

基于Arduino的GY-271磁场模块角度测量 (HMC5883L)

基于Arduino的GY-271磁场模块角度测量 (HMC5883L)

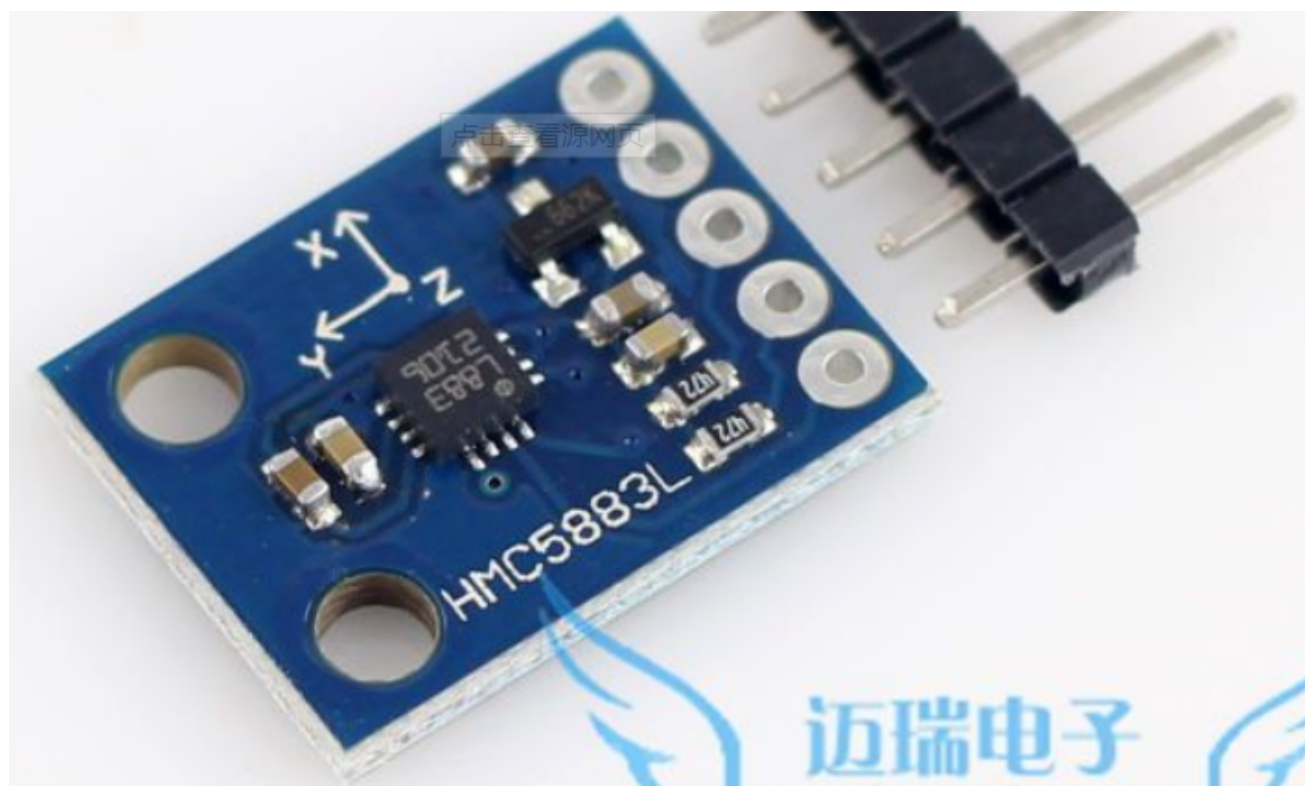
1. 3轴磁场模块
 - 1.1 GY-271
 - 1.2 HMC5883L芯片
2. HMC5883L测量原理
3. 基于Arduino的角度测量
 - 3.1 硬件连接
 - 3.2 软件设计
 - 3.3 测试结果

1. 3轴磁场模块

[GY-271模块介绍](#)

[HMC5883L芯片资料](#)

1.1 GY-271



- Power 3-5V (3.3V/5V , Recommend 3.3V)
- Chipset HMC5883L
- Communication via I2C protocol
- Dimensions 14.8 x 13.5 x 3.5mm

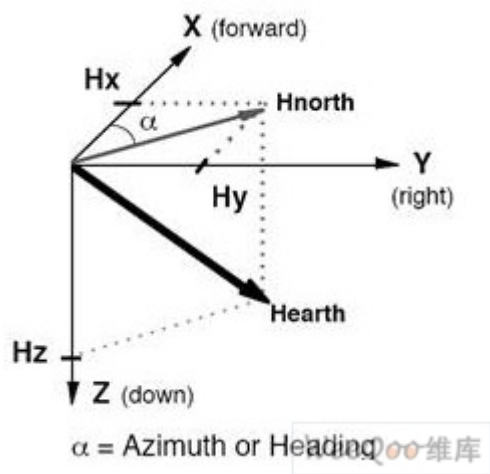
1.2 HMC5883L芯片

HMC5883L作为一个IIC兼容装置，该装置包含一个7-bit串行地址，并支持IIC协议。HMC5883L可以支持标准和快速模式，分别为100kHz和400kHz，但不支持高速模式（Hs）。

