STM32之在oled上实现文字滚动

原创
小张从未入门
于 2021-11-18 22:58:57 发布 5040 收藏 2
分类专栏： stm32 文章标签： stm32 arm 嵌入式硬件
版权

stm32 专栏收录该内容 0 订阅 17 篇文章 订阅专栏

> 本文内容：本文主要介绍怎么通过 单片机 控制7针的oled上实现文字滚动。

一、oled与SPI简介

（一）oled简介

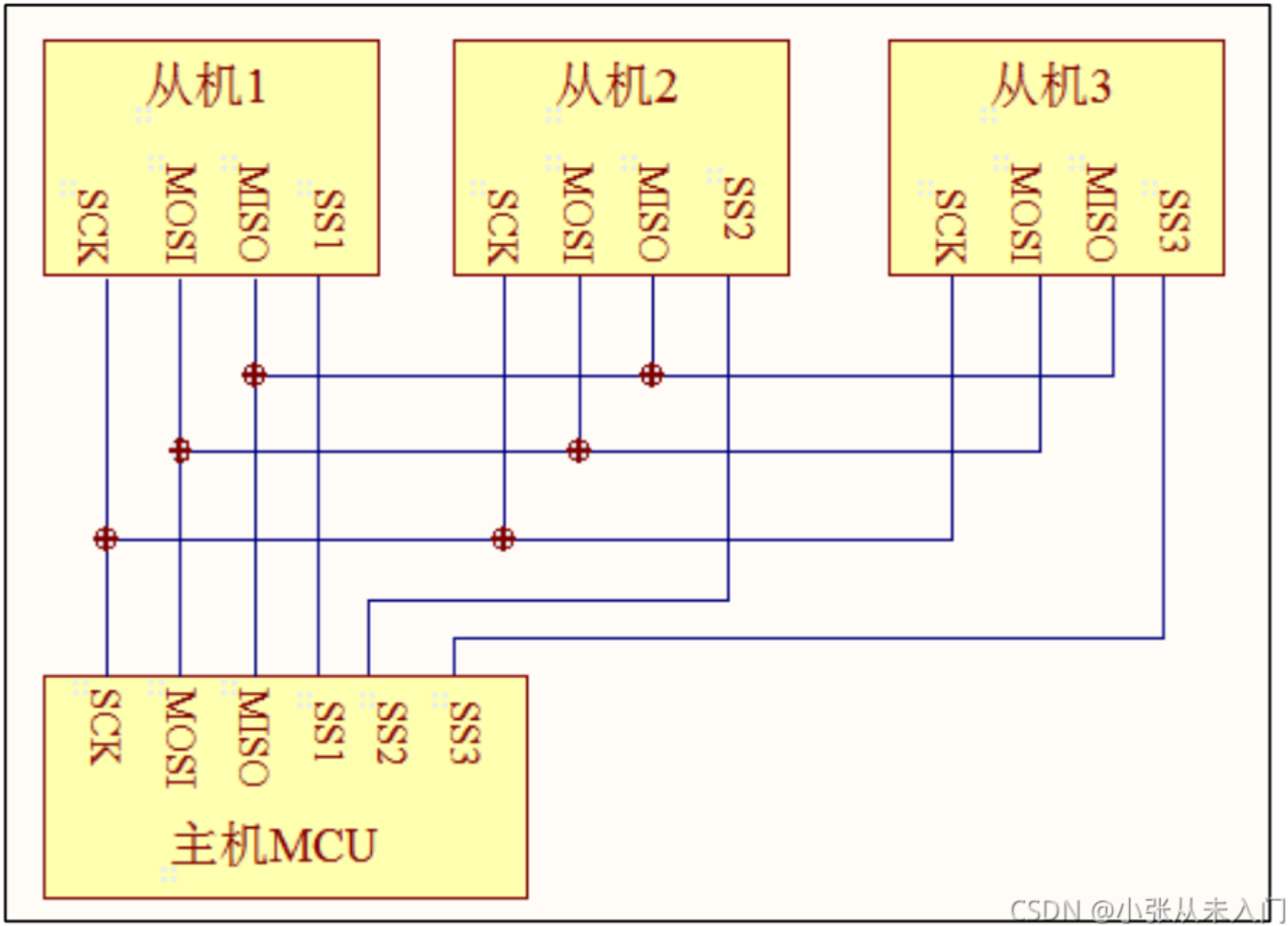
有机发光二极管（OrganicLight-Emitting Diode，OLED），又称为有机电激光显示、有机发光半导体（OrganicElectroluminescence Display，OLED），是指有机半导体材料和发光材料在电场驱动下，通过载流子注入和复合导致发光的现象。

