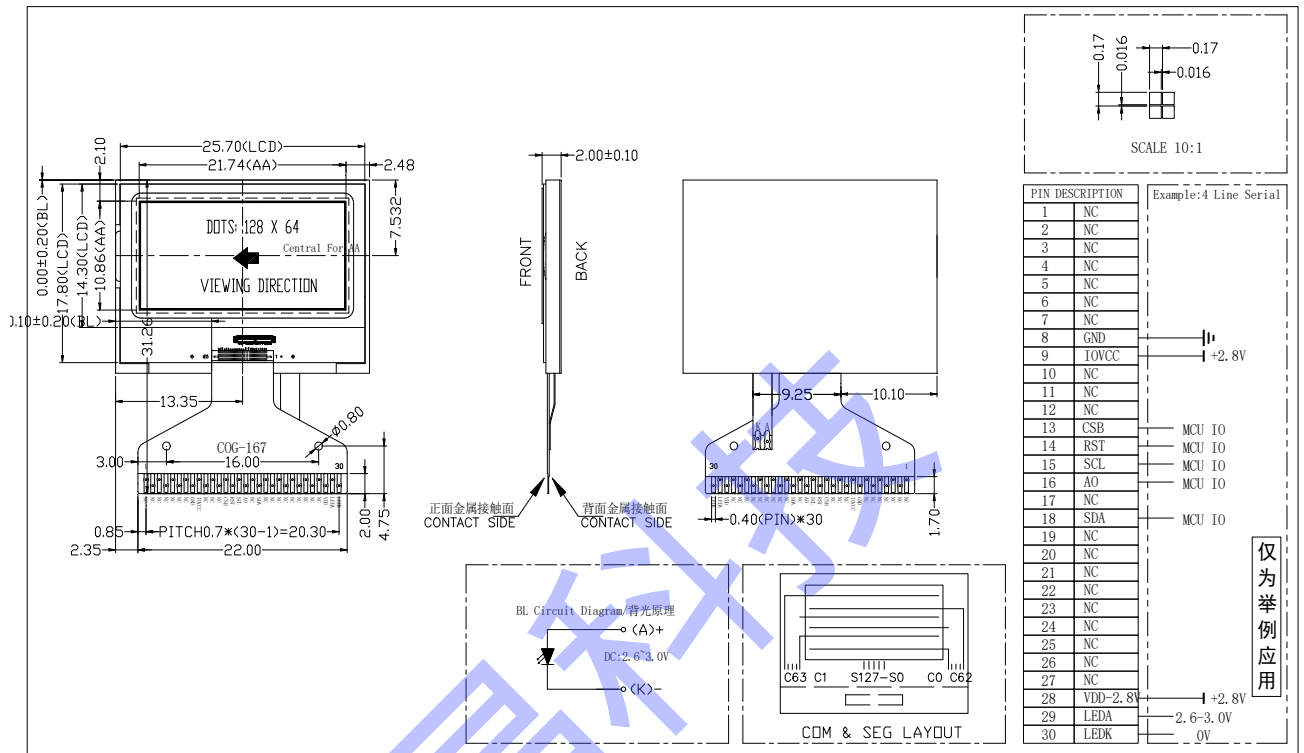




## 目 录

第一章、	显示器外形结构尺寸图.....	1
第二章、	显示器基本功能介绍.....	2
第三章、	显示器接口定义说明.....	3
第一节、	显示器管脚说明.....	3
第四章、	显示器的电性参数.....	4
第一节、	直流供电参数.....	4
第二节、	级限参数.....	4
第五章、	显示器的显示结构原理.....	5
第一节、	显示器控制器方框图.....	5
第二节、	玻璃走线映射图.....	6
第三节、	显示内存映射图.....	6
第四节、	写入数据流程图.....	7
第六章、	驱动程序时序图说明.....	8
第一节、	模块串行接口时序图.....	8
第二节、	写串口时序的流程图.....	9
第三节、	写复位时序.....	9
第七章、	驱动程序的指令说明.....	10
第一节、	显示模块指令表:.....	10
第二节、	显示模块指令详细说明.....	11
一、	显示开关设置.....	11
二、	起始行设置.....	11
三、	面地址设置.....	11
四、	列地址设置.....	11
五、	读状态设置.....	12
六、	电源控制设置.....	12
七、	调节对比度设置.....	12
八、	玻璃微调比度设置.....	13
第八章、	单片机与显示器连接说明.....	14
第一节、	接口串口应用原理图.....	14
第九章、	单片机驱动程序源代码.....	15
第一节、	源代码解释定义声明.....	15
第二节、	接口时序函数.....	16
第三节、	液晶模块初始化.....	17
第四节、	应用函数.....	18
第五节、	主调用函数.....	22
第六节、	动态显示函数.....	23
第七节、	标准字符数据表.....	25
第十章、	版本信息.....	25

# 第一章、显示器外形结构尺寸图



背光 1 支灯并联

项目	参考值
单位 (英寸)	0.96in
LCM 尺寸 (长×宽×厚)	26.7×19.26×2.0
可视区域 (长×宽)	22.34×11.46
点尺寸 (长×宽)	0.154×0.154
点间距 (长×宽)	0.17×0.17

