

LCD 128x64 屏幕如何驱动



主讲人：程飞
i@feieeee.com

简介

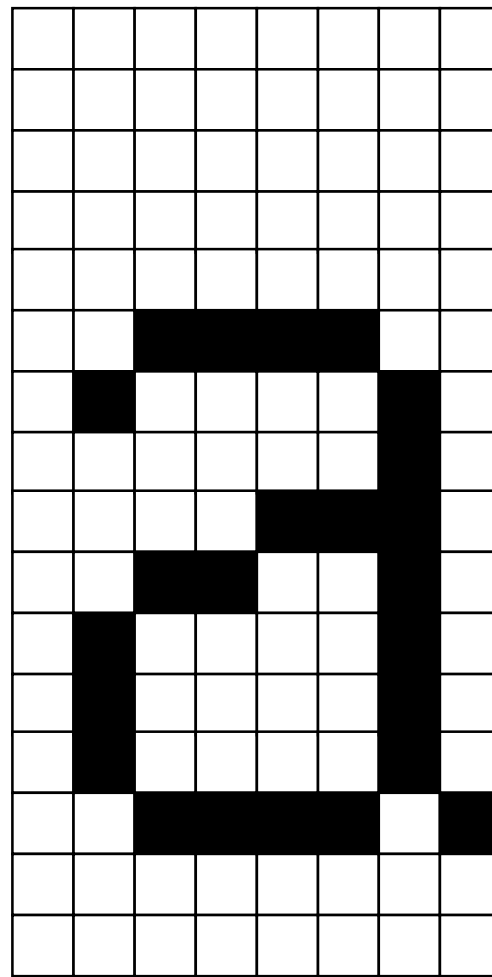
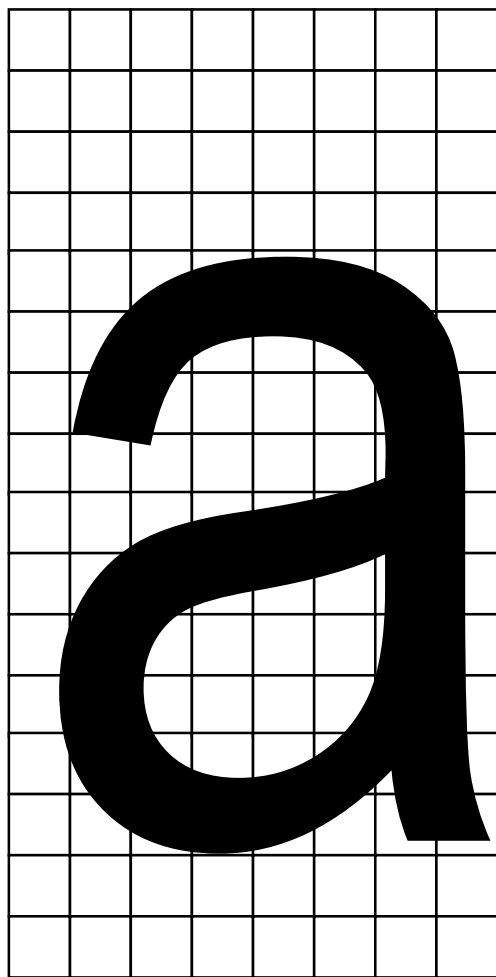
TREEE

