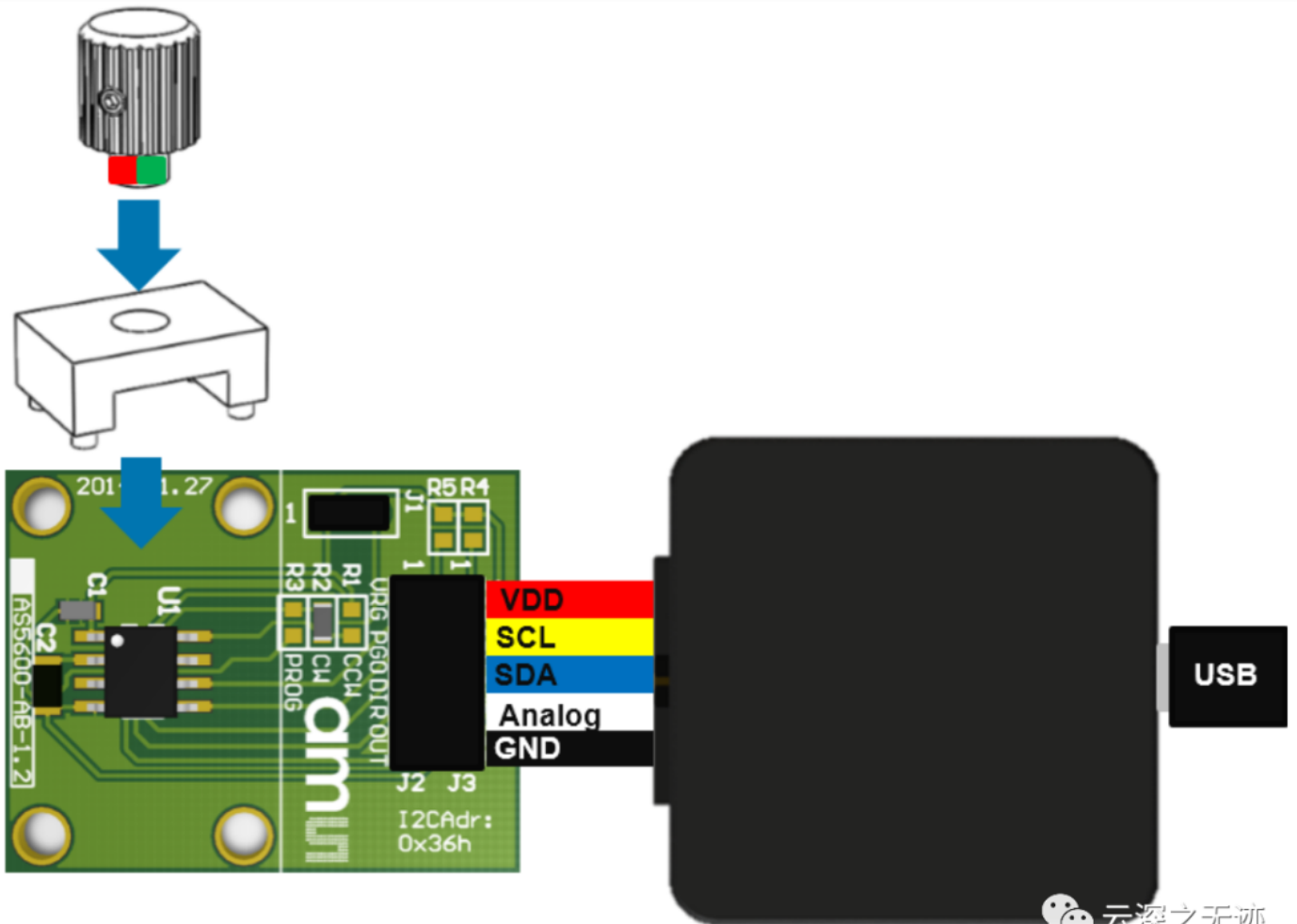




云深之无迹

我喜欢这个，官网的东西



云深之无迹







这个也是一个接线图


1 | <https://ams.com/zh/as5600#tab/documents>

官网在此

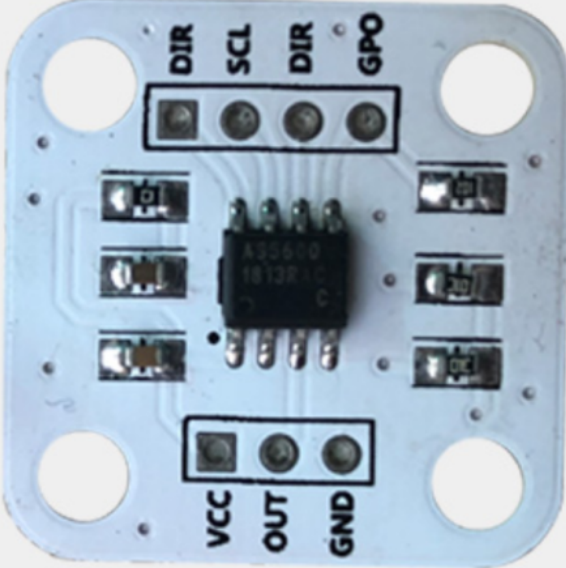
1 | <https://pan.baidu.com/share/init?sur1=Mh203kjsxmiBXPXie9F6-w>