## 第二章、显示器基本功能介绍

- ◇ COG: 将驱动 IC、FPC 软板直接压贴在 LCD 玻璃上的封装形式叫做 COG, 有性价比高、低功耗、体积小、软件指令丰富的优势
- ◇ 工作电压 3.3V, 背光电压 2.8V/15MA
- ◇ 通讯方式: 4 线串行通讯
- ◇ 显示器驱动芯片为 SC5260, 兼容 ST7567A, AIP31567A 等的内存 (DDRAM132\*65)
  - 可自由编写图形或者汉字
- ◇ 功能指令
  - 行列扫描顺序设置            起始行、页、列地址设置
  - 显示开/关                    反白显示
  - 电源控制                    待机模式
  - 偏压比设置                  LCD 倍压设置
  - 显示对比度设置            软件复位
- ◇ 低功率省电设计 (除背光 15MA/非绝对值)
  - 正常模式 (147uA typ VDD=3.3V)
  - 待机模式 (4uA max VDD=3.3V)
- ◇ 显示对比度由内部寄存器调节 (软件调对比度)
- ◇ 占空比 1/64, 偏压比 1/9
- ◇ 工作温度为 -20°C ~ +70°C
- ◇ 存储温度为 -30°C ~ +80°C
- ◇ 视角方向为 3 点钟
- ◇ 显示颜色为蓝底白字
- ◇ 显示器驱动芯片 SC5260 (兼容 ST7567A、AIP31567A 等)

## 第三章、显示器接口定义说明

### 第一节、显示器管脚说明

引脚	名称	方向	说明
1	NC	--	空脚
2	NC	--	空脚
3	NC	--	空脚
4	NC	--	空脚
5	NC	--	空脚
6	NC	--	空脚
7	NC	--	空脚
8	GND	--	电源负端(0V)
9	IOVCC	--	电源正端
10	NC	--	空脚
11	NC	--	空脚
12	NC	--	空脚
13	CSB	I	片选信号，低电平有效。
14	RESET	I	低电平复位，复位完成后，回到高电平，液晶显示器开始工作
15	SCL	I	显示器串行时钟
16	AO	I	数据/指令寄存器选择（高为数据，低为指令）
17	NC	--	空脚
18	SDA	I/O	显示器串行数据
19	NC	--	空脚
20	NC	--	空脚
21	NC	--	空脚
22	NC	--	空脚
23	NC	--	空脚
24	NC	--	空脚
25	NC	--	空脚
26	NC	--	空脚
27	NC	--	空脚
28	VDD	--	电源正端
29	A	--	背光电源的正极
30	K	--	背光电源的负极

## 第四章、显示器的电性参数

### 第一节、直流供电参数

名称	符号	测试条件	参数范围			单位
			最小	标准	最大	
模块工作电压	VDD	-	3.2	3.3	3.4	V
背光工作电压	VLED	-	2.7	2.8	2.9	V
液晶驱动电压	VOP/VLCD	VO-XVO	7	8.8	10.5	V
I/O 输入高电平	VIH	-	2.2	-	VDD	V
I/O 输入低电平	VIL	-	-0.3	-	0.6	V
LCM 输出高电平	VOH	-	2.4	-	-	V
LCM 输出低电平	VOL	-	-	-	0.4	V
模块工作电流	IDD	=VDD	-	-	1.0	MA
模块待机电流	ID0	=VDD	-	-	10	uA
背光工作电流	ILED	=VLED	-	15	20	MA

### 第二节、级限参数

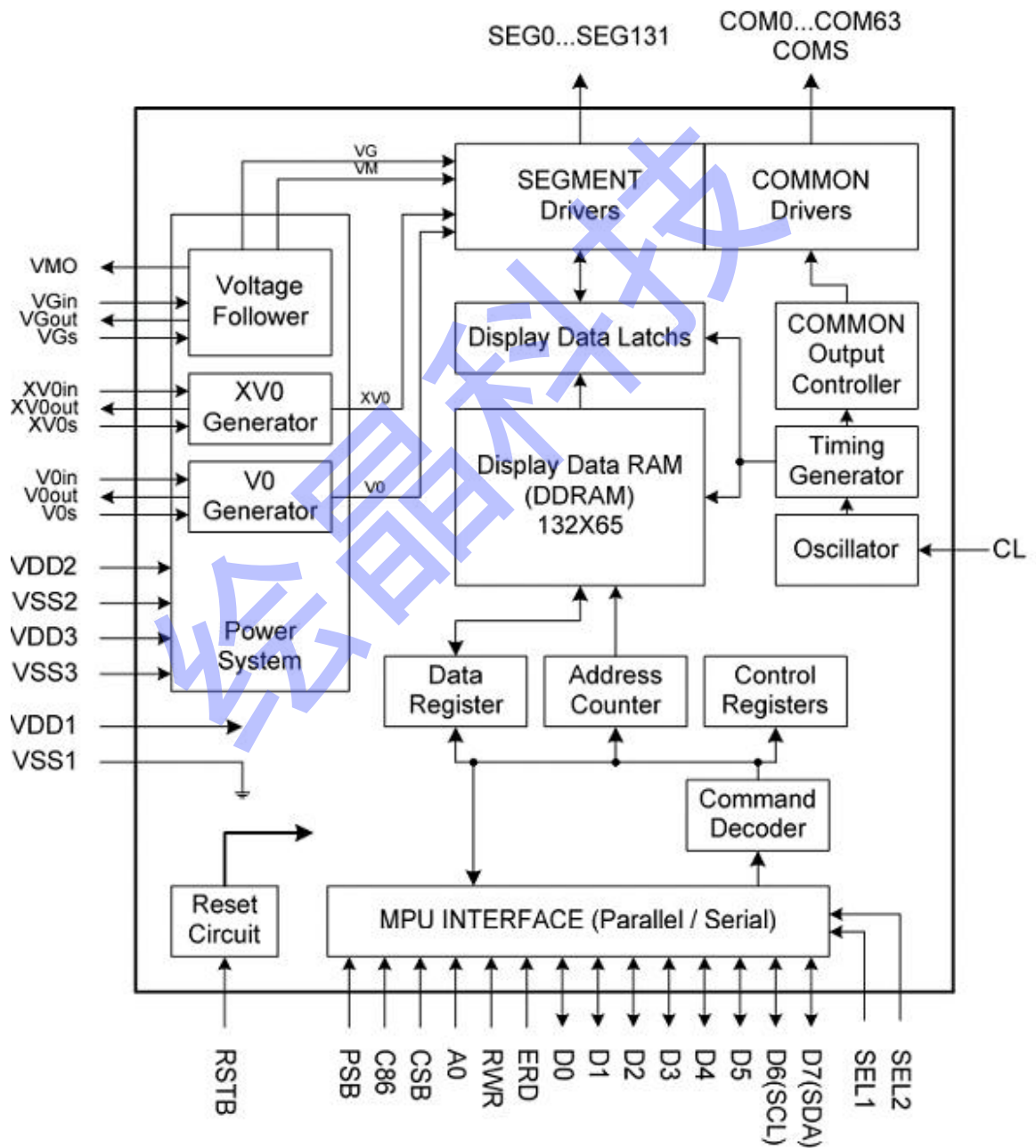
Parameter	Sysbol	Conditions	Uint
Digital power supply voltage	Vdd	-0.3~3.6	V
Analog power supply voltage	Vdd2	-0.3~3.6	V
LCD power supply voltage	VO-XVO	-0.3~13.5	V
LCD power supply voltage	VG	-0.3~3.3	V
Operating temperature	Topr	-20 to +70	°C
Storage temperature	Tstr	-30 to +80	°C

