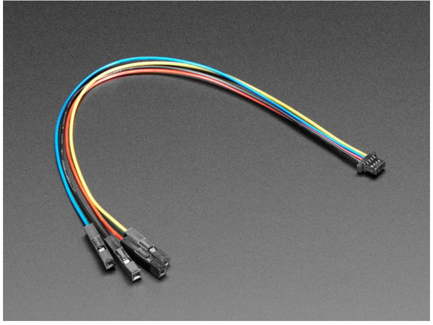
STEMMA QT / Qwiic JST SH 4-pin Cable - 100mm Long
Out of Stock



- STEMMA QT / Qwiic JST SH 4-pin Cable with Premium Female Sockets
  Out of Stock

21
Beginner
Product guide

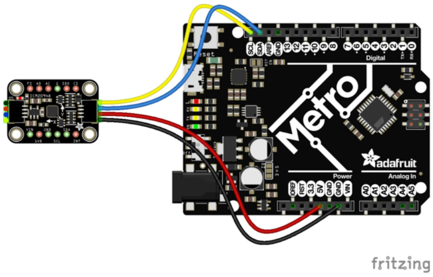
Arduino

Using the ICM20948 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit ICM20X library we've written, and running the provided example code.

I2C Wiring

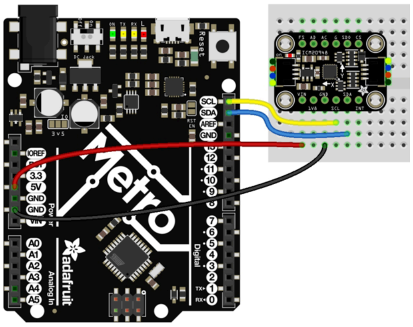
Use this wiring if you want to connect via I2C interface. The default I2C address for the ICM20948 is **0x69** but it can be switched to **0x68** by pulling the address pin low to GND.

Here is how to wire up the sensor using one of the **STEMMA QT** connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.



- Connect **board VIN (red wire)** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V,** connect to that instead.
- Connect **board GND (black wire)** to **Arduino GND**
- Connect **board SCL (yellow wire)** to **Arduino SCL**
- Connect **board SDA (blue wire)** to **Arduino SDA**

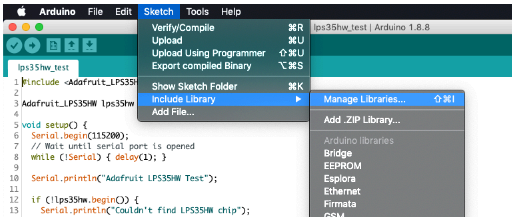
Here is how to wire the sensor to a board using a solderless breadboard:



- Connect **board VIN (red wire)** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V,** connect to that instead.
- Connect **board GND (black wire)** to **Arduino GND**
- Connect **board SCL (yellow wire)** to **Arduino SCL**
- Connect **board SDA (blue wire)** to **Arduino SDA**

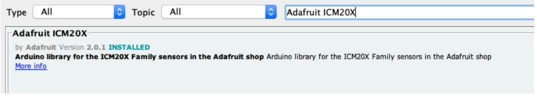
Library Installation

You can install the **Adafruit ICM20X l**ibrary for Arduino using the Library Manager in the Arduino IDE. This library is compatible with both the ICM20948 and it's sister sensor, the ICM20649



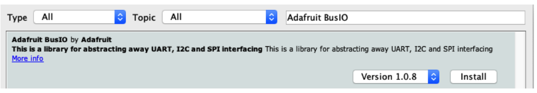
```
#include <Adafruit_LPS35H

Adafruit_LPS35HW lps35hw

void setup() {
  Serial.begin(115200);
  // Wait until serial port is opened
  while (!Serial) { delay(1); }

  Serial.println("Adafruit LPS35HW Test");

  if (!lps35hw.begin()) {
    Serial.println("Couldn't find LPS35HW chip");
```

Click the **Manage Libraries ...** menu item, search for **Adafruit ICM20X**, and select the **Adafruit ICM20X** library:



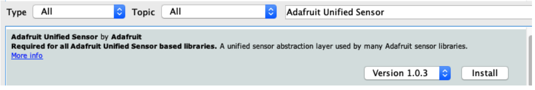
Adafruit ICM20X
by **Adafruit** Version **2.0.1** **INSTALLED**
Arduino library for the ICM20X Family sensors in the Adafruit shop Arduino library for the ICM20X Family sensors in the Adafruit shop
More info

Follow the same process for the **Adafruit BusIO** library.