这是找到的一些程序代码

	AMS_5600_example.zip	2021/11/7 12:00	ZIP 压缩文件
	as5600.zip	2021/11/7 12:00	ZIP 压缩文件
	AS5600_Datasheet_EN_v2 (1).pdf	2021/11/7 12:00	WPS PDF 文档
	AS5600-POTUINO_Operation-Manual_R...	2021/11/7 12:00	WPS PDF 文档
	模块3D模型.step	2021/11/7 12:00	STEP 文件
	原理图.JPG	2021/11/7 12:00	JPG 文件

云深之无迹

这么多

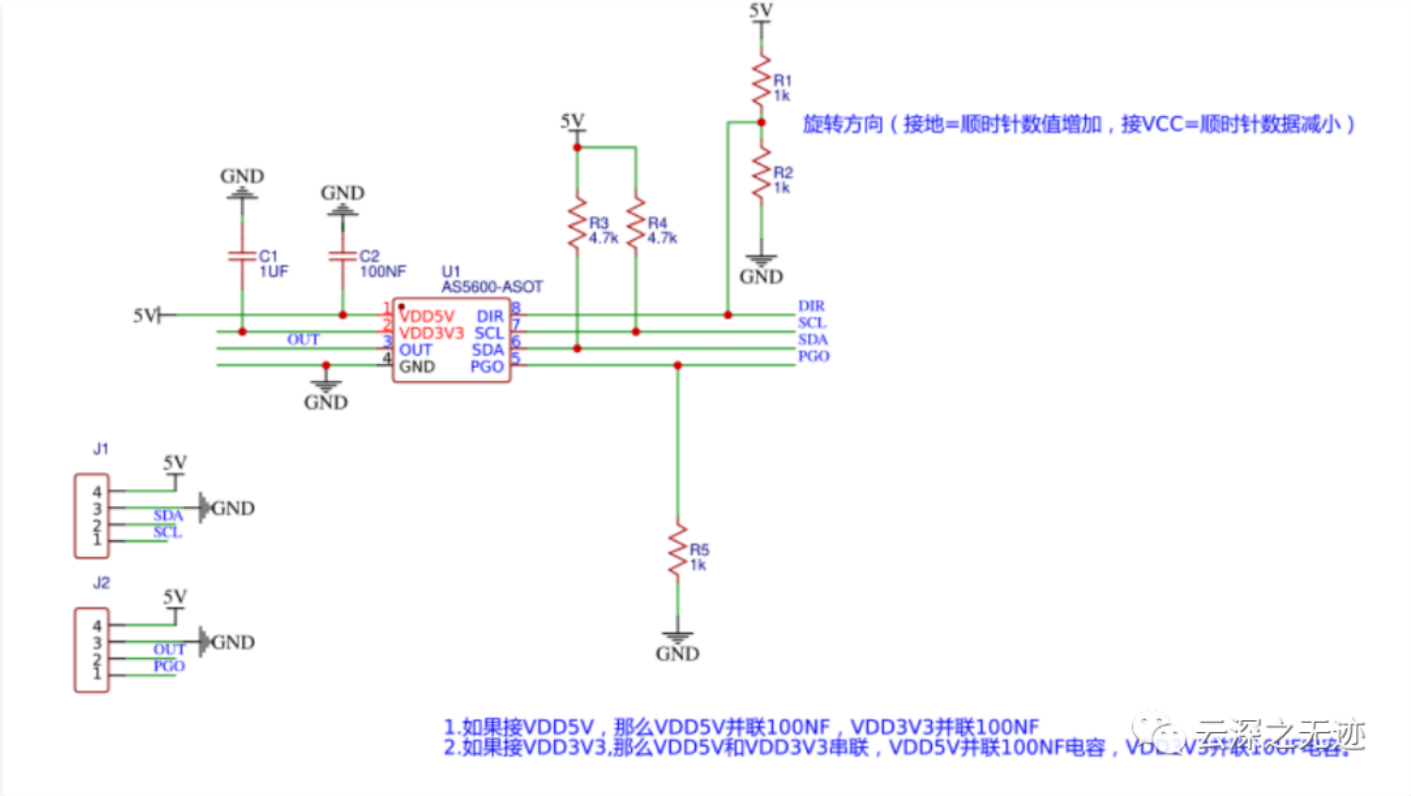


【产品名称】AS5600高精度
12bit角度传感器模块

【电气参数】

VCC	3.3V
GND	电源地
Out	PWM/模拟电压输出
DIR	旋转方向（接地=顺时针数值增加；接VCC=顺时针数值减小）
SCL	IIC 通信时钟线
SDA	IIC 通信数据线
GPO	模式选择（内部上拉，接VCC=模式B；接地=模式A）