#### Notes

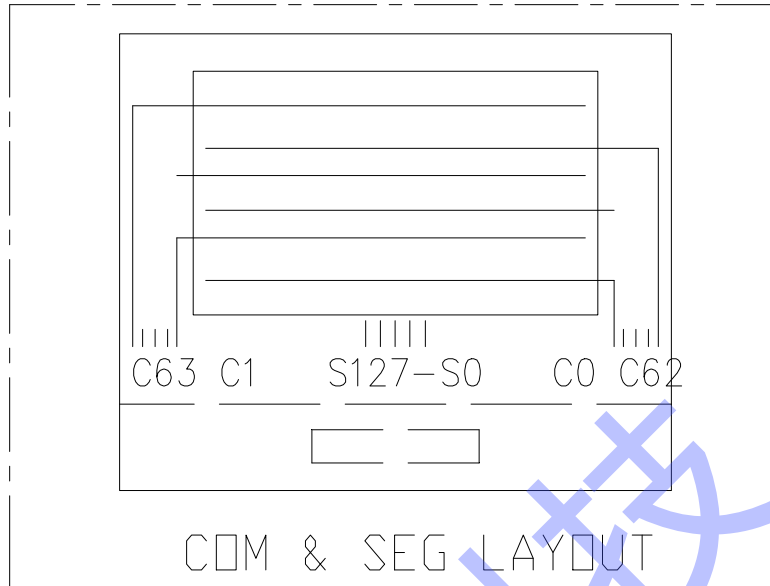
1. Stresses above those listed under Limiting Values may cause permanent damage to the device.
2. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to VSS unless otherwise noted.

## 第五章、显示器的显示结构原理

### 第一节、 显示器控制器方框图



## 第二节、 玻璃走线映射图

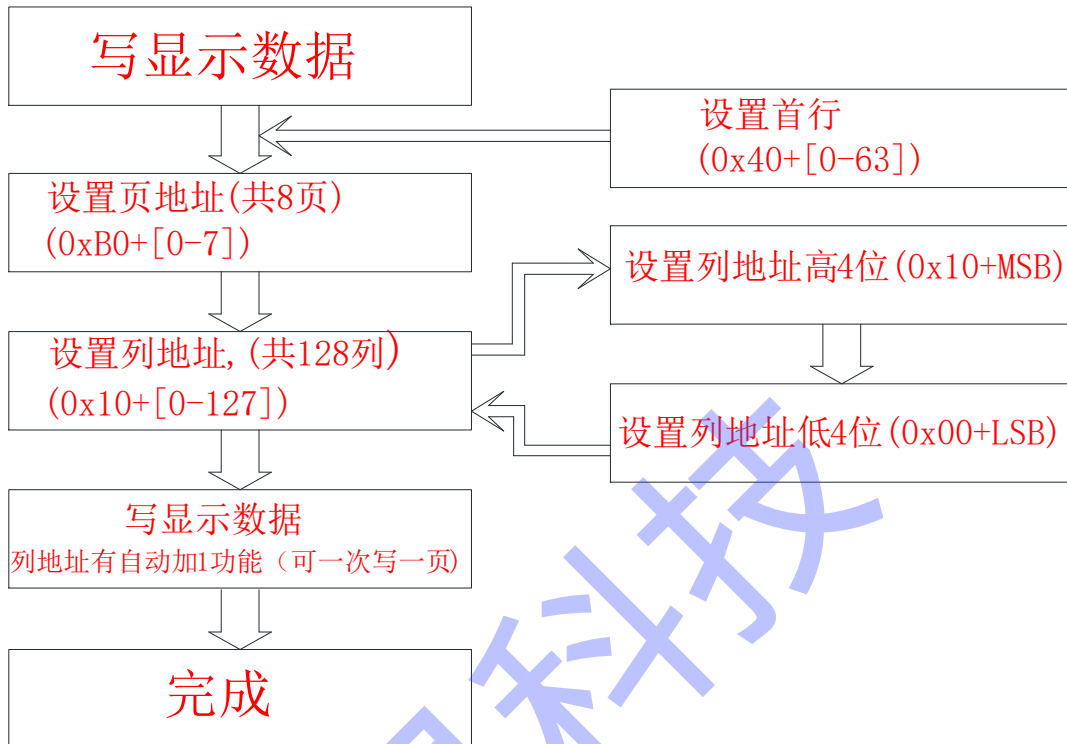


## 第三节、 显示内存映射图

Y =	0-127列										行号
	0	1	...	62	63	64	65	...	126	127	
X=0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0(63)
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	7
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	8
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	55
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	63(0)

