

# JLX12864G-573-PN 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	5~6
6	时序特性	6~7
7	指令功能及硬件接口与编程案例	7~29

## 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 JLX12864G-573-PN 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。

JLX12864G-573-PN 可以显示 128 列\*64 行点阵单色图片，或显示 8 个/行\*4 行 16\*16 点阵的汉字，或显示 16 个/行\*8 行 8\*8 点阵的英文、数字、符号。

## 2. JLX12864G-573-PN 图像型点阵液晶模块的特性

2.1 结构牢：插式 FPC。

2.2 IC 采用矽创公司 ST7565R, 功能强大，稳定性好

2.3 功耗低:5 - 100mW (不带背光 5mW, 带背光不大于 100mW) ;

2.4 显示内容:

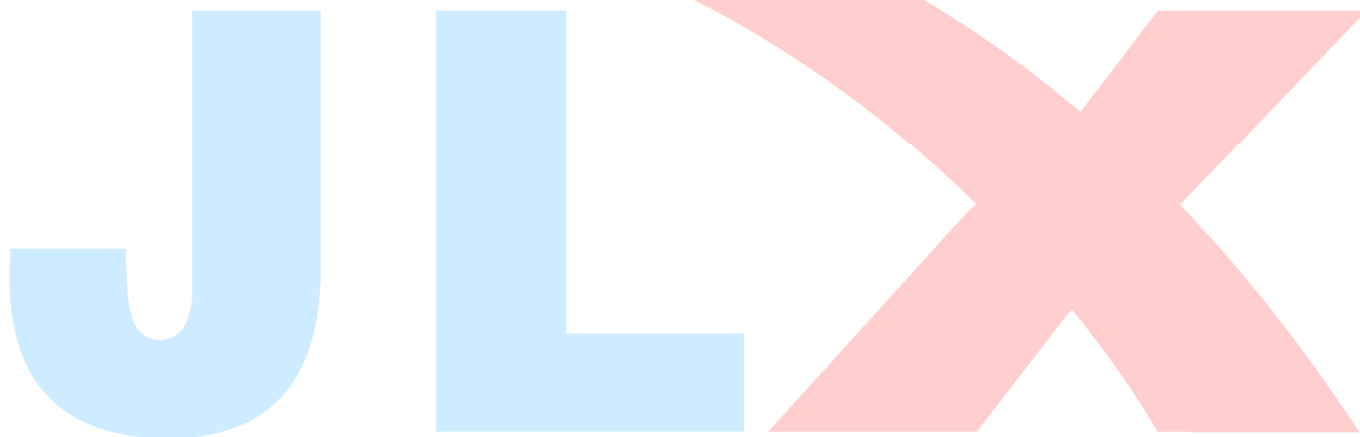
- 128\*64 点阵单色图片;

- 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/行\*4 行。按照 12\*12 点阵汉字来计算可显示 10 字/行\*4 行。

2.5 指令功能强:可组合成各种输入、显示、移位方式以满足不同的要求;

2.6 接口简单方便:串行接口。

2.7 工作温度宽:-0℃ - 50℃;