Adafruit BusIO by **Adafruit**
This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing
More info

Finally follow the same process for the **Adafruit Unified Sensor** library:

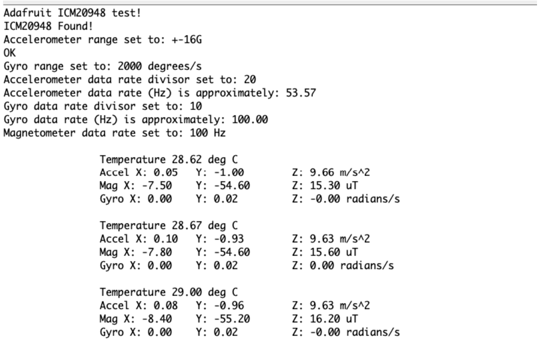


Adafruit Unified Sensor by **Adafruit**
Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used by many Adafruit sensor libraries.
More info

Load Example

Open up **File -> Examples -> Adafruit ICM20X -> adafruit_ICM20948_test**

After opening the demo file, upload to your Arduino wired up to the sensor. Once you upload the code, you will see the **Temperature** as well as **X, Y, and Z values** for the **Gyro Accelerometer, and Magnetometer** being printed when you open the Serial Monitor (**Tools->Serial Monitor**) at **115200 baud**, similar to this:



```
Adafruit ICM20948 test!
ICM20948 Found!
Accelerometer range set to: +-16G
OK
Gyro range set to: 2000 degrees/s
Accelerometer data rate divisor set to: 20
Accelerometer data rate (Hz) is approximately: 53.57
Gyro data rate divisor set to: 10
Gyro data rate (Hz) is approximately: 100.00
Magnetometer data rate set to: 100 Hz

                Temperature 28.62 deg C
                Accel X: 0.05   Y: -1.00        Z: 9.66 m/s^2
                Mag X: -7.50    Y: -54.60       Z: 15.30 uT
                Gyro X: 0.00    Y: 0.02         Z: -0.00 radians/s

                Temperature 28.67 deg C
                Accel X: 0.10   Y: -0.93        Z: 9.63 m/s^2
                Mag X: -7.80    Y: -54.60       Z: 15.60 uT
                Gyro X: 0.00    Y: 0.02         Z: 0.00 radians/s

                Temperature 29.00 deg C
                Accel X: 0.08   Y: -0.96        Z: 9.63 m/s^2
                Mag X: -8.40    Y: -55.20       Z: 16.20 uT
                Gyro X: 0.00    Y: 0.02         Z: -0.00 radians/s
```