这是一款 128\*64 的纯图形点阵屏，地址结构如图，列地址范围以 1 个像素单位定义为 (0-127) 列，行地址范围以 8 个像素为一个单位定义为 (0-7) 页，首行地址范围为一个像素单位，设置范围为 (0-63)

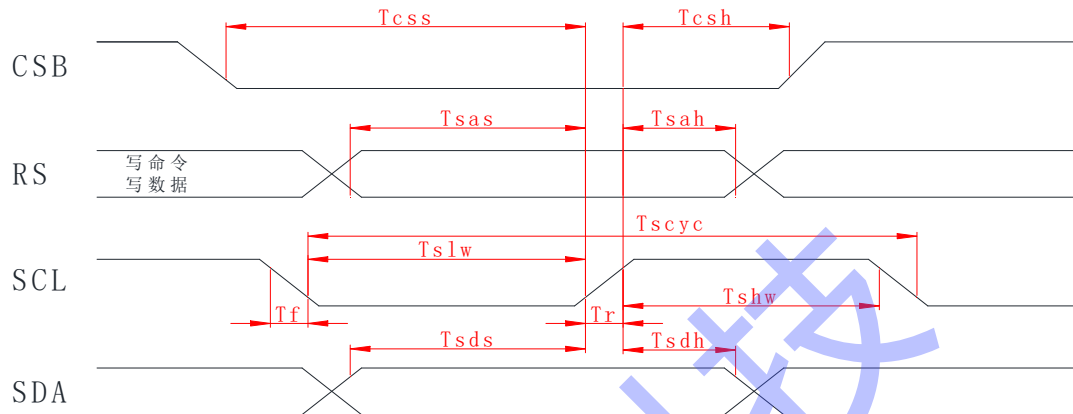
## 第四节、 写入数据流程图





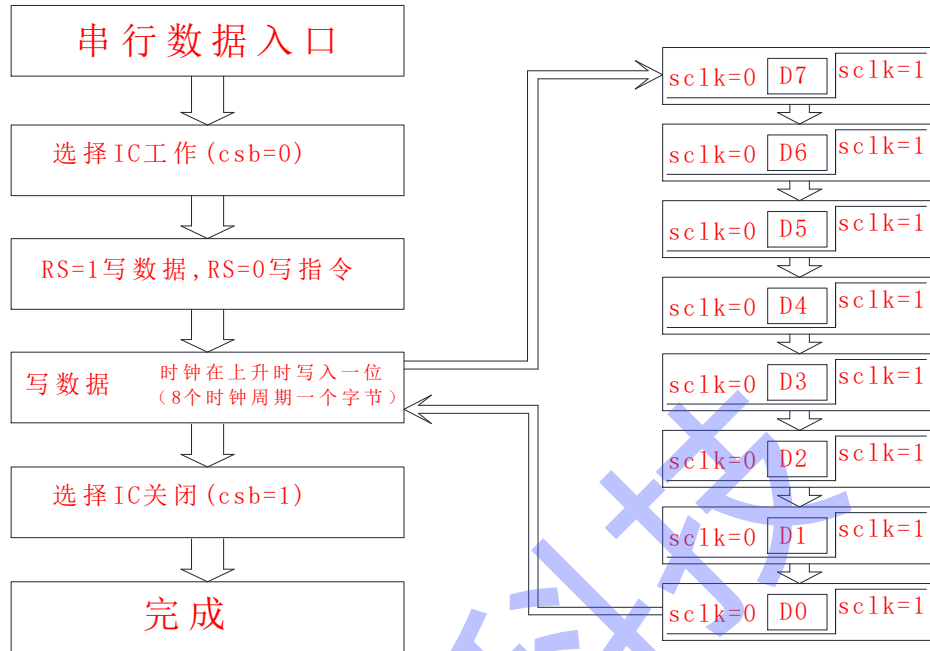
## 第六章、驱动程序时序图说明

### 第一节、 模块串行接口时序图

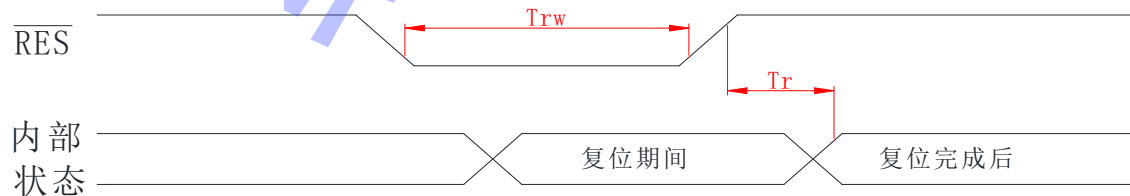


项目	信号	标识	条件	最小	最大	单位
串行时钟	SCL	Tscyc		50	-	纳秒
时钟高电平宽度		Tshw		25	-	
时钟高电平宽度		Tslw		25	-	
指令建立时间	RS	Tsas		20	-	
指令宽度		Tsah		10	-	
数据建立时间	SDA	Tsds		20	-	
数据宽度		Tsdh		10	-	
片选建立时间	CSB	Tcss		20	-	
片选保持时间		Tcsh		40	-	

## 第二节、 写串口时序的流程图



## 第三节、 写复位时序



项目	符号	测试条件	范围			单位
			最小	最佳	最大	
复位保持低电平时间	$T_{rw}$	引脚: RES	-	-	3.0	us
复位到内部状态延时	$T_r$	引脚和IC内部状态	3.0	-	-	us