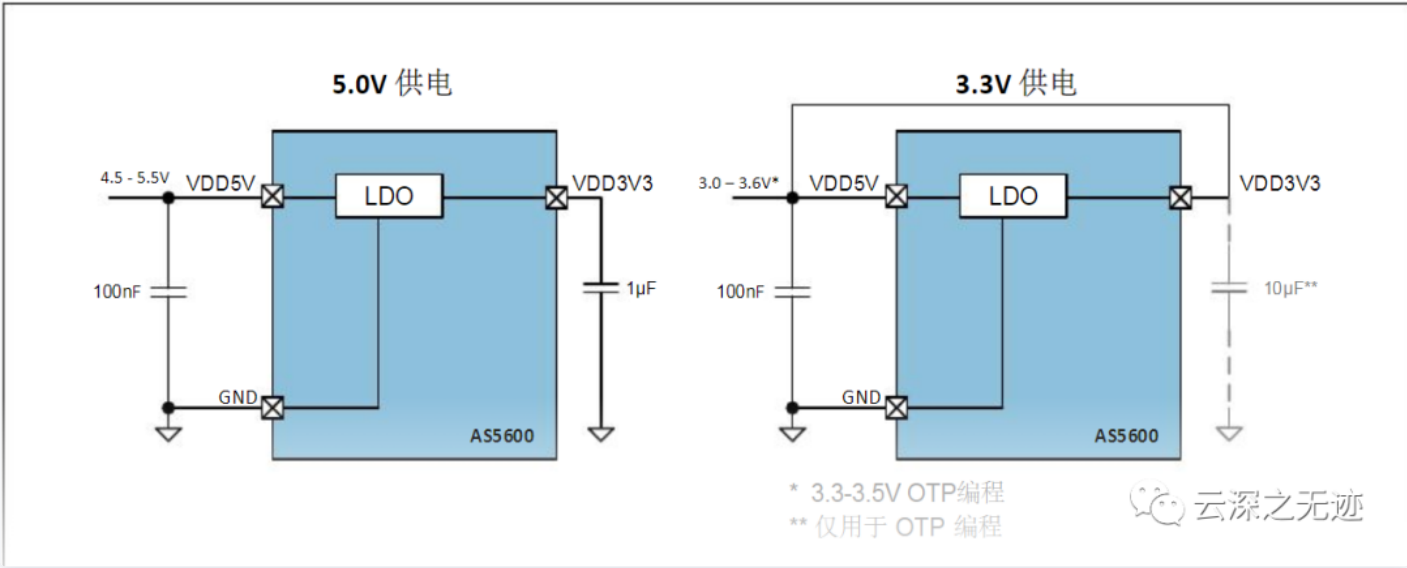
接线图



引脚号	名称	类型	描述
1	VDD5V	Supply	+5V 电压供应模式(需要100 nF的退耦电容)
2	VDD3V3	Supply	+3.3V 电压供应模式(在+5V模式的基础上额外需要一个1-μF退耦电容)
3	OUT	模拟/数字输出	模拟/PWM 输出
4	GND	Supply	接地端
5	PGO	数字输入	编程选项 (内部上拉, 连接至GND = 编程选项B)
6	SDA	数字输入/输出	I ² C 数据端 (consider external pull-up)
7	SCL	数字输入	I ² C 时钟端 (consider external pull-up)
8	DIR	数字输入	方向极性 (GND = 数值顺时针方向增加, VDD = 数值逆时针方向增加)

芯片的引脚

5.0V 和 3.3V 电源供应选择



供电情况

📁 AS5600资料	2021/11/7 12:00
📄 AnglePositionOnAxis_AN000271_2-00.pdf	2021/11/7 12:27
📄 AS5047D_47P_55A_AS5600_FS000181_1-00.pdf	2021/11/7 11:48
📄 AS5047P_AS5311_AS5510_AS5600_FS000182_1-00.pdf	2021/11/7 11:48
📄 AS5600_AB-POT_UG000378_1-00.pdf	2021/11/7 12:21
🖼️ AS5600_BD000233_1-00.png	2021/11/7 12:28
📄 AS5600_Datasheet_EN_v3.pdf	2021/11/7 11:48
📄 AS5600_DS000365_5-00.pdf	2021/11/7 12:19
📄 AS5600_FS000133_2-00.pdf	2021/11/7 11:48
📄 AS5600_FS000146_CHIN_1-00.pdf	2021/11/7 11:48
🖼️ AS5600_IM000266_1-00.png	2021/11/7 12:21
📄 AS5600_UG000240_1-00.pdf	2021/11/7 12:29
📄 MDF_191240003_AS5600-ASOT_MD000326_1-00.pdf	2021/11/7 12:28
📄 Position Sensors_UG000359_1-00.pdf	2021/11/7 12:28
📄 PositionSensors_AN000304_1-00.pdf	2021/11/7 11:48
📄 USB_I-P_Box_AN001012_1-08.pdf	2021/11/7 12:27

