

# PZ-HMC5883L 电子罗盘模块开发手册

本手册将教大家如何在普中 Z100&Z200&Z300&Z500 开发板上使用 PZ-HMC5883L 电子罗盘模块。本章分为如下几部分内容：

- 1 PZ-HMC5883L 模块简介
- 2 硬件设计
- 3 软件设计
- 4 实验现象

# 1 PZ-HMC5883L 模块简介

## 1.1 特性参数

PZ-HMC5883L 是深圳普中科技推出的一款高性能 3 轴数字罗盘模块。该模块采用霍尼韦尔的 HMC5883L 芯片作为核心，是一种表面贴装的高集成模块，并带有数字接口的弱磁传感器芯片，应用于低成本罗盘和磁场检测领域。HMC5883L 包括最先进的高分辨率 HMC118X 系列磁阻传感器，并附带霍尼韦尔专利的集成电路包括放大器、自动消磁驱动器、偏差校准、能使罗盘精度控制在  $1^{\circ} \sim 2^{\circ}$  的 16 位模数转换器、简易的 I2C 系列总线接口。HMC5883L 是采用无铅表面封装技术，带有 16 引脚，尺寸为 3.0X3.0X0.9mm。HMC5883L 的所应用领域有手机、笔记本电脑、消费类电子、汽车导航系统和个人导航系统。

HMC5883L 采用霍尼韦尔各向异性磁阻 (AMR) 技术，该技术的优点是其他磁传感器技术所无法企及。这些各向异性传感器具有在轴向高灵敏度和线性高精度的特点。传感器带有的对于正交轴低敏感度的固相结构能用于测量地球磁场的方向和大小，其测量范围从毫高斯到 8 高斯 (gauss)。霍尼韦尔的磁传感器在低磁场传感器行业中是灵敏度最高和可靠性最好的传感器。

PZ-HMC5883L 模块具有：体积小、测量精度高、多测量工作模式、支持 IIC 从机地址设置和中断、兼容 3.3V/5V 系统、使用方便等特点。

PZ-HMC5883L 模块各项参数如下所示：

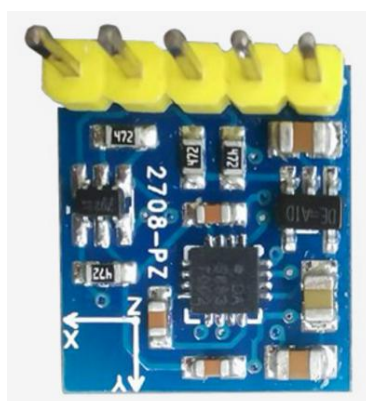
特性	条件*	最小	标准	最大	单位
磁场范围	满量程 (FS) - 全部施加磁场 (典型)	-8		+8	高斯 (gauss)
磁动态范围	3-bit 增益控制	±1		±8	高斯
线性	±2.0 高斯输入范围			0.1	±%满量程
解析度	VDD=3.0V, GN=2		5		毫高斯
启动时间	I <sup>2</sup> C 控制准备时间		200		μs
磁滞	±2.0 高斯输入范围		±25		ppm
纵向灵敏度	测试条件: 纵向区域=0.5gauss, Happlied=±3gauss		±2%		%FS/Gauss
输出速率	持续测量模式 单一测量模式	0.75		75 160	Hz Hz
开关时间	I <sup>2</sup> C 指令等待		200		us
测量周期	从接收指令到数据准备		6		ms
增益公差	所有增益/动态范围设置		±5		%
I <sup>2</sup> C 地址	7-bit 地址 8-bit 读取地址 8-bit 写入地址		0x1E 0x3D 0x3C		hex (十六进制) hex (十六进制) hex (十六进制)
I <sup>2</sup> C 率	由 I <sup>2</sup> C 主机控制			400	kHz
I <sup>2</sup> C 滞后性	施密特触发器脉冲输入在 SCL 和 SDA 上的滞后性- 下降 (VDDIO=1.8V) 上升 (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		V V
自测试	X&Y 轴 Z 轴		±1.16 ±1.08		高斯
	X&Y 轴 (GN=100) Z 轴 (GN=100)		510		LSb

