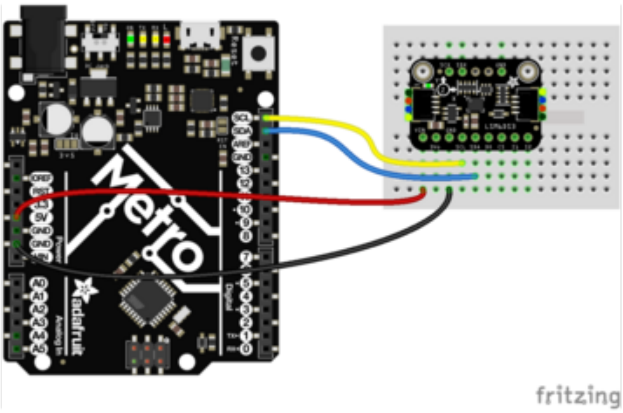
Wire as shown for a 5V board like an Uno. If you are using a 3V board, like an Adafruit Feather, wire the board's 3V pin to the LSM6DS3TR-C VIN.

Here is an Adafruit Metro wired up to the LSM6DS3TR-C using the STEMMA QT connector:



Board 5V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

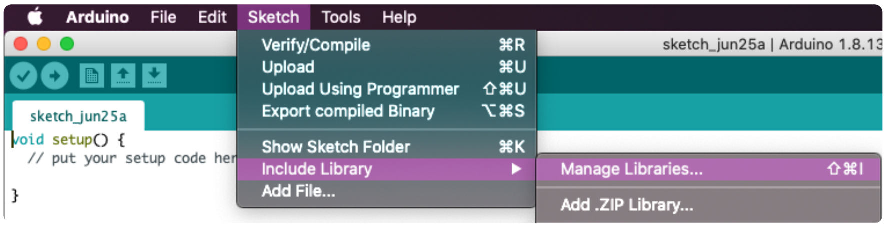
Here is an Adafruit Metro wired up using a solderless breadboard:



Board 5V to sensor VIN (red wire)
Board GND to sensor GND (black wire)
Board SCL to sensor SCL (yellow wire)
Board SDA to sensor SDA (blue wire)

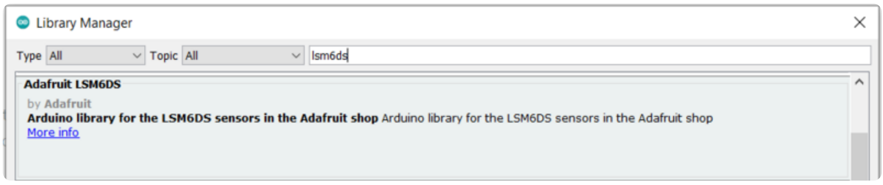
Library Installation

You can install the LSM6DS library for Arduino using the Library Manager in the Arduino IDE.



```
void setup() {
  // put your setup code her
}
```

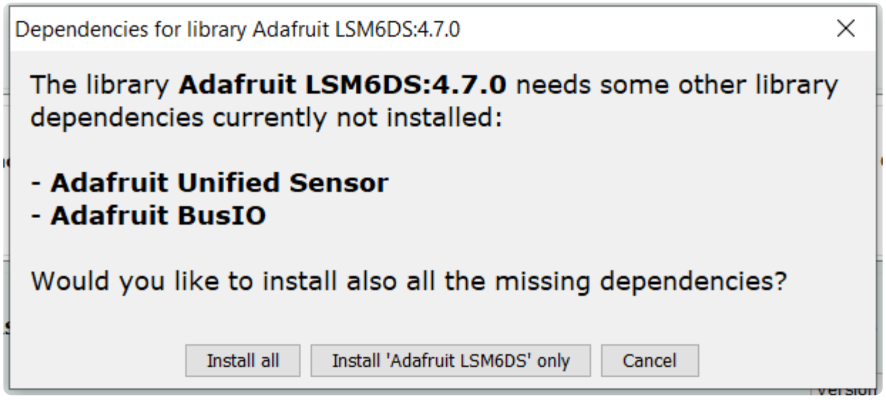
Click the Manage Libraries ... menu item, search for LSM6DS, and select the Adafruit LSM6DS library:



lsm6ds

Adafruit LSM6DS
by Adafruit
Arduino library for the LSM6DS sensors in the Adafruit shop Arduino library for the LSM6DS sensors in the Adafruit shop
More info

If asked about dependencies, click "Install all".