云深之无迹

所有资料

AS5600模块	STM32F0系统板
VCC	3.3V
GND	GND
DIR	GND
SCL	PA9
SDA	PA10
GPO	3.3V

云深之无迹

AS5600支持Fast-mode Plus模式，SCL能达到1MHz。

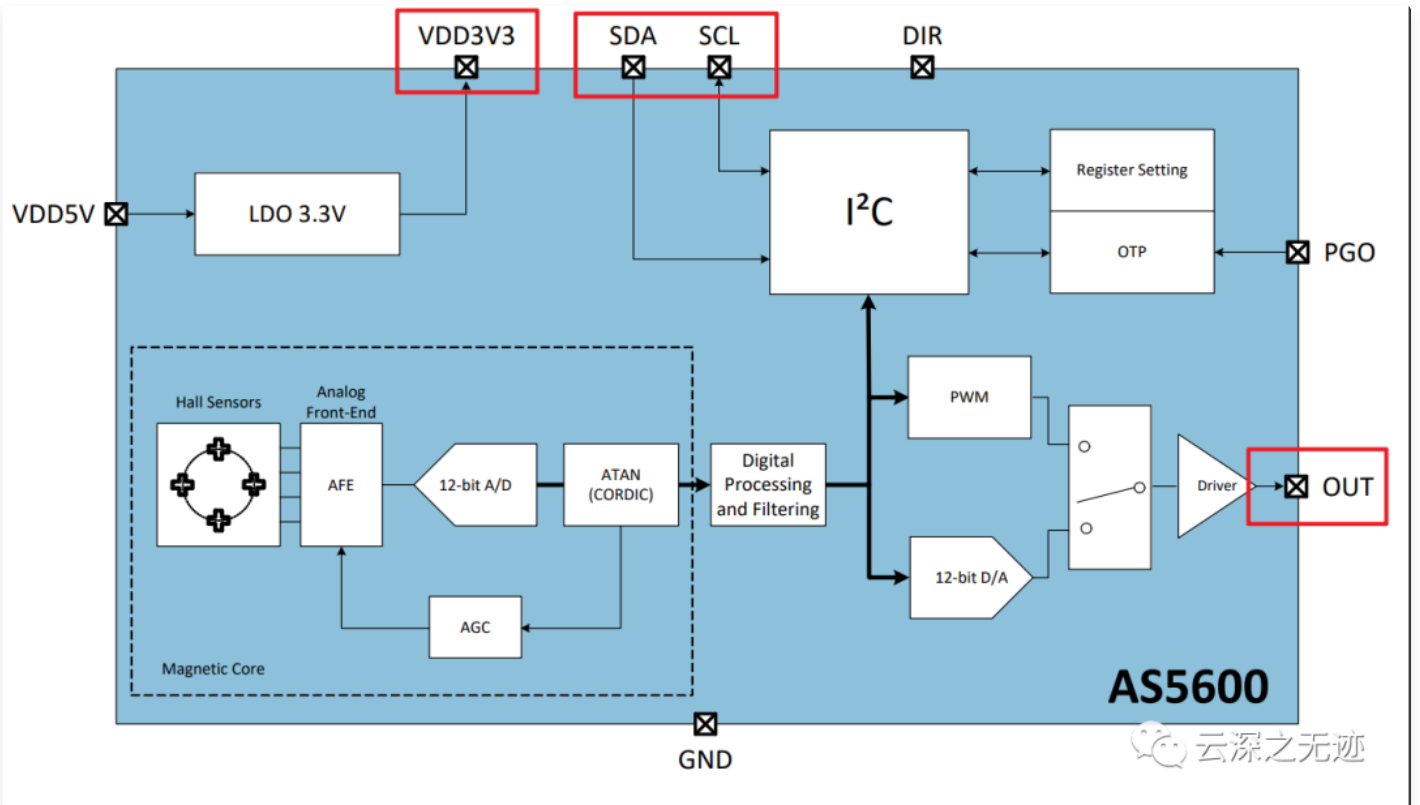
- 1 | https://blog.csdn.net/SC_Electronic/article/details/86105409
- 1 | https://github.com/capella-ben/microPython_AS5600L