3. 外形尺寸及接口引脚功能

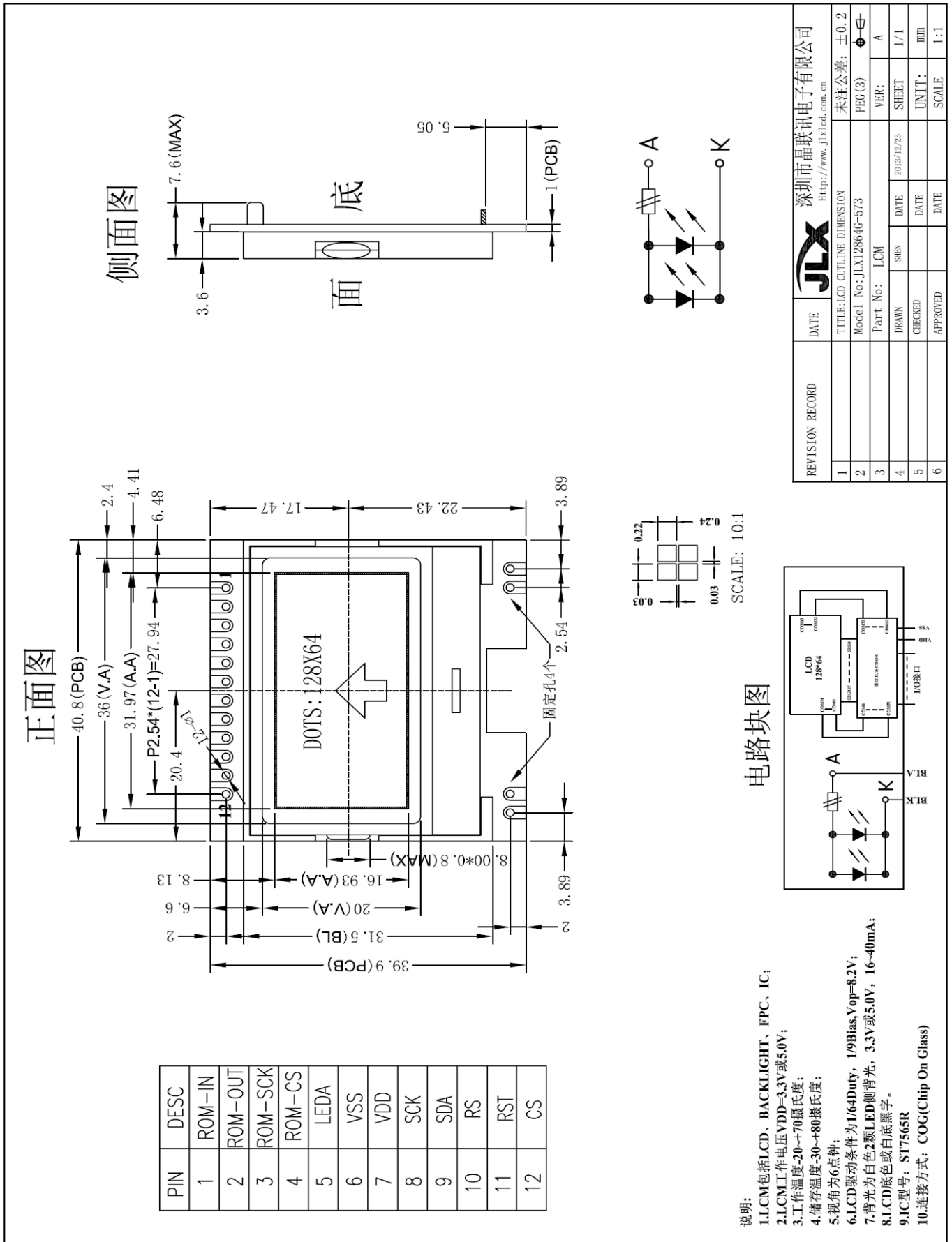


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	NC	NC	
2	NC	NC	
3	NC	NC	
4	NC	NC	
5	LDEA	背光电源	背光电源正极、同 VDD 电压 (5V 或 3.3V)
6	VSS	接地	0V
7	VDD	电源电路	5V, 或 3.3V 可选
8	SCLK	I/O	串行时钟
9	SDA	I/O	串行数据
10	RS	寄存选择信号	H: 数据存储器 0: 指令存储 (IC 资料上缩写为 “CD”)
11	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
12	CS	片选	低电平片选

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 128×64 点阵, 128 个列信号与驱动 IC 相连, 64 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 2 是 JLX12864G-573-PN 图像点阵型模块的电路框图, 它由驱动 IC ST7565R 及几个电阻电容组成。