Dependencies for library Adafruit LSM6DS:4.7.0

The library **Adafruit LSM6DS:4.7.0** needs some other library dependencies currently not installed:

- **Adafruit Unified Sensor**
- **Adafruit BusIO**

Would you like to install also all the missing dependencies?

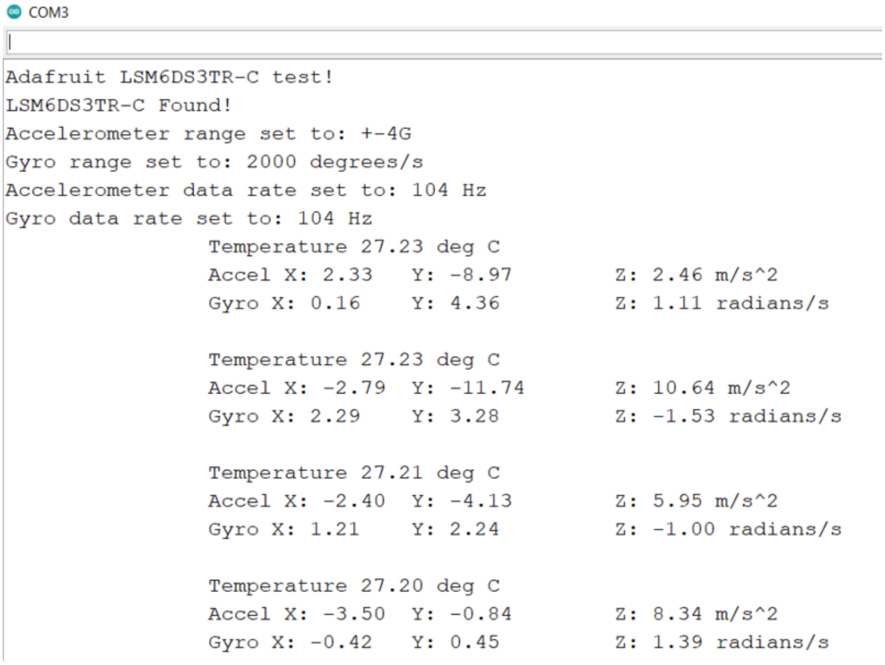
If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

