SmartElex Triple-axis Magnetometer - MMC5603

Sense the magnetic fields surrounding us with this handy triple-axis magnetometer (compass) module. Magnetometers can sense where the strongest magnetic force is coming from, generally used to detect magnetic north, but can also be used for measuring magnetic fields. This sensor tends to be paired with a 6-DoF (degree of freedom) accelerometer/gyroscope to create a 9-DoF inertial measurement unit that can detect its orientation in real-space, thanks to Earth's stable magnetic field. It's a great match for the LSM6DSOX from ST!

We based this breakout on the MMC5603, a great general purpose magnetometer with a very wide range and both I2C and SPI interfaces. This compact sensor is very easy to use. Simply download our library and connect the SCL pin to your I2C clock pin, and SDA pin to your I2C data pin and upload our test program to read out magnetic field data. If you'd like, you can also use SPI to receive data (we just happen to prefer I2C here).

This sensor can sense ranges from ±30 Gauss (±3000uT or ±3mT) with no range-setting required and full 20 bit output, up to 1000 Hz rate reading. The range makes it good for reading Earth's magnetic field (which maxes at about 0.6 Gauss) or some basic magnets. It isn't good for very strong rare earth magnets, for that check out the TLV293.

To make life easier, so you can focus on your important work, we've taken the MMC5603 and put it onto a breakout PCB along with support circuitry to let you use this little wonder with 3.3V (Feather/Raspberry Pi) or 5V (Arduino/ Metro328) logic levels. Additionally, since it speaks I2C, you can easily connect it up with two wires (plus power and ground!).

It's fully assembled and tested. Comes with a bit of 0.1" standard header in case you want to use it with a breadboard or perfboard. Four 2.5mm (0.1") mounting holes for easy attachment.

The default I2C address is **0x30**.

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.

Power LED

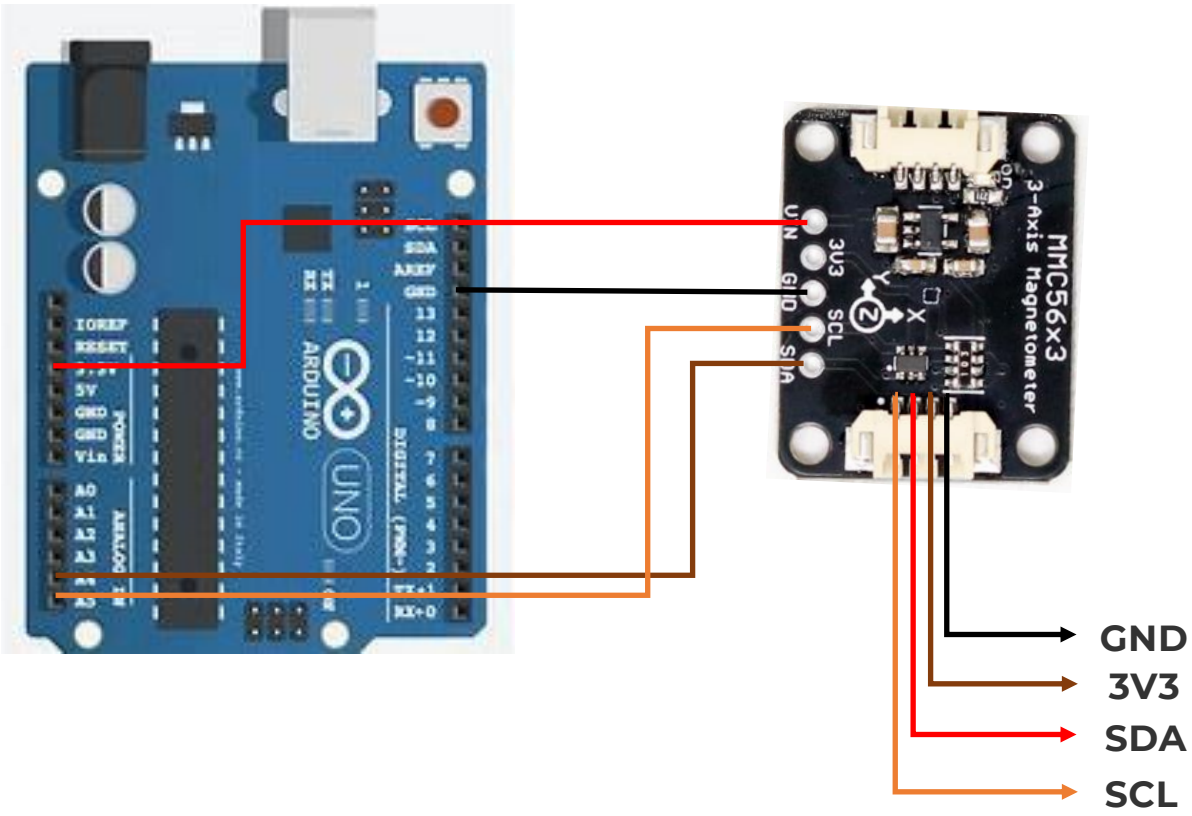
- **Power LED -** In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled **on**. It is the green LED.

Arduino

Using the MMC5603 with Arduino involves wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit_MMC56x3 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 MMC5603 VIN.