## 第七章、驱动程序的指令说明

### 第一节、显示模块指令表：

NO	指令	指令码										HEX	说明
		A0	RWR	D7	D6	D5	D4	D3	D2	D1	D0		
1	显示开/关	0	0	1	0	1	0	1	1	1	D	AF	D=1, 显示开 D=0, 显示关
2	起始行设置 (COM0)	0	0	0	1	S5	S4	S3	S2	S1	S0	40	设置显示屏第一行的位置
3	页地址设置	0	0	1	0	1	1	Y3	Y2	Y1	Y0	B0	设置页地址
4	列地址设置	0	0	0	0	0	1	X7	X6	X5	X4	10	列地址高4位 (MSB)
		0	0	0	0	0	0	X3	X2	X1	X0	00	列地址低4位 (LSB)
5	读状态	0	1	BUS Y	MX	D	RST	0	0	0	0		读出模块的IC内部工作状态
6	写数据	1	0	D7	D6	D5	D4	D3	D2	D1	D0	00	向模块内写显示数据
7	读数据	1	1	D7	D6	D5	D4	D3	D2	D1	D0	00	读出模块IC的数据
8	列方向设置 (SEG)	0	0	1	0	1	0	0	0	0	MX	A0	设置列的扫描方向 MX=1, 反方向 (右到左) MX=0, 正方向 (左到右)
9	显示方向	0	0	1	0	1	0	0	1	1	INV	A6	INV=1, 反向显示 INV=0, 正常显示
10	全部点阵打开	0	0	1	0	1	0	0	1	0	AP	A4	AP=1, 点阵全部打开 AP=0, 正常
11	偏压选择	0	0	1	0	1	0	0	0	1	BS	A2	偏压比设置 0=1/9; 1=1/7 (1/64duty)
12	Read-modify-Write	0	0	1	1	1	0	0	0	0	0	E0	Columnaddress increment: Read:+0, Write:+1
13	END	0	0	1	1	1	0	1	1	1	0	EE	Exit Read-modify-Write mode
14	复位	0	0	1	1	1	0	0	0	1	0	E2	软件复位
15	行方向 (COM)	0	0	1	1	0	0	MY	-	-	-	C0	设置行方向 MY=1, 反方向 (下到上) MY=0, 正方向 (上到下)
16	LCD电源控制	0	0	0	0	1	0	1	VB	VR	VF	28	供应LCD电压的电路控制 =1, 打开; =0, 关闭
17	Regulation Ratio	0	0	0	0	1	0	0	RR2	RR1	RR0	20	选择调节电阻比
18	LCD电压设置 (对比度)	0	0	1	0	0	0	0	0	0	1	81	双指令模式, 6位精度调节LCD 电压 (显示双比度)
		0	0	0	0	EV5	EV4	EV3	EV2	EV1	EVO	00	
19	睡眠模式设置	0	0	1	0	1	0	1	1	0	MD	AC	MD=0, 睡眠模式 MD=1, 正常模式
		0	0	0	0	0	0	0	0	0	0	00	
20	省电	0	0	复合指令									显示关+全部点阵开
21	设置升压倍数 (LCD 电压)	0	0	1	1	1	1	1	0	0	0	F8	双指令模式 设置升压倍数 BL (0 0) 2倍, 3倍, 4倍 BL (0 1) 5倍 BL (1 0) 6倍
		0	0	0	0	0	0	0	0	BL1	BLO	00	
22	空指令	0	0	1	1	1	0	0	0	1	1	E3	空操作
23	测试	0	0	1	1	1	1	-	-	-	-	F0	不要使用, 保留测试

## 第二节、显示模块指令详细说明

### 一、显示开关设置

D 标志选择显示模式

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	1	0	1	1	1	D	AE

D=1; 正常显示模式

D=0; 显示关闭, 所以SEG/COM输出低电平

### 二、起始行设置

显示内存的第一行对应显示屏顶部的第一行地址设置

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	1	S5	S4	S3	S2	S1	S0	40

### 三、面地址设置

128\*8点阵为一页,  $12864 = (1286 * 8) * 8$ 页, 与起始行决定行在显示屏上的位置

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	1	1	Y3	Y2	Y1	Y0	B0

### 四、列地址设置

设置列地址

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	0	1	X7	X6	X5	X4	10

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	0	0	X3	X2	X1	X0	00

第1列为	0x10(高4位)	0x00(低4位)	列地址需要写两次
第2列为	0x10(高4位)	0x01(低4位)	
第128列为	0x17(高4位)	0x0F(低4位)	

## 五、读状态设置

读显示器内部状态。串行接口模式读取功能不可用。

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	BUSY	MX	D	RST	0	0	0	0	

标志	描述
BUSY	BUSY= 0: 命令可以接受 BUSY= 1: 命令或复位程序执行
MX	MX=0; 反向 (SEG131--SEG0) MX=1; 正常 (SEG0--SEG131)
D	D=0; 显示开 D=1; 显示关
RST	RST=1在复位 (硬件或软件复位) RES=0; 正常运行

## 六、电源控制设置

一般3个标志位同时打开

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	1	0	1	VB	VR	VF	28

标志	描述
VB	VB=0; 内置升压关闭 VB=1; 内置升压打开
VR	VR=0; 内置稳压关闭 VR=1; 内置稳压打开
VF	VF=0; 内置降压关闭 VF=1; 内置降压打开

## 七、调节对比度设置

LCD 对比度档位选择

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	1	0	0	RR2	RR1	RR0	20
	标志		调节比								
	0x20		3.0								
	0x21		3.5								
	0x22		4.0								
	0x23		4.5								
	0x24		5.0								
	0x25		5.5								
	0x26		6.0								
	0x27		6.5								

工作电压 (V0) 的计算公式如下: (RR 为调节比, EV 为 LCD 电压设置)

$$V0 = RR \times [1 - (63 - EV) / 162] \times 2.1, \text{ 或 } V0 = RR \times [(99 + EV) / 162] \times 2.1$$

