

JLX12832G-040-BN 使用说明书

目 录

序号	内 容 标 题	页码
1	概述	2
2	特点	2
3	外形及接口引脚功能	3~4
4	基本原理	4~5
5	技术参数	6
6	时序特性	6~8
7	指令功能及硬件接口与编程案例	8~末页

1. 概述

晶联讯电子专注于液晶屏及液晶模块的研发、制造。所生产 JLX12832G-040 型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

JLX12832G-040 可以显示 128 列*32 行点阵单色图片，或显示 8 个/行*2 行 16*16 点阵的汉字，或显示 16 个/行*4 行 8*8 点阵的英文、数字、符号。

2. JLX12832G-040 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙。插接式 FPC

2.2 IC 采用矽创公司 ST7567, 功能强大，稳定性好

2.3 功耗低:10 - 100mW (不带背光 10mW, 带背光不大于 100mW) ;

2.4 显示内容:

- 128*64 点阵单色图片;

- 可選用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 8 字*4 行。

- 按照 12*12 点阵汉字来计算可显示 10 字/行*2 行。

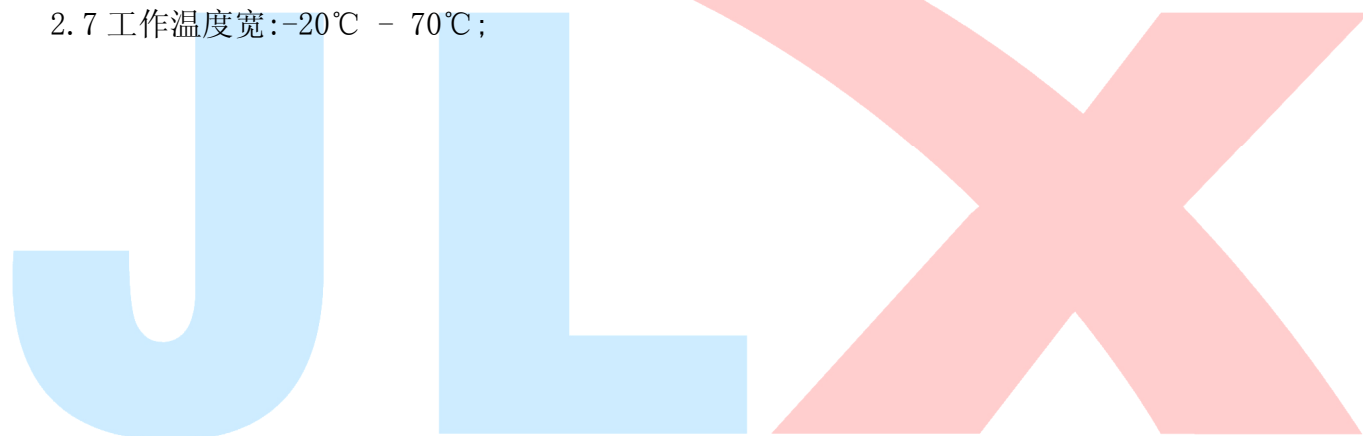
- 可显示 16 个*2 行 8*16 点阵的英文、数字、符号。

- 可显示 21 个*4 行 5*8 点阵的英文、数字、符号。

2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改 (可旋转 180 度使用)。

2.6 接口简单方便:采用 4 线 SPI 串口。

2.7 工作温度宽:-20℃ - 70℃;



3. 外形尺寸及接口引脚功能

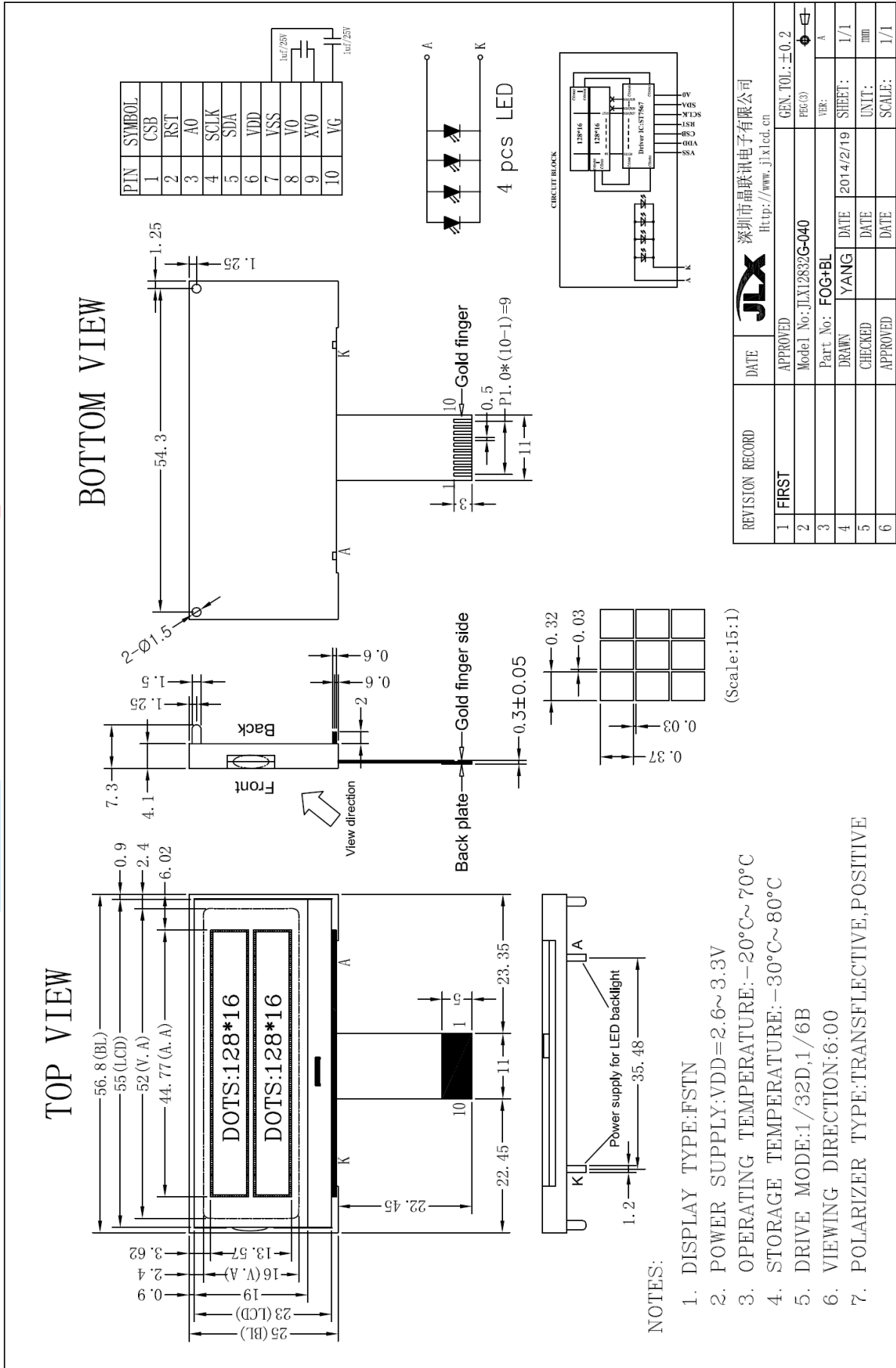


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	CS1	片选	低电平片选
2	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶模块开始工作
3	A0	寄存器选择信号	H: 数据寄存器 0: 指令寄存器
4	SCLK	I/O	串行时钟
5	SDA	I/O	串行数据
6	VDD	电路电源	3.3V
7	VSS	接地	0V
8	V0	倍压电路	两脚之间接一个 1uF 电容
9	XV0	倍压电路	
10	VG	偏置电压	LCD 驱动偏置电压。与 VSS 之间接一个 1uF 电容

表 1: 模块的接口引脚功能

4. 基本原理

4.1 液晶屏 (LCD)

在 LCD 上排列着 128×32 点阵, 128 个列信号与驱动 IC 相连, 32 个行信号也与驱动 IC 相连, IC 邦定在 LCD 玻璃上 (这种加工工艺叫 COG)。

4.2 工作电路图:

图 2 是 JLX12832G-040 图像点阵型模块的电路框图, 它由驱动 IC ST7567 及几个电阻电容组成。

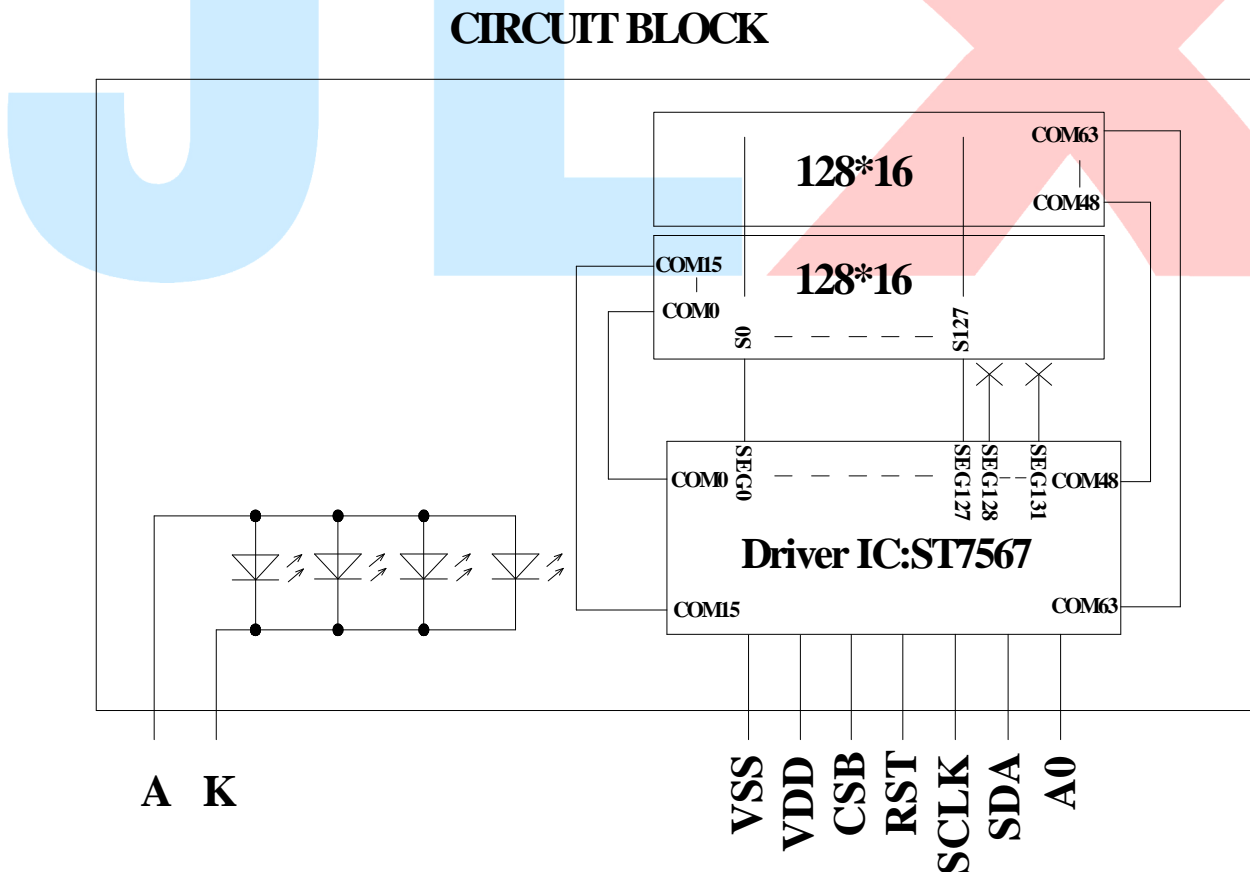


图 2: JLX12832G-040 图像点阵型液晶模块的电路框图

4.2 升压电路图：

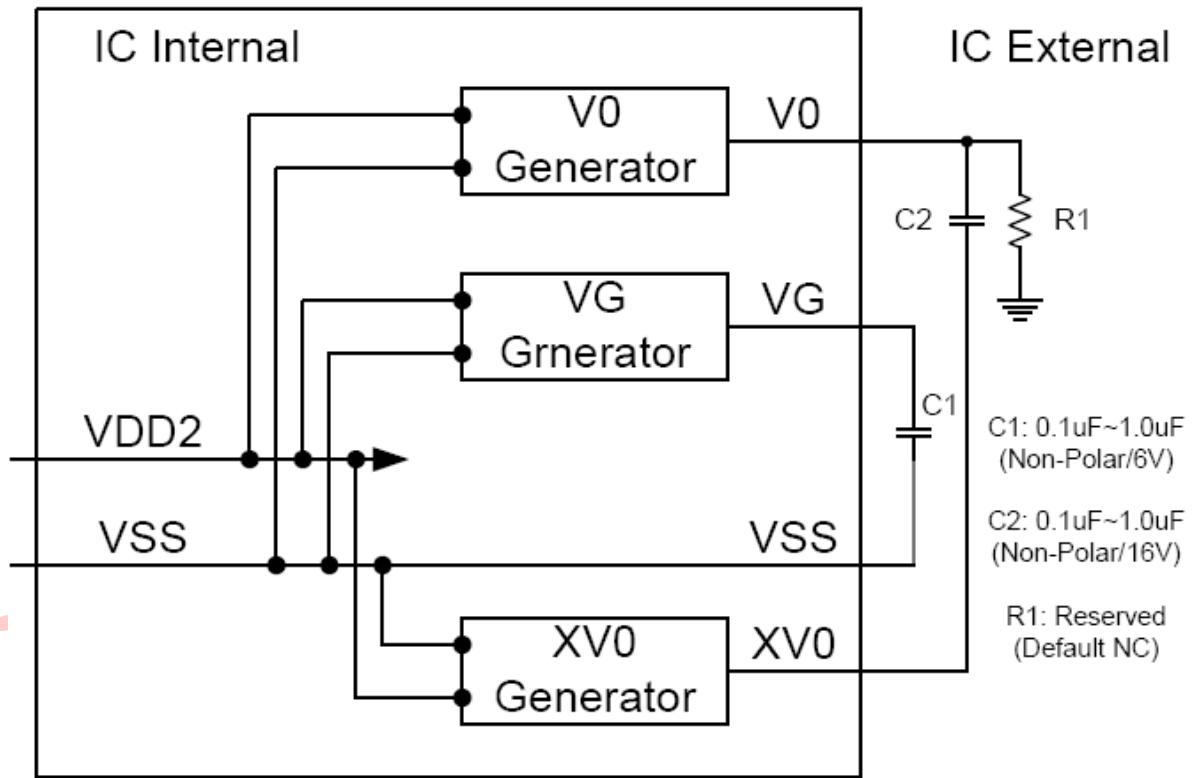


Fig 11. Power Circuit

图 3：3 倍升压电路图

4.3 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下：
 工作温度： $-20\sim+70^{\circ}\text{C}$ ；
 存储温度： $-30\sim+80^{\circ}\text{C}$ ；
 正常工作电流为： $40\sim 80\text{mA}$ （LED 灯数共 4 颗）；
 工作电压： 3.0V ；

5. 技术参数

5.1 最大极限参数 (超过极限参数则会损坏液晶模块)

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3	-0.3~3.6		V
LCD 驱动电压	V0 - XV0		-0.3~16		V
LCD 数字电压	VG		-0.3~3.6		V
工作温度		-20		+70	°C
储存温度		-30		+80	°C

表 2: 最大极限参数

5.2 直流 (DC) 参数

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
工作电压	VDD		2.4	-	3.3	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIO	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOO	I00 = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.0V	-		1.0	mA
背光工作电流	ILED	VLED=3.0V	50	75	100	mA

表 3: 直流 (DC) 参数

6. 读写时序特性

6.1 串行接口:

从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

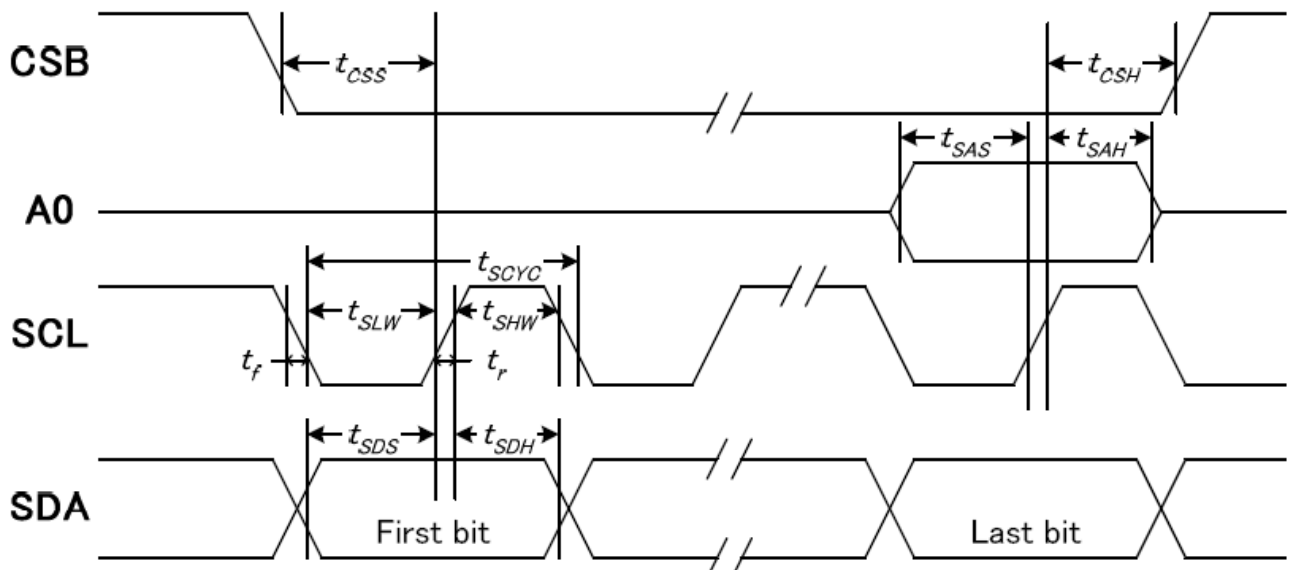


图 4. 从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

6.2 串行接口: 时序要求 (AC 参数):

写数据到 ST7567 的时序要求:

表 4.

