



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Triple Axis Magnetometer - HMC5883L Breakout Quickstart Guide

by JordanDee | June 30, 2011 |

Skill Level: ★ Beginner

Overview:

This is a breakout board for Honeywell's HMC5883L, a 3-axis magnetometer. Magnetometers have a wide range of uses. The most common include using the chip as a digital compass to sense direction or using them to detect ferrous (magnetic) metals.



Triple Axis Magnetometer Breakout - HMC5883L

Requirements:

This breakout board can be hooked up to a number of microcontrollers, as long as they have an I²C interface . However, for this guide you will find the following helpful:

- Arduino Uno
- Breadboard
- soldering iron
- solder
- male headers or other connection method
- (optional) needle nose pliers
- (optional) third hand

How it Works:

Magnetic fields and current go hand-in-hand. When current flows through a wire, a magnetic field is created. This is the basic principle behind electromagnets. This is also the principle used to measure magnetic fields with a magnetometer. The direction of Earth's magnetic fields affects the flow of electrons in the sensor, and those changes in current can be measured and calculated to derive a compass heading or other useful information.

How to Use it:

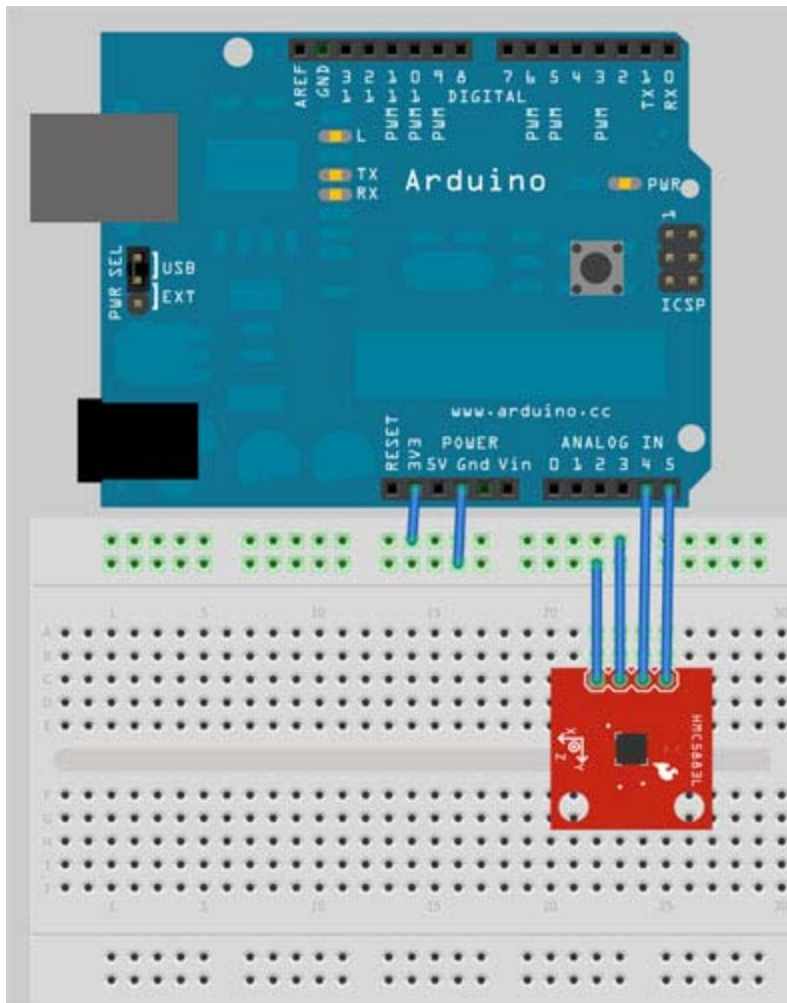
So you're ready to start using this chip with your Arduino are you? Well lucky for you, we have everything you need to get your project rolling.

Hardware:

First we'll need to solder on some headers to the breakout board so it will fit into a breadboard. For soldering suggestions, this tutorial is helpful.

The breakout board includes the HMC5883 sensor and all filtering capacitors necessary. There are two unpopulated pads in case you need pull up resistors (no need for them if you're using the Arduino/ATmega328).

Communication with the HMC5883 is simple and all done through an I²C interface. We will go into this in further detail momentarily. All you need to know for now is how to wire it up. Attach the SDA line to A4 and the SCL line to A5. Also attach Vcc to 3.3V and GND to GND.