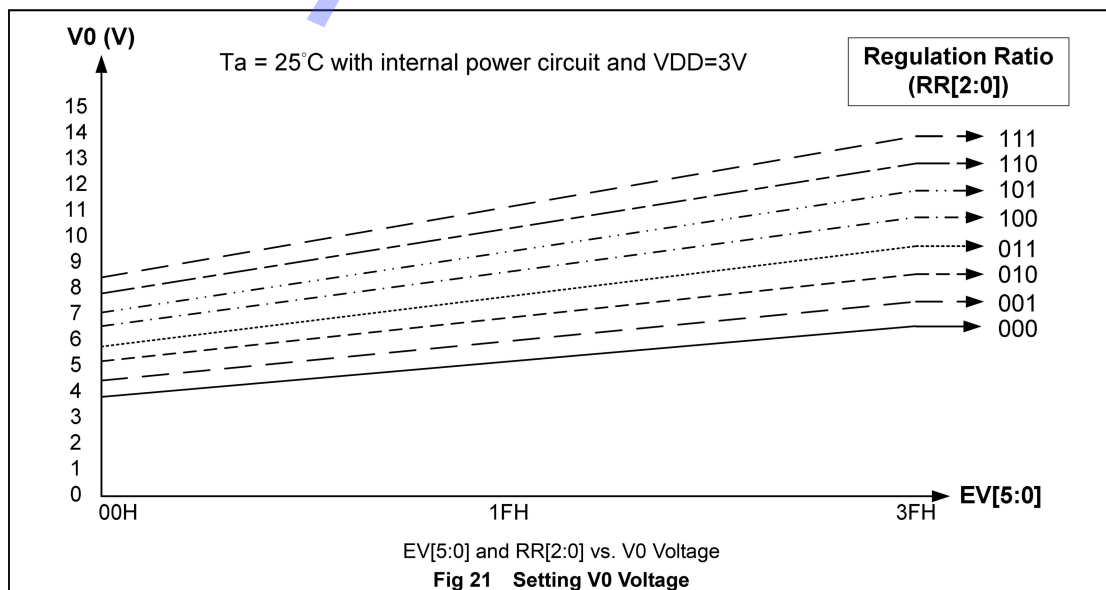
标志	寄存器	数值
RR	RR[2: 0]	3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
EV	EV[5: 0]	0-63

## 八、玻璃微调比度设置

双字节指令, 两条指令要一起使用可以调节LCD的深浅显示

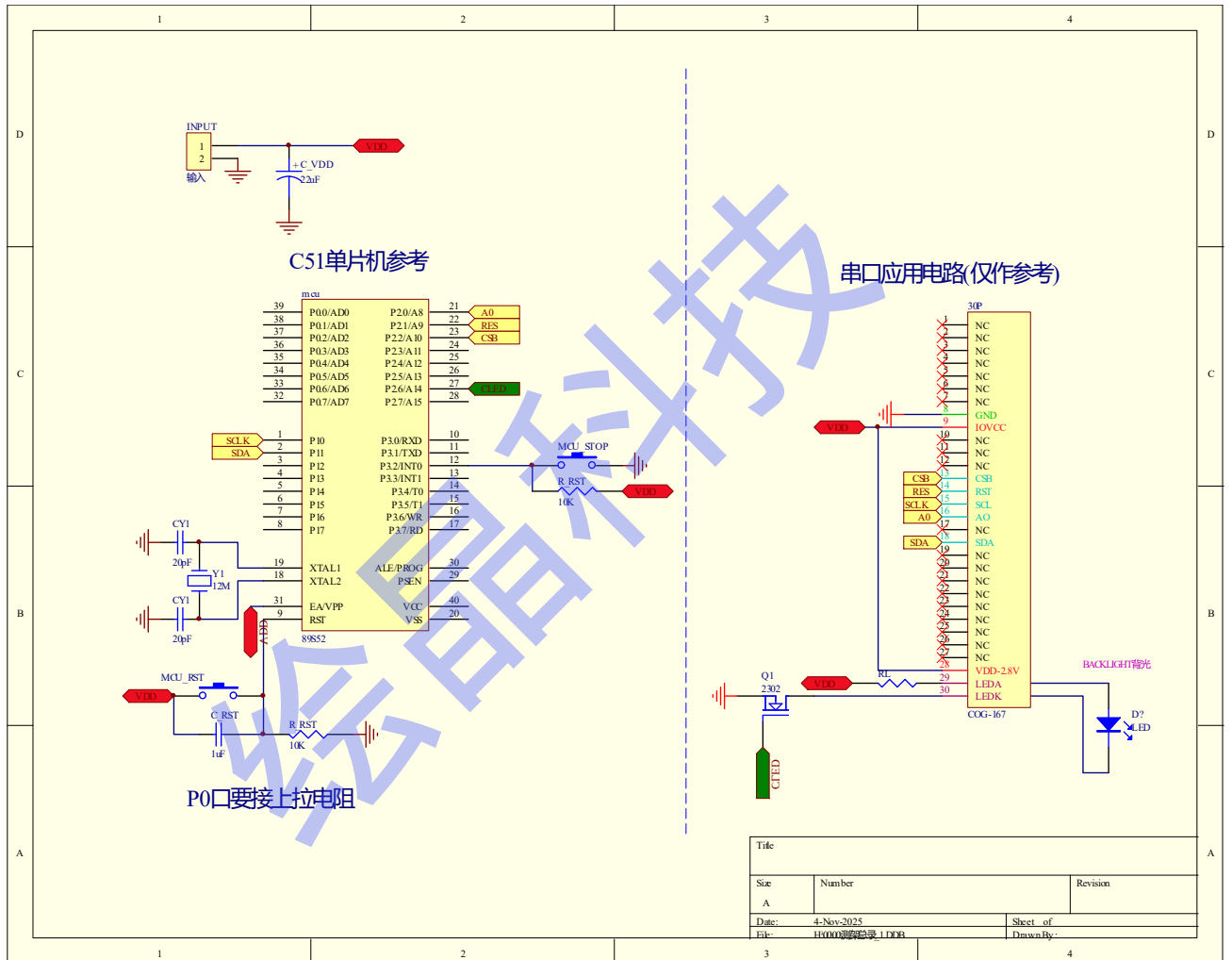
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	0	0	0	0	0	1	81
	0	0	0	0	EV5	EV4	EV3	EV2	EV1	EV0	0