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
4线 SPI串口时钟周期 (4-line SPI Clock Period)	T _{scyc}	引脚: SCK	50	--	--	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T _{shw}	引脚: SCK	25			ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T _{slw}	引脚: SCK	25			ns
地址建立时间 (Address setup time)	T _{sas}	引脚: RS	20	--	--	ns
地址保持时间 (Address hold time)	T _{sah}	引脚: RS	10	--	--	ns
数据建立时间 (Data setup time)	T _{sds}	引脚: SI	20	--	--	ns
数据保持时间 (Data hold time)	T _{sdh}	引脚: SI	10	--	--	ns
片选信号建立时间 (CS-SCL time)	T _{css}	引脚: CS	20			ns
片选信号保持时间 (CS-SCL time)	T _{csh}	引脚: CS	40			ns

VDD = 3.0V ± 5%, Ta = 25°C

6.5 电源启动后复位的时序要求 (RESET CONDITION AFTER POWER UP):

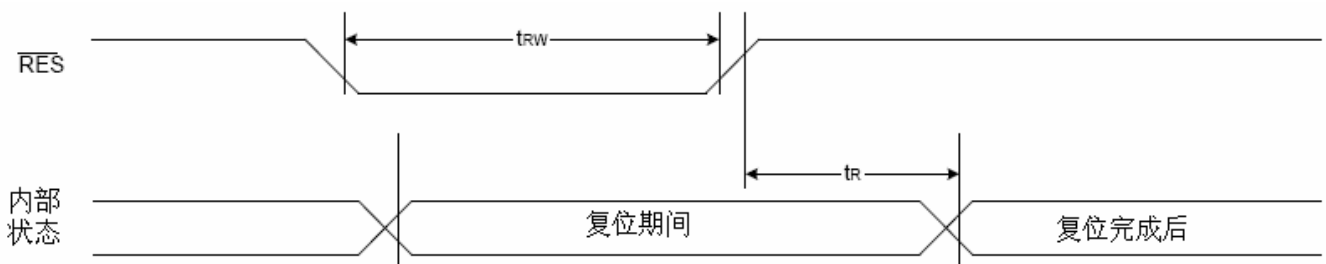


图 7: 电源启动后复位的时序

表 6: 电源启动后复位的时序要求

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	tr		--	--	1.0	us
复位保持低电平的时间	trw	引脚: RES	1.0	--	--	us

7. 指令功能:

7.1 指令表

指 令 表
表 8.

指令名称	指 令 码									说 明	
	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
(1)显示开/关 (display on/off)	0	1	0	1	0	1	1	1	0 1	显示开/关: 0XAE :关, 0XAF : 开	
(2)显示初始行设置 (Display start line set)	0	0	1	显示初始行地址, 共 6 位						设置显示存储器的显示初始行,可设置值为 0X40~0X7F ,分别代表第 0~63 行, 针对该液晶屏一般设置为 0x60	
(3)页地址设置 (Page address set)	0	1	0	1	1	显示页地址, 共 4 位				设置页地址。每 8 行为一个页, 64 行分为 8 个页, 可设置值为: 0XB0~0XB8 分别对应第一页到第九页, 第九页是一个单独的一行图标, 本液晶屏没有这一行图标, 所以设置值为 0XB0~0XB7 分别对应第一页~第八页。	
(4) 列地址高4位设置 列地址低4位设置	0	0	0	0	1	列地址的高 4 位			高 4 位与低 4 位共同组成列地址, 指定 128 列中的其中一列。比如液晶模块的第 101 列地址十六进制为 0x64 , 那么此指令由 2 个字节来表达: 0x16, 0x04		
		0	0	0	0	列地址的低 4 位					
(5) 读状态 (Status read)	0	状态			0	0	0	0		并口时: 读驱动 IC 的当前状态,串口时不能用此指令	
(6)写显示数据到液晶屏 (Display data write)	1	8 位显示数据									从 CPU 写数据到液晶屏, 每一位对应一个点阵, 1 个字节对应 8 个竖置的点阵
(7)读液晶屏的显示数据 (Display data read)	1	8 位显示数据									并口时: 读已经显示到液晶屏上的点阵数据。串口时不能用此指令
(8) 显示列地址增减 (ADC select)		1	0	1	0	0	0	0	0 1	显示列地址增减: 0xA0 : 常规: 列地址从左到右, 0xA1 : 反转: 列地址从右到左	
(9)显示正显/反显 (Display normal/reverse)	0	1	0	1	0	0	1	1	0 1	显示正显/反显: 0xA6 : 常规: 正显 0xA7 : 反显	
(10)显示全部点阵 (Display all points)	0	1	0	1	0	0	1	0	0 1	显示全部点阵: 0xA4 : 常规 0xA5 : 显示全部点阵	
(11)LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	0 1	设置偏压比: 0XA2 : BIAS=1/9 (常用) 0XA3 : BIAS=1/7	
(12) 读-改-写	0	1	1	1	0	0	0	0	0	0XE0 : “读-改-写” 开始。	

(Read-modify-write)										本液晶模块使用串行接口, 不具备此功能。 详情请参考IC资料	
(13) 退出上述“读-改-写”指令(End)		0	1	1	1	0	1	1	1	0	0xEE:上述“读-改-写”指令结束 本液晶模块使用串行接口, 不具备此功能。 详情请参考 IC 资料
(14) 软件复位 (Reset)		0	1	1	1	0	0	0	1	0	0XE2:软件复位。
(15) 行扫描顺序选择 (Common output mode select)			1	1	0	0	0	0	0	0	1 行扫描顺序选择: 0XC0:普通扫描顺序: 从上到下 0XC8:反转扫描顺序: 从下到上
(16) 电源控制 (Power control set)			0	0	1	0	1				选择内部电压供应操作模式: D2、D1、D0 位分别对应内部升压是否打开 (1 为打开, 0 为不打开), 电压调整电路是否打开(1 为打开, 0 为不打开), 电压跟随器是否打开(1 为打开, 0 为不打开)。 通常是 0x2C,0x2E,0x2F 三条指令按顺序紧接着写, 表示依次打开内部升压、电压调整电路、电压跟随器。也可以单写 0x2F , 一次性打开三部分电路。
(17) 选择内部电阻比例		0	0	0	1	0	0				内部电压值电阻设置 选择内部电阻比例 (Rb/Ra):可以理解为 粗调 对比度值。可设置范围为: 0x20~0x27 , 数值越大对比度越浓, 越小越淡
(18)	内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调, 可以理解为 微调 对比度值, 此两个指令需紧接着使用。上面一条指令 0x81 是不改的, 下面一条指令可设置范围为: 0x00~0x3F , 数值越大对比度越浓, 越小越淡
	设置的电压值	6 位电压值数据, 0~63 共 64 级									
(19)静态图标显示: 开/关		0	1	0	1	0	1	1	0	0	1 静态图标的开关设置: 0xAC : 关, 0xAD : 开。 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)		0	1	1	1	1	1	0	0	0	2 位数设置 升压倍数 00: 2 倍, 3 倍, 4 倍 01: 5 倍 11: 6 倍。本模块外部已设置升压倍数为 4 倍, 不必使用此指令
(21) 省电模式 (Power save)											省电模式, 此非一条指令, 是由“(10)显示全部点阵”、(19)静态图标显示: 开/关等指令合成一个“省电功能”。详细看 IC 规格书第 47 页“POWER SAVE”
(22)空指令 (NOP)		0	1	1	1	0	0	0	1	1	空操作
(23) 测试 (Test)		0	1	1	1	1	*	*	*	*	内部测试用, 千万别用!

请详细参考 IC 资料“ST7567.PDF”的第 21~28 页。

7.3 点阵与 DD RAM 地址的对应关系

请留意页的定义: PAGE, 与平时所讲的“页”并不是一个意思, 在此表示 8 个行就是一个“页”, 一个 128*64 点阵的屏分为 8 个“页”, 从第 0“页”到第 7“页”。

DB7--DB0 的排列方向: 数据是从下向上排列的。最低位 D0 是在最上面, 最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵, 通常“1”代表点亮该点阵, “0”代表关掉该点阵. 如下图所示:

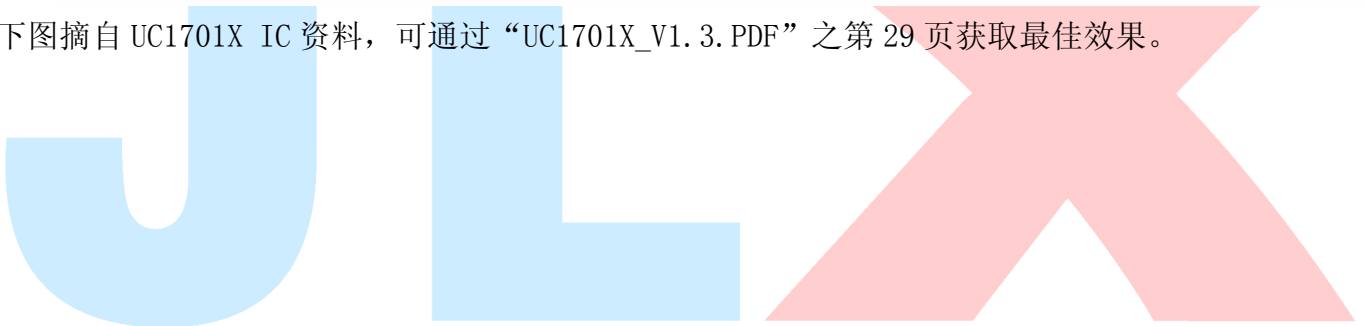
D0	0	1	1	1		0
D1	1	0	0	0		0
D2	0	0	0	0		0
D3	0	1	1	1		0
D4	1	0	0	0		0
-						