microPython_AS5600L 磁铁旋转位置传感器

这是一个很棒的小型磁性位置传感器，带有 I2C 接口和 12 位输出。

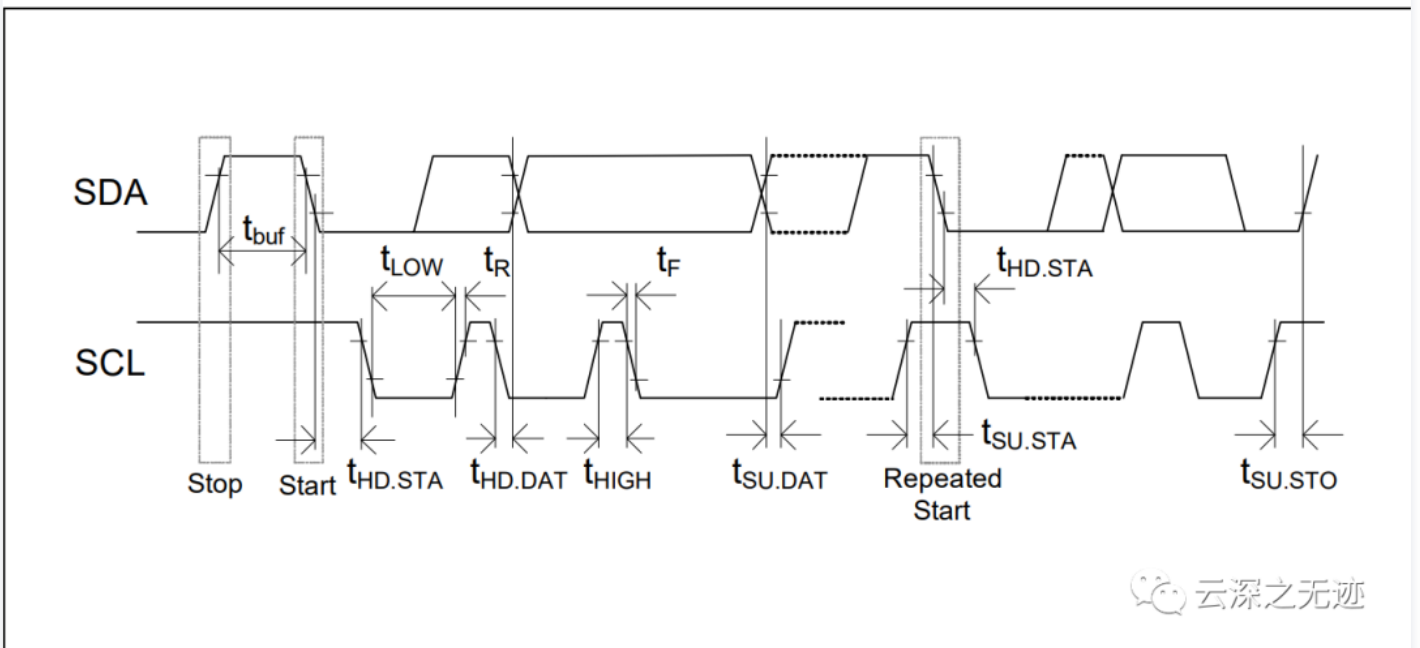
仅在 Raspberry Pi Pico 上测试过，但应该与其他人一起使用。

云深之无迹



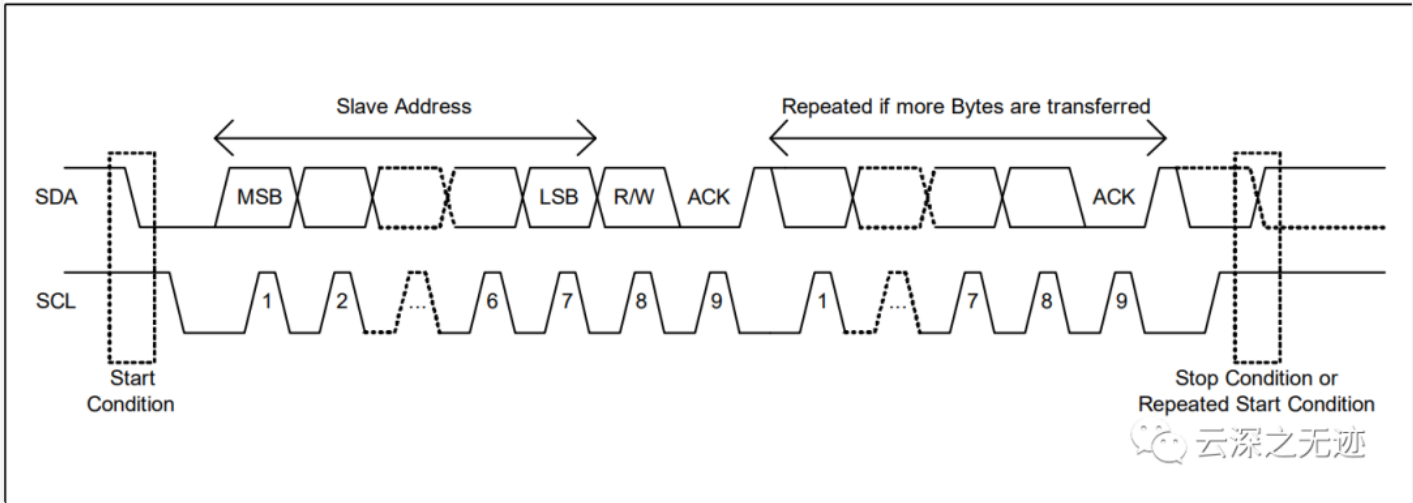
芯片框图，注意供电和控制以及输出

I²C Timing Diagram



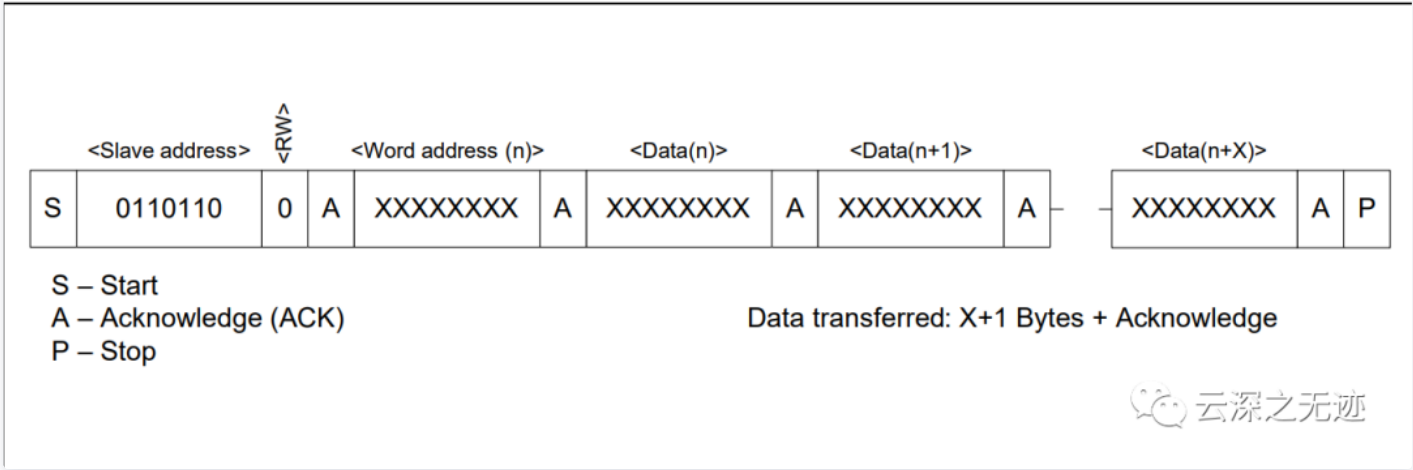
I2C的时序

Data Read



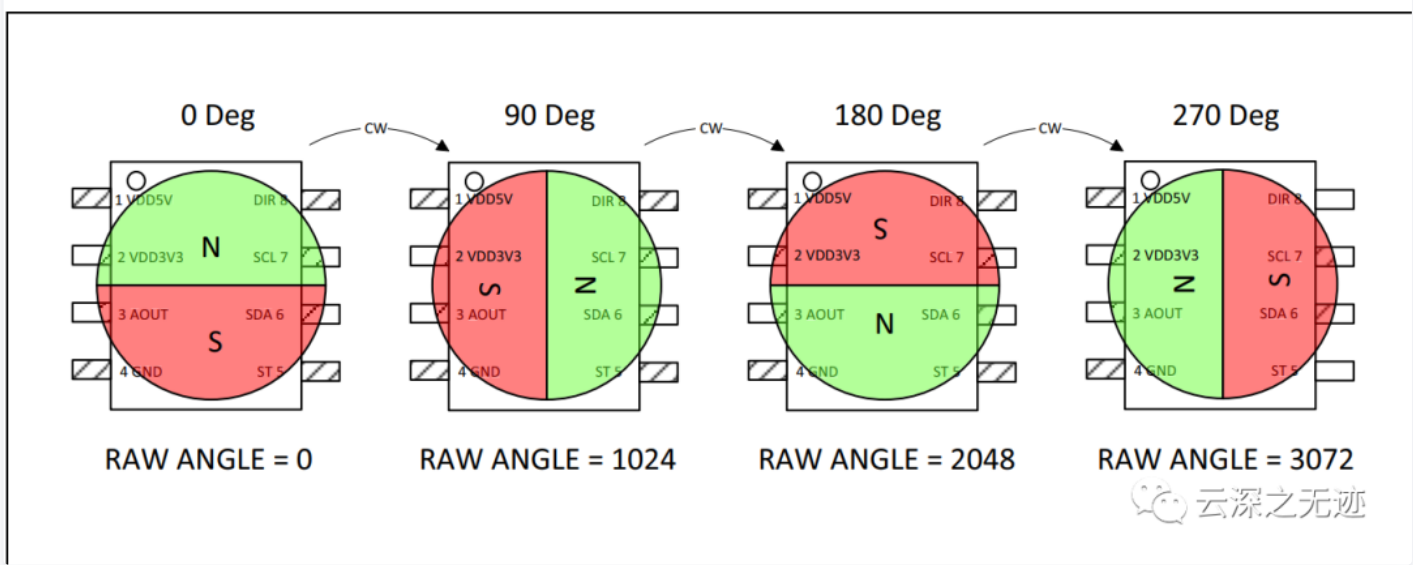
数据读

Data Write (Slave Receiver Mode)



数据写

Raw Angle in Clockwise Direction



角度输出

```
1 | int _raw_ang_hi = 0x0c;  
2 | int _raw_ang_lo = 0x0d;  
3 | int _ams5600_Address = 0x36;
```

复制

高低位和数据的地址

```
1 | int readOneByte(int in_addr)
2 | {
3 |     int retVal = -1;
4 |     Wire.beginTransmission(_ams5600_Address); //开始传输
5 |     Wire.write(in_addr); //需要读的地址
