

# 基于SPI通信方式的OLED显示

原创

不#曾 & 轻听

于 2020-12-26 13:54:48 发布

3618

收藏

44

分类专栏：

嵌入式开发

 文章标签：

stm32

嵌入式

单片机

版权



嵌入式开发 专栏收录该内容

1 订阅 25 篇文章

订阅专栏

本文目的是通过SPI的 通信方式 进行OLED的显示，同时与之前学的I2C通信方式进行对比，在实践中学习SPI通信。

## 目录

### （一）SPI简介

- 1.SPI时序
- 2.SPI工作模式
- 3.SPI 优缺点

### （二）OLED简介

- 1.OLED原理
- 2.点阵编码原理与显示

### （三）OLED显示实验

- 1.实验目的
- 2.实验主要程序

### 3.实验说明

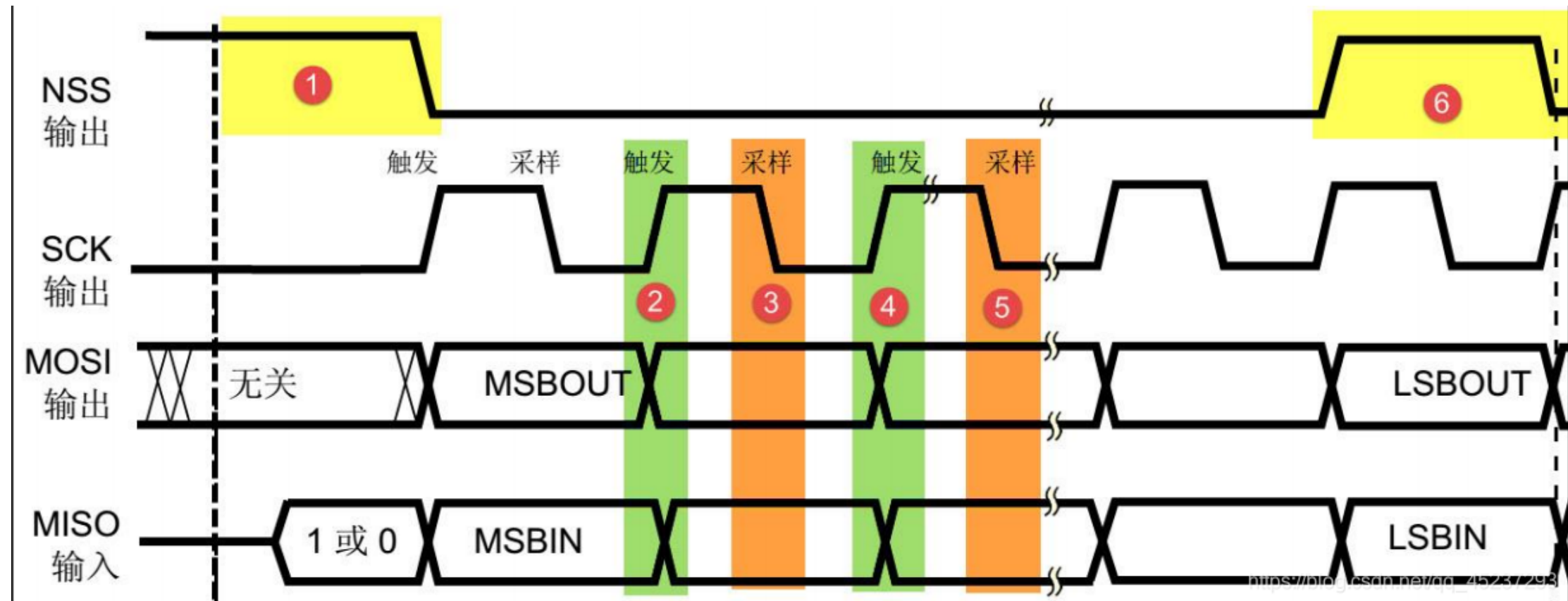
### 4.实验结果

## （四）总结

## （一）SPI简介

SPI是串行外设接口（Serial Peripheral Interface）的缩写，是由 Motorola 公司提出的一种高速的，全双工，同步的通信总线，被广泛地使用在 ADC、LCD 等设备与 MCU 间要求通讯速率较高的场合。SPI总线系统可直接与各个厂家生产的多种标准外围器件连接，该接口一般使用4条线：串行时钟线（SCK）、主机输入/从机输出数据线MISO、主机输出/从机输入数据线MOST和低电平有效的从机选择线C/S(有的SPI接口芯片带有中断信号线INT或INT、有的SPI接口芯片没有主机输出/从机输入数据线MOSI)。

### 1.SPI时序



上图中的时序只是 SPI 其中一种通讯模式，SPI 一共有四种通讯模式，它们的主要区别是总线空闲时 SCK 的时钟状态以及数据采样时刻。为方便说明，在此引入“时钟极性 CPOL”和“时钟相位 CPHA”的概念。

时钟极性 CPOL 是指 SPI 通讯设备处于空闲状态时，SCK 信号线的电平信号(即 SPI 通讯开始前、NSS 线为高电平时 SCK 的状态)。CPOL=0 时，SCK 在空闲状态时为低电平，CPOL=1 时，则相反。

时钟相位 CPHA 是指数据的采样的时刻，当 CPHA=0 时，MOSI 或 MISO 数据线上的信号将会在 SCK 时钟线的“奇数边沿”被采样。当 CPHA=1 时，数据线在 SCK 的“偶数边沿”采样。