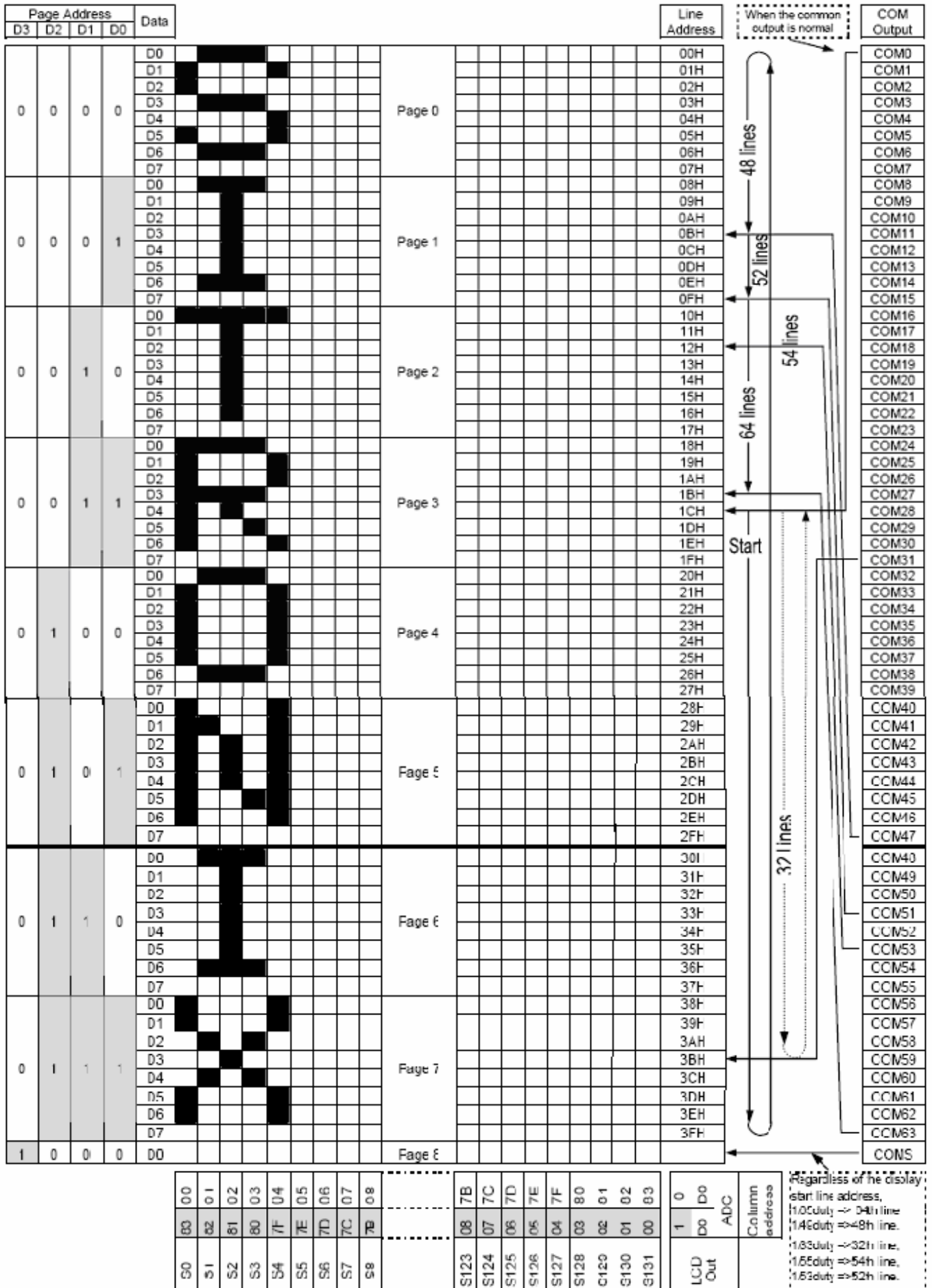
Display data RAM
(显示数据存储器)

COM0		■	■	■		
COM1	■					
COM2						
COM3		■	■	■		
COM4	■					
-						

Liquid crystal display
(液晶屏)

下图摘自 UC1701X IC 资料, 可通过“UC1701X_V1.3.PDF”之第 29 页获取最佳效果。

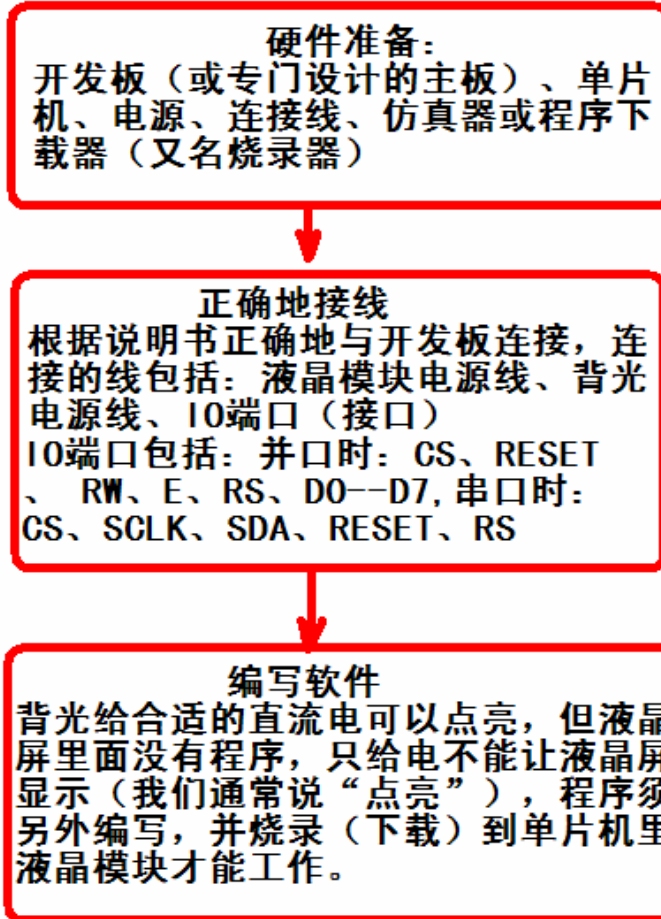




7.4 初始化方法

用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 过程请参考程序

点亮液晶模块的步骤



7.5 程序举例：

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下：

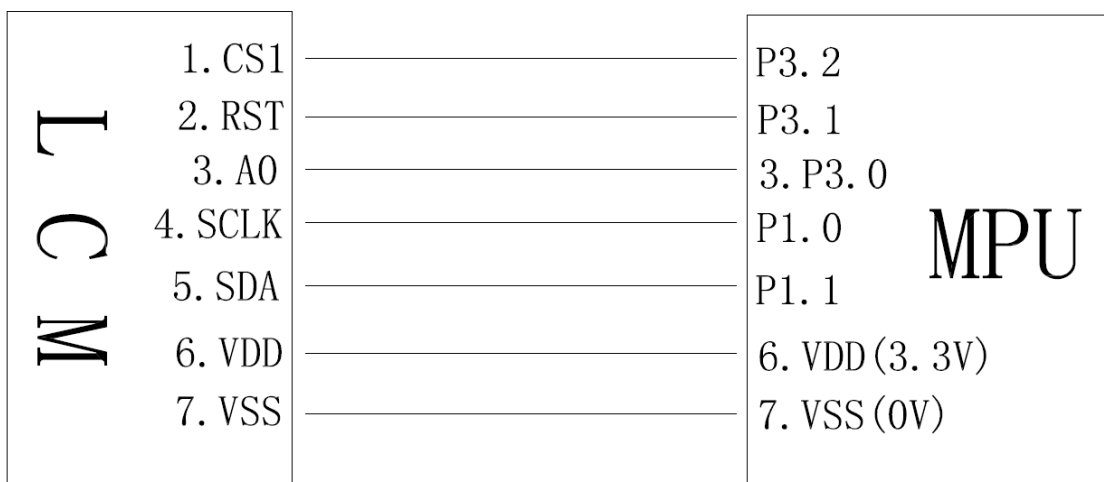
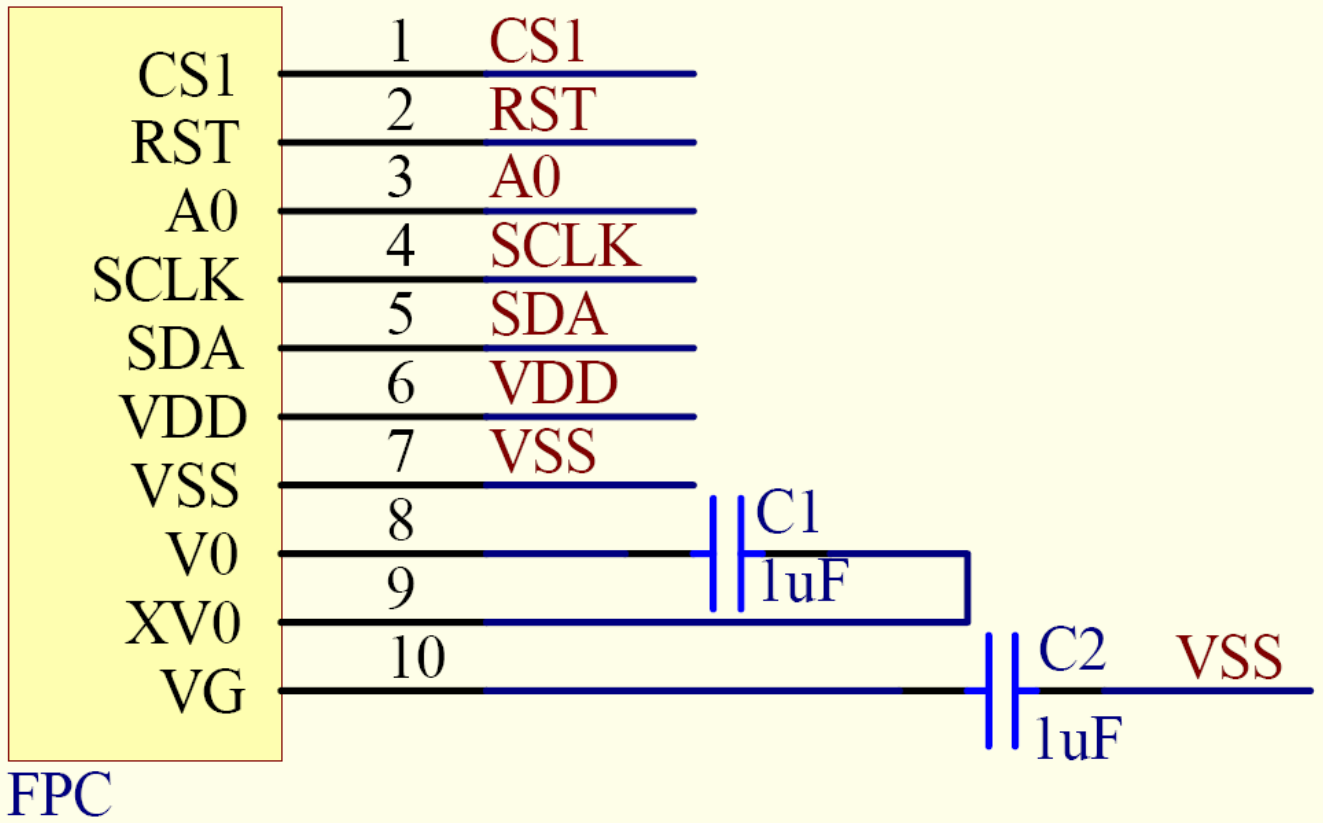


