SmartElex Micro Magnetometer - MMC5983MA

The MMC5983MA is a micro-sized. sensor that utilizes the highly sensitive triple-axis magnetometer by MEMSIC. The MMC5983MA is capable of sensing down to 0.4mG, enabling a heading accuracy of ±0.5°. The MMC5983MA IMU communicates over I^2C.

Saturation is a problem for all mag sensors. The MMC5983MA has built-in degaussing circuitry to clear any residual magnetization. Output rates of 1000Hz, ±8G FSR, and 18-bit resolution make the MMC5983MA a phenomenal magnetic sensor for electronic compass applications.

Hardware Overview

The MMC5983MA is an AEC-Q100 qualified, fully integrated 3-axis magnetic sensor with on-chip signal processing and integrated I^2C/SPI bus. It has superior dynamic range and accuracy with ±8G FSR, 18bit operation, 0.4mG total RMS noise, and can enable heading accuracy of 0.5º. More information can be found in the datasheet.

Pins

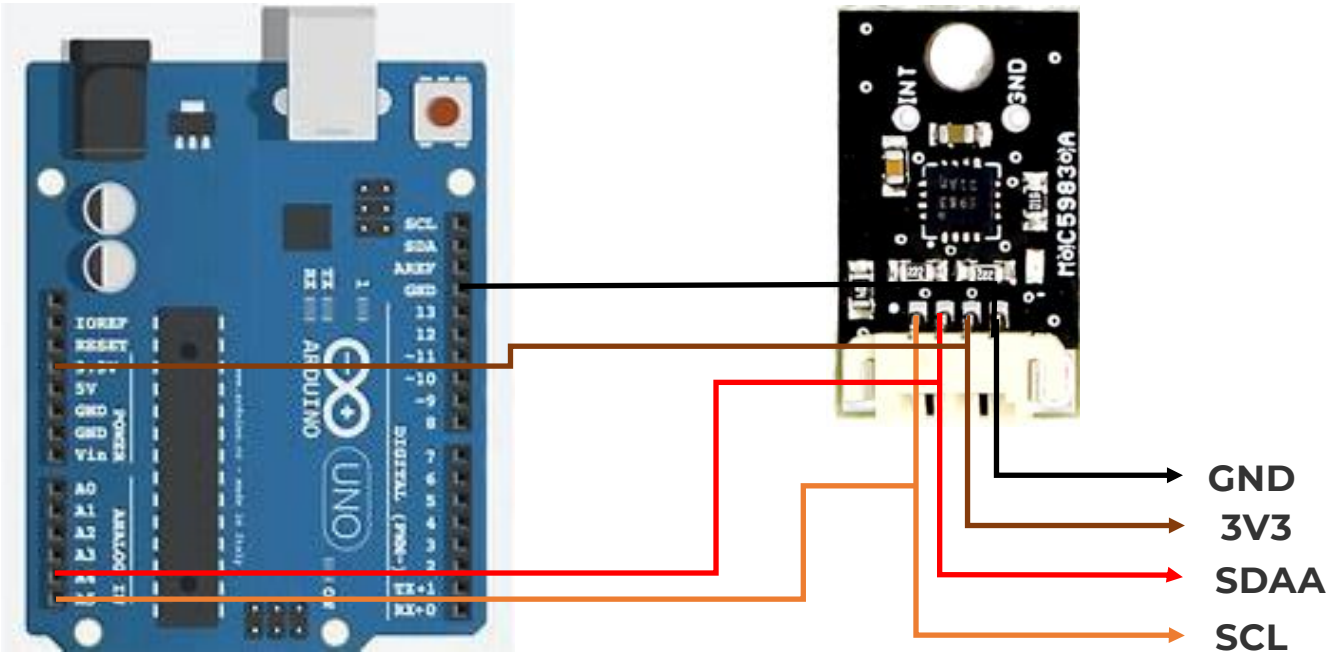
We've broken out the interrupt and ground pins to PTH on either side of the board. The interrupt pin is active high - writing “1” will enable the interrupt for completed measurements. Once a measurement is finished, either magnetic field or temperature, an interrupt will be sent to the host.