（二）SPI协议

> 7针oled需要用到SPI协议

SPI 协议是由摩托罗拉公司提出的通讯协议(Serial Peripheral Interface)，即串行外围设备接口，是一种高速全双工的通信总线。它被广泛地使用在 ADC、LCD 等设备与 MCU 间，
要求通讯速率较高的场合。
SPI物理层
SPI通讯使用 3 条总线及片选线，3条总线分别为 SCK、MOSI、MISO

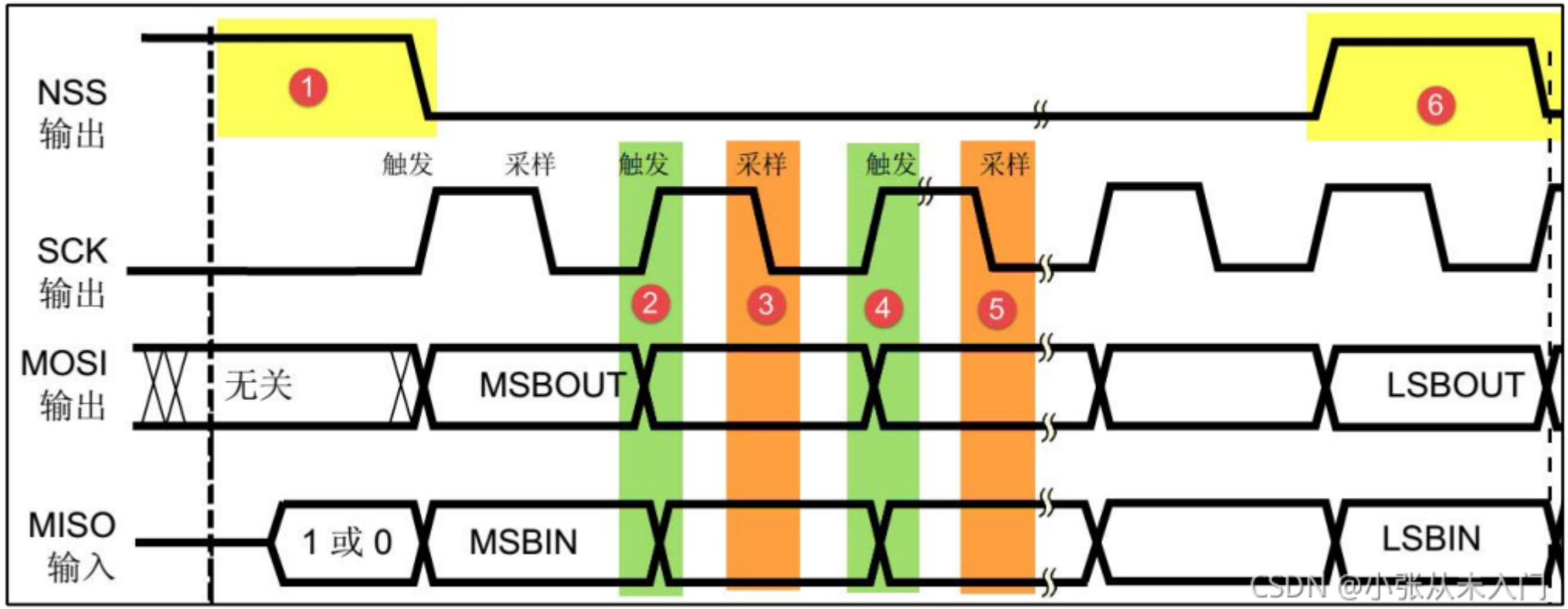
SPI 通讯设备之间的常用连接方式见图 25-1。



CSDN @小张从未入门