## 2.SPI工作模式

根据 CPOL 及 CPHA 的不同状态，SPI 分成了四种模式，见下图，主机与从机需要工作在相同的模式下才可以正常通讯，实际中采用较多的是“模式 0”与“模式 3”。

SPI 模式	CPOL	CPHA	空闲时 SCK 时钟	采样时刻
0	0	0	低电平	奇数边沿
1	0	1	低电平	偶数边沿
2	1	0	高电平	奇数边沿
3	1	1	高电平	偶数边沿

[https://blog.csdn.net/qq\\_4269293](https://blog.csdn.net/qq_4269293)

### 3.SPI 优缺点

- SPI 优点

支持全双工通信  
通信简单  
数据传输速率快

- SPI 缺点

没有指定的流控制，没有应答机制确认是否接收到数据，所以跟IIC总线协议比较在数据可靠性上有一定的缺陷。

## (二) OLED简介

这里以7针SPI通信方式的OLED为例进行说明

# 1.OLED原理

OLED（OrganicLight-Emitting Diode），又称为有机电激光显示、有机发光半导体（OrganicElectroluminesence Display，OLED）。OLED属于一种电流型的有机发光器件，是通过载流子的注入和复合而致发光的现象，发光强度与注入的电流成正比。OLED在电场的作用下，阳极产生的空穴和阴极产生的电子就会发生移动，分别向空穴传输层和电子传输层注入，迁移到发光层。当二者在发光层相遇时，产生能量激子，从而激发发光分子最终产生可见光。

下面是7针SPI通信OLED和4针I2C通信OLED的图片。

7针OLED



4针OLED



## 2.点阵编码原理与显示

- 汉字点阵编码

在汉字的点阵字库中，每个字节的每个位都代表一个汉字的一个点，每个汉字都是由一个矩形的点阵组成，0 代表没有点，1 代表有点，将 0 和 1 分别用不同颜色画出，就形成了一个汉字，常用的点阵矩阵有 12\*12, 14\*14, 16\*16 三 种字库。

字库根据字节所表示点的不同有分为横向矩阵和纵向矩阵，目前多数的字库都是横向矩阵的存储方式(用得最多的应该是早期 UC DOS 字库)，纵向矩阵一般是因为有某些液晶是采用纵向扫描显示法，为了提高显示速度，于是便把字库 矩阵做成纵向，省得在显示时还要做矩阵转换。