LCD 电压与 EV 和 RR 之前的关系图



## 第八章、单片机与显示器连接说明

### 第一节、接口串口应用原理图



## 第九章、单片机驱动程序源代码

### 第一节、源代码解释定义声明

```
//串口驱动程序
//产品内核在SC5260（兼容ST7567A，AIP31567A等）（兼容ST7565R）
//MCU :ATMEL 89S52（STC完全兼容）
//

#include <reg52.H>
#include <intrins.h>

//显示I/O口定义
sbit lcd_sclk=P1^0; //对应显示器SCLK
sbit lcd_sid =P1^1; //对应显示器SDA
sbit rs      =P2^0; //对应显示器RS
sbit reset  =P2^1; //对应显示器RST
sbit cs1    =P2^2; //对应显示器CS0

//数据指针
#define uchar unsigned char
#define uint unsigned int

//定义变量
uchar ml,z,zl,d,d1,s,s1,s10,s100;

//函数声明
uchar code ascii_table_8x16[95][16]; //95个标准8*16的字符码
uchar code ascii_table_5x8[95][5]; //95个标准5*8的字符码
uchar code hui1[]; //取模16*16汉字”绘“
uchar code jing1[]; //取模16*16汉字”晶“
uchar code ke1[]; //取模16*16汉字”科“
uchar code ji1[]; //取模16*16汉字”技“

//调用中断函数，单片机P3.2低电平时程序暂停
void ini_int1(void)
{
    EA=1;
    EX0=1; //允许外部INT0的中断
    IT0=1; //允许中断
}
```



```
}  
  
int scankey1() interrupt 0 using 1 //使用外部中断1,寄存器组3  
{  
    while(P3^2==0){for(;;);}  
    IE1=0;//中断标志清零  
}
```

## 第二节、接口时序函数

```
//=====transfer command to LCM=====串口
```

```
void transfer_command(int data1)
```

```
{  
    char i;  
    cs1=0;  
    rs=0;  
    for(i=0;i<8;i++)  
    {  
        lcd_sclk=0;  
        //delay_us(10); //加少量延时  
        if(data1&0x80) lcd_sid=1;  
        else lcd_sid=0;  
        lcd_sclk=1;  
        //delay_us(10); //加少量延时  
        data1=data1<<=1;  
    }  
    cs1=1;  
}
```

```
//-----transfer data to LCM-----
```

```
void transfer_data(int data1)
```

```
{  
    char i;  
  
    cs1=0;  
    rs=1;  
    for(i=0;i<8;i++)  
    {  
        lcd_sclk=0;
```

```
        if(data1&0x80) lcd_sid=1;
        else lcd_sid=0;
        lcd_sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

//=====transfer command to LCM=====

/*延时*/
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<110;k++);
}
```

### 第三节、液晶模块初始化

```
/*LCD 模块初始化*/
void initial_lcd()
{
    cs1=0;
    reset=0; /*低电平复位*/
    delay(20);
    reset=1; /*复位完毕*/
    delay(20);
    transfer_command(0xe2); /*软复位*/
    delay(50);
    transfer_command(0x2c); /*升压步聚1*/
    delay(50);
    transfer_command(0x2e); /*升压步聚2*/
    delay(50);
    transfer_command(0x2f); /*升压步聚3*/
    delay(50);
    transfer_command(0x24); /*粗调对比度, 可设置范围0x20~0x27*/
    transfer_command(0x81); /*微调对比度*/
    transfer_command(0x20); /*0x1a, 微调对比度的值, 可设置范围0x00~0x3f 1f*/
    transfer_command(0xa2); /*1/9 偏压比 (bias) */
    transfer_command(0xc0); /*行扫描顺序: 从上到下*/
    transfer_command(0xa0); /*列扫描顺序: 从左到右*/
}
```

```
transfer_command(0x40); /*起始行：第一行开始*/
transfer_command(0xaf); /*开显示*/
cs1=1;
}

void lcd_address(uchar page,uchar column)
{
    cs1=0;
    column=column; //
    page=page-1;
    transfer_command(0xb0+page);
    transfer_command(((column>>4)&0x0f)+0x10); //设置列地址的高4 位
    transfer_command(column&0x0f); //设置列地址的低4 位
}
```

## 第四节、应用函数

```
/*全屏清屏*/
void clear_screen()
{
    unsigned char i,j;
    cs1=0;
    for(i=0;i<9;i++)
    {
        lcd_address(1+i,0);
        for(j=0;j<132;j++)
        {
            transfer_data(0x00);
        }
    }
    cs1=1;
}

void test_display(uchar data1,uchar data2)
{
    int i,j;
    for(j=0;j<8;j++)
    {
```

```
        cs1=0;
        lcd_address(j+1,0); //页, 列地址
        for(i=0; i<64; i++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}

void display_string_8x16(uchar fb, uint page, uint column, uchar *text)
{
    uint i=0, j, k, n;
    cs1=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;

            for(n=0; n<2; n++)
            {
                lcd_address(page+n, column);
                for(k=0; k<8; k++)
                {
                    if (fb==1) transfer_data(ascii_table_8x16[j][k+8*n]);
                    else transfer_data(~(ascii_table_8x16[j][k+8*n]));
                }
            }
            i++;
            column+=8;
        }
        else
            i++;
    }
}

void display_string_8x16_t(uchar fb, uint page, uint column, uchar text)
{
    uint j, k, n;
    j=text+16;
```

```
for(n=0;n<2;n++)
{
    lcd_address(page+n,column);
    for(k=0;k<8;k++)
    {
        if(fb==1) transfer_data(ascii_table_8x16[j][k+8*n]);
        else transfer_data(~(ascii_table_8x16[j][k+8*n]));
    }
}

void display_string_5x8(uint page,uint column,uchar *text)
{
    uint i=0,j,k;
    cs1=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<0x7e))
        {
            j=text[i]-0x20;
            lcd_address(page,column);
            for(k=0;k<5;k++)
            {
                transfer_data(ascii_table_5x8[j][k]);
            }
            i++;
            column+=6;
        }
        else
            i++;
    }
}

void display_graphic_16x16(uchar fb,uchar page,uchar column,uchar *dp)
{
    uint i,j;
    for(j=0;j<2;j++)
```

```
{  
  lcd_address(page+j,column);  
  for (i=0;i<16;i++)  
  {  
    if(fb==1) transfer_data(*dp);  
    else transfer_data(~(*dp));  
    dp++;  
  }  
}  
}
```