LED

If power consumption is an issue, cutting this jumper will disable the Power LED on the front of the board.

Wiring

Connecting the MMC5983MA to Arduino:



Arduino	MMC5983MA
SCL(A5)	SCL
SDA(A4)	SDA
5v OR 3.3v	VIN
GND	GND

Example 1: I2C Simple Measurement

This first example just does some basic measurements. To find Example 1, go to **File -> Examples -> SparkFun MMC5983MA Magnetometer Arduino Library -> Example1-I2C_Simple_measurement**.

```
#include <Wire.h>


#include <SparkFun_MMC5983MA_Arduino_Library.h> //Click here to get the library: http://librarymanager/All#SparkFun_MMC5983MA
```

```
SFE_MMC5983MA myMag;

void setup()
{
  Serial.begin(115200);
  Serial.println("MMC5983MA Example");
  Wire.begin();

  if (myMag.begin() == false)
  {
    Serial.println("MMC5983MA did not respond - check your wiring. Freezing.");
    while (true)
      ;
  }
  myMag.softReset();

  Serial.println("MMC5983MA connected");

  int celsius = myMag.getTemperature();
  float fahrenheit = (celsius * 9.0f / 5.0f) + 32.0f;

  Serial.print("Die temperature: ");
  Serial.print(celsius);
  Serial.print("°C or ");
  Serial.print(fahrenheit, 0);
  Serial.println("°F.");
}
void loop()
{
  uint32_t currentX = 0;
  uint32_t currentY = 0;
  uint32_t currentZ = 0;
  double scaledX = 0;
```

```
double scaledY = 0;
double scaledZ = 0;

// This reads the X, Y and Z channels consecutively
// (Useful if you have one or more channels disabled)
currentX = myMag.getMeasurementX();
currentY = myMag.getMeasurementY();
currentZ = myMag.getMeasurementZ();

// Or, we could read all three simultaneously
//myMag.getMeasurementXYZ(&currentX, &currentY, &currentZ);

Serial.print("X axis raw value: ");
Serial.print(currentX);
Serial.print("\tY axis raw value: ");
Serial.print(currentY);
Serial.print("\tZ axis raw value: ");
Serial.println(currentZ);

// The magnetic field values are 18-bit unsigned. The _approximate_ zero (mid) point is 2^17 (131072).
// Here we scale each field to +/- 1.0 to make it easier to convert to Gauss.
//
// Please note: to properly correct and calibrate the X, Y and Z channels, you need to determine true
// offsets (zero points) and scale factors (gains) for all three channels. Futher details can be found at:
// https://thecavepearlproject.org/2015/05/22/calibrating-any-compass-or-accelerometer-for-arduino/
scaledX = (double)currentX - 131072.0;
scaledX /= 131072.0;
scaledY = (double)currentY - 131072.0;
scaledY /= 131072.0;
scaledZ = (double)currentZ - 131072.0;
scaledZ /= 131072.0;

// The magnetometer full scale is +/- 8 Gauss
```

```
  // Multiply the scaled values by 8 to convert to Gauss
  Serial.print("X axis field (Gauss): ");
  Serial.print(scaledX * 8, 5); // Print with 5 decimal places

  Serial.print("\tY axis field (Gauss): ");
  Serial.print(scaledY * 8, 5);

  Serial.print("\tZ axis field (Gauss): ");
  Serial.println(scaledZ * 8, 5);

  Serial.println();
  delay(100);
}
```

Once you're ready to go, go ahead and hit the upload button (the right facing arrow button under the "Edit" menu item). Once your code is uploaded, open the Serial Monitor and you'll see X, Y, and Z values start printing out.