- OLED点阵显示

点阵屏像素按128列\*64行组织，每一行128个像素单元的阴极是连接在一起，作为公共极（COM），每一列64个像素单元的阳极也连接在一起，作为一段（SEG）。行列交叉点上的LED就是一个显示单元，即一个像素。要点亮一个像素，只要在该像素所在列电极上加上正电压、行电极接地。同样，要驱动一整行图像，就需要同时把128列信号加载到列电极上，把该行行电极接地。该行显示时，其他63行均不能显示，其行电极应为高电平或悬空。

可见，整屏的显示，只能分时扫描进行，一行一行的显示，每次显示一行。行驱依次产生低电平扫描各行，列驱动读取显示数据依次加载到列电极上。扫描一行的时间称为行周期，完成一次全屏扫描，就叫做一帧。一般帧频大于60，人眼观察不到逐行显示。每行扫描显示用时叫占空比，占空比小，为达到相同的显示亮度，驱动电流就大。SSD1306段驱动最大电流为100uA，当整行128个像素全部点亮时，行电极就要流过12.8mA的电流。

## （三）OLED显示实验

### 1.实验目的

- 显示自己的学号和姓名；

- 显示AHT20的温度和湿度；
- 上下或左右的滑动显示(使用硬件刷屏模式)

## 2.实验主要程序

- main.c

```
1  #include "delay.h"
2  #include "sys.h"
3  #include "oled.h"
4  #include "gui.h"
5  #include "test.h"
6  #include "temhum.h"
7
8  int main(void)
9  {
10     u8 i;
11     delay_init();           // 延时函数初始化
12     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
13     AHT20_Init();
14     OLED_Init();           // 初始化OLED
15     OLED_Clear(0);         // 清屏 (全黑)
16
17     GUI_ShowCHinese(0,20,32,"通信二班",1);
18
19     OLED_Display_scroll();
20     delay_ms(1000);
21     delay_ms(1000);
22     delay_ms(1000);
23     delay_ms(1000);
24     delay_ms(1000);
25     delay_ms(1000);
26
```

```

27     delay_ms(1000);
28
29     while(1)
30     {
31         TEST_Chinese();           // 学号、姓名显示
32         OLED_Clear(0);
33
34         for(i=0;i<5;i++)
35         {
36             TEST_Menu2();         // AHT20 温湿度显示
37             delay_ms(500);
38         }
39
40         OLED_Clear(0);
41     }
}

```

- OLED主要函数

```

1  //OLED控制用函数
2  void OLED_WR_Byte(unsigned dat,unsigned cmd);
3  void OLED_Display_On(void);
4  void OLED_Display_Off(void);
5  void OLED_Set_Pos(unsigned char x, unsigned char y);
6  void OLED_Reset(void);
7  void OLED_Init_GPIO(void);
8  void OLED_Init(void);
9  void OLED_Set_Pixel(unsigned char x, unsigned char y,unsigned char color);
10 void OLED_Display(void);
11 void OLED_Clear(unsigned dat);
12
13
//OLED 清屏函数

```



```
void OLED_Display_scroll(void);
```

### 3.实验说明

#### • 硬件连线

```
//=====电源接线=====//
// OLED模块          STM32单片机
//  VCC          接      DC 5V/3.3V      //OLED屏电源正
//  GND          接      GND              //OLED屏电源地
//=====液晶屏数据线接线=====//
//本模块默认数据总线类型为4线制SPI
// OLED模块          STM32单片机
//  D1          接      PB15              //OLED屏SPI写信号
//=====液晶屏控制线接线=====//
// OLED模块          STM32单片机
//  CS          接      PB11              //OLED屏片选控制信号
//  RES         接      PB12              //OLED屏复位控制信号
//  DC          接      PB10              //OLED屏数据/命令选择控制信号
//  D0          接      PB13              //OLED屏SPI时钟信号
```

[https://blog.csdn.net/qq\\_45237293](https://blog.csdn.net/qq_45237293)