Example Code

Download File
Copy Code

```
// Basic demo for accelerometer readings from Adafruit ICM20948

#include <Adafruit_ICM20X.h>
#include <Adafruit_ICM20948.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_ICM20948 icm;
uint16_t measurement_delay_us = 65535; // Delay between measurements for testing
// For SPI mode, we need a CS pin
#define ICM_CS 10
// For software-SPI mode we need SCK/MOSI/MISO pins
#define ICM_SCK 13
#define ICM_MISO 12
#define ICM_MOSI 11

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

  Serial.println("Adafruit ICM20948 test!");

  // Try to initialize!
  if (!icm.begin_I2C()) {
    // if (!icm.begin_SPI(ICM_CS)) {
    // if (!icm.begin_SPI(ICM_CS, ICM_SCK, ICM_MISO, ICM_MOSI)) {

    Serial.println("Failed to find ICM20948 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("ICM20948 Found!");
  // icm.setAccelRange(ICM20948_ACCEL_RANGE_16_G);
  Serial.print("Accelerometer range set to: ");
  switch (icm.getAccelRange()) {
  case ICM20948_ACCEL_RANGE_2_G:
    Serial.println("+-2G");
    break;
  case ICM20948_ACCEL_RANGE_4_G:
    Serial.println("+-4G");
    break;
  case ICM20948_ACCEL_RANGE_8_G:
    Serial.println("+-8G");
    break;
  case ICM20948_ACCEL_RANGE_16_G:
    Serial.println("+-16G");
    break;
  }
  Serial.println("OK");

  // icm.setGyroRange(ICM20948_GYRO_RANGE_2000_DPS);
```

```
  Serial.print("Gyro range set to: ");
  switch (icm.getGyroRange()) {
  case ICM20948_GYRO_RANGE_250_DPS:
    Serial.println("250 degrees/s");
    break;
  case ICM20948_GYRO_RANGE_500_DPS:
    Serial.println("500 degrees/s");
    break;
  case ICM20948_GYRO_RANGE_1000_DPS:
    Serial.println("1000 degrees/s");
    break;
  case ICM20948_GYRO_RANGE_2000_DPS:
    Serial.println("2000 degrees/s");
    break;
  }

  //  icm.setAccelRateDivisor(4095);
  uint16_t accel_divisor = icm.getAccelRateDivisor();
  float accel_rate = 1125 / (1.0 + accel_divisor);

  Serial.print("Accelerometer data rate divisor set to: ");
  Serial.println(accel_divisor);
  Serial.print("Accelerometer data rate (Hz) is approximately: ");
  Serial.println(accel_rate);

  //  icm.setGyroRateDivisor(255);
  uint8_t gyro_divisor = icm.getGyroRateDivisor();
  float gyro_rate = 1100 / (1.0 + gyro_divisor);

  Serial.print("Gyro data rate divisor set to: ");
  Serial.println(gyro_divisor);
  Serial.print("Gyro data rate (Hz) is approximately: ");
  Serial.println(gyro_rate);

  // icm.setMagDataRate(AK09916_MAG_DATARATE_10_HZ);
  Serial.print("Magnetometer data rate set to: ");
  switch (icm.getMagDataRate()) {
  case AK09916_MAG_DATARATE_SHUTDOWN:
    Serial.println("Shutdown");
    break;
  case AK09916_MAG_DATARATE_SINGLE:
    Serial.println("Single/One shot");
    break;
  case AK09916_MAG_DATARATE_10_HZ:
    Serial.println("10 Hz");
    break;
  case AK09916_MAG_DATARATE_20_HZ:
    Serial.println("20 Hz");
    break;
  case AK09916_MAG_DATARATE_50_HZ:
    Serial.println("50 Hz");
    break;
  case AK09916_MAG_DATARATE_100_HZ:
    Serial.println("100 Hz");
    break;
  }
  Serial.println();

}

void loop() {

  //  /* Get a new normalized sensor event */
  sensors_event_t accel;
  sensors_event_t gyro;
  sensors_event_t mag;
  sensors_event_t temp;
  icm.getEvent(&accel, &gyro, &temp, &mag);

  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 ");

  Serial.print("\t\tMag X: ");
  Serial.print(mag.magnetic.x);
  Serial.print(" \tY: ");
  Serial.print(mag.magnetic.y);
  Serial.print(" \tZ: ");
  Serial.print(mag.magnetic.z);
  Serial.println(" uT");

  /* Display the results (acceleration is measured in m/s^2) */
  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.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.print(",");
  //  Serial.print(mag.magnetic.x);
  //  Serial.print(","); Serial.print(mag.magnetic.y);
  //  Serial.print(","); Serial.print(mag.magnetic.z);

  //  Serial.println();
  //
  //  delayMicroseconds(measurement_delay_us);
}
```

View on GitHub

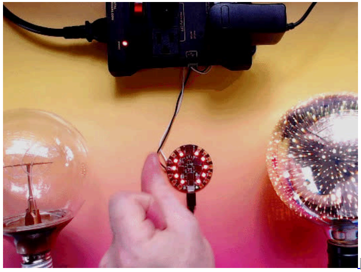
Page last edited January 22, 2025

Text editor powered by tinymce.

Pinouts Python & CircuitPython

Related Guides

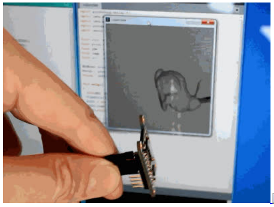
Wave Shield
Wave Shield
By lady ada
86



Your First Power Switch Relay Project - Circuit Playground Smart Plug
Your First Power Switch Relay Project - Circuit...
By Erin St Blaine
29

LSM6DSOX, ISM330DHC, & LSM6DSO32 6 DoF IMUs
LSM6DSOX, ISM330DHC, & LSM6DSO32 6 DoF IMUs
By Bryan Siepert
21

UTi165 Thermal Fever Scanner Camera
UTi165 Thermal Fever Scanner Camera
By lady ada
2



AHRS for Adafruit's 9-DOF, 10-DOF, LSM9DS0 Breakouts
AHRS for Adafruit's 9-DOF, 10-DOF, LSM9DS0 Breakouts
By Kevin Townsend
24

Adafruit Metro Mini
Adafruit Metro Mini
By lady ada
17

Network Interface Failover using FONA