- 分辨率：宽128px 高64px
- 尺寸：很小
- 驱动芯片：UC1701（SPI）
- 字库芯片：GT20L16S1Y（SPI）



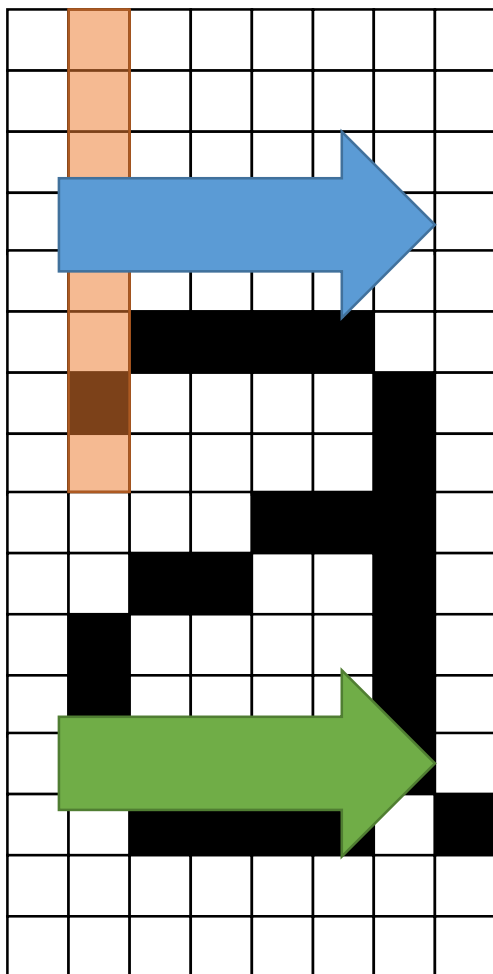
点阵屏的特点

TREEE



存储字模

TREEE

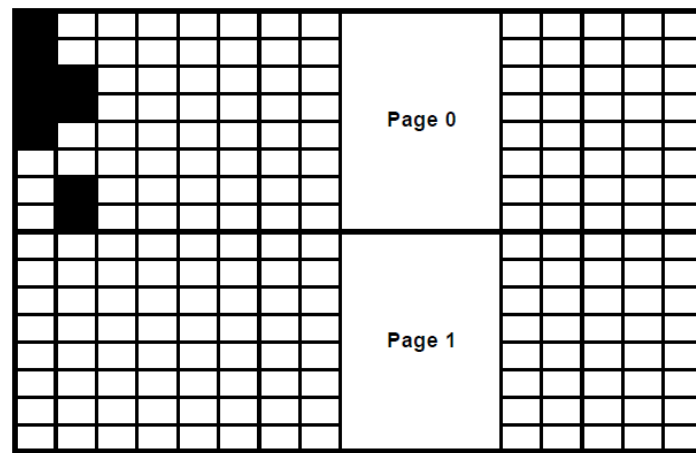


0x00

0x40

01000000

PA[3:0]	0	Line AddeCss
0000	D0	00H
	D1	01H
	D2	02H
	D3	03H
	D4	04H
	D5	05H
	D6	06H
	D7	07H
0001	D0	08H
	D1	09H
	D2	0AH
	D3	0BH
	D4	0CH
	D5	0DH
	D6	0EH
	D7	0FH