#### • 使用硬件SPI通信

```
//本测试程序使用的是硬件SPI接口驱动
//SPI时钟信号以及SPI读、写信号引脚不可更改
//SPI的时钟引脚定义固定为PB13
//SPI的读数据引脚定义固定为PB14
//SPI的写数据引脚定义固定为PB15

u8 SPI_WriteByte(SPI_TypeDef* SPIx,u8 Byte);
void SPI2_Init(void);
void SPI_SetSpeed(SPI_TypeDef* SPIx,u8 SpeedSet);
```

这里使用的是SPI2硬件通信，固定SPI的时钟引脚为PB13，读数据引脚为PB14，写数据引脚为PB15。

#### • 显示姓名学号

```

1 void TEST_Chinese(void)
2 {
3     OLED_Clear(0);
4     GUI_ShowString(50,50,"***",8,1);          // 此处显示学号
5     GUI_ShowCHinese(10,10,32,"***",1);  // 此处显示姓名
6     delay_ms(1000);
7     OLED_Clear(0);
8 }

```

这里使用的是已经编写好的GUI\_ShowString()函数和GUI\_ShowCHinese()函数显示学号和姓名。

需要注意的是，如果字库oledfont.h里没有相应的中文点阵编码，那么OLED将不会显示该中文。

- 显示温湿度

将之前[博客](#)的温湿度显示程序移植到该工程中，并将串口发送改为OLED显示即可。

```

1 void TEST_Menu2(void)
2 {
3     u32 CT_data[2];
4     volatile int c1=0,t1=0;
5     srand(123456);
6     delay_ms(40);
7     AHT20_Read_CTdata(CT_data);          // 不经过CRC校验，直接读取AHT20的温度和湿度数据    推荐每隔大于1S读一次
8     c1 = CT_data[0]*100*10/1024/1024;    // 湿度
9     t1 = CT_data[1]*200*10/1024/1024-500; // 温度
10
11     GUI_DrawLine(0, 10, WIDTH-1, 10,1);
12     GUI_DrawLine(WIDTH/2-1,11,WIDTH/2-1,HEIGHT-1,1);
13     GUI_DrawLine(WIDTH/2-1,10+(HEIGHT-10)/2-1,WIDTH-1,10+(HEIGHT-10)/2-1,1);
14

```

```

15     GUI_ShowString(0,1,"2020-12-23",8,1);
16     GUI_ShowString(74,1,"Wednesday",8,1);
17     GUI_ShowString(14,HEIGHT-1-10,"Cloudy",8,1);
18     GUI_ShowString(WIDTH/2+1,13,"TEMP",8,1);           // 显示温度
19     GUI_DrawCircle(WIDTH-20, 25, 1,2);
20     GUI_ShowString(WIDTH-15,20,"C",16,1);
21     GUI_ShowNum(WIDTH/2+8,20,t1/10,2,16,1);
22     GUI_ShowString(WIDTH-41,26,".",8,1);
23     GUI_ShowNum(WIDTH-35,20,t1%10,1,16,1);
24
25     GUI_ShowString(WIDTH/2+1,39,"HUM",8,1);           // 显示湿度
26     GUI_ShowNum(WIDTH/2+8,46,c1/10,2,16,1);
27     GUI_ShowString(WIDTH-41,52,".",8,1);
28     GUI_ShowNum(WIDTH-35,46,c1%10,1,16,1);
29     GUI_ShowString(WIDTH-21,46,"%",16,1);
30     GUI_DrawBMP(6,16,51,32, BMP5, 1);
31     delay_ms(1000);
32 }