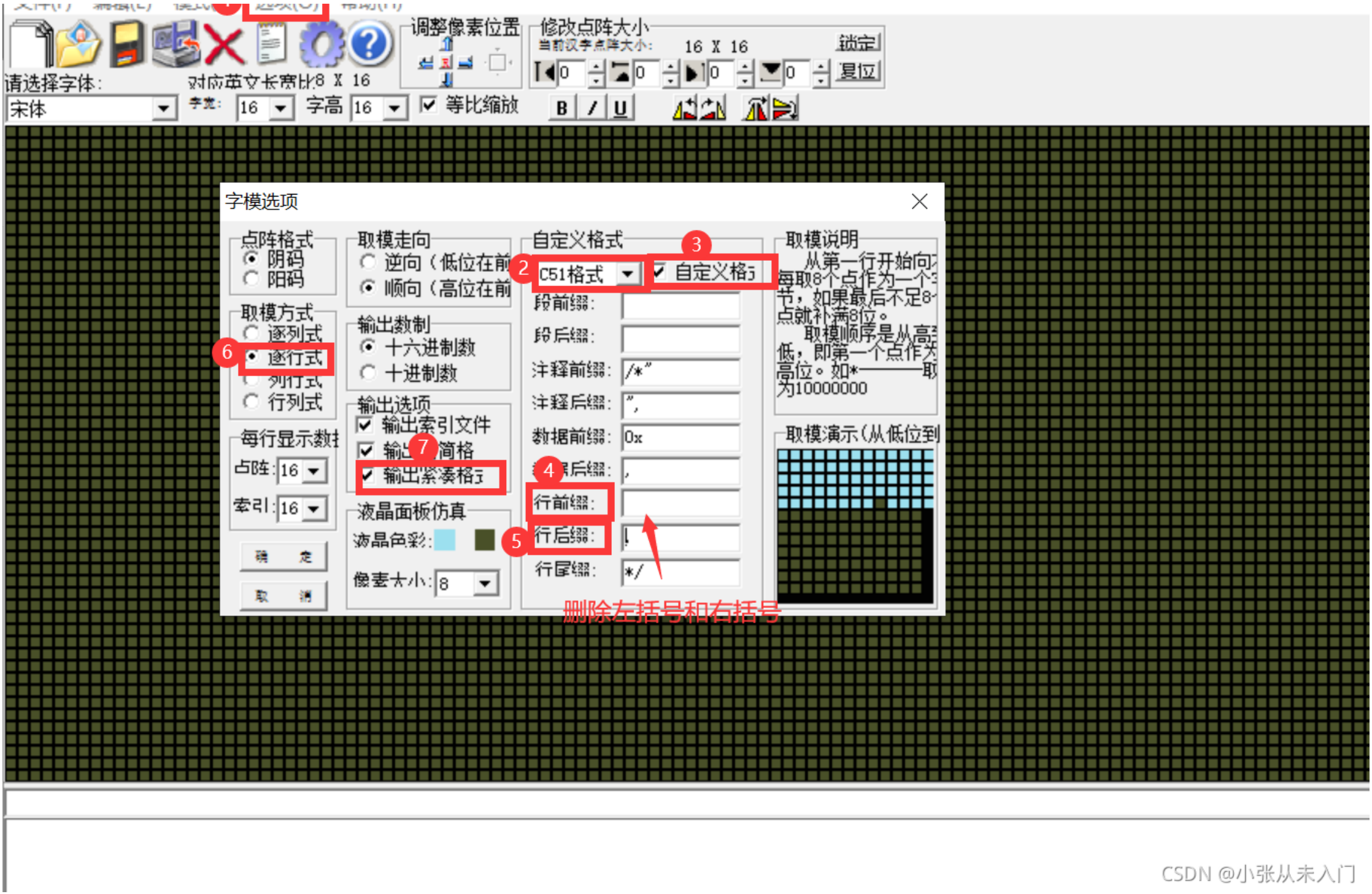
SPI协议层

与 I2C的类似，SPI协议定义了通讯的起始和停止信号、数据有效性、时钟同步等环节。
SPI通信时序



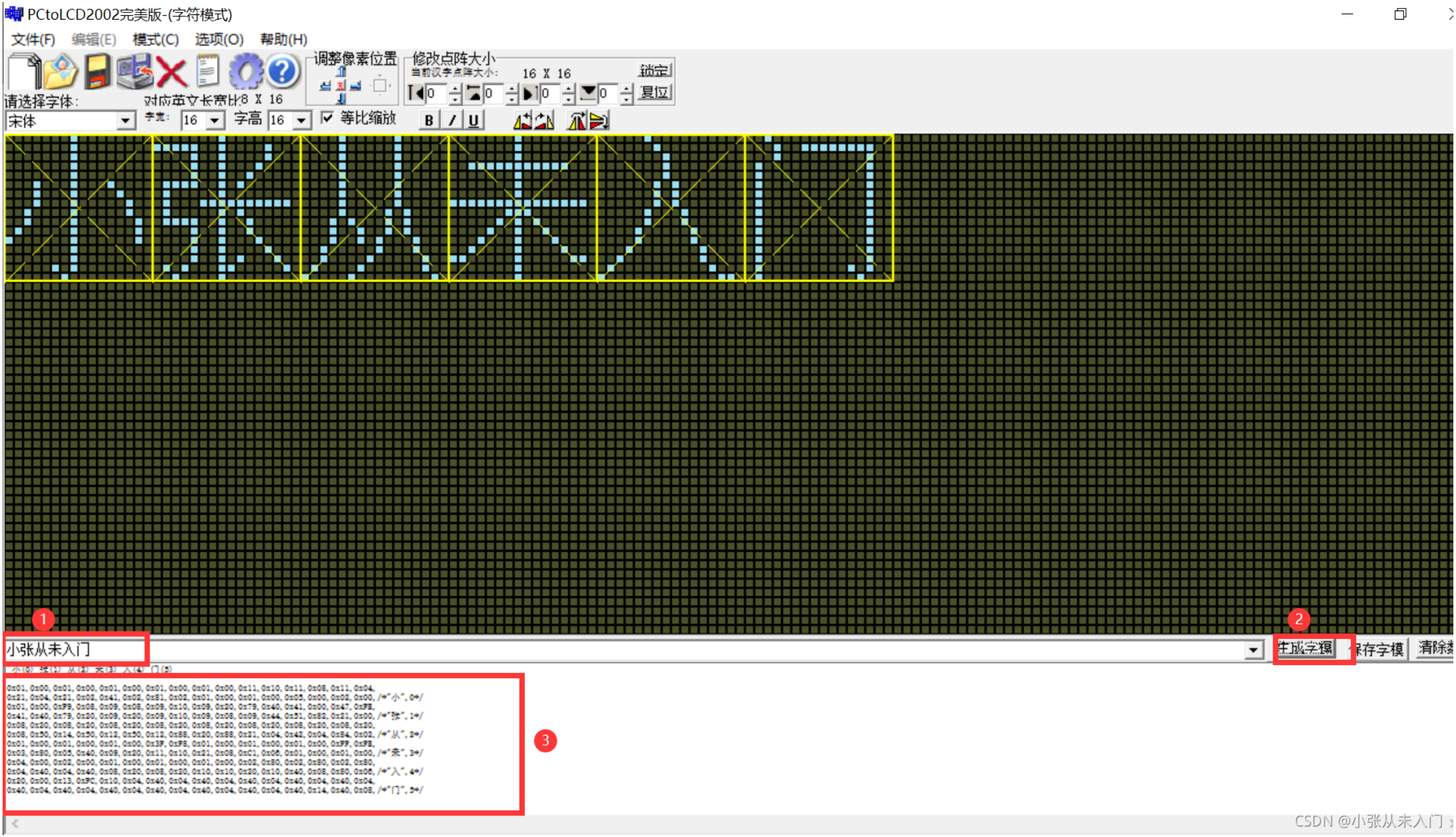
二、获取自模

下载字模软件PCtoLCD
字模软件PCtoLCD
打开后先进行设置，如下图：


调整像素位置
修改点阵大小
当前汉字点阵大小: 16 X 16
锁定
复位
请选择字体:
对应英文长宽比8 X 16
宋体
字宽: 16
字高 16
等比缩放
B I U
字模选项
点阵格式
阴码
阳码
取模走向
逆向（低位在前
顺向（高位在前
自定义格式
C51格式
自定义格式
取模说明
从第一行开始向
每取8个点作为一个
节，如果最后不足8
点就补满8位。
取模顺序是从高
低，即第一个点作为
高位。如*———取
为10000000
取模方式
逐列式
逐行式
列行式
行列式
输出数制
十六进制数
十进制数
输出选项
输出索引文件
输出紧凑格式
每行显示数据
点阵: 16
索引: 16
液晶面板仿真
液晶色彩:
像素大小: 8
段前缀:
段后缀:
注释前缀: /*"
注释后缀: ",
数据前缀: 0x
数据后缀: ,
行前缀:
行后缀:
行尾缀: */
取模演示(从低位到
确 定
取 消
2
3
4
5
6
7
删除左括号和右括号

