Phi Robotics

Product Manual

HMC5883L-BB

Version 1.1

Phi Robotics Research Pvt. Ltd.

www.phi-robotics.com

Table of Contents

1 Introduction

HMC5889L-BB is a breakout board of HMC5883L, a 3-axis digital magnetometer designed for low-field magnetic sensing. The HMC5883L includes high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass accuracy. The sensor has a full-scale range of ±8 gauss and a resolution of up to 5 milligauss. Communication with the HMC5883L is simple and done through an I2C interface. It comes in a low-height, LCC surface mount package. An on board 3.3V power regulator is provided to the board hence no external regulator is required.

2 Board Features

- On board regulator of 3.3V
- Low power consumption (0.74mA Measurement Mode, 0.34mA Idle Mode)
- I2C interface
- Built-In strap drive circuits
- 12-Bit ADC Coupled with Low Noise AMR Sensors achieves 5 milligauss resolution in ±8 gauss fields
- 1° to 2° compass accuracy
- Self-test mode
- Dynamic range (1 gauss to 8 gauss gain)
- Wide magnetic field range
- Fast 160Hz maximum output range

3 Specifications

- Supply voltage: 5.0V
- Field range: -8 gauss to +8 gauss
- Digital resolution: 0.73 milligauss to 4.35 milligauss
- Sensitivity: 230 LSb/gauss to 1370 LSb/gauss.

3.1 PCB Details

- PCB size: 24.64 mm x 24.64 mm
- PCB type: FR4
- Solder mask: Black
- Board thickness: 1.6 mm
- Surface finish: Immersion gold

4 Hardware Connections

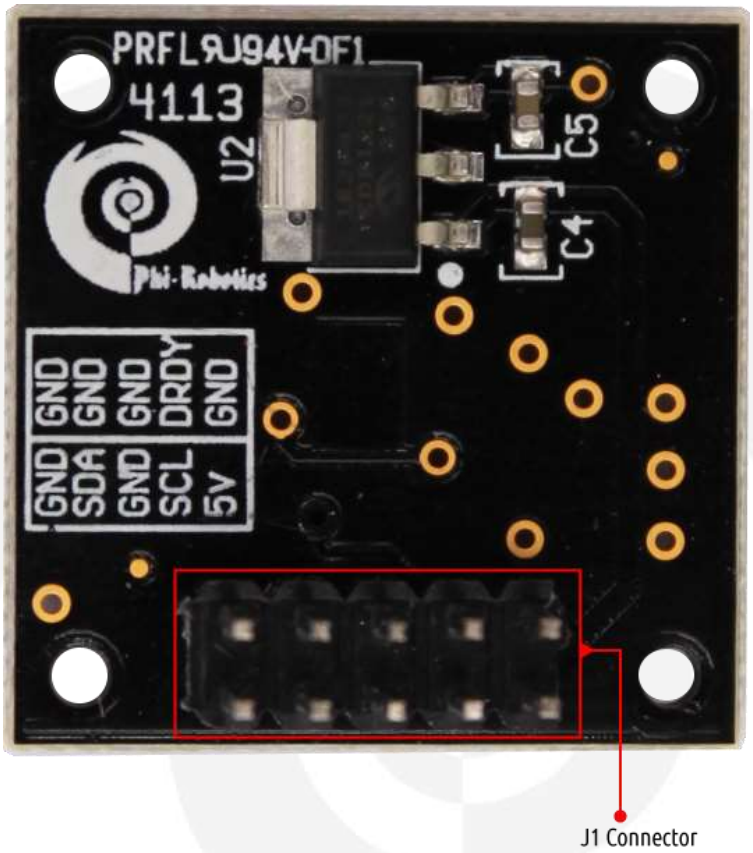

Figure 1 - HMC5883L-BB hardware layout

The HMC5883L-BB board can be directly connected to the microcontroller's I2C pins. SDA and SCL pins from microcontroller can be connected to I2C_SDA and I2C_SCL of the module respectively.

Figure 2 below shows the pin layout for J1 header. The breakout board header has I2C pins for interfacing with microcontroller.