6 |     Wire.endTransmission();
7 |     Wire.requestFrom(_ams5600_Address, 1); //从5600处获取数据
8 |     while(Wire.available() == 0); //等待I2C总线停止占用（传输结束）
9 |     retVal = Wire.read(); //读到的数据
10 |
11 |     return retVal;
12 | }
```

读取字节的函数

10位的数据，要高位和地位，然后组合起来


```
1 | word readTwoBytes(int in_addr_hi, int in_addr_lo)
2 | {
3 |     word retVal = -1;
4 |
5 |     /* 先读取低位 */
6 |     Wire.beginTransmission(_ams5600_Address);
7 |     Wire.write(in_addr_lo);
8 |     Wire.endTransmission();
9 |     Wire.requestFrom(_ams5600_Address, 1);
10 |    while(Wire.available() == 0);
11 |    int low = Wire.read();
12 |
13 |    /* 再读取高位 */
14 |    Wire.beginTransmission(_ams5600_Address);
15 |    Wire.write(in_addr_hi);
16 |    Wire.endTransmission();
17 |    Wire.requestFrom(_ams5600_Address, 1);
18 |
19 |    while(Wire.available() == 0);
20 |
21 |    word high = Wire.read();
22 |
23 |    high = high << 8; //高位数据左移
24 |    retVal = high | low; //高位数据左移动8位之后，和低位数据或
25 |
26 |    //或计算就是尽量等于1，任何一边有1就是1
27 |
28 |    return retVal;
29 | }
```

读取两个数值组合后就是一个磁编码器数据，10bit，从0-4096的那种，根据角度划分， $360/4096=0.087$ 度，精度达到0.1度。

把这个东西输出：

```
1 | int val = getRawAngle()*0.087; //获取10bit数值，乘以角度后转成实际角度
2 |     Serial.println(val);
3 |     //delay(10);
4 |     digitalWrite(OUTPUT, HIGH);
5 |     delayMicroseconds(400);
6 |     digitalWrite(OUTPUT, LOW);
7 |     delay(3);
```

```
#define OUTPUT 8 //ver2
```