//显示128x64 点阵边框

```
void display_bk()  
{  
  uint i,j;  
  
  //左框  
  for(j=0;j<8;j++)  
  {  
    lcd_address(j+1,0);  
    transfer_data(0xff);  
  }  
  
  //右框  
  for(j=0;j<8;j++)  
  {  
    lcd_address(j+1,127);  
    transfer_data(0xff);  
  }  
  
  //上框  
  lcd_address(1,1);  
  for (i=0;i<126;i++)  
  {  
    transfer_data(0x01);  
  }  
  
  //下框  
  lcd_address(8,1);  
  for (i=0;i<126;i++)  
  {
```

```
transfer_data(0x80);  
}  
  
}
```

## 第五节、主调用函数

```
//~~~~~@~~~~&~~~~@~~~~程序开始  
void main(void)  
{  
    ini_int1();//开中断  
    for(m1=0;m1<2;m1++)  
    {  
  
        initial_lcd();//去初始化显示模块的寄存器  
  
        clear_screen();//clear all dots//显示屏清屏  
  
        test_display(0xaa,0xaa);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~1屏,横  
        test_display(0x55,0x55);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~2屏,横  
        test_display(0xff,0x00);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~3屏,竖  
        test_display(0x00,0xff);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~4屏,竖  
        test_display(0x55,0xaa);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~5屏,点  
        test_display(0xaa,0x55);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~6屏,点  
        test_display(0xff,0xff);  
        delay(700);  
        //~~~~~@~~~~&~~~~@~~~~7屏,黑
```

```
clear_screen(); //clear all dots
display_string_8x16(0,1,8, " HUIJINGKEJI");
display_string_8x16(1,3,1, "WELCOME TO HUIJI");
display_string_5x8(5,1, "TEL: 0755-23146001");
display_string_5x8(6,1, "FAX: 0755-23146002");
display_string_5x8(7,1, "www.huijinglcm.com");
display_string_5x8(8,1, "huijinglcm@sina.com");
delay(700);
//~~~~~@~~~~&~~~~~@~~~~~8屏, 自建字符演示
```

```
clear_screen(); //clear all dots
display_bk();
display_graphic_16x16(1,3,33, hui1);
display_graphic_16x16(1,3,49, jing1);
display_graphic_16x16(1,3,65, ke1);
display_graphic_16x16(1,3,81, ji1);
display_string_8x16(1,5,40, "128*64");
delay(700);
//~~~~~@~~~~&~~~~~@~~~~~9屏, 自建汉字图形
```

```
clear_screen(); //clear all dots

transfer_command(0xa7); //反显
display_graphic_16x16(1,3,33, hui1);
display_graphic_16x16(1,3,49, jing1);
display_graphic_16x16(1,3,65, ke1);
display_graphic_16x16(1,3,81, ji1);
display_string_8x16(1,5,40, "128*64");
delay(700);
//~~~~~@~~~~&~~~~~@~~~~~10屏, 反白演示
```

```
clear_screen(); //clear all dots
transfer_command(0xa6);
}
```

## 第六节、动态显示函数

```
initial_lcd();
clear_screen(); //clear all dots
display_graphic_16x16(0,1,0, hui1);
display_graphic_16x16(0,3,0, jing1);
display_string_8x16(0,6,0, "TEL: 755-23146001");
display_string_5x8(8,4, ". WWW. HUIJINGLCM. COM. ");
```



```
display_string_5x8(1,40,"~~~~~");
display_string_5x8(4,40,"~~~~~");
for(z=0;z<10;z++)
{
    display_string_8x16_t(0,2,40,z);
    for(z1=0;z1<10;z1++)
    {
        display_string_8x16_t(0,2,48,z1);
        for(d=0;d<6;d++)
        {
            display_string_8x16_t(0,2,56,10);
            display_string_8x16_t(0,2,64,d);
            for(d1=0;d1<10;d1++)
            {
                display_string_8x16_t(0,2,72,d1);
                for(s=0;s<6;s++)
                {
                    display_string_8x16_t(0,2,80,10);
                    display_string_8x16_t(0,2,88,s);
                    for(s1=0;s1<10;s1++)
                    {
                        display_string_8x16_t(0,2,96,s1);
                        delay(10);////延时 x ms
                        display_string_8x16_t(0,2,104,10);
                        for(s10=0;s10<10;s10++)
                        {
                            display_string_8x16_t(0,2,112,s10);
                            delay(9);////延时 x ms
                            for(s100=0;s100<9;s100++)
                            {
                                display_string_8x16_t(0,2,120,s100);
                                delay(9);////延时 x ms
                            }
                        }
                    }
                }
            }
        }
    }
}
//~~~~~@&~~~~~11屏，动态变量时钟演示
```

}

## 第七节、标准字符数据表

uchar code ascii\_table\_8x16[95][16]={

因篇太长，有需要的用户和业务联系

## 第十章、版本信息

更新日期	更新内容说明	版本
2025 11 04	第一版	V1.0