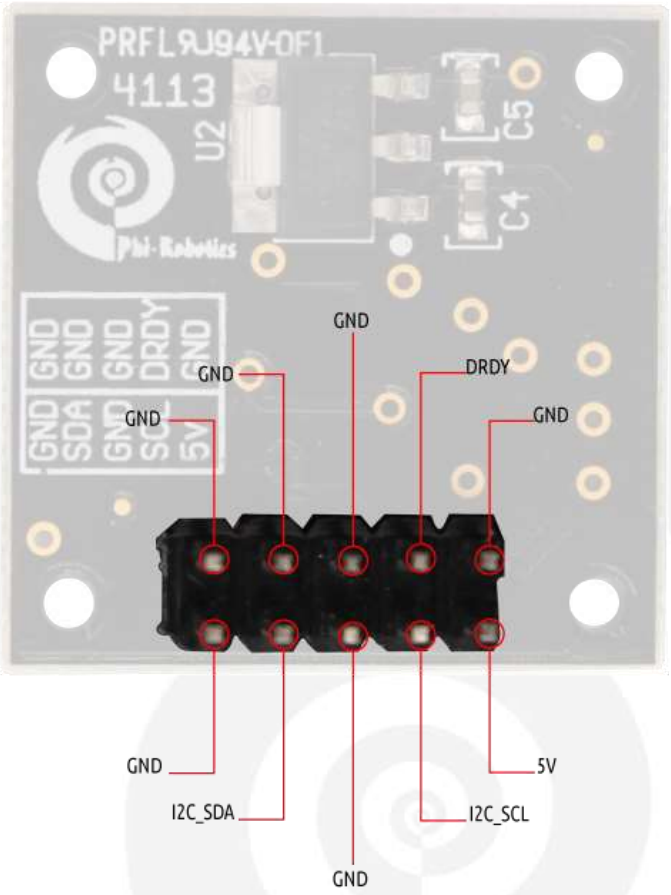

Figure 2 - HMC5883L-BB pin layout

5 Pseudo Code

5.1 Register addresses and configuration parameters

```
// HMC5883 I2C slave address
HMC5883_SLAVE_ADDR = 0x3C

// HMC5883 register addresses
HMC5883_REG_CONFIG_A = 0x00
HMC5883_REG_CONFIG_B = 0x01
HMC5883_REG_MODE = 0x02
HMC5883_REG_DATAX_H = 0x03
HMC5883_REG_DATAX_L = 0x04
HMC5883_REG_DATAZ_H = 0x05
HMC5883_REG_DATAZ_L = 0x06
HMC5883_REG_DATAY_H = 0x07
HMC5883_REG_DATAY_L = 0x08

// HMC5883 configuration
// 3 dimensional axis
X_AXIS = 1
Y_AXIS = 2
Z_AXIS = 3
```

```
// HMC5883 configuration
// update rate = 75 Hz
HMC_UPDATE_RATE = 0x18
// measurement mode = normal (No positive/negative biasing)
HMC_MODE_NORMAL = 0x00
// number of samples averaged per measurement output = 8
HMC_SAMPLE_RATE = 0x60
// gain = 1090 LSb/Gauss
HMC_GAIN_1090 = 0x01
// disable high speed I2C, enable single measurement mode
HMC_MODE = 0x01
// scaling factor for gain = 1090 LSb/Gauss
SCALING_FACTOR = 0.73
// PI value
PI = 3.14159265358979323846
```

5.2 Accessing Registers and Configuring the Device

```
uint8_t hmcReadRegister(uint8_t regAddr)
{
      uint8_t data;
      i2cStart();                   // I2C start signal
      // send HMC5883 I2C slave address with R/W bit set as 0
      i2cWriteByte(HMC5883_SLAVE_ADDR);
      // send register address to read
      i2cWriteByte(regAddr);
      i2cStart();                   // I2C repeated start signal
      // send HMC5883 I2C address with R/W bit set as 1
      i2cWriteByte(HMC5883_SLAVE_ADDR | 0x01);
      data = i2cReadByte();         // read a byte from I2C
      i2cStop();                    // I2C stop signal
      return data;
}

void hmcWriteRegister(uint8_t regAddr, uint8_t data)
{
      i2cStart();                   // I2C start signal
      // send HMC5883 I2C slave address with R/W bit set as 0
      i2cWriteByte(HMC5883_SLAVE_ADDR);
      i2cWriteByte(regAddr);        // write register address
      i2cWriteByte(data);           // write data byte
      i2cStop();                    // I2C stop signal

}

void hmcInit()
{
      uint8_t cfgA_data, cfgB_data;
      cfgA_data = HMC_UPDATE_RATE | HMC_MODE_NORMAL | HMC_SAMPLE_RATE;
      cfgB_data = HMC_GAIN_1090;
      // set update rate, measurement mode and sample rate
      hmcWriteRegister(HMC5883_REG_CONFIG_A, cfgA_data);
      // set device gain
      hmcWriteRegister(HMC5883_REG_CONFIG_B, cfgB_data);
      // set operating mode of the device
      hmcWriteRegister(HMC5883_REG_MODE, HMC_MODE );
}
```

5.3 Reading Axis Components and Calculating Angle

```
uint16_t hmcGetAxisComponent(uint8_t axis)
{
	uint8_t data[2];
	uint16_t value;
	switch(axis)
	{
		X_AXIS:
			data[0] = hmcReadRegister(HMC5883_REG_DATAX_H);
			data[1] = hmcReadRegister(HMC5883_REG_DATAX_L);
			value = (data[0] << 8) | data[1];
			break;

		Y_AXIS:
			data[0] = hmcReadRegister(HMC5883_REG_DATAY_H);
			data[1] = hmcReadRegister(HMC5883_REG_DATAY_L);
			value = (data[0] << 8) | data[1];
			break;

		Z_AXIS:
			data[0] = hmcReadRegister(HMC5883_REG_DATAZ_H);
			data[1] = hmcReadRegister(HMC5883_REG_DATAZ_L);
			value = (data[0] << 8) | data[1];
			break;
	}
	return value;
}

uint32_t hmcGetHeadingAngle(void)
{
	uint16_t comp[3];
	uint32_t angle;

	// read X axis component
	comp[0] = hmcGetAxisComponent(X_AXIS);
	// read Y axis component
	comp[1] = hmcGetAxisComponent(Y_AXIS);
	// read Z axis component
	comp[2] = hmcGetAxisComponent(Z_AXIS);
	// calculate heading angle.Function atan2 defined in math.h
	angle = (((atan2((int16_t)comp[0] ,(int16_t)comp[1])) * 180) / PI)
+ 180;

	return angle;
}

uint32_t hmcGetMagneticFieldStrength(uint8_t axis)
{
	uint16_t axisVal;
	uint32_t fieldStrength;

	// read axis component
	axisVal = hmcGetAxisComponent(axis);
	// multiply raw value by scaling factor
	fieldStrength = axisVal * SCALING_FACTOR;

	return fieldStrength;
}
```

6 Reference

MCP1826S Datasheet: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en531455

HMC5883L Datasheet: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf

Phi Robotics Research Pvt. Ltd.

www.phi-robotics.com