CSDN @小张从未入门

输入想要获取的文字，点击生成字模，下面会出现字码



三、oled主要代码

主函数

```
#include "delay.h"
#include "sys.h"
#include "oled.h"
#include "gui.h"
#include "test.h"
int main(void)
{
        delay_init();                   //延时函数初始化
        NVIC_Configuration();      //设置NVIC中断分组2:2位抢占优先级，2位响应优先级
        OLED_Init();                            //初始化OLED
        OLED_Clear(0);             //清屏（全黑）
        OLED_WR_Byte(0x2E,OLED_CMD);         //关闭滚动
        OLED_WR_Byte(0x27,OLED_CMD);         //水平向左或者右滚动 26/27
        OLED_WR_Byte(0x00,OLED_CMD);         //虚拟字节
        OLED_WR_Byte(0x00,OLED_CMD);         //起始页 0
        OLED_WR_Byte(0x07,OLED_CMD);         //滚动时间间隔
        OLED_WR_Byte(0x07,OLED_CMD);         //终止页 7
        OLED_WR_Byte(0x00,OLED_CMD);         //虚拟字节
        OLED_WR_Byte(0xFF,OLED_CMD);         //虚拟字节
        TEST_MainPage();
        OLED_WR_Byte(0x2F,OLED_CMD);         //开启滚动
        while(1)
        {

        }
}
```

输出文字代码

```
void TEST_MainPage(void)
{
        GUI_ShowCHinese(10,20,16,"小张从未入门",1);
        GUI_ShowString(4,48,"631904110130",16,1);
        delay_ms(1500);
        delay_ms(1500);
}
```