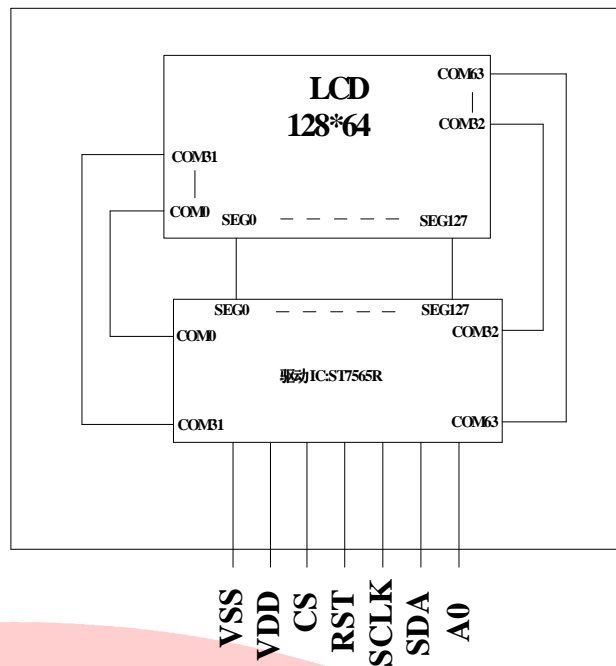


图 2: JLX12864G-573-PN 图像点阵型液晶模块的电路框图

## 4.2 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下：  
 背光板可选择白色。  
 正常工作电流为：16~40mA（LED 灯数共 2 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		7.0	V
LCD 驱动电压	VDD - V0	VDD - 13.5		VDD + 0.3	V
静电电压		-	-	100	V
工作温度		-10		+60	°C
储存温度		-20		+70	°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	3.6	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	IOO = 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	16	30	40	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 ST7565R (Writing Data from CPU to ST7565R)

The 4-line SPI Interface

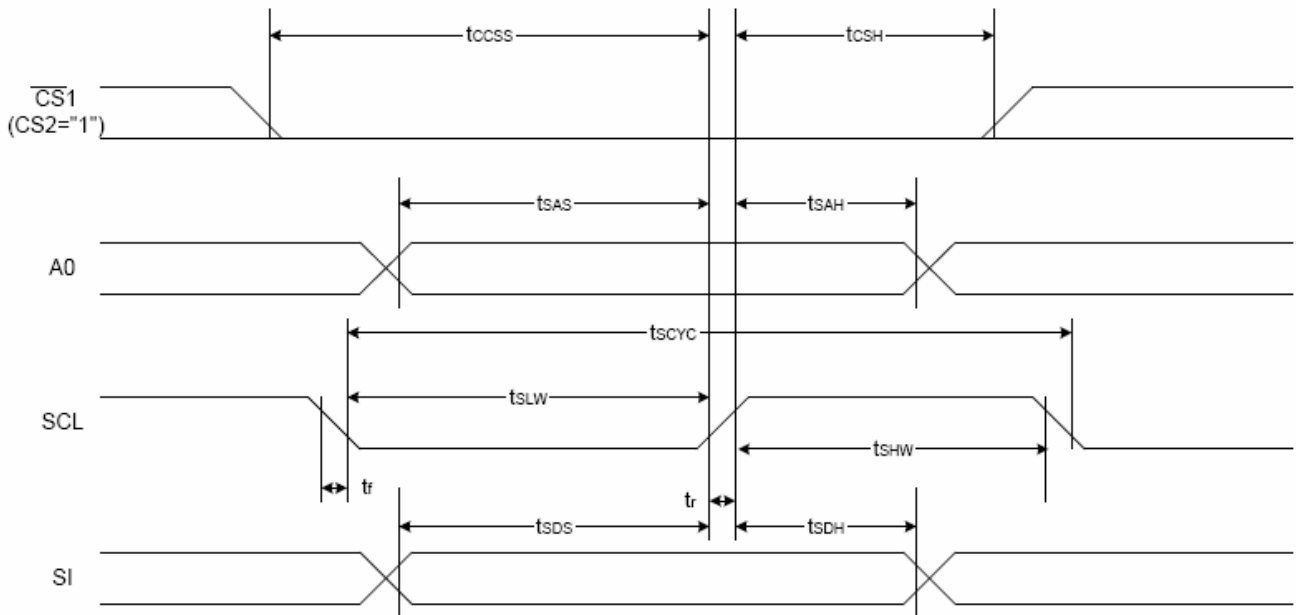


图 4. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

### 6.2 串行接口: 时序要求 (AC 参数):

写数据到 ST7565R 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 <sub>scyc</sub>	引脚: SCK	50	--	25	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T <sub>shw</sub>	引脚: SCK	25			ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T <sub>slw</sub>	引脚: SCK	25			ns
地址建立时间 (Address setup time)	T <sub>sas</sub>	引脚: RS	20	--	--	ns
地址保持时间 (Address hold time)	T <sub>sah</sub>	引脚: RS	10	--	--	ns
数据建立时间 (Data setup time)	T <sub>sds</sub>	引脚: SI	20	--	--	ns
数据保持时间 (Data hold time)	T <sub>sdh</sub>	引脚: SI	10	--	--	ns