Load Example

Open up File -> Examples -> Adafruit LSM6DS -> adafruit_lsm6ds3trc_test and upload to your Arduino wired to the sensor.

```
// Basic demo for accelerometer/gyro readings from Adafruit LSM6DS3TR-C

#include <Adafruit_LSM6DS3TRC.h>

// For SPI mode, we need a CS pin
#define LSM_CS 10
// For software-SPI mode we need SCK/MOSI/MISO pins
#define LSM_SCK 13
#define LSM_MISO 12
#define LSM_MOSI 11

Adafruit_LSM6DS3TRC lsm6ds3trc;

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

  Serial.println("Adafruit LSM6DS3TR-C test!");

  if (!lsm6ds3trc.begin_I2C()) {
    // if (!lsm6ds3trc.begin_SPI(LSM_CS)) {
    // if (!lsm6ds3trc.begin_SPI(LSM_CS, LSM_SCK, LSM_MISO, LSM_MOSI)) {
    Serial.println("Failed to find LSM6DS3TR-C chip");
    while (1) {
      delay(10);
    }
  }

  Serial.println("LSM6DS3TR-C Found!");

  // lsm6ds3trc.setAccelRange(LSM6DS_ACCEL_RANGE_2_G);
  Serial.print("Accelerometer range set to: ");
  switch (lsm6ds3trc.getAccelRange()) {
  case LSM6DS_ACCEL_RANGE_2_G:
    Serial.println("+-2G");
    break;
  case LSM6DS_ACCEL_RANGE_4_G:
    Serial.println("+-4G");
    break;
  case LSM6DS_ACCEL_RANGE_8_G:
    Serial.println("+-8G");
    break;
  case LSM6DS_ACCEL_RANGE_16_G:
    Serial.println("+-16G");
    break;
  }
  // lsm6ds3trc.setGyroRange(LSM6DS_GYRO_RANGE_250_DPS);
  Serial.print("Gyro range set to: ");
  switch (lsm6ds3trc.getGyroRange()) {
  case LSM6DS_GYRO_RANGE_125_DPS:
    Serial.println("125 degrees/s");
    break;
  case LSM6DS_GYRO_RANGE_250_DPS:
    Serial.println("250 degrees/s");
    break;
  case LSM6DS_GYRO_RANGE_500_DPS:
    Serial.println("500 degrees/s");
    break;
  case LSM6DS_GYRO_RANGE_1000_DPS:
    Serial.println("1000 degrees/s");
    break;
  case LSM6DS_GYRO_RANGE_2000_DPS:
    Serial.println("2000 degrees/s");
    break;
  case ISM330DHCX_GYRO_RANGE_4000_DPS:
    break; // unsupported range for the DS33
  }

  // lsm6ds3trc.setAccelDataRate(LSM6DS_RATE_12_5_HZ);
  Serial.print("Accelerometer data rate set to: ");
  switch (lsm6ds3trc.getAccelDataRate()) {
  case LSM6DS_RATE_SHUTDOWN:
    Serial.println("0 Hz");
    break;
  case LSM6DS_RATE_12_5_HZ:
    Serial.println("12.5 Hz");
    break;
  case LSM6DS_RATE_26_HZ:
    Serial.println("26 Hz");
    break;
  case LSM6DS_RATE_52_HZ:
    Serial.println("52 Hz");
    break;
  case LSM6DS_RATE_104_HZ:
    Serial.println("104 Hz");
    break;
  case LSM6DS_RATE_208_HZ:
    Serial.println("208 Hz");
    break;
  case LSM6DS_RATE_416_HZ:
    Serial.println("416 Hz");
    break;
  case LSM6DS_RATE_833_HZ:
    Serial.println("833 Hz");
    break;
  case LSM6DS_RATE_1_66K_HZ:
    Serial.println("1.66 KHz");
    break;
  case LSM6DS_RATE_3_33K_HZ:
    Serial.println("3.33 KHz");
    break;
  case LSM6DS_RATE_6_66K_HZ:
    Serial.println("6.66 KHz");
    break;
  }

  // lsm6ds3trc.setGyroDataRate(LSM6DS_RATE_12_5_HZ);
  Serial.print("Gyro data rate set to: ");
  switch (lsm6ds3trc.getGyroDataRate()) {
  case LSM6DS_RATE_SHUTDOWN:
    Serial.println("0 Hz");
    break;
  case LSM6DS_RATE_12_5_HZ:
    Serial.println("12.5 Hz");
    break;
  case LSM6DS_RATE_26_HZ:
    Serial.println("26 Hz");
    break;
  case LSM6DS_RATE_52_HZ:
    Serial.println("52 Hz");
    break;
  case LSM6DS_RATE_104_HZ:
    Serial.println("104 Hz");
    break;
  case LSM6DS_RATE_208_HZ:
    Serial.println("208 Hz");
    break;
  case LSM6DS_RATE_416_HZ:
    Serial.println("416 Hz");
    break;
  case LSM6DS_RATE_833_HZ:
    Serial.println("833 Hz");
    break;
  case LSM6DS_RATE_1_66K_HZ:
    Serial.println("1.66 KHz");
    break;
  case LSM6DS_RATE_3_33K_HZ:
    Serial.println("3.33 KHz");
    break;
  case LSM6DS_RATE_6_66K_HZ:
    Serial.println("6.66 KHz");
    break;
  }

  lsm6ds3trc.configInt1(false, false, true); // accelerometer DRDY on INT1
  lsm6ds3trc.configInt2(false, true, false); // gyro DRDY on INT2
}

void loop() {
  // Get a new normalized sensor event
  sensors_event_t accel;
  sensors_event_t gyro;
  sensors_event_t temp;
  lsm6ds3trc.getEvent(&accel, &gyro, &temp);

  Serial.print("\t\tTemperature ");
  Serial.print(temp.temperature);
  Serial.println(" deg C");

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

  /* Display the results (rotation is measured in rad/s) */
  Serial.print("\t\tGyro X: ");
  Serial.print(gyro.gyro.x);
  Serial.print(" \tY: ");
  Serial.print(gyro.gyro.y);
  Serial.print(" \tZ: ");
  Serial.print(gyro.gyro.z);
  Serial.println(" radians/s ");
  Serial.println();

  delay(100);

  //  // serial plotter friendly format

  //  Serial.print(temp.temperature);
  //  Serial.print(",");
  //  Serial.print(accel.acceleration.x);
  //  Serial.print(","); Serial.print(accel.acceleration.y);
  //  Serial.print(","); Serial.print(accel.acceleration.z);
  //  Serial.print(",");

  // Serial.print(gyro.gyro.x);
  // Serial.print(","); Serial.print(gyro.gyro.y);
  // Serial.print(","); Serial.print(gyro.gyro.z);
  // Serial.println();
  //  delayMicroseconds(10000);
}
```



```
Adafruit LSM6DS3TR-C test!
LSM6DS3TR-C Found!
Accelerometer range set to: +-4G
Gyro range set to: 2000 degrees/s
Accelerometer data rate set to: 104 Hz
Gyro data rate set to: 104 Hz
                Temperature 27.23 deg C
                Accel X: 2.33   Y: -8.97        Z: 2.46 m/s^2
                Gyro X: 0.16    Y: 4.36         Z: 1.11 radians/s

                Temperature 27.23 deg C
                Accel X: -2.79  Y: -11.74       Z: 10.64 m/s^2
                Gyro X: 2.29    Y: 3.28         Z: -1.53 radians/s

                Temperature 27.21 deg C
                Accel X: -2.40  Y: -4.13        Z: 5.95 m/s^2
                Gyro X: 1.21    Y: 2.24         Z: -1.00 radians/s

                Temperature 27.20 deg C
                Accel X: -3.50  Y: -0.84        Z: 8.34 m/s^2
                Gyro X: -0.42   Y: 0.45         Z: 1.39 radians/s
```