 云深之无迹

这里接电机

```
1 | #include "Wire.h"
2 | #define OUTPUT 8 //ver2
```



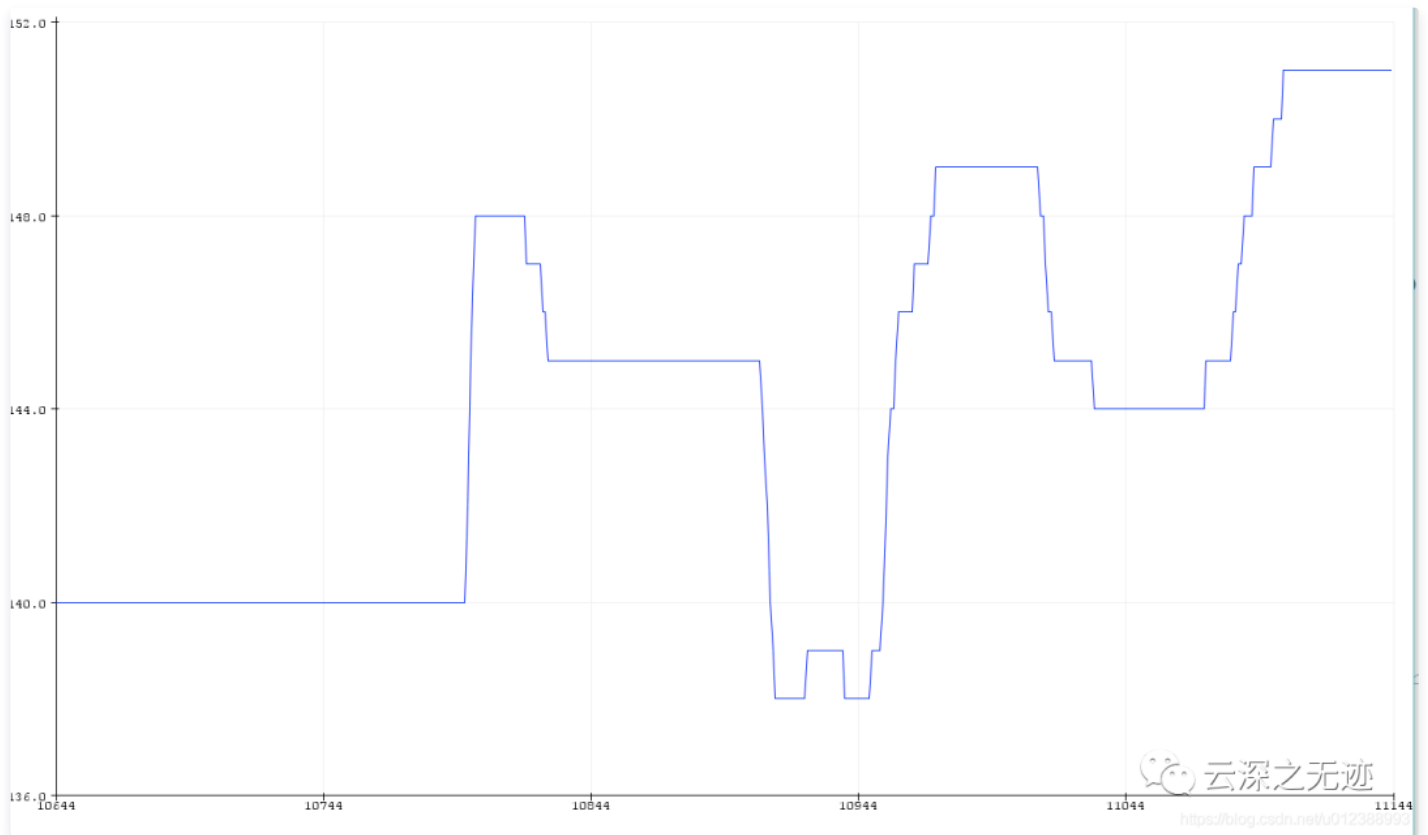
```

3  int _raw_ang_hi = 0x0c;
4  int _raw_ang_lo = 0x0d;
5  int _ams5600_Address = 0x36;
6  void setup() {
7
8  Serial.begin(115200);
9  Wire.begin();
10  pinMode(OUTPORT,OUTPUT);
11  }
12
13  void loop() {
14
15      //Serial.println(String(convertRawAngleToDegrees(getRawAngle()),DEC));
16      int val = getRawAngle()*0.087;
17      Serial.println(val);
18      //delay(10);
19      digitalWrite(OUTPORT,HIGH);
20      delayMicroseconds(400);
21      digitalWrite(OUTPORT,LOW);
22      delay(3);
23  }
24
25
26  float convertRawAngleToDegrees(word newAngle)
27  {
28      /* Raw data reports 0 - 4095 segments, which is 0.087 of a degree */
29      float retVal = newAngle * 0.087;
30      return retVal;
31  }
32
33  word getRawAngle()
34  {
35      return readTwoBytes(_raw_ang_hi, _raw_ang_lo);
36  }
37
38
39  int readOneByte(int in_adr)
40  {
41      int retVal = -1;
42      Wire.beginTransmission(_ams5600_Address);
43      Wire.write(in_adr);
44      Wire.endTransmission();
45      Wire.requestFrom(_ams5600_Address, 1);
46      while(Wire.available() == 0);
47      retVal = Wire.read();
48
49      return retVal;
50  }
51
52  word readTwoBytes(int in_adr_hi, int in_adr_lo)
53  {
54      word retVal = -1;
55
56      /* Read Low Byte */
57      Wire.beginTransmission(_ams5600_Address);
58      Wire.write(in_adr_lo);
59      Wire.endTransmission();
60      Wire.requestFrom(_ams5600_Address, 1);
61      while(Wire.available() == 0);
62      int low = Wire.read();
63
64      /* Read High Byte */
65      Wire.beginTransmission(_ams5600_Address);
66      Wire.write(in_adr_hi);
67      Wire.endTransmission();
68      Wire.requestFrom(_ams5600_Address, 1);
69
70      while(Wire.available() == 0);
71

```



```
72 | word high = Wire.read();
73 |
74 | high = high << 8;
75 | retVal = high | low;
76 |
77 | return retVal;
78 | }
```



串口绘图

1 | <https://blog.csdn.net/u012388993/article/details/113590623>

以上代码来自这里。