片选信号建立时间 (CS-SCL time)	T <sub>css</sub>	引脚: CS	20			ns
片选信号保持时间 (CS-SCL time)	T <sub>csh</sub>	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):

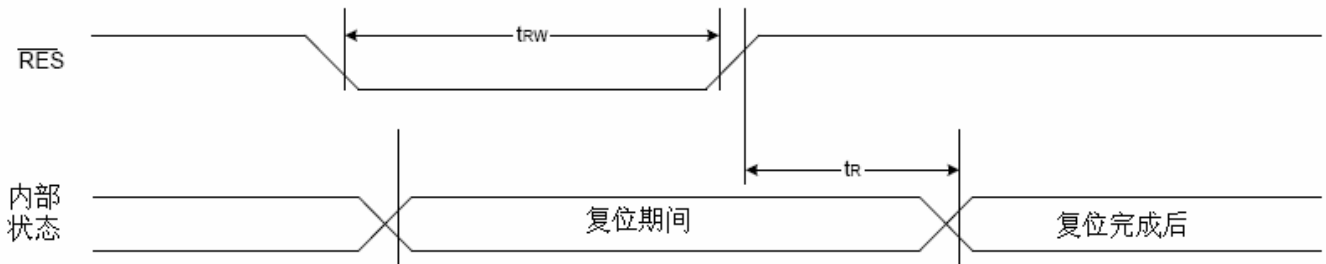


图 7: 电源启动后复位的时序

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: 0: 关, 1: 开
(2) 显示初始行设置 (Display start line set)	0	0	1	显示初始行地址, 共 5 位						设置显示存储器的显示初始行
(3) 页地址设置 (Page address set)	0	1	0	1	1	显示页地址, 共 4 位				设置显示页地址 (注: 每 8 行为一个页, 64 行分为 8 个页, 例 0000 为第一页, 0001 为第二页)
(4) 列地址高 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, 分别指定 128 列中任对应列。本液晶模块的第一列的地址为 00000001, 所以此指令表达为:
		0	0	0	0	列地址的低 4 位				

										0x10, 0x01	
(5) 读状态 (Status read)	0	状态				0	0	0	0		在本型号液晶模块不用此指令
(6) 写数据 (Display data write)	1	<b>8 位显示数据</b>									从 CPU 写数据到液晶模块
(7) 读数据 (Display data read)	1	<b>8 位显示数据</b>									在本型号液晶模块不用此指令
(8) 显示列地址增减 (ADC select)		1	0	1	0	0	0	0	<b>0</b> <b>1</b>		显示列地址增减: 0: 常规: 从左到右, 1: 反转: 从右到左
(9) 显示正显/反显 (Display normal/reverse)	0	1	0	1	0	0	1	1	<b>0</b> <b>1</b>		显示正显/反显: 0: 常规: 正显 1: 反显
(10) 显示全部点阵 (Display all points)	0	1	0	1	0	0	1	0	<b>0</b> <b>1</b>		显示全部点阵: 0: 常规 1: 显示全部点阵
(11) LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	<b>0</b> <b>1</b>		设置偏压比: 0: 1/9 BIAS 1: 1/7 BIAS
(12) Read-modify-write	0	1	1	1	0	0	0	0	0		Column address increment At write: +1 At read: 0
13) 退出上述指令 (End)	0	1	1	1	0	1	1	1	0		退出上述 “read/modify/write” 指令
(14) 软件复位 (Reset)	0	1	1	1	0	0	0	1	<b>0</b>		软件复位。
(15) 行扫描顺序选择 (Common output mode select)		1	1	0	0	0	0	0	<b>0</b>		行扫描顺序选择: 0: 普通顺序 1: 反向扫描
(16) 电源控制 (Power control set)		0	0	1	0	1	<b>电压操作模式选择, 共 3 位</b>				选择内部电压供应操作模式
(17) 选择内部电阻比例	0	0	0	1	0	0	<b>内部电压值电阻设置</b>				选择内部电阻比例 (Rb/Ra), 本液晶模块通过外置电阻设置, 此指令失效
(18) 内部设置液晶电压模式	内部设置液晶电压模式	0	1	0	0	0	0	0	1		设置内部电阻微调, 以设置液晶电压, 此两个指令需紧接着使用
	设置的电压值		0	0	<b>6 位电压值数据, 0~63 共 64 级</b>						
(19) 静态图标显示: 开/关	0	1	0	1	0	1	1	0	<b>0</b> <b>1</b>		0: 关, 1: 开。本液晶屏无此图标。此指令在进入及退出睡眠模式时起作用