Upload the sketch to your board and open up the Serial Monitor (Tools -> Serial Monitor) at 115200 baud. You should see the values from the embedded temperature sensor, accelerometer and gyroscope being printed out. You'll see the values change depending on the movement of the sensor.

Arduino Docs

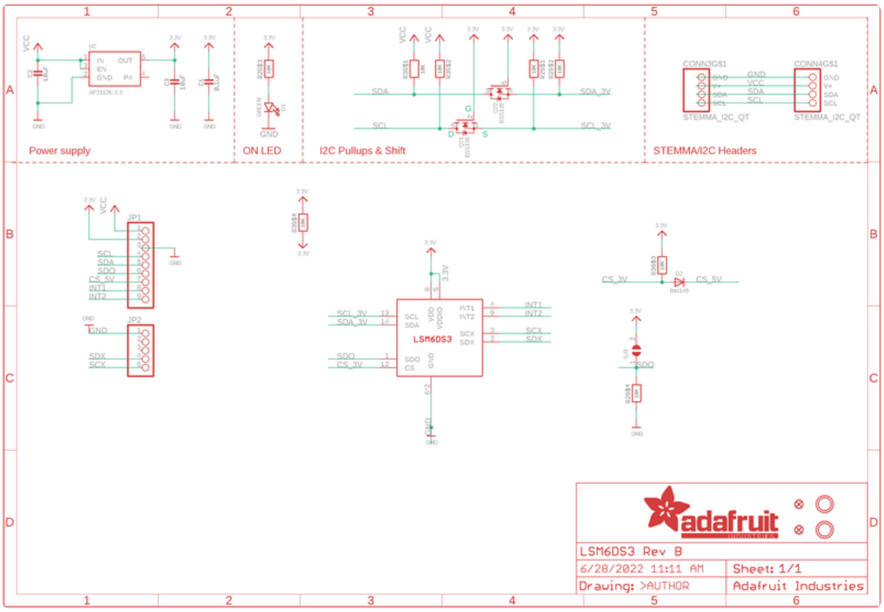
Arduino Docs ()