图 8. 串行接口

JLX12832G-040



串行例程序:

```
//Test program:For LCD module,model:JLX12832G-040
//Interface mode:4-wire SPI serial interface
//LCD driver ST7567
//Copyright:ShenZhen JLX electronics Co.,Ltd.
//Web:http://www.jlxlcd.cn;
//MCU:AT89C51
```

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

```
sbit key=P2^0;
```

```
sbit sid=P1^1;
sbit sclk=P1^0;
sbit rs=P3^0;
sbit reset=P3^1;
sbit cs1=P3^2;
```

```

#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long

char code graphic0[];
char code graphic1[];
char code ascii_table_8x16[95][16];
char code ascii_table_5x8[95][5];

//Chinese code
char code cheng1[]={
/*-- Chinese: cheng --*/
/*-- Font:song, 23W*H=32x32 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C,
0xFC, 0xFC, 0x88, 0x00, 0x00, 0x1C, 0x78, 0xF0, 0xE0, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x83, 0x83, 0x83, 0x83, 0x83, 0x83, 0xC3, 0xC3, 0x03, 0x1F,
0xFF, 0xFF, 0x83, 0x03, 0x03, 0x03, 0xC3, 0xF3, 0xF3, 0x63, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xFC, 0xFF, 0x3F, 0x00, 0x80, 0x00, 0x00, 0x80, 0xFF, 0xFF, 0x03, 0x00, 0x00, 0x03,
0x9F, 0xFF, 0xF8, 0xF8, 0xBE, 0x1F, 0x07, 0x01, 0x00, 0x00, 0xE0, 0x20, 0x00, 0x00, 0x20, 0x38,
0x1F, 0x07, 0x01, 0x00, 0x00, 0x01, 0x01, 0x07, 0x07, 0x23, 0x31, 0x18, 0x0C, 0x0E, 0x07, 0x03,
0x01, 0x01, 0x01, 0x03, 0x07, 0x0F, 0x0E, 0x1C, 0x1F, 0x3F, 0x30, 0x00, 0x00, 0x00, 0x00};

char code zhuang1[]={
/*-- Chinese: zhuang --*/
/*-- Font:song, 12, W*H=16x16 --*/
0x08, 0x30, 0x00, 0xFF, 0x20, 0x20, 0x20, 0x20, 0xFF, 0x20, 0xE1, 0x26, 0x2C, 0x20, 0x20, 0x00,
0x04, 0x02, 0x01, 0xFF, 0x40, 0x20, 0x18, 0x07, 0x00, 0x00, 0x03, 0x0C, 0x30, 0x60, 0x20, 0x00};

char code tail[]={
0x00, 0x04, 0x04, 0x04, 0x84, 0x44, 0x34, 0x4F, 0x94, 0x24, 0x44, 0x84, 0x84, 0x04, 0x00, 0x00,
0x00, 0x60, 0x39, 0x01, 0x00, 0x3C, 0x40, 0x42, 0x4C, 0x40, 0x40, 0x70, 0x04, 0x09, 0x31, 0x00};

char code shi1[]={
0x40, 0x20, 0xF0, 0x1C, 0x07, 0xF2, 0x94, 0x94, 0x94, 0xFF, 0x94, 0x94, 0x94, 0xF4, 0x04, 0x00,
0x00, 0x00, 0x7F, 0x00, 0x40, 0x41, 0x22, 0x14, 0x0C, 0x13, 0x10, 0x30, 0x20, 0x61, 0x20, 0x00};

char code yong1[]={
0x00, 0x00, 0x00, 0xFE, 0x22, 0x22, 0x22, 0x22, 0xFE, 0x22, 0x22, 0x22, 0x22, 0xFE, 0x00, 0x00,
0x80, 0x40, 0x30, 0x0F, 0x02, 0x02, 0x02, 0x02, 0xFF, 0x02, 0x02, 0x42, 0x82, 0x7F, 0x00, 0x00};

char code mao_hao[]={
0x00, 0x00, 0x00, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00};
    
```

```

char code num0[]={
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x10, 0x0F, 0x00
};
char code num1[]={
0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00
};
char code num2[]={

0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00
};
char code num3[]={

0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00
};
char code num4[]={

0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00
};

char code num5[]={

0x00, 0xF8, 0x08, 0x88, 0x88, 0x08, 0x08, 0x00, 0x00, 0x19, 0x21, 0x20, 0x20, 0x11, 0x0E, 0x00
};

char code num6[]={

0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00
};
char code num7[]={

0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00
};
char code num8[]={

0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00
};

char code num9[]={

0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00
};

```

```

//Write command to LCM(LCD MODULE)
void transfer_command(int data1)
{
    char i;

```

```

    cs1=0;
    rs=0;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

```

//Write data to LCM(LCD MODULE)

```

void transfer_data(int data1)
{
    char i;
    cs1=0;
    rs=1;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

```

//Delay

```

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

```

//wait a key, there is a key between P2.0 and GND in my mainboard

```

void waitkey()
{
    repeat:
        if(key==1) goto repeat;
        else delay(500);
}

```



```
//LCD 模块初始化
void initial_lcd()
{
    csl=0;
    reset=0;          //低电平复位
    delay(100);
    reset=1;         //复位完毕
    delay(50);
    transfer_command(0xe2); //软复位
    delay(5);
    transfer_command(0x2c); //升压步骤 1
    delay(5);
    transfer_command(0x2e); //升压步骤 2
    delay(5);
    transfer_command(0x2f); //升压步骤 3
    delay(5);
    transfer_command(0x22); //粗调对比度, 可设置范围 0x20~0x27
    transfer_command(0x81); //微调对比度
    transfer_command(0x15); //0x1a, 微调对比度的值, 可设置范围 0x00~0x3f
    transfer_command(0xa2); //1/7 偏压比 (bias)
    transfer_command(0xc0); //行扫描顺序: 从上到下
    transfer_command(0xa1); //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
}

//set the page address and column address of LCD
void lcd_address(char page,char column)
{
    csl=0;
    column=column+3;          //Because column address is from 0 to 127,so column=column-1
    page=page-1;
    transfer_command(0xb0+page); //set page address, LCD panel is divided to 8 pages, 1 page is include
8 lines.
    transfer_command(((column>>4)&0x0f)+0x10); //set the upper 4 bits of column address
    transfer_command(column&0x0f); //set the lower 4 bits of column address
}

//clear lcd screen
void clear_screen()
{
    unsigned char i,j;
    for(i=0;i<9;i++)
```

```

    {
        lcd_address(1+i, 1);
        for(j=0; j<132; j++)
        {
            transfer_data(0x00);           //send "0x00" to lcm, clear all dots
        }
    }
}

```

//=====display a picture of 128*64 dots=====

```
void full_display(char data_left, char data_right)
```

```

{
    int i, j;
    for(i=0; i<4; i++)
    {
        lcd_address(i+1, 1);
        for(j=0; j<64; j++)
        {
            transfer_data(data_left);
            transfer_data(data_right);
        }
    }
}

```

//=====display a picture of 32*32 dots=====

```
void display_graphic_32x32(char page, char column, char *dp)
```

```

{
    char i, j;
    for(j=0; j<4; j++)
    {
        lcd_address(page+j, column);
        for (i=0; i<31; i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

//=====display a picture of 16*16 dots=====

```
void display_graphic_16x16(char page, char column, char *dp)
```

```

{
    char i, j;
    for(j=0; j<2; j++)

```

```

{
    lcd_address(page+j, column);
    for (i=0;i<16;i++)
    {
        transfer_data(*dp);
        dp++;
    }
}
}

```

//=====display a picture of 8*16 dots=====

```
void display_graphic_8x16(char page, char column, char *dp)
```

```

{
    char i, j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<8;i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

//=====display a string of ASCII character, font style:8*16 dots=====

```
void display_string_8x16(uint page, uint column, char *text)
```

```

{
    uint i=0, j, k, n;
    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++)
                {
                    transfer_data(ascii_table_8x16[j][k+8*n]);
                }
            }
            i++;
            column+=8;
        }
    }
}

```

```

        else
        i++;
    }
}

//=====display a string of ASCII character,font style:5*8 dots=====
void display_string_5x8(uint page,uint column,char *text)
{
    uint i=0,j,k;
    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]);/*显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x 为列地址, 最后为
数据*/
            }
            i++;
            column+=6;
        }
        else
        i++;
    }
}

//=====display a picture of 128*64 dots=====
void disp_grap(char *dp)
{
    int i,j;
    for(j=0;j<8;j++)
    {
        lcd_address(1+j,1);
        for(i=0;i<128;i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}
}