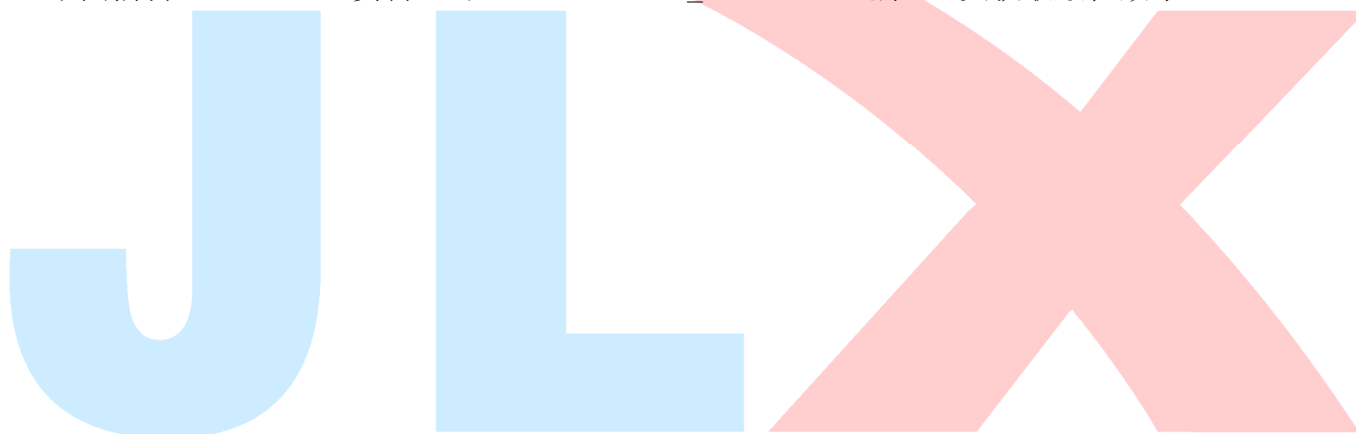
(20) 升压倍数选择 (Booster ratio set)	0	1	1	1	1	1	0	0	0	选择升压倍数: 00: 2 倍, 3 倍, 4 倍 01: 5 倍 11: 6 倍。本模块外部已设置升压倍数为 4 倍, 不必使用此指令
		0	0	0	0	0	0	2 位数设置 升压倍数		
(21) 省电模式 (Power save)										省电模式, 此非一条指令, 是由“(10)显示全部点阵”、(19)静态图标显示: 开/关等指令合成一个“省电功能”。详细看 IC 规格书第 47 页“POWER SAVE”
(22) 空指令 (NOP)	0	1	1	1	0	0	0	1	1	空操作
(23) 测试 (Test)	0	1	1	1	1	*	*	*	*	内部测试用, 千万别用!