## 1.2 使用说明

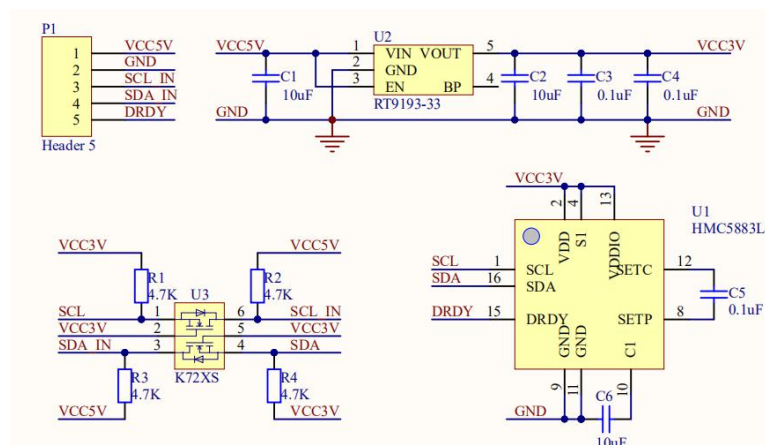
### 1.2.1 模块引脚说明

PZ-HMC5883L 模块通过 1\*5 的排针 (2.54mm 间距) 同外部连接, 模块可以与 Z100、Z200、Z300、Z500 开发板通过杜邦线连接进行测试。所有普中 STM32F1 系列开发板都可使用该例程, 用户可以直接在这些开发板上, 对模块进行测试。

PZ-HMC5883L 模块外观如下图所示:



PZ-HMC5883L 模块原理图如下图所示:



从图中可以看出，模块自带了 3.3V 超低压差稳压芯片，给 HMC5883L&QMC5883L 芯片供电，因此外部供电可以选择：3.3V/5V 都可以的。模块通过 P1 排针与外部连接，引出了 VCC、GND、SDA\_IN、SCL\_IN、DRDY 信号。其中，SCL、SDA 带了 4.7K 上拉电阻，外部可以不用再加上拉电阻了。

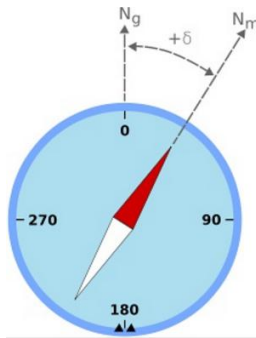
PZ-HMC5883L 模块通过一个 1\*5 的排针（P1）同外部电路连接，各引脚的详细描述如下所示：（PZ-HMC5883L 模块-->STM32 IO）

VCC-->5V  
GND-->GND  
SCL-->PB6  
SDA-->PB7

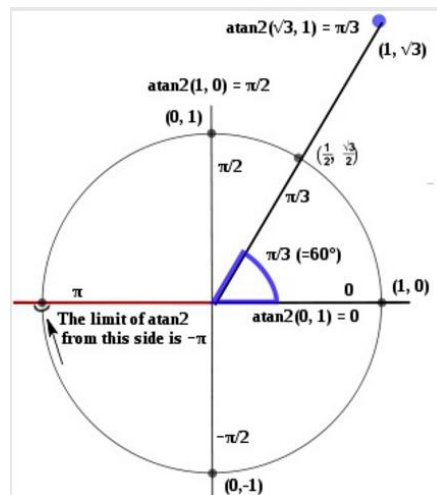
模块通过 IIC 接口与外部进行通信，上电时，默认的 IIC 从机地址为：0X1A，关于模块详细资料，可以参考“\3--芯片数据手册\HMC5883L 中文规格书.pdf”，因为 PZ-HMC5883L 模块使用的是高精度的 QMC5883L 芯片，而 QMC5883L 和 HMC5883L 会有一点细微差别，主要表现在寄存器地址不同，所以用户可以参考“\4--参考资料\QMC5883L 寄存器对比及参考设置.pdf”。

## 1.2.2 磁偏角和方位公式简介

磁偏角，即现实中指北针指向的磁极（地磁极）和我们地图上标注的传统意义的磁极是不重合的，有一个夹角。你看地球仪上的地球也不是歪脖子转动么，就是这个意思。如下图所示：