```

```
void main(void)
```

```

{
while(1)
{
    cs1=0;
    initial_lcd();
    clear_screen(); //clear all dots
    display_graphic_32x32(1, 1, cheng1); //display 1 Chinese character on (1, 1). (page 1, column 1)
    waitkey();
    clear_screen(); //clear all dots
    display_graphic_16x16(1, 1, zhuang1);
    display_graphic_16x16(1, (1+16), tail);
    display_graphic_8x16(1, (1+16*2), mao_hao);
    display_graphic_16x16(1, (1+16*2+8), shi1);
    display_graphic_16x16(1, (1+16*3+8), yong1);
    display_graphic_8x16(1, (89), num0);
    display_graphic_8x16(1, (89+8*1), num0);
    display_graphic_8x16(1, (89+8*2), mao_hao);
    display_graphic_8x16(1, (89+8*3), num0);
    display_graphic_8x16(1, (89+8*4), num0);
    waitkey();
    clear_screen(); //clear all dots
    display_string_8x16(1, 1, "0123456789abcdef");//display a string of ASCII character.
    display_string_8x16(3, 1, "~`!@#%^&*()_-=");
    waitkey();
    clear_screen(); //clear all dots
    display_string_5x8(1, 1, "0123456789abcdef");//display a string of ASCII character.
    display_string_5x8(3, 1, "~`!@#%^&*()_-=");
    waitkey();
    disp_grap(graphic1); //display a picture of 128*32 dots
    waitkey();
    disp_grap(graphic0); //display a picture of 128*32 dots
    waitkey();
    full_display(0xff, 0xff);
    waitkey();
    full_display(0xaa, 0xaa);
    waitkey();
    full_display(0x55, 0x55);
    waitkey();
    full_display(0xff, 0x00);
    waitkey();
    full_display(0x00, 0xff);
    waitkey();
}
}

```

```
char code ascii_table_8x16[95][16]={
```

```

/*-- char:  --*/
/*-- Comic Sans MS12; Font format:=8x16  --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

/*-- character: !  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x30, 0x00, 0x00, 0x00,

```

```

/*-- character: "  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0x10, 0x0C, 0x06, 0x10, 0x0C, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

/*-- character: #  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x40, 0xC0, 0x78, 0x40, 0xC0, 0x78, 0x40, 0x00, 0x04, 0x3F, 0x04, 0x04, 0x3F, 0x04, 0x04, 0x00,

```

```

/*-- character: $  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0x70, 0x88, 0xFC, 0x08, 0x30, 0x00, 0x00, 0x00, 0x18, 0x20, 0xFF, 0x21, 0x1E, 0x00, 0x00,

```

```

/*-- character: %  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0xF0, 0x08, 0xF0, 0x00, 0xE0, 0x18, 0x00, 0x00, 0x00, 0x21, 0x1C, 0x03, 0x1E, 0x21, 0x1E, 0x00,

```

```

/*-- character: &  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0xF0, 0x08, 0x88, 0x70, 0x00, 0x00, 0x00, 0x1E, 0x21, 0x23, 0x24, 0x19, 0x27, 0x21, 0x10,

```

```

/*-- character: '  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x10, 0x16, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

/*-- character: (  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0x00, 0x00, 0xE0, 0x18, 0x04, 0x02, 0x00, 0x00, 0x00, 0x00, 0x07, 0x18, 0x20, 0x40, 0x00,

```

```

/*-- character: )  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x00, 0x02, 0x04, 0x18, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x40, 0x20, 0x18, 0x07, 0x00, 0x00, 0x00,

```

```

/*-- character: *  --*/
/*-- Comic Sans MS12; Font format:WxH=8x16  --*/
0x40, 0x40, 0x80, 0xF0, 0x80, 0x40, 0x40, 0x00, 0x02, 0x02, 0x01, 0x0F, 0x01, 0x02, 0x02, 0x00,

```

```

/*-- character: +  --*/

```

/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x1F, 0x01, 0x01, 0x01, 0x00,

/*-- character: , --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xB0, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- character: - --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,

/*-- character: . --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- character: / --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x18, 0x04, 0x00, 0x60, 0x18, 0x06, 0x01, 0x00, 0x00, 0x00,

/*-- character: 0 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x10, 0x0F, 0x00,

/*-- character: 1 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/*-- character: 2 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00,

/*-- character: 3 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,

/*-- character: 4 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00,

/*-- character: 5 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0xF8, 0x08, 0x88, 0x88, 0x08, 0x08, 0x00, 0x00, 0x19, 0x21, 0x20, 0x20, 0x11, 0x0E, 0x00,

/*-- character: 6 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,

```

/*-- character: 7 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00,

```

```

/*-- character: 8 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00,

```

```

/*-- character: 9 --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00,

```

```

/*-- character: : --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00,

```

```

/*-- character: ; --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x00, 0x00, 0x00, 0x00,

```

```

/*-- character: < --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x40, 0x20, 0x10, 0x08, 0x00, 0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00,

```

```

/*-- character: = --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x00, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x00,

```

```

/*-- character: > --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00, 0x00, 0x00, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01, 0x00,

```

```

/*-- character: ? --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x70, 0x48, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x30, 0x36, 0x01, 0x00, 0x00,

```

```

/*-- character: @ --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xC0, 0x30, 0xC8, 0x28, 0xE8, 0x10, 0xE0, 0x00, 0x07, 0x18, 0x27, 0x24, 0x23, 0x14, 0x0B, 0x00,

```

```

/*-- character: A --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0xC0, 0x38, 0xE0, 0x00, 0x00, 0x00, 0x20, 0x3C, 0x23, 0x02, 0x02, 0x27, 0x38, 0x20,

```

```

/*-- character: B --*/

```


/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x88, 0x88, 0x88, 0x70, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,

/*-- character: C --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xC0, 0x30, 0x08, 0x08, 0x08, 0x08, 0x38, 0x00, 0x07, 0x18, 0x20, 0x20, 0x20, 0x10, 0x08, 0x00,

/*-- character: D --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x10, 0x0F, 0x00,

/*-- character: E --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x88, 0x88, 0xE8, 0x08, 0x10, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x23, 0x20, 0x18, 0x00,

/*-- character: F --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x88, 0x88, 0xE8, 0x08, 0x10, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x03, 0x00, 0x00, 0x00,

/*-- character: G --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xC0, 0x30, 0x08, 0x08, 0x08, 0x38, 0x00, 0x00, 0x07, 0x18, 0x20, 0x20, 0x22, 0x1E, 0x02, 0x00,

/*-- character: H --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x08, 0x00, 0x00, 0x08, 0xF8, 0x08, 0x20, 0x3F, 0x21, 0x01, 0x01, 0x21, 0x3F, 0x20,

/*-- character: I --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/*-- character: J --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x00, 0xC0, 0x80, 0x80, 0x80, 0x7F, 0x00, 0x00, 0x00,

/*-- character: K --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x88, 0xC0, 0x28, 0x18, 0x08, 0x00, 0x20, 0x3F, 0x20, 0x01, 0x26, 0x38, 0x20, 0x00,

/*-- character: L --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x08, 0x00, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x20, 0x30, 0x00,

/*-- character: M --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0xF8, 0x00, 0xF8, 0xF8, 0x08, 0x00, 0x20, 0x3F, 0x00, 0x3F, 0x00, 0x3F, 0x20, 0x00,

```

/*-- character: N --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x30, 0xC0, 0x00, 0x08, 0xF8, 0x08, 0x20, 0x3F, 0x20, 0x00, 0x07, 0x18, 0x3F, 0x00,

```

```

/*-- character: O --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xE0, 0x10, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x20, 0x10, 0x0F, 0x00,

```

```

/*-- character: P --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x08, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x20, 0x3F, 0x21, 0x01, 0x01, 0x01, 0x00, 0x00,

```

```

/*-- character: Q --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xE0, 0x10, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x0F, 0x18, 0x24, 0x24, 0x38, 0x50, 0x4F, 0x00,

```

```

/*-- character: R --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x88, 0x88, 0x88, 0x88, 0x70, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x03, 0x0C, 0x30, 0x20,

```

```

/*-- character: S --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x70, 0x88, 0x08, 0x08, 0x08, 0x38, 0x00, 0x00, 0x38, 0x20, 0x21, 0x21, 0x22, 0x1C, 0x00,

```

```

/*-- character: T --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x18, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x18, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x00, 0x00,

```

```

/*-- character: U --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x08, 0x00, 0x00, 0x08, 0xF8, 0x08, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x20, 0x1F, 0x00,

```

```

/*-- character: V --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0x78, 0x88, 0x00, 0x00, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x07, 0x38, 0x0E, 0x01, 0x00, 0x00,

```

```

/*-- character: W --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0xF8, 0x08, 0x00, 0xF8, 0x00, 0x08, 0xF8, 0x00, 0x03, 0x3C, 0x07, 0x00, 0x07, 0x3C, 0x03, 0x00,

```

```

/*-- character: X --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0x18, 0x68, 0x80, 0x80, 0x68, 0x18, 0x08, 0x20, 0x30, 0x2C, 0x03, 0x03, 0x2C, 0x30, 0x20,

```

```

/*-- character: Y --*/

```

/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0x38, 0xC8, 0x00, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x00, 0x00,

/*-- character: Z --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x10, 0x08, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x20, 0x38, 0x26, 0x21, 0x20, 0x20, 0x18, 0x00,

/*-- character: [--*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0xFE, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x40, 0x40, 0x40, 0x00,

/*-- character: \ --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x0C, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x06, 0x38, 0xC0, 0x00,

/*-- character:] --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x02, 0x02, 0x02, 0xFE, 0x00, 0x00, 0x00, 0x40, 0x40, 0x40, 0x7F, 0x00, 0x00, 0x00,

/*-- character: ^ --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x04, 0x02, 0x02, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- character: _ --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,

/*-- character: ` --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x02, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- character: a --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x19, 0x24, 0x22, 0x22, 0x22, 0x3F, 0x20,

/*-- character: b --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,

/*-- character: c --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x20, 0x11, 0x00,

/*-- character: d --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x80, 0x88, 0xF8, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x10, 0x3F, 0x20,

```

/*-- character: e --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x22, 0x22, 0x22, 0x22, 0x13, 0x00,

```

```

/*-- character: f --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x80, 0x80, 0xF0, 0x88, 0x88, 0x88, 0x18, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

```

```

/*-- character: g --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x6B, 0x94, 0x94, 0x94, 0x93, 0x60, 0x00,