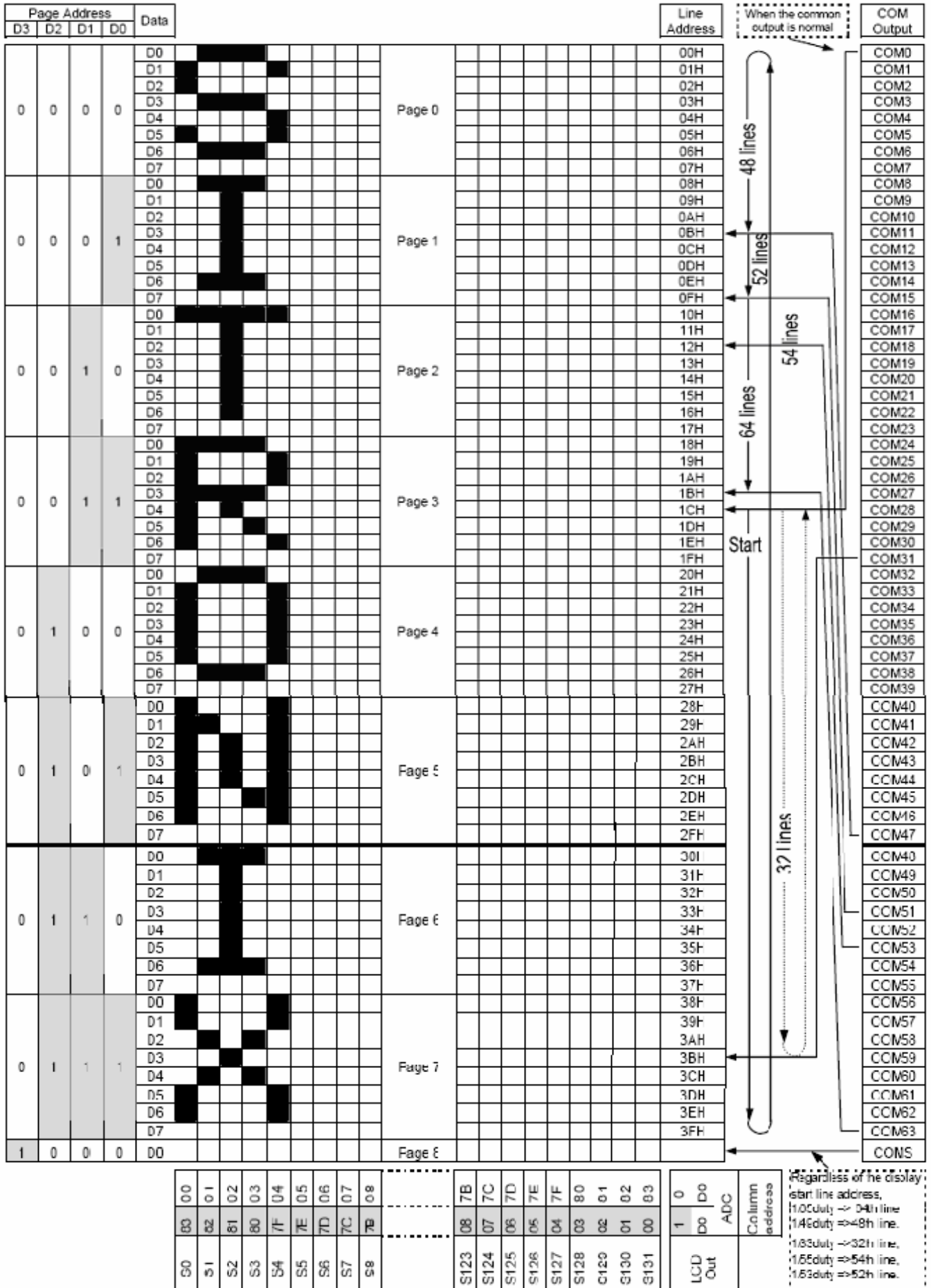
请详细参考 IC 资料“ST7565R\_V15.PDF”的第 42~49 页。

### 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。下图摘自 ST7565R IC 资料, 可通过“ST7565R\_V15.PDF”之第 27 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