```

- 滑动显示字符

采用SSD1306芯片控制OLED显示屏硬件滑动，这样的滑动效果显得很顺畅。

```

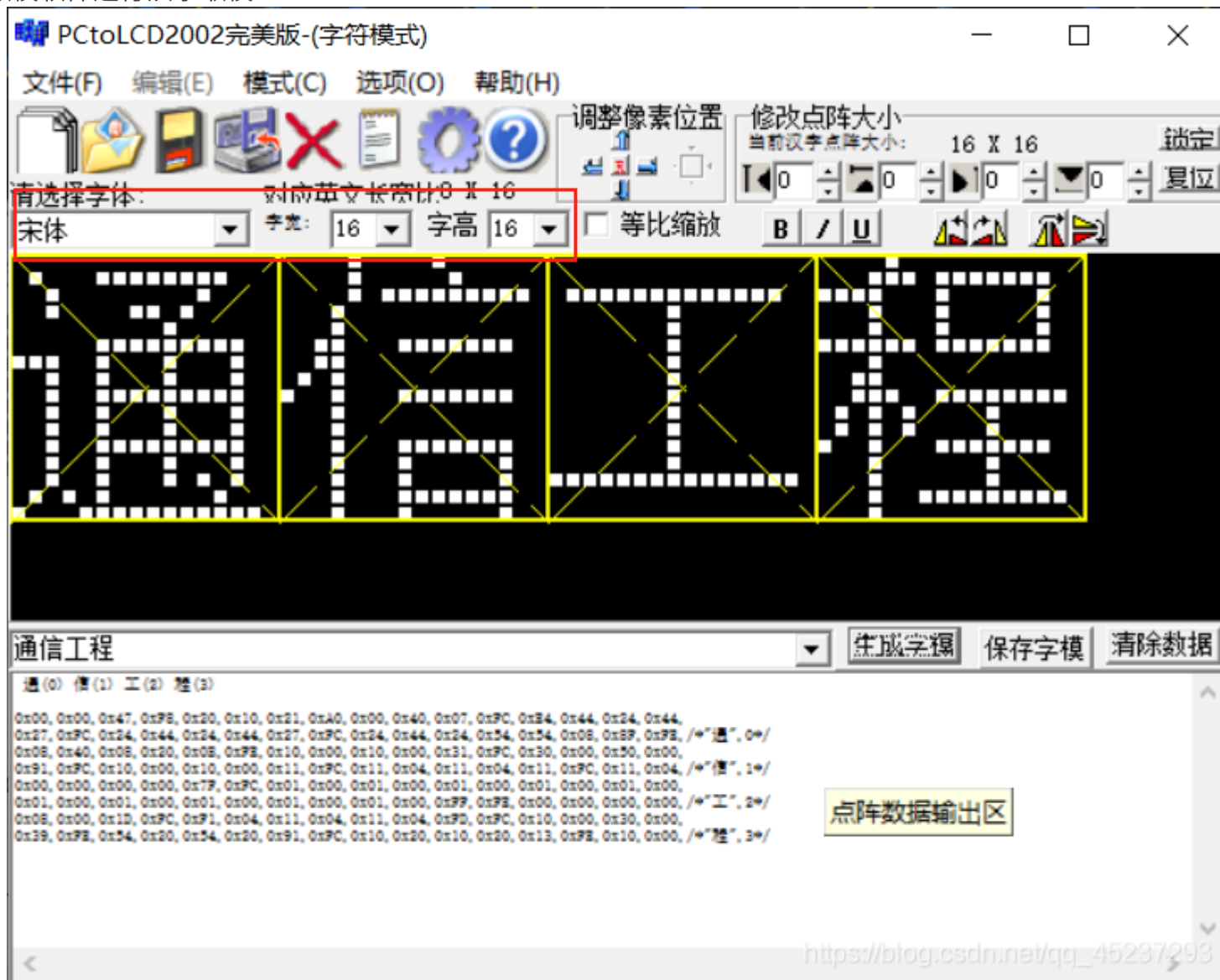
1  void OLED_Display_scroll(void)
2  {
3      OLED_WR_Byte(0x2e,OLED_CMD);// 关滚动
4      OLED_WR_Byte(0x2a,OLED_CMD);// 2a向右, 2a向左( 垂直水平滚动)
5      OLED_WR_Byte(0x00,OLED_CMD);// A: 空字节