VCC	2.16V-3.6V
GND	
SCL	串行时钟——IIC总线主/从时钟
SDA	串行数据——IIC总线主/从数据
DRDY	数据准备，中断引脚，内部被拉高，选项为连接，当数据位于输出寄存器上时会在低电位上停250us

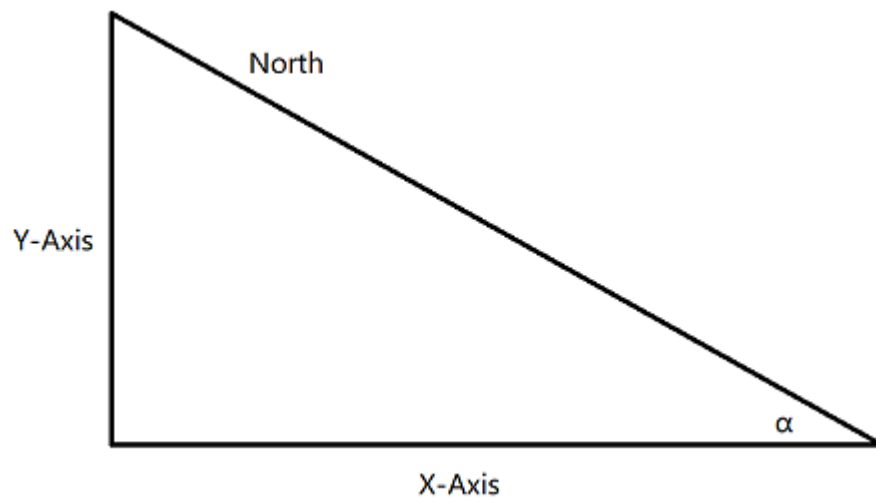
2. HMC5883L测量原理

地磁场是一个矢量，对于一个固定的地点来说，这个矢量可以被分解为两个与当地水平面平行的分量和一个与当地水平面垂直的分量。如果保持电子罗盘和当地的水平面平行，那么罗盘中磁力计的三个轴就和这三个分量对应起来，如图所示。



实际上对水平方向的两个分量来说，他们的矢量和总是指向磁北的。罗盘中的航向角（Azimuth）就是当前方向和磁北的夹角。由于罗盘保持水平，只需要用磁力计水平方向两轴（通常为X轴和Y轴）的检测数据就可以用式计算出航向角。当罗盘水平旋转的时候，航向角在0- 360度之间变化。因此，有以下两种计算方法：

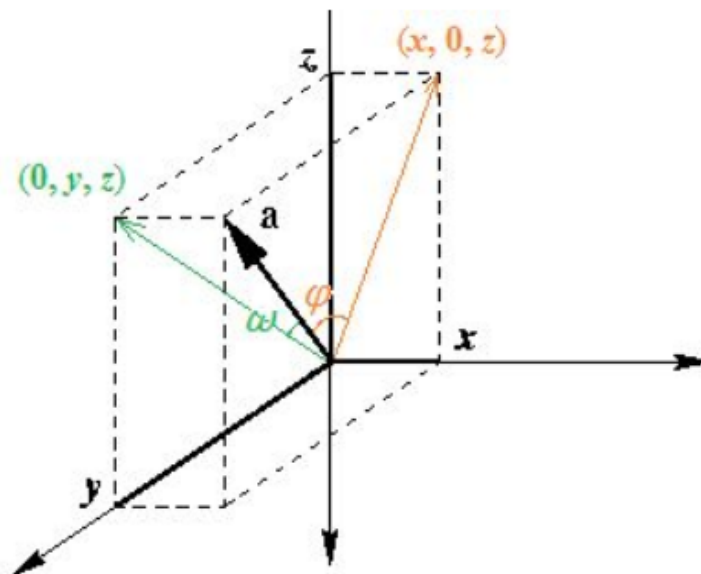
- 当磁场模块与水平面平行时，Z轴同地磁场垂直分量，水平分量被映射到X轴和Y轴上。



所以X轴与地磁水平分量的夹角 $\alpha = \arctan \frac{Y}{X}$, 代码为 `angle=atan2(y,x) * (180 / PI) + 180;` , `atan2()` 函数返回弧度角, 范围为 $\pm\pi$, 化为0-360度。

- 当磁场模块在空间中任意角度放置时, 需要利用X,Y,Z轴数据来修正。

[知乎：三轴螺旋仪空间修正](#)