### 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:

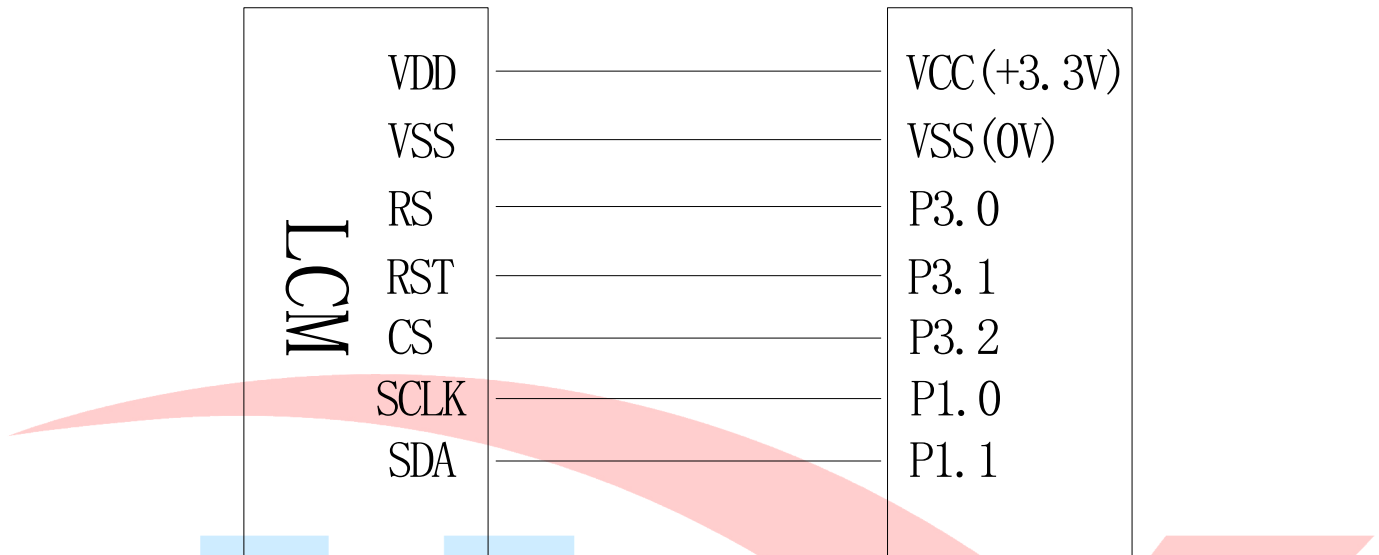


图 8 串行接口

```

/* Test program for JLX12864G-573-PN-S, 串行接口
   驱动 IC 是:ST7565R(or compatible)
   晶联讯电子: 网址 http://www.jlxlcd.cn; http://www.jlxlcd.com.cn
*/
#include <reg51.h>
#include <intrins.h>
#include<string.h>

sbit rs=P3^0;    /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit reset=P1^0; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P1^1;  /*接口定义:lcd_cs1 就是 LCD 的 cs1*/
sbit sclk=P3^2;
sbit sid=P3^1;

#define uchar unsigned char
#define uint unsigned int

void transfer_data(int data1);
void transfer_command(int data1);

void delay(int i);
    
```

```
void Delay1(int i);

uchar code ascii_table_8x16[95][16];
uchar code ascii_table_5x8[95][5];
char code bmp1[];
char code bmp2[];
char code bmp3[];
uchar code zhuang[];
uchar code tai[];
uchar code shi[];
uchar code yong[];
uchar code mon[];
uchar code cheng[];
uchar code gong[];

//=====initial
void initial_lcd()
{
    reset=0;          /*低电平复位*/
    delay(20);
    reset=1;         /*复位完毕*/
    delay(20);
    transfer_command(0xe2); /*软复位*/
    delay(5);
    transfer_command(0x2c); /*升压步聚 1*/
    delay(5);
    transfer_command(0x2e); /*升压步聚 2*/
    delay(5);
    transfer_command(0x2f); /*升压步聚 3*/
    delay(5);
    transfer_command(0x23); /*粗调对比度，可设置范围 0x20~0x27*/
    transfer_command(0x81); /*微调对比度*/
    transfer_command(0x16); /*微调对比度的值，可设置范围 0x00~0x3f*/
    transfer_command(0xa2); /*1/9 偏压比 (bias) */
    transfer_command(0xc8); /*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*/
}

//=====clear all dot martrics=====
void clear_screen()
{
```

```
unsigned char i, j;
for(i=0; i<9; i++)
{
    transfer_command(0xb0+i);
    transfer_command(0x10);
    transfer_command(0x00);
    for(j=0; j<132; j++)
    {
        transfer_data(0x00);
    }
}
}
```

```
/*=====写指令=====*/
```

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

```
{
    char i;
    cs1=0;
    rs=0;
    for(i=0; i<8; i++)
    {
        sclk=0;
        Delay1(2);
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        Delay1(2);
        data1=data1<<=1;
    }
    cs1=1;
}
```

```
/*-----写数据-----*/
```

```
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;
}

void address(uchar page,uchar column)
{
    column=column-1;
    page=page-1;
    transfer_command(0xb0+page);
    transfer_command(((column>>4)&0x0f)+0x10);
    transfer_command(column&0x0f);
}

//=====delay time=====
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<990;k++);
}

//=====delay time=====
void Delay1(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<10;k++);
}

//-----wait a switch, jump out if P2.0 get a signal"0"-----
void waitkey()
{
    repeat:
        if (P2&0x01) goto repeat;
        else delay(60);
        if (P2&0x01) goto repeat;
        else
            delay(400);;
}

//=====display a picture of 128*64 dots=====
void display_graphic(char page, char column, char *dp)
{
    int i, j;
    for(i=0;i<8;i++)
```

```

{
    address(page+i, column);
    for(j=0; j<128; j++)
    {
        transfer_data(*dp);
        dp++;
    }
}
}