下面是几何时间，如何定义方位。这里采用了反正切角的表达方式。



$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

上述中 atan2 为反正切角。 $\pi$  是 180 度， $2\pi$  是 360 度， $1/2\pi$  是 90 度... 其实这个反正切角就是射线  $(x, y)$  和  $x$  轴正向间的夹角。如果  $y < 0$ ，射线在  $x$  轴下方，那么角度从  $x$  轴正向顺时针旋转，而且是负值；如果  $y > 0$ ，角度从  $x$  轴正向逆时针旋转。关于这些公式，大家不理解也没关系，最终我们直接使用下面的公式即可。

我们来分析 honeywell 的 HMC5883 三轴数位罗盘对角度的计算公式。这个公式才是我们最终使用的。

$$\text{angle} = \text{atan2}(y, x) * (180 / 3.14159265) + 180;$$

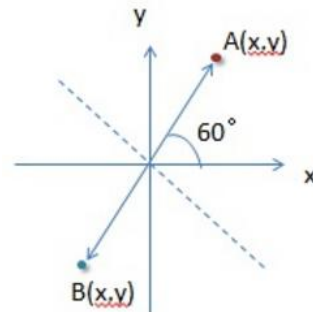
其实就是这样的：

角度 =  $\text{atan2}(y, x) * (180 / \pi) + 180$ ，这里角度用  $0 \sim 360^\circ$  表示

因为  $\text{atan2}(y, x)$  算出来的会是多少  $\pi$ ，所以后面用了个  $180/\pi$  来换算成度

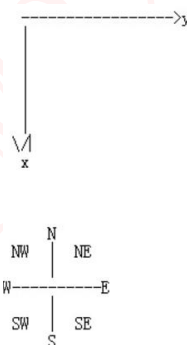
数( $0^{\circ}\sim 360^{\circ}$ )。但是，但是后面那个  $+180$  是干嘛？

如果是  $60^{\circ}$  的角， $+180$  就变成了  $240^{\circ}$ ，从 A 点变成了 B 点变成了下面这个图。



因为这种反正切角的范围是从  $-\pi$  到  $\pi$  ( $-180^{\circ}\sim 180^{\circ}$ )，而我们通常读出的指南针角度是  $0^{\circ}\sim 360^{\circ}$ ，所以需要把这个范围进行平移  $180^{\circ}$ ，到指南针的角度。但是这样一来，原来  $60^{\circ}$  的角度变成了  $240^{\circ}$ ，也就是，整个象限沿着虚线被“镜像”了一下。

下面我们介绍下如何根据磁偏角换算为方位，先看下图：X 轴指向北方位。



根据获取的 XY 周磁偏角，我们可以根据下列关系得到方位。

```
1.  if((angle < 22.5) || (angle > 337.5 ))
2.      printf("南(South)\r\n");
3.  if((angle > 22.5) && (angle < 67.5 ))
4.      printf("西南(SouthWest)\r\n");
5.  if((angle > 67.5) && (angle < 112.5 ))
6.      printf("西(West)\r\n");
7.  if((angle > 112.5) && (angle < 157.5 ))
8.      printf("西北(NorthWest)\r\n");
9.  if((angle > 157.5) && (angle < 202.5 ))
10.     printf("北(North)\r\n");
11. if((angle > 202.5) && (angle < 247.5 ))
12.     printf("东北(NorthEast)\r\n");
13. if((angle > 247.5) && (angle < 292.5 ))
```

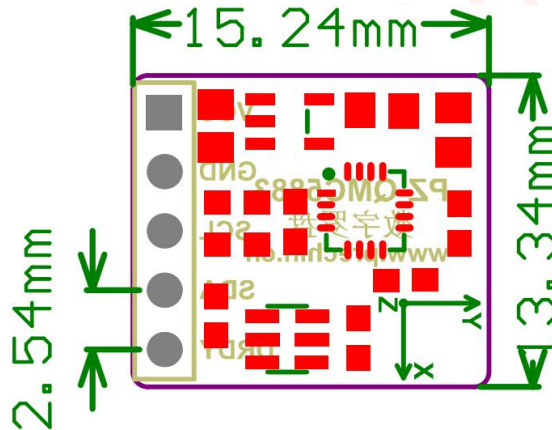
```
14.     printf("东(East)\r\n");
15.     if((angle > 292.5) && (angle < 337.5 ))
16.         printf("东南(SouthEast)\r\n");
```