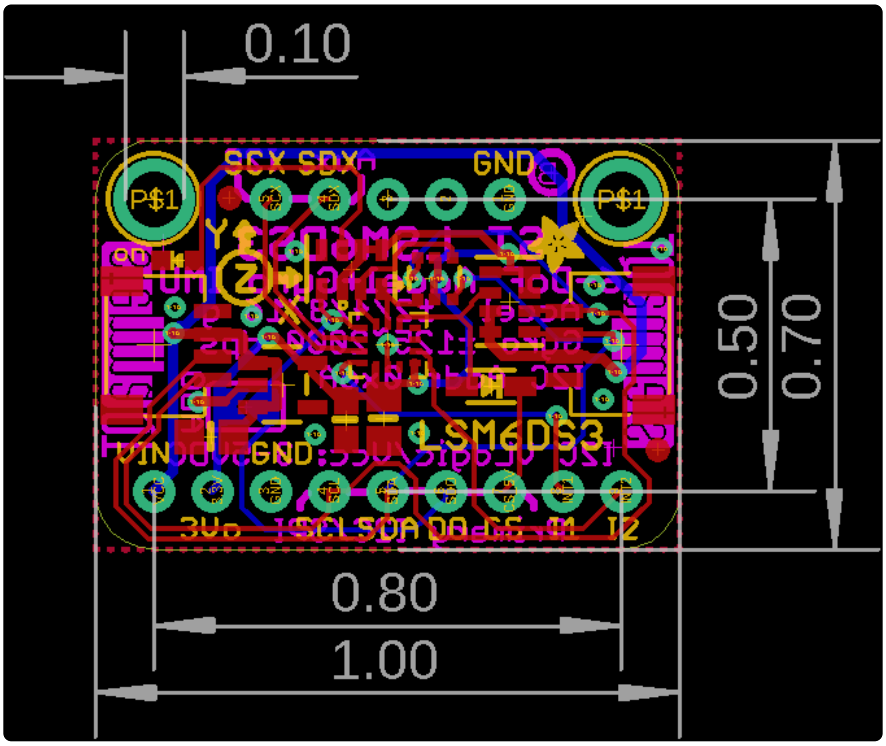
Downloads

Files

- LSM6DS3TR-C Datasheet ()
- EagleCAD PCB files on GitHub ()
- Fritzing object in the Adafruit Fritzing Library ()

Schematic and Fab Print


Power supply
ON LED
I2C Pullups & Shift
STEMMA/I2C Headers
VCC
IN
EN
GND
OUT
AP2112K-3.3
3.3V
GND
GREEN
SDA
SCL
SDA_3V
SCL_3V
G
D
S
BSS138
CONN3G$1
CONN4G$1
GND
VCC
SDA
SCL
V+
STEMMA_I2C_QT
JP1
JP2
SDO
CS_5V
INT1
INT2
SDX
SCX
LSM6DS3
VDD
VDDIO
CS
CS_3V
CS_5V
1N4148
SJ1
adafruit
LSM6DS3 Rev B
6/28/2022 11:11 AM
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries
1
2
3
4
5
6
A
B
C
D


0.10
0.50
0.70
0.80
1.00
SCX SDX
GND
P$1
P$1
VIN
GND
LSM6DS3
3Vo
SCL
SDA
DO
CS
I1
I2