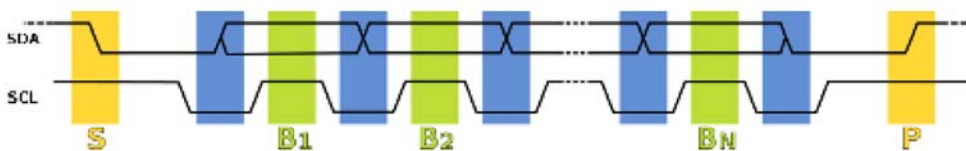


Software:

If you haven't done so already, please download and install the Arduino IDE.

Firmware:

The HMC5883L speaks a kind of funky serial language called I²C. I²C requires just two wires for communication – one for a clock (SCL), and one for data (SDA). Be aware that you probably should avoid using those two pins for anything else (except for maybe other I²C devices, if you want to get tricky). Don't try using them as analog inputs!



The Arduino code example we have, HMC5883.pde, will continuously reads data from the magnetometer for each of the magnetometer's axes using I²C. If you need help uploading the sketch to your Arduino, their site has lots of helpful information.

Let's take a look at the first part of the code:

```
#include <Wire.h> //I2C Arduino Library

#define address 0x1E //0011110b, I2C 7bit address of HMC5883

void setup(){
  //Initialize Serial and I2C communications
  Serial.begin(9600);
  Wire.begin();

  //Put the HMC5883 IC into the correct operating mode
  Wire.beginTransmission(address); //open communication with HMC
5883
  Wire.send(0x02); //select mode register
  Wire.send(0x00); //continuous measurement mode
  Wire.endTransmission();
```

The code that resides within the *setup()* function is ran once at the beginning. It initializes the serial communication at 9600 baud. We use the serial communication to send data for each axis back to the computer. The I2C is also initialized. We then do a 'write' operation to the HMC5883L. The purpose of this 'write' operation is to adjust the value in the configuration register of the HMC5883L to tell it to be in continuous operation mode. This lets us make continuous reads of the axis data. By default the chip is in single read mode meaning after reading from it once, it will go idle to save power. Once idle, we have write to it to turn it on before we can read from it again. The datasheet has all the information about the registers in addition to other useful information. Of course, we always encourage you to RTFM... just look at the datasheet...

Moving on, here's where we actually request and receive the data:

The `loop()` function will run over and over again as long as the board has power. Each time through, we do a quick write to tell it where we want to begin reading the data for the axes. Register 3 is selected and happens to be the most significant byte of the X axis. We request to read 6 bytes one by one. The chip auto-increments the number of which register we are reading from so we don't have to select each register manually before we read from it. The x, y, and z axis data is then sent back serially to the computer. You can see these values by using the 'Serial Monitor' of the Arduino development environment as shown below.

The screenshot shows the Arduino IDE interface. The main window displays the code for the HMC5883 sensor. The `setup()` function initializes the serial port at 9600 baud and the I2C bus. The `loop()` function sends a command to read data from register 3. The Serial Monitor window, titled 'COM4', shows the output of the program, which consists of three integers representing the x, y, and z axis coordinates. The data is updated every few lines, showing values like x: 112, y: -169, z: 383.

```

HMC5883 | Arduino 0022
File Edit Sketch Tools Help
HMC5883
void setup(){
  //Initialize Serial and I2C commu
  Serial.begin(9600);
  Wire.begin();

  //Put the HMC5883 IC into the con
  Wire.beginTransmission(address);
  Wire.send(0x02); //select mode re
  Wire.send(0x00); //continuous nee
  Wire.endTransmission();
}

void loop(){

  int x,y,z; //triple axis data

  //Tell the HMC5883 where to begin reading data
  Wire.beginTransmission(address);
  Wire.send(0x03); //select register 3, X MSB register
}
Done uploading
Binary sketch size: 3784 bytes (of a 32256 byte maximum)
12
COM4
x: 112 y: -169 z: 383
x: 108 y: -145 z: 362
x: 112 y: -149 z: 361
x: 128 y: -154 z: 376
x: 112 y: -138 z: 361
x: 109 y: -159 z: 400
x: 112 y: -157 z: 362
x: 113 y: -153 z: 361
x: 108 y: -165 z: 392
x: 104 y: -174 z: 352
x: 113 y: -138 z: 363
x: 105 y: -158 z: 358
x: 112 y: -148 z: 366
x: 108 y: -150 z: 388
x: 108 y: -147 z: 384
x: 105 y: -168 z: 360
Autoscroll No line ending 9600 baud

```

All you need now is a creative idea of how to use this information in your next project. Always feel free to share it with us, it just might end up on the Sparkfun homepage!

Resources:

- Schematic
- HMC5883L Datasheet
- Arduino Code
- Example C code (ATmega328)
- Digital Compass How-To Video

Conclusion:

Enjoy your new magnetometer! If you have any problems, feel free to contact SparkFun Technical Support at techsupport@sparkfun.com