假设我们是  $60^\circ$  角，也就是指北针正北顺时针转  $60$  度位置，应该在 NE（东北）位置，套用上面的公式， $60 + 180 = 240^\circ$ ，然后套用上面的条件判断： $\text{angle} > 202.5^\circ$  并且  $\text{angle} < 247.5^\circ$ ，处于 NorthEast（东北）位置，正确。

此处是作者对于磁偏角与方位关系的理解，可能不一定百分百准确，如有错误还请联系我们修正。

## 1.3 结构尺寸

PZ-HMC5883L 模块尺寸结构如下图所示：



## 2 硬件设计

### 2.1 硬件准备

本实验所需要的硬件资源如下：

- ① Z100&Z200&Z300&Z500 开发板 1 个
- ② PZ-HMC5883L 模块 1 个
- ③ USB 线一条（用于供电和模块与电脑串口调试助手通信）

## 2.2 模块与开发板连接

PZ-HMC5883L 模块可直接使用杜邦线与 Z100&Z200&Z300&Z500 开发板进行连接，该接口与 STM32 IO 连接关系如下所示：（PZ-HMC5883L 模块-->STM32 IO）

```
VCC-->5V
GND-->GND
SCL-->PB6
SDA-->PB7
```

## 3 软件设计

本实验在 Z100&Z200&Z300&Z500 开发板的 IIC-EEPROM 实验基础上进行修改，在 APP 文件夹内新建了 qmc5883 文件夹，qmc5883 文件夹内新建 qmc5883.c、qmc5883.h 文件。在工程目录 APP 中添加 qmc5883.c 源文件，最后添加对应头文件路径。

本例程是基于 IIC 通信，所以 qmc5883.c 驱动程序与 EEPROM 的代码有很大相似处。下面我们只介绍部分代码，详细代码可参考源码文件。

### 3.1 qmc5883.c

qmc5883.c，该文件为 QMC5883 的驱动，是基于 IIC 总线的再次封装。它调用了 iic.c 文件的 iic 读写数据函数，实现了对 QMC5883 读写操作，同时文件也实现对罗盘角度 XY 的获取功能，具体代码如下：

```
1.  #include "qmc5883.h"
2.  #include "SysTick.h"
3.  #include "math.h"
4.
5.  //QMC5883L 初始化
6.  void QMC5883L_Init(void)
7.  {
8.      IIC_Init();//IIC 初始化
9.      QMC5883L_WriteOneByte(QMC5883L_Addr,QMC58X3_R_CONF1,0x0D); //0x09 = 0x0d/0x1d; OSR=512,
      RNG=+/-2G or +/-8G, ODR=200Hz, MODE= continuous
```



```

10.         QMC5883L_WriteOneByte(QMC5883L_Addr,QMC58X3_R_SETCLEAR_PERIOD,0x01);// 设置清除时间寄存
    器
11.         QMC5883L_WriteOneByte(QMC5883L_Addr,0x20,0x40);
12.         QMC5883L_WriteOneByte(QMC5883L_Addr,0x21,0x01);
13.         delay_ms(300);
14.     }
15.
16.     //在 QMC5883L 指定地址写入一个寄存器数据
17.     //WriteAddr: 地址
18.     //Reg: 寄存器
19.     //RegData: 寄存器值
20.     void QMC5883L_WriteOneByte(u8 WriteAddr,u8 Reg,u8 RegData)
21.     {
22.         IIC_Start();
23.         IIC_Send_Byte(WriteAddr);
24.         IIC_Wait_Ack();
25.         IIC_Send_Byte(Reg);
26.         IIC_Wait_Ack();
27.         IIC_Send_Byte(RegData);
28.         IIC_Stop();//产生一个停止条件
29.         delay_ms(10);
30.     }
31.
32.     //在 QMC5883L 指定地址写入多个寄存器数据
33.     //WriteAddr: 地址
34.     //Reg: 寄存器
35.     //NumToWrite: 写入数据的长度
36.     //pBuffer: 数据数组首地址
37.     void QMC5883L_Write(u8 WriteAddr,u8 Reg,u8 NumToWrite,u8 *pBuffer)
38.     {
39.         while(NumToWrite--)
40.         {
41.             QMC5883L_WriteOneByte(WriteAddr,Reg++,*pBuffer);
42.             pBuffer++;
43.         }
44.     }
45.
46.     //在 QMC5883L 指定地址读一个寄存器数据
47.     //WriteAddr: 地址
48.     //Reg: 寄存器
49.     //返回值: 读到的数据
50.     u8 QMC5883L_ReadOneByte(u8 WriteAddr,u8 Reg)
51.     {