3. 基于Arduino的角度测量

3.1 硬件连接

- VCC-GND: 3.3V(5V兼容, 推荐3.3V)
- SDA: A4
- SCL: A5

3.2 软件设计

工程文件见文件夹 GY271_Arduino

```

2  * HMC5883L Demo.
3  * dipmicro electronics, 2014
4  *
5  * Hardware Used: GY-271 HMC5883L
6  *
7  * Arduino UNO/compatible/Nano
8  * GY271 module (dipmicro part DE4196
9  * Arduino GND -> GY271/HMC5883L GND
10 * Arduino 3.3V -> GY271/HMC5883L VCC
11 * Arduino A4 (SDA) -> GY271/HMC5883L SDA
12 * Arduino A5 (SCL) -> GY271/HMC5883L SCL
13 */
14
15 #include <Wire.h>           //I2C Arduino Library
16
17 #define HMC5883L_ADDR 0x1E //0011110b, I2C 7bit address of HMC5883
18
19 bool haveHMC5883L = false;
20
21 bool detectHMC5883L ()
22 {
23     // read identification registers
24     wire.beginTransmission(HMC5883L_ADDR); //open communication with HMC5883
25     wire.write(10); //select Identification register A
26     wire.endTransmission();
27     wire.requestFrom(HMC5883L_ADDR, 3);
28     if(3 == wire.available()) {
29         char a = wire.read();
30         char b = wire.read();
31         char c = wire.read();
32         if(a == 'H' && b == '4' && c == '3')
33             return true;
34     }
35
36     return false;
37 }
38
39 void setup()
40 {
41     //Initialize Serial and I2C communications
42     Serial.begin(9600);
43     Serial.println("GY271 TEST");
44     wire.begin();
45     // lower I2C clock http://www.gammon.com.au/forum/?id=10896
46     TWBR = 78; // 25 kHz
47     TWSR |= _BV (TWPS0); // change prescaler
48 }
49
50 void loop()
51 {
52     bool detect = detectHMC5883L();
53
54     if(!haveHMC5883L)

```

```

55 {
56     if(detect)
57     {
58         haveHMC5883L = true;
59         Serial.println("We have HMC5883L, moving on");
60         // Put the HMC5883 IC into the correct operating mode
61         wire.beginTransaction(HMC5883L_ADDR); //open communication with HMC5883
62         wire.write(0x02); //select mode register
63         wire.write(0x00); //continuous measurement mode
64         wire.endTransmission();
65     }
66     else
67     {
68         Serial.println("No HMC5883L detected!");
69         delay(2000);
70         return;
71     }
72 }
73 else
74 {
75     if(!detect) {
76         haveHMC5883L = false;
77         Serial.println("Lost connection to HMC5883L!");
78         delay(2000);
79         return;
80     }
81 }
82
83 int x,y,z; //triple axis data
84
85 //Tell the HMC5883 where to begin reading data
86 wire.beginTransaction(HMC5883L_ADDR);
87 wire.write(0x03); //select register 3, X MSB register
88 wire.endTransmission();
89
90 //Read data from each axis, 2 registers per axis
91 wire.requestFrom(HMC5883L_ADDR, 6);
92 if(6<=wire.available()){
93     x = wire.read()<<8; //X msb
94     x |= wire.read(); //X lsb
95     z = wire.read()<<8; //Z msb
96     z |= wire.read(); //Z lsb
97     y = wire.read()<<8; //Y msb
98     y |= wire.read(); //Y lsb
99 }
100
101 //Print out values of each axis
102 Serial.print("x: ");
103 Serial.print(x);
104 Serial.print(" y: ");
105 Serial.print(y);
106 Serial.print(" z: ");
107 Serial.print(z);

```

```

108 float angle=atan2(y,x) * (180 / PI) + 180;
109 Serial.print("    Current Angle: ");
110 Serial.println(angle);
111
112 delay(250);
113 }
114

```

3.3 测试结果

波特率设置为9600，打开串口获取当前3轴磁场强度和方位角。

ATX XCOM V2.0

x	y	z	Current Angle
-286	-563	437	63.07
-60	-685	494	84.99
-94	-685	464	82.19
-196	-613	440	72.27
-130	-717	472	79.72
-362	-658	382	61.18
-530	-364	212	34.48
-494	-151	221	17.00
-253	29	457	353.46
-125	64	493	332.89
-124	63	483	333.07
-86	77	484	318.16
-78	80	481	314.27
-91	75	476	320.51
-47	64	489	306.29
9	36	476	255.96
48	35	468	216.10
113	18	448	189.05
49	-40	503	140.77
-58	-135	634	66.75
-358	-233	554	33.06
-352	-221	562	32.12
-299	-200	588	33.78
-240	-178	611	36.56
-172	-169	646	44.50

串口选择: COM4: USB-SERIAL

波特率: 9600

停止位: 1

数据位: 8

奇偶校验: 无

串口操作: ☒ 打开串口

保存窗口 清除接收

☐ 16进制显示 ☒ 白底黑字

☐ RTS ☐ DTR

☐ 时间戳(以换行回车断帧)

单条发送 多条发送 协议传输 帮助