```

```

void display_32x32(uchar page, uchar column, uchar reverse, uchar *dp)
{
    uint i, j;
    for(i=0; i<4; i++)
    {
        address(page+i, column);
        for(j=0; j<31; j++)
        {
            if(reverse==1)
                transfer_data(~*dp);
            else
                transfer_data(*dp);
            dp++;
        }
    }
}

```

```

void display_16x16(uchar page, uchar column, uchar reverse, uchar *dp)
{
    uint i, j;
    for(i=0; i<2; i++)
    {
        address(page+i, column);
        for(j=0; j<16; j++)
        {
            if(reverse==1)
                transfer_data(~*dp);
            else
                transfer_data(*dp);
            dp++;
        }
    }
}

```

```
void display_8x16(char page, char column, char reverse, char *dp)
{
    int i, j;
    for(i=0; i<2; i++)
    {
        address(page+i, column);
        for(j=0; j<8; j++)
        {
            if(reverse==1)
                transfer_data(~*dp);
            else
                transfer_data(*dp);
            dp++;
        }
    }
}
```

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

```
{
    uint i=0, j, k, n;
    csl=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;
            for(n=0; n<2; n++)
            {
                address(page+n, column);
                for(k=0; k<8; k++)
                {
                    transfer_data(ascii_table_8x16[j][k+8*n]); /*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+=8;
        }
        else
            i++;
    }
}
```

```
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;
        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*/
        }
        transfer_data(0x00);
        i++;
        column+=6;
    }
    else
        i++;
}
}

//-----main program-----
void main(void)
{
    initial_lcd();
    while(1)
    {
        clear_screen();           //clear all dots
        display_graphic(1, 1, bmp1);    //display a picture of 128*64 dots
        waitkey();
        display_graphic(1, 1, bmp2);    //display a picture of 128*64 dots
        waitkey();
        display_graphic(1, 1, bmp3);    //display a picture of 128*64 dots
        waitkey();
        clear_screen();
        display_32x32(1, (1+32*1), 0, cheng);
        display_32x32(1, (1+32*2), 0, gong);
        display_16x16(6, (1+16*0), 1, zhuang);
        display_16x16(6, (1+16*1), 1, tai);
        display_8x16(6, 33, 1, mon);
        display_16x16(6, 41, 0, shi);
        display_16x16(6, 57, 0, yong);
    }
}

```

```

waitkey();
clear_screen(); //clear all dots
display_string_8x16(1,1," !\"#$%&'()*+,-./"); //括号里的参数分别为（页地址，列地址，要显示的字符），注意：
为了显示双引号（"），在双引号前面加一个斜杠（\），斜杠（\）本身并不显示出来
display_string_8x16(3,1,"0123456789:;<=>?"); //括号里的参数分别为（页地址，列地址，要显示的字符）
display_string_8x16(5,1,"@ABCDEFGHIJKLMNO");
display_string_8x16(7,1,"PQRSTUVWXYZ[\\]^_"); //注意：为了显示斜杠（\），在斜杠（\）前面再加一个斜杠（\），两个
斜杠（\\）只显示一个出来
waitkey();

clear_screen();
display_string_8x16(1,1,"`abcdefghijklmnop`");
display_string_8x16(3,1,"pqrstuvwxyz{|}~`");
display_string_5x8(5,1," !\"#$%&'()*+,-./01234");
display_string_5x8(6,1,"56789:;<=>?@ABCDEFGHI");
display_string_5x8(7,1,"JKLMNOPQRSTUVWXYZ[\\]^");
display_string_5x8(8,1,"_`abcdefghijklmnopqrs`");
waitkey();

}
}

```

```

uchar code cheng[]={
/*-- 文字： 成 --*/
/*-- 宋体 23； 此字体下对应的点阵为： 宽 x 高=31x31 --*/
/*-- 高度不是 8 的倍数