Arduino	MMC5603
SCL(A5)	SCL
SDA(A4)	SDA
5v OR 3.3v	VIN
GND	GND

- **Board 5V** to **sensor VIN**
- **Board GND** to **sensor GND**
- **Board SCL** to **sensor SCL**
- **Board SDA** to **sensor SDA**

Library Installation

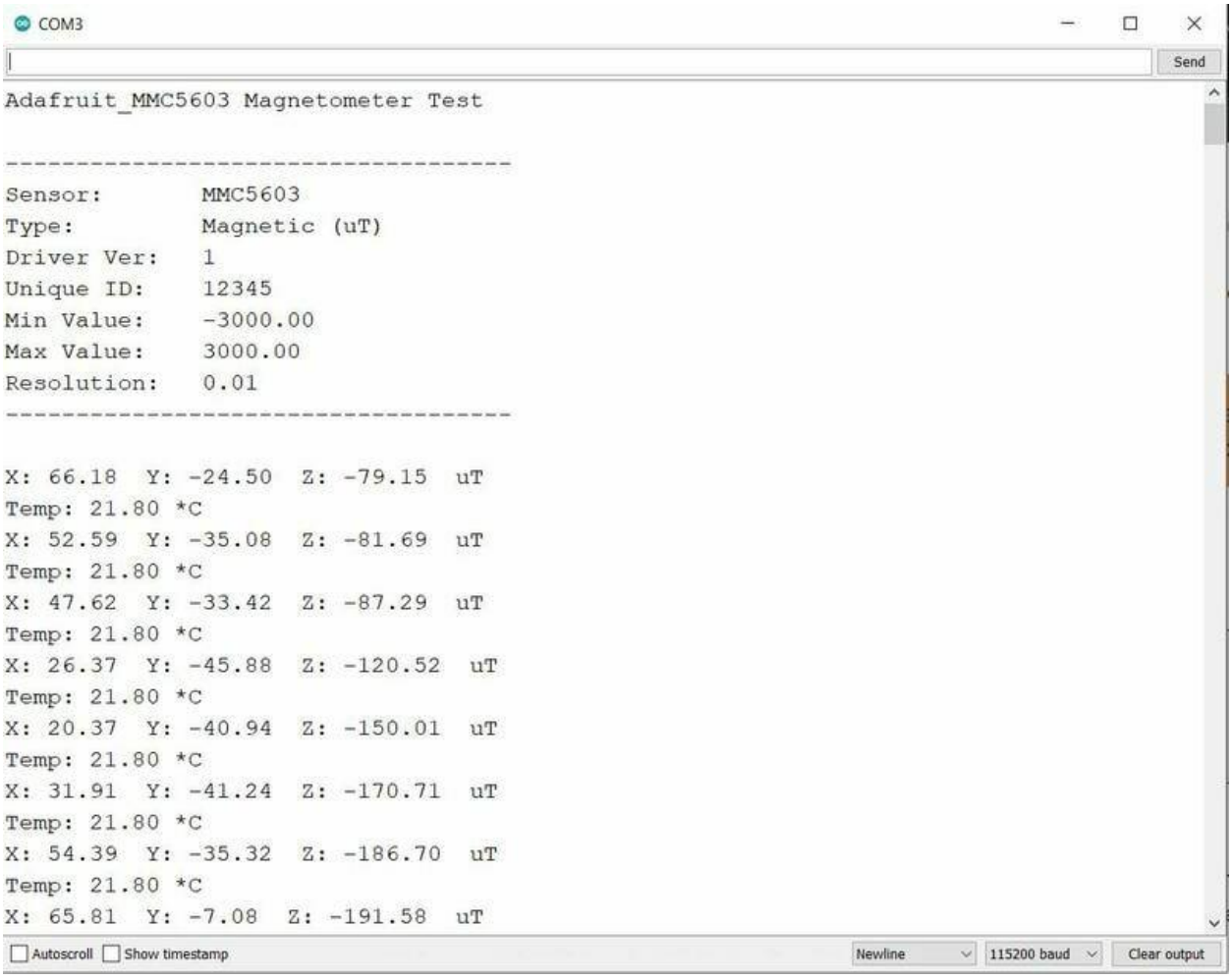
You can install the **Adafruit MMC56x3** library for Arduino using the Library Manager in the Arduino IDE.

Click the **Manage Libraries ...** menu item, search for **Adafruit MMC56x3**, and select the **Adafruit MMC56x3** library:

Example Code

```
#include <Adafruit_MMC56x3.h>

/* Assign a unique ID to this sensor at the same time */
Adafruit_MMC5603 mmc = Adafruit_MMC5603(12345);

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

  Serial.println("Adafruit_MMC5603 Magnetometer Test");
  Serial.println("");

  /* Initialise the sensor */
  if (!mmc.begin(MMC56X3_DEFAULT_ADDRESS, &Wire)) {  // I2C mode
    /* There was a problem detecting the MMC5603 ... check your connections */
    Serial.println("Ooops, no MMC5603 detected ... Check your wiring!");
    while (1) delay(10);
  }

  /* Display some basic information on this sensor */
  mmc.printSensorDetails();
}
void loop(void) {
  // Get a new sensor event
  sensors_event_t event;
  mmc.getEvent(&event);

  // Display the results (magnetic vector values are in micro-Tesla (uT))
  Serial.print("X: ");
  Serial.print(event.magnetic.x);
  Serial.print("  ");
```

```
  Serial.print("Y: ");
  Serial.print(event.magnetic.y);
  Serial.print("  ");
  Serial.print("Z: ");
  Serial.print(event.magnetic.z);
  Serial.print("  ");
  Serial.println("uT");

  // Read and display temperature
  float temp_c = mmc.readTemperature();
  Serial.print("Temp: "); Serial.print(temp_c); Serial.println(" *C");
  // Delay before the next sample
  delay(100);
}
```



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 and magnetometer being printed out. You'll see the values change depending on the movement of the sensor.