如何获取和存储字模

通过Matlab的基本思路

- 将字符显示到figure上去
- 保存成图片
- 读取图片，然后判断每一个像素，然后组成16进制数
- 按照扫描顺序存储到文本

- 然后按照ASCII码表的顺序依次取字模，存储

接线方式

TREEE



11

13

9

8

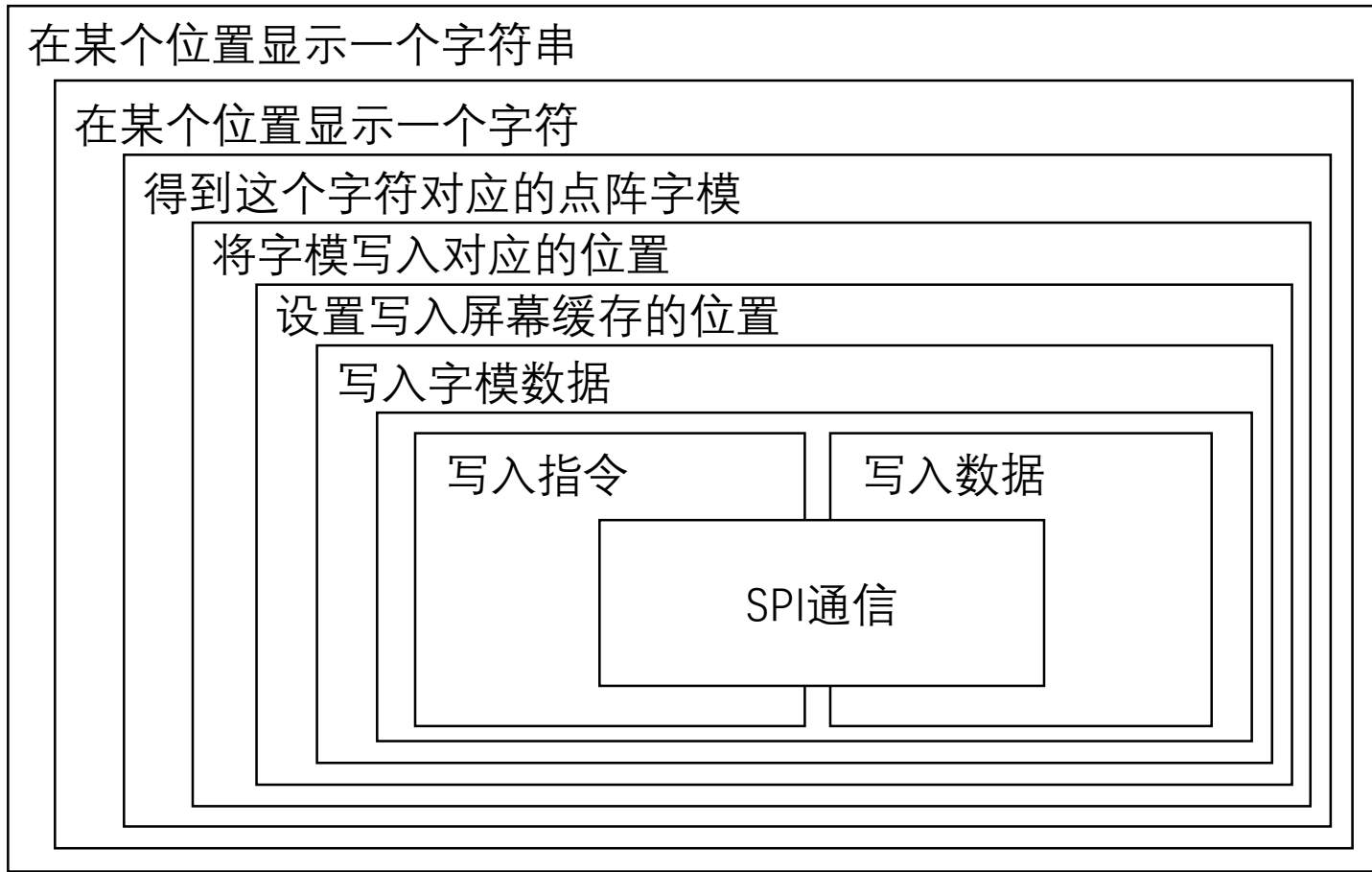
10

5V

GND

点阵屏的显示字符串的原理

TREEE



写驱动程序流程

1. SPI写入
2. 向屏幕写入命令
3. 向屏幕写入数据
4. 设定写入的位置
5. 在某位置上显示一个字符
6. 在某位置显示一个字符串
7. 屏幕初始化
8. 其他相关功能
9. 调用与测试

#define的灵活使用

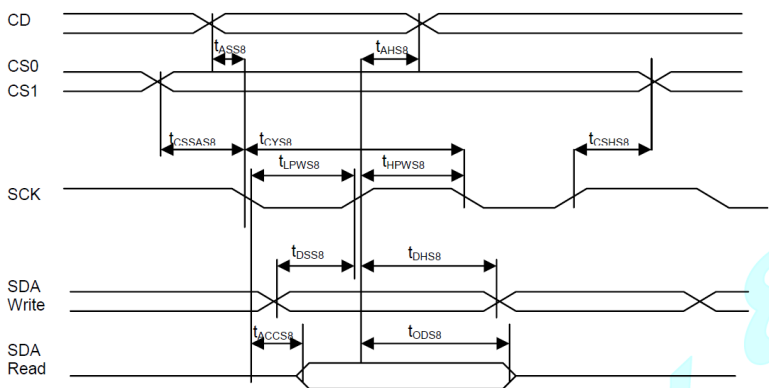
TREEE

```
#define LCD_SCK 13
#define LCD_SI 11
#define LCD_CS 10
#define LCD_CD 9
#define LCD_RST 8

#define Set(x) digitalWrite(x, 1)
#define Clr(x) digitalWrite(x, 0)
#define Out(x) pinMode(x, 1)
#define In(x) pinMode(x, 0)
```