添加的字模代码

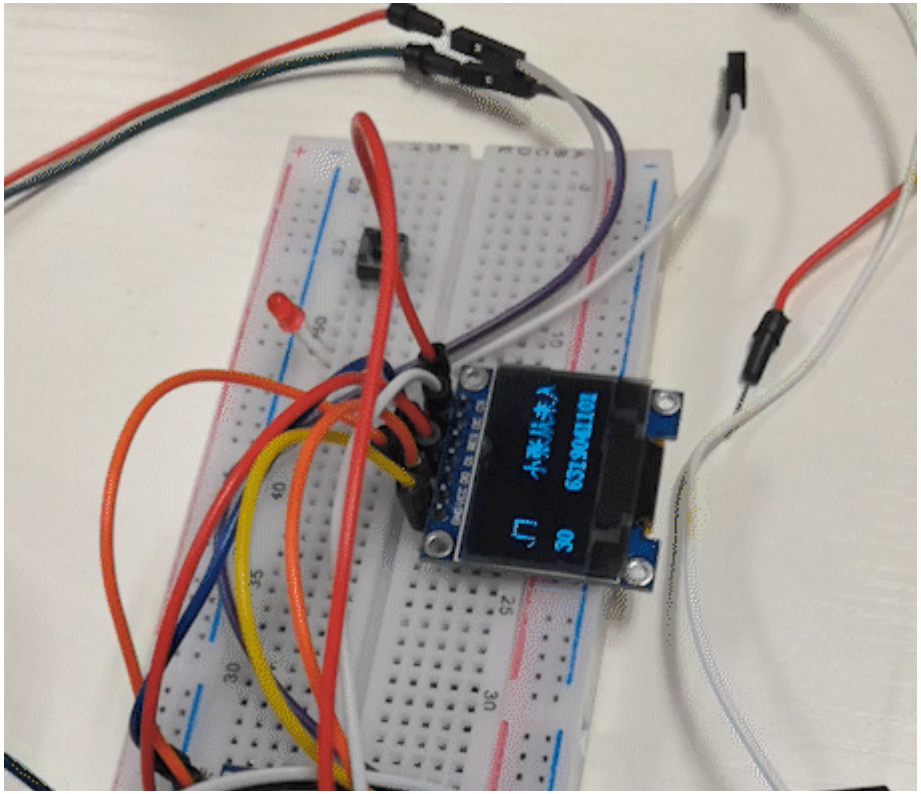
> 字模在oledfont.c文件中

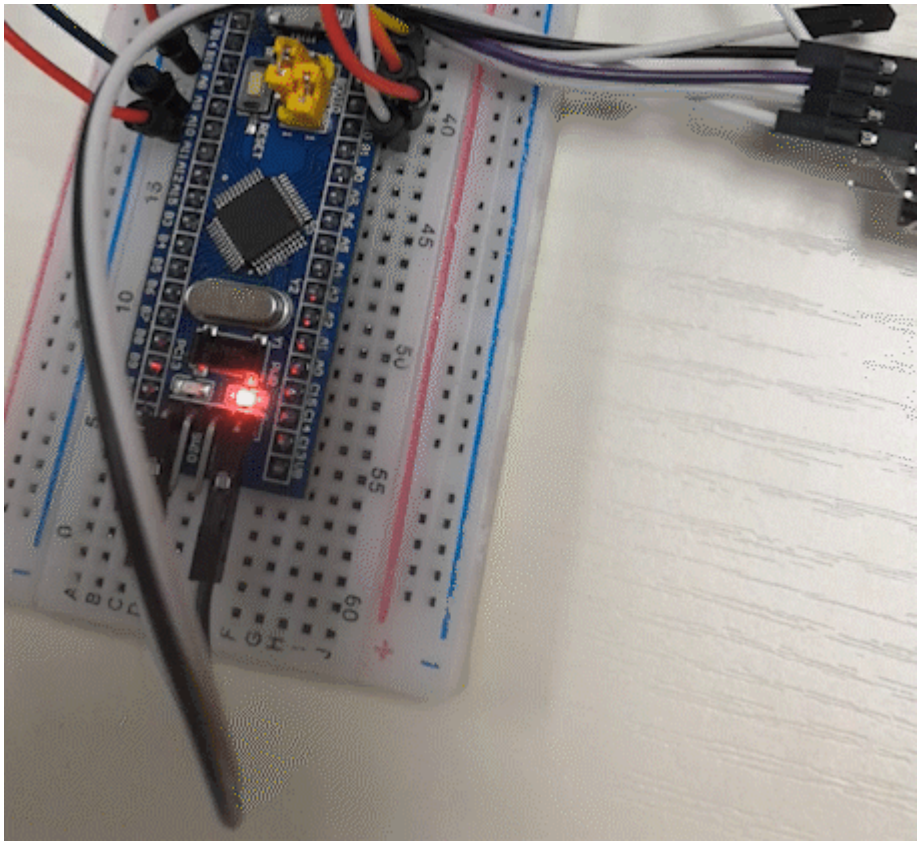
```
"小",0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x11,0x10,0x11,0x08,0x11,0x04,
        0x21,0x04,0x21,0x02,0x41,0x02,0x81,0x02,0x01,0x00,0x01,0x00,0x05,0x00,0x02,0x00,/*"小",0*/
        "张",0x01,0x00,0xF9,0x08,0x09,0x08,0x09,0x10,0x09,0x20,0x79,0x40,0x41,0x00,0x47,0xFE,
        0x41,0x40,0x79,0x20,0x09,0x20,0x09,0x10,0x09,0x08,0x09,0x44,0x51,0x82,0x21,0x00,/*"张",1*/
        "从",0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,
        0x08,0x50,0x14,0x50,0x12,0x50,0x12,0x88,0x20,0x88,0x21,0x04,0x42,0x04,0x84,0x02,/*"从",2*/
        "未",0x01,0x00,0x01,0x00,0x01,0x00,0x3F,0xF8,0x01,0x00,0x01,0x00,0x01,0x00,0xFF,0xFE,
        0x03,0x80,0x05,0x40,0x09,0x20,0x11,0x10,0x21,0x08,0xC1,0x06,0x01,0x00,0x01,0x00,/*"未",3*/
        "入",0x04,0x00,0x02,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x02,0x80,0x02,0x80,0x02,0x80,
        0x04,0x40,0x04,0x40,0x08,0x20,0x08,0x20,0x10,0x10,0x20,0x10,0x40,0x08,0x80,0x06,/*"入",4*/
        "门",0x20,0x00,0x13,0xFC,0x10,0x04,0x40,0x04,0x40,0x04,0x40,0x04,0x40,0x04,0x40,0x04,
        0x40,0x04,0x40,0x04,0x40,0x04,0x40,0x04,0x40,0x04,0x40,0x14,0x40,0x08,/*"门",5*/
```

四、运行效果

连线如下

```
// 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
```





运行效果如图

五、总结

本文的代码使用的其他大佬的代码改的，主要加了一些自己用到的字模，这次用到的是7针的oled的SPI协议，与4针的oled有所不同，4针为I2C的协议。oled里面的函数需要根据用到的字长和想要的位置进行相应的参数更改，否则可能达不到想要的效果。

六、参考文章

0.96寸OLED在STM32f103上实现滚动显示长字符

基于SPI通信方式的OLED显示

完整代码