```

```

52.     u8 temp=0;

53.     IIC_Start();
54.     IIC_Send_Byte(WriteAddr);
55.     IIC_Wait_Ack();
56.     IIC_Send_Byte(Reg);
57.     IIC_Wait_Ack();
58.     IIC_Start();
59.     IIC_Send_Byte(WriteAddr|0x01);           //进入接收模式
60.     IIC_Wait_Ack();
61.     temp=IIC_Read_Byte(0);
62.     IIC_Stop();//产生一个停止条件
63.     return temp;
64. }

65.
66. //在 QMC5883L 指定地址读多个寄存器数据
67. //WriteAddr: 地址
68. //Reg: 寄存器
69. //NumToRead: 读取数据的长度
70. //pBuffer: 数据数组首地址
71. void QMC5883L_Read(u8 WriteAddr,u8 Reg,u8 NumToRead,u8 *pBuffer)
72. {
73.     while(NumToRead)
74.     {
75.         *pBuffer++=QMC5883L_ReadOneByte(WriteAddr,Reg++);
76.         NumToRead--;
77.     }
78. }
79.
80. //读取 QMC5883L 角度数据
81. //返回值: XY 角度
82. float QMC5883L_Read_Angle(void)
83. {
84.     int x,y,z;
85.     float angle;
86.     u8 BUF[6],flag[1];    //接收数据缓存区
87.
88.     QMC5883L_Read(QMC5883L_Addr,QMC58X3_R_STATUS,1,flag); //查看 QMC5883L 数据存放状态
89.     if(flag[0]&0x01)                                           //判断 QMC5883L 数据是否存放完
成
90.     {
91.         QMC5883L_Read(QMC5883L_Addr,QMC58X3_R_X_LSB,6,BUF); //读取 QMC5883L 的 X,Y,Z 轴数据
(OUT_X_L_A>OUT_X_H_A>OUT_Z_L_A>OUT_Z_H_A>OUT_Y_L_A>OUT_Y_H_A)

```

```

92.         x=(BUF[1] << 8) | BUF[0];                                //合并 X 轴寄存器输出数据的高
           位与低位
93.         z=(BUF[5] << 8) | BUF[4];                                //合并 Z 轴寄存器输出数据的高
           位与低位
94.         y=(BUF[3] << 8) | BUF[2];                                //合并 Y 轴寄存器输出数据的高
           位与低位
95.         if(x>0x7fff)x-=0xffff;                                  //X 轴数据限幅
96.         if(z>0x7fff)z-=0xffff;                                  //Z 轴数据限幅
97.         if(y>0x7fff)y-=0xffff;                                  //Y 轴数据限幅
98.         angle= atan2(y,x) * (180 / 3.14159265) + 180;          //换算成角度
99.     }
100.    return angle;
101. }

```

以上就是 qmc5883.c 文件的全部代码。都是 QMC5883 底层操作的函数，比较简单，与开发板的 IIC-EEPROM 实验类似。这里就不多说。

QMC5883L\_Read\_Angle 函数为获取 XY 角度，即磁偏角，根据磁偏角即可获得方位，这个在前面已经介绍。

## 3.2 main.c

我们再来看看 main.c，该文件里面就 2 个 函数，Get\_Position 函数和 main 函数，代码如下：

```

1.  #include "system.h"
2.  #include "SysTick.h"
3.  #include "led.h"
4.  #include "usart.h"
5.  #include "key.h"
6.  #include "qmc5883.h"
7.
8.
9.  //获取罗盘方位
10. void Get_Position(float angle)
11. {
12.     if((angle < 22.5) || (angle > 337.5 ))
13.         printf("南(South)\r\n");
14.     if((angle > 22.5) && (angle < 67.5 ))
15.         printf("西南(SouthWest)\r\n");
16.     if((angle > 67.5) && (angle < 112.5 ))
17.         printf("西(West)\r\n");
18.     if((angle > 112.5) && (angle < 157.5 ))
19.         printf("西北(NorthWest)\r\n");