SPI写入

TREEE



```
void SPI_write(unsigned char dat)
{
    for(unsigned char i = 0; i < 8; i++)
    {
        Clr(LCD_SCK);
        if (dat & 0x80)
            Set(LCD_SI);
        else
            Clr(LCD_SI);
        dat <<= 1;
        Set(LCD_SCK);
    }
}
```

写入数据或者指令

TREEE

```
void write_cmd(unsigned char dat)
{
    Clr(LCD_CD); //Wrtie CMD
    SPI_write(dat);
}

void write_dat(unsigned char dat)
{
    Set(LCD_CD); //Wrtie Data
    SPI_write(dat);
}
```

设定写入的位置

4. Set Column Address

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Column Address LSB, CA[3:0]	0	0	0	0	0	0	CA3	CA2	CA1	CA0
Set Column Address MSB, CA[7:4]	0	0	0	0	0	1	CA7	CA6	CA5	CA4

Set the SRAM column address before Write/Read memory from host interface.

CA value range: 0~131

7. Set Page Address

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Page Address, PA[3:0]	0	0	1	0	1	1	PA3	PA2	PA1	PA0

Set the SRAM page address before write/read memory from host interface. Each page of SRAM corresponds to 8 COM lines on LCD panel, except for the last page. The last page corresponds to the icon output CIC.

Possible value = 0~8.

```
void LCD_add(unsigned page, unsigned column)
{
    write_cmd(0xb0 + page);
    write_cmd(0x10 + ((column>>4) & 0x0f));
    write_cmd(column & 0x0f);
}
```

显示一个字符串

TREEE

```
void Dis_char(unsigned char line, unsigned char column, char *chr)
{
    unsigned char page = line * 2;
    Clr(LCD_CS);
    for (unsigned char i = 0; i < 2; i++)
    {
        LCD_add(page + i, column);
        for (unsigned j = 0; j < 8; j++)
        {
            int loc = (*chr - 32) * 16 + j + i * 8;
            write_dat(pgm_read_byte_near(ascii+loc));
        }
    }
    Set(LCD_CS);
}
```

Arduino中如何存储大量数据

TREEE

- `#include <avr/pgmspace.h>`

```
const dataType variableName[] PROGMEM = {data0, data1, data3...};
```

- 如何读取数据: `pgm_read_byte_near(address)`

屏幕初始化

TREEE

```
void LCD_init() {
    Clr(LCD_CS);
    Clr(LCD_RST); delay(20);
    Set(LCD_RST); //Reset the LCD
    write_cmd(0xe2); /*软复位*/ delay(5);
    write_cmd(0x2c); /*升压步聚1*/ delay(5);
    write_cmd(0x2e); /*升压步聚2*/ delay(5);
    write_cmd(0x2f); /*升压步聚3*/ delay(5);
    write_cmd(0x23); /*粗调对比度, 可设置范围0x20~0x27*/
    write_cmd(0x81); /*微调对比度*/
    write_cmd(0x2f); /*0x28, 微调对比度的值, 可设置范围0x00~0x3f*/
    write_cmd(0xa2); /*1/9偏压比 (bias) */
    write_cmd(0xc8); /*行扫描顺序: 从上到下*/
    write_cmd(0xa0); /*列扫描顺序: 从左到右*/
    write_cmd(0x40); /*起始行: 第一行开始*/
    write_cmd(0xaf); /*开显示*/
    Set(LCD_CS);
}
```

清屏

TREEE

```
void Clr_LCD()
{
    unsigned char i, j;
    Clr(LCD_CS);
    for(i=0; i<9; i++)
    {
        write_cmd(0xb0 + i);
        write_cmd(0x10);
        write_cmd(0x00);
        for(j=0; j<132; j++)
        {
            write_dat(0x00);
        }
    }
    Set(LCD_CS);
}
```


谢谢

TREEE

- 有问题可以访问tree.feieee.com, 或者发邮件到i@feieee.com

T Together
R Reach
E Experienced
E Elegant
E Excellent