6      OLED_WR_Byte(0x00,OLED_CMD);// B: 水平起始页
7      OLED_WR_Byte(0x00,OLED_CMD);// C: 水平滚动速度
8      OLED_WR_Byte(0x07,OLED_CMD);// D: 水平结束页
9      OLED_WR_Byte(0x01,OLED_CMD);// E: 每次垂直滚动位移
10

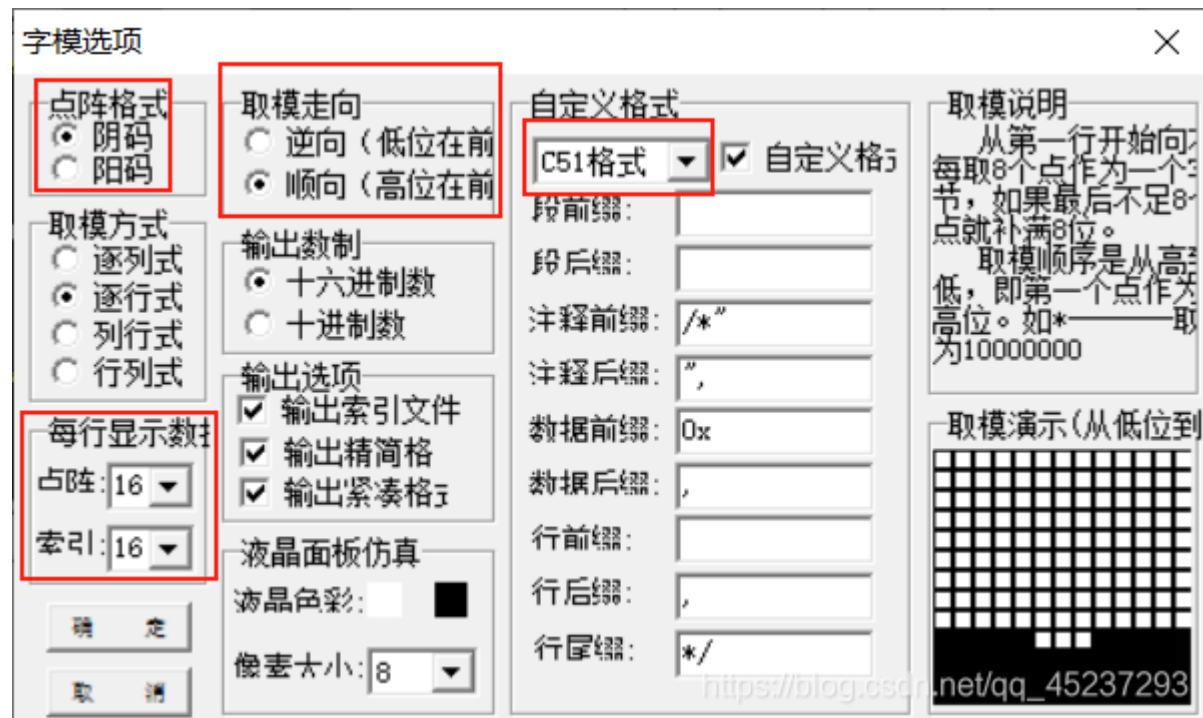
```