```

```

/*-- character: h --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x3F, 0x21, 0x00, 0x00, 0x20, 0x3F, 0x20,

```

```

/*-- character: i --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x80, 0x98, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

```

```

/*-- character: j --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x98, 0x98, 0x00, 0x00, 0x00, 0xC0, 0x80, 0x80, 0x80, 0x7F, 0x00, 0x00,

```

```

/*-- character: k --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x08, 0xF8, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x20, 0x3F, 0x24, 0x02, 0x2D, 0x30, 0x20, 0x00,

```

```

/*-- character: l --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

```

```

/*-- character: m --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x3F, 0x20, 0x00, 0x3F,

```

```

/*-- character: n --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x3F, 0x21, 0x00, 0x00, 0x20, 0x3F, 0x20,

```

```

/*-- character: o --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x20, 0x1F, 0x00,

```

```

/*-- character: p --*/

```

/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x80, 0xFF, 0xA1, 0x20, 0x20, 0x11, 0x0E, 0x00,

/*-- character: q --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0xA0, 0xFF, 0x80,

/*-- character: r --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x20, 0x20, 0x3F, 0x21, 0x20, 0x00, 0x01, 0x00,

/*-- character: s --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x33, 0x24, 0x24, 0x24, 0x24, 0x19, 0x00,

/*-- character: t --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x80, 0x80, 0xE0, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x00, 0x00,

/*-- character: u --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x10, 0x3F, 0x20,

/*-- character: v --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x01, 0x0E, 0x30, 0x08, 0x06, 0x01, 0x00,

/*-- character: w --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80, 0x80, 0x0F, 0x30, 0x0C, 0x03, 0x0C, 0x30, 0x0F, 0x00,

/*-- character: x --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x31, 0x2E, 0x0E, 0x31, 0x20, 0x00,

/*-- character: y --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x80, 0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x81, 0x8E, 0x70, 0x18, 0x06, 0x01, 0x00,

/*-- character: z --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x21, 0x30, 0x2C, 0x22, 0x21, 0x30, 0x00,

/*-- character: { --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0x80, 0x7C, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x40, 0x40,

```

/*-- character: | --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,

/*-- character: } --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x02, 0x02, 0x7C, 0x80, 0x00, 0x00, 0x00, 0x00, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00,

/*-- character: ~ --*/
/*-- Comic Sans MS12; Font format:WxH=8x16 --*/
0x00, 0x06, 0x01, 0x01, 0x02, 0x02, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

};

//-----
char code graphic0[]={
/*-- 调入了一幅图像: E:\work\图片收藏夹\JLX12832G-508.bmp --*/
/*-- Picture format:WxH=128x32 --*/
0xFF, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x09, 0x09, 0xF9, 0xF9, 0x09, 0x09, 0x01, 0x09, 0xF9,
0xF9, 0x09, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x09, 0x19, 0x79, 0xE9, 0x81, 0xE9, 0x79, 0x19, 0x09,
0x01, 0x11, 0x11, 0xF9, 0xF9, 0x01, 0x01, 0x01, 0x01, 0x01, 0x71, 0x79, 0x09, 0x09, 0x89, 0xF9,
0x71, 0x01, 0x01, 0x71, 0xF9, 0x89, 0x09, 0x89, 0xF9, 0x71, 0x01, 0x01, 0x31, 0x39, 0x89, 0x89,
0xC9, 0x79, 0x31, 0x01, 0x01, 0x71, 0x79, 0x09, 0x09, 0x89, 0xF9, 0x71, 0x01, 0xC1, 0xF1, 0x39,
0x09, 0x09, 0x39, 0x39, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0xF9, 0xF9, 0x89, 0x89, 0x89, 0x09, 0x09, 0x01, 0x01, 0xE1, 0xF1, 0x19, 0x09, 0x19, 0xF1, 0xE1,
0x01, 0x01, 0x71, 0xF9, 0x89, 0x09, 0x89, 0xF9, 0x71, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0xC0, 0xC0, 0x80, 0x80, 0xFF, 0x7F, 0x00, 0x00, 0x00, 0x20, 0x3F,
0x3F, 0x20, 0x20, 0x20, 0x30, 0x30, 0x00, 0x20, 0x30, 0x3C, 0x2F, 0x03, 0x2F, 0x3C, 0x30, 0x20,
0x00, 0x20, 0x20, 0x3F, 0x3F, 0x20, 0x20, 0x00, 0x00, 0x00, 0x30, 0x38, 0x2C, 0x26, 0x23, 0x31,
0x30, 0x00, 0x00, 0x1C, 0x3E, 0x23, 0x21, 0x23, 0x3E, 0x1C, 0x00, 0x00, 0x18, 0x38, 0x20, 0x20,
0x31, 0x1F, 0x0E, 0x00, 0x00, 0x30, 0x38, 0x2C, 0x26, 0x23, 0x31, 0x30, 0x00, 0x07, 0x1F, 0x38,
0x20, 0x22, 0x3E, 0x1E, 0x02, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,
0x19, 0x39, 0x21, 0x20, 0x31, 0x1F, 0x0E, 0x00, 0x00, 0x0F, 0x1F, 0x30, 0x20, 0x30, 0x1F, 0x0F,
0x00, 0x00, 0x1C, 0x3E, 0x23, 0x21, 0x23, 0x3E, 0x1C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0xF8,
0xF8, 0x08, 0x08, 0x18, 0xF0, 0xE0, 0x00, 0xE0, 0xF0, 0x18, 0x08, 0x08, 0x18, 0xF0, 0xE0, 0x00,
0x18, 0x18, 0x08, 0xF8, 0xF8, 0x08, 0x18, 0x18, 0x00, 0x00, 0x70, 0xF8, 0x88, 0x08, 0x08, 0x38,
0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0xF8, 0xF8,
0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x78, 0x08, 0x08, 0x88, 0xF8, 0x70, 0x00, 0x00, 0x70, 0xF8,
0x88, 0x08, 0x88, 0xF8, 0x70, 0x00, 0x40, 0x40, 0xC0, 0xF0, 0xF0, 0xC0, 0x40, 0x40, 0x00, 0x00,
0x30, 0x38, 0x88, 0x88, 0xC8, 0x78, 0x30, 0x00, 0x00, 0x70, 0x78, 0x08, 0x08, 0x88, 0xF8, 0x70,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xA0, 0xBF,
0xBF, 0xA0, 0xA0, 0xB0, 0x9F, 0x8F, 0x80, 0x8F, 0x9F, 0xB0, 0xA0, 0xA0, 0xB0, 0x9F, 0x8F, 0x80,