```

```

20.     if((angle > 157.5) && (angle < 202.5 ))
21.         printf("北(North)\r\n");
22.     if((angle > 202.5) && (angle < 247.5 ))
23.         printf("东北(NorthEast)\r\n");
24.     if((angle > 247.5) && (angle < 292.5 ))
25.         printf("东(East)\r\n");
26.     if((angle > 292.5) && (angle < 337.5 ))
27.         printf("东南(SouthEast)\r\n");
28. }
29.
30. int main()
31. {
32.     u8 i=0;
33.     float xy_angle;
34.
35.     SysTick_Init(72);
36.     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); //中断优先级分组 分2组
37.     LED_Init();
38.     USART1_Init(115200);
39.     KEY_Init();
40.     QMC5883L_Init();
41.
42.
43.     while(1)
44.     {
45.         xy_angle=QMC5883L_Read_Angle();
46.         i++;
47.         if(i%20==0)
48.         {
49.             LED1=!LED1;
50.         }
51.         if(i%50==0)
52.         {
53.             printf("xy_angle=%0.1f\r\n",xy_angle);
54.             Get_Position(xy_angle);
55.         }
56.         delay_ms(10);
57.     }
58. }

```

代码非常简单，初始化相应外设，比如 LED、串口 1、QMC5883L 模块等。然后读取角度，根据角度得到方位，最后间隔一段时间由串口 1 输出。

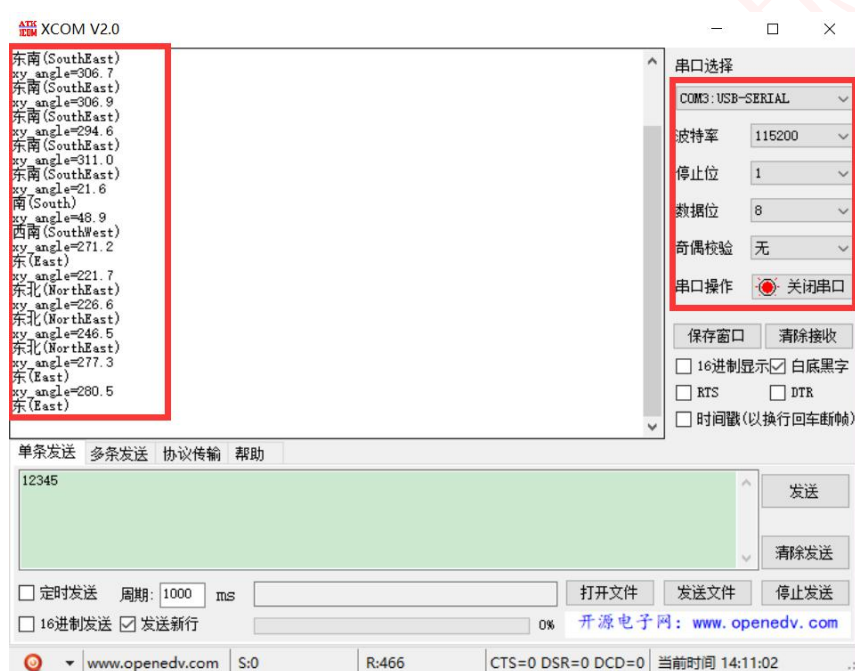
## 4 实验现象

首先，请先确保硬件都已经连接好：

①PZ-HMC5883L 模块与 Z100&Z200&Z300&Z500 开发板连接（可参考硬件设计小节）

②使用 USB 线给开发板供电。

代码编译成功之后，我们将代码下载到 STM32 开发板上。打开串口调试助手，输出显示如下图所示：



可以试试，用手机自带指南针与检测结果对比下。

**注意：当转动模块，如果输出结果始终不变，请将开发板复位或检查线路是否正确。**

## 5 其他

（1）购买地址（普中授权店铺）

<http://www.prechin.net/forum.php?mod=viewthread&tid=38746&extra=>

（2）资料下载

<http://prechin.net/forum.php?mod=viewthread&tid=35264&extra=page%3D1>

(3) 技术支持

普中官网: [www.prechin.cn](http://www.prechin.cn)

普中论坛: [www.prechin.net](http://www.prechin.net)

技术电话: 0755-36564227 (转技术)

普中STM32开发板