}

- 字库取模

由于OLED显示中使用了中文字符，因此需要将中文进行取模得到中文的点阵编码并存到oledfont.h中，方便程序调用并显示到OLED上去。下面演示如何使用汉字取模软件进行汉字取模：





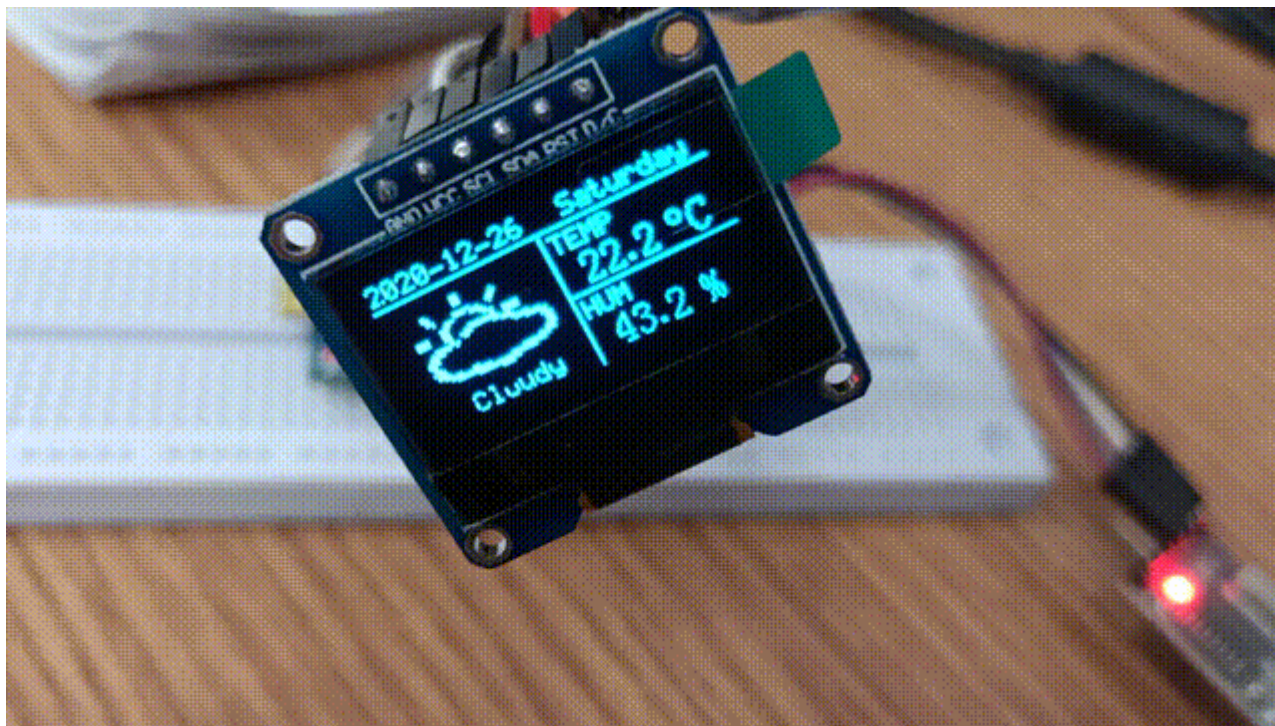
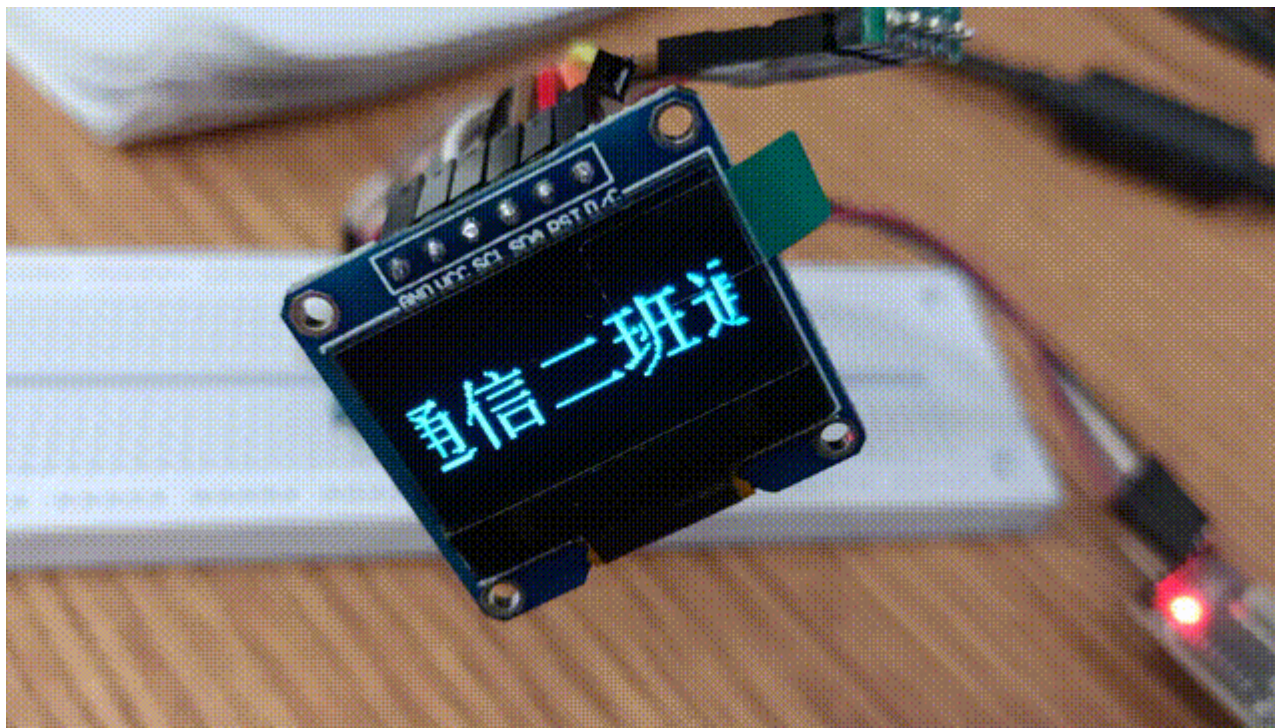
配置好选项后，输入需要取模的汉字，点击生成字模即可得到中文汉字的字模。