```

```
0x80, 0x80, 0xA0, 0xBF, 0xBF, 0xA0, 0x80, 0x80, 0x80, 0x80, 0xB8, 0xB8, 0xA1, 0xA1, 0xA3, 0xBE,
0x9C, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xA0, 0xA0, 0xBF, 0xBF,
0xA0, 0xA0, 0x80, 0x80, 0x80, 0xB0, 0xB8, 0xAC, 0xA6, 0xA3, 0xB1, 0xB0, 0x80, 0x80, 0x9C, 0xBE,
0xA3, 0xA1, 0xA3, 0xBE, 0x9C, 0x80, 0x82, 0x82, 0x83, 0x8F, 0x8F, 0x83, 0x82, 0x82, 0x80, 0x80,
0x98, 0xB8, 0xA0, 0xA0, 0xB1, 0x9F, 0x8E, 0x80, 0x80, 0xB0, 0xB8, 0xAC, 0xA6, 0xA3, 0xB1, 0xB0,
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xFF
```

```
};
```

```
char code graphic1[]={
```

```
/*-- 调入了一幅图像: D:\e\新开发部\显示图案收藏\YJD12834.bmp --*/
```

```
/*-- 宽度 x 高度=128x32 --*/
```

```
0x00, 0x80, 0x40, 0x60, 0x50, 0x48, 0x44, 0xC3, 0x44, 0x48, 0x50, 0x70, 0x60, 0x20, 0x00, 0x00,
0x40, 0x44, 0x24, 0x24, 0xF4, 0x5C, 0x57, 0x54, 0x54, 0x5C, 0x54, 0x24, 0x64, 0xC4, 0x44, 0x00,
0x10, 0x61, 0x06, 0xE0, 0x18, 0x84, 0xE4, 0x1C, 0x84, 0x65, 0xBE, 0x24, 0xA4, 0x64, 0x04, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7E, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x7E, 0x00, 0x00, 0x00, 0x00,
0x08, 0xF4, 0x57, 0x54, 0xFC, 0x54, 0xF0, 0x42, 0xA2, 0x1E, 0x02, 0xA2, 0x62, 0x3E, 0x00, 0x00,
0x02, 0x04, 0xD8, 0x08, 0x00, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,
0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0xF9, 0x4A, 0x4C, 0x48, 0x48, 0xC8, 0x08, 0x08, 0x08, 0x00,
0x00, 0x20, 0x2C, 0x24, 0x64, 0x74, 0xAD, 0xA6, 0xE4, 0x34, 0x24, 0x24, 0x2C, 0x24, 0x00, 0x00,
0x00, 0x40, 0x40, 0x44, 0x44, 0x44, 0x44, 0x7F, 0x44, 0x44, 0x44, 0x46, 0x44, 0x60, 0x40, 0x00,
0x04, 0x04, 0x44, 0x64, 0x57, 0x4D, 0x45, 0x65, 0x25, 0x25, 0x35, 0xE4, 0x44, 0x04, 0x04, 0x00,
0x04, 0x04, 0xFF, 0x00, 0x01, 0x00, 0xFF, 0x41, 0x21, 0x12, 0x0C, 0x1B, 0x61, 0xC0, 0x40, 0x00,
0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00,
0x80, 0x7F, 0x02, 0x02, 0x5F, 0x82, 0x7F, 0x0A, 0x09, 0x09, 0x09, 0xFF, 0x09, 0x09, 0x09, 0x00,
0x02, 0xFE, 0x01, 0x80, 0x41, 0x21, 0x11, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00,
0x40, 0x40, 0x20, 0x10, 0x0C, 0x03, 0x00, 0x00, 0x20, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x24, 0x24, 0x25, 0x15, 0x15, 0x0D, 0xFE, 0x04, 0x0D, 0x17, 0x14, 0x24, 0x64, 0x24, 0x00,
0x00, 0x00, 0xFE, 0x02, 0x0A, 0xCA, 0x4A, 0x6A, 0x5E, 0x49, 0x49, 0xC9, 0x09, 0x01, 0x00, 0x00,
0x40, 0x40, 0x40, 0xDF, 0x55, 0x55, 0x55, 0xD5, 0x55, 0x55, 0x55, 0xDF, 0x40, 0x40, 0x40, 0x00,
0x02, 0x02, 0xFE, 0x92, 0x92, 0x92, 0xFE, 0x02, 0x02, 0x7C, 0x84, 0x04, 0x84, 0x7C, 0x04, 0x00,
0x00, 0x00, 0xFE, 0x12, 0x12, 0xFE, 0x20, 0x1C, 0x10, 0x10, 0xFF, 0x10, 0x10, 0x10, 0x00, 0x00,
0x40, 0x20, 0xD0, 0x4C, 0x43, 0x44, 0x48, 0xD8, 0x30, 0x10, 0x00, 0xFC, 0x00, 0x00, 0xFF, 0x00,
0x00, 0x42, 0x62, 0x5A, 0xC6, 0x10, 0x54, 0x54, 0x54, 0xFF, 0x54, 0x54, 0x7C, 0x10, 0x10, 0x00,
0x00, 0x00, 0xFC, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0xFC, 0x00, 0x00, 0x00,
0x00, 0x82, 0xE2, 0x5E, 0x42, 0xC2, 0x02, 0xFC, 0x24, 0xA6, 0x7D, 0x24, 0x24, 0xFC, 0x00, 0x00,
0x80, 0x60, 0x1F, 0x00, 0x80, 0x4F, 0x20, 0x10, 0x0F, 0x10, 0x20, 0xEF, 0x40, 0x00, 0x00, 0x00,
0x40, 0x40, 0x40, 0x57, 0x55, 0x55, 0x55, 0x7F, 0x55, 0x55, 0x55, 0x57, 0x50, 0x40, 0x40, 0x00,
0x10, 0x10, 0x0F, 0x08, 0x08, 0x04, 0xFF, 0x04, 0x22, 0x10, 0x09, 0x06, 0x09, 0x30, 0x10, 0x00,
0x40, 0x30, 0x0F, 0x21, 0x41, 0x3F, 0x20, 0x21, 0x21, 0x21, 0x3F, 0x21, 0x21, 0x21, 0x20, 0x00,
0x00, 0x00, 0x3F, 0x40, 0x40, 0x42, 0x44, 0x43, 0x78, 0x00, 0x00, 0x07, 0x20, 0x40, 0x3F, 0x00,
0x40, 0x23, 0x14, 0x08, 0x17, 0x28, 0x29, 0x49, 0x49, 0x7F, 0x49, 0x49, 0x49, 0x49, 0x40, 0x00,
0x00, 0x00, 0x3F, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x3F, 0x00, 0x00, 0x00,
0x01, 0x00, 0x7F, 0x10, 0x10, 0x3F, 0x08, 0x0D, 0x0B, 0x09, 0xFF, 0x09, 0x09, 0x09, 0x08, 0x00,
```

};

```
char code ascii_table_5x8[95][5]={
/*all ASCII list:5x8 dots*/
0x00,0x00,0x00,0x00,0x00, //space
0x00,0x00,0x4f,0x00,0x00, //!
0x00,0x07,0x00,0x07,0x00, //"
0x14,0x7f,0x14,0x7f,0x14, //#
0x24,0x2a,0x7f,0x2a,0x12, //$
0x23,0x13,0x08,0x64,0x62, //%
0x36,0x49,0x55,0x22,0x50, //&
0x00,0x05,0x07,0x00,0x00, //]
0x00,0x1c,0x22,0x41,0x00, //(
0x00,0x41,0x22,0x1c,0x00, //)
0x14,0x08,0x3e,0x08,0x14, //*
0x08,0x08,0x3e,0x08,0x08, //+
0x00,0x50,0x30,0x00,0x00, //,
0x08,0x08,0x08,0x08,0x08, //-
0x00,0x60,0x60,0x00,0x00, //.
0x20,0x10,0x08,0x04,0x02, ///
0x3e,0x51,0x49,0x45,0x3e, //0
0x00,0x42,0x7f,0x40,0x00, //1
0x42,0x61,0x51,0x49,0x46, //2
0x21,0x41,0x45,0x4b,0x31, //3
0x18,0x14,0x12,0x7f,0x10, //4
0x27,0x45,0x45,0x45,0x39, //5
0x3c,0x4a,0x49,0x49,0x30, //6
0x01,0x71,0x09,0x05,0x03, //7
0x36,0x49,0x49,0x49,0x36, //8
0x06,0x49,0x49,0x29,0x1e, //9
0x00,0x36,0x36,0x00,0x00, //:
0x00,0x56,0x36,0x00,0x00, //;
0x08,0x14,0x22,0x41,0x00, //<
0x14,0x14,0x14,0x14,0x14, // =
0x00,0x41,0x22,0x14,0x08, //>
0x02,0x01,0x51,0x09,0x06, //?
0x32,0x49,0x79,0x41,0x3e, //@
0x7e,0x11,0x11,0x11,0x7e, //A
0x7f,0x49,0x49,0x49,0x36, //B
0x3e,0x41,0x41,0x41,0x22, //C
0x7f,0x41,0x41,0x22,0x1c, //D
0x7f,0x49,0x49,0x49,0x41, //E
0x7f,0x09,0x09,0x09,0x01, //F
0x3e,0x41,0x49,0x49,0x7a, //G
```



0x7f, 0x08, 0x08, 0x08, 0x7f, //H
0x00, 0x41, 0x7f, 0x41, 0x00, //I
0x20, 0x40, 0x41, 0x3f, 0x01, //J
0x7f, 0x08, 0x14, 0x22, 0x41, //K
0x7f, 0x40, 0x40, 0x40, 0x40, //L
0x7f, 0x02, 0x0c, 0x02, 0x7f, //M
0x7f, 0x04, 0x08, 0x10, 0x7f, //N
0x3e, 0x41, 0x41, 0x41, 0x3e, //O
0x7f, 0x09, 0x09, 0x09, 0x06, //P
0x3e, 0x41, 0x51, 0x21, 0x5e, //Q
0x7f, 0x09, 0x19, 0x29, 0x46, //R
0x46, 0x49, 0x49, 0x49, 0x31, //S
0x01, 0x01, 0x7f, 0x01, 0x01, //T
0x3f, 0x40, 0x40, 0x40, 0x3f, //U
0x1f, 0x20, 0x40, 0x20, 0x1f, //V
0x3f, 0x40, 0x38, 0x40, 0x3f, //W
0x63, 0x14, 0x08, 0x14, 0x63, //X
0x07, 0x08, 0x70, 0x08, 0x07, //Y
0x61, 0x51, 0x49, 0x45, 0x43, //Z
0x00, 0x7f, 0x41, 0x41, 0x00, //[
0x02, 0x04, 0x08, 0x10, 0x20, //\
0x00, 0x41, 0x41, 0x7f, 0x00, //]
0x04, 0x02, 0x01, 0x02, 0x04, //^
0x40, 0x40, 0x40, 0x40, 0x40, //_
0x01, 0x02, 0x04, 0x00, 0x00, //`
0x20, 0x54, 0x54, 0x54, 0x78, //a
0x7f, 0x48, 0x48, 0x48, 0x30, //b
0x38, 0x44, 0x44, 0x44, 0x44, //c
0x30, 0x48, 0x48, 0x48, 0x7f, //d
0x38, 0x54, 0x54, 0x58, //e
0x00, 0x08, 0x7e, 0x09, 0x02, //f
0x48, 0x54, 0x54, 0x54, 0x3c, //g
0x7f, 0x08, 0x08, 0x08, 0x70, //h
0x00, 0x00, 0x7a, 0x00, 0x00, //i
0x20, 0x40, 0x40, 0x3d, 0x00, //j
0x7f, 0x20, 0x28, 0x44, 0x00, //k
0x00, 0x41, 0x7f, 0x40, 0x00, //l
0x7c, 0x04, 0x38, 0x04, 0x7c, //m
0x7c, 0x08, 0x04, 0x04, 0x78, //n
0x38, 0x44, 0x44, 0x44, 0x38, //o
0x7c, 0x14, 0x14, 0x14, 0x08, //p
0x08, 0x14, 0x14, 0x14, 0x7c, //q
0x7c, 0x08, 0x04, 0x04, 0x08, //r
0x48, 0x54, 0x54, 0x54, 0x24, //s
0x04, 0x04, 0x3f, 0x44, 0x24, //t
0x3c, 0x40, 0x40, 0x40, 0x3c, //u



```
0x1c, 0x20, 0x40, 0x20, 0x1c, //v
0x3c, 0x40, 0x30, 0x40, 0x3c, //w
0x44, 0x28, 0x10, 0x28, 0x44, //x
0x04, 0x48, 0x30, 0x08, 0x04, //y
0x44, 0x64, 0x54, 0x4c, 0x44, //z
0x08, 0x36, 0x41, 0x41, 0x00, //{
0x00, 0x00, 0x77, 0x00, 0x00, //|
0x00, 0x41, 0x41, 0x36, 0x08, //}
0x04, 0x02, 0x02, 0x02, 0x01, //~
};
```