## 4.实验结果

- SPI通信OLED显示效果

这里的OLED是六针的原因是因为CS硬件上直接接地，没有使用（只有一个SPI从机时CS可以不使用）



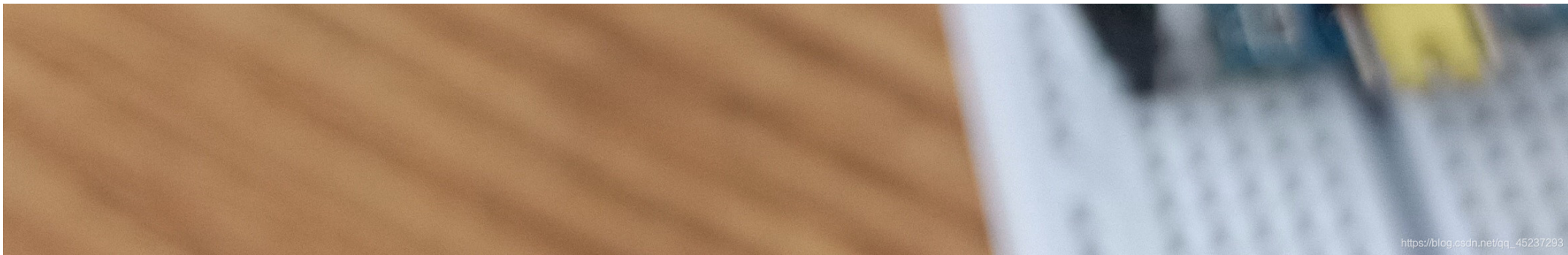


- I2C通信OLED显示效果









## （四）总结

通过使用SPI通信方式进行OLED的显示，接触到了许多新鲜的事物，比如说中文字库点阵、OLED屏幕滑动显示等，同时与I2C通信方式进行OLED显示进行了对比，发现SPI的速度确实要比I2C的传输速度更快，更适合用作高速传输通信。由于我也是初学SPI，如有错误希望大家批评指正。

参考文章：

- 1.SPI通信协议（SPI总线）学习
- 2.什么是SPI通信？
- 3.OLED–百度百科
- 4.OLED显示模块驱动原理及应用
- 5.《零死角玩转STM32—F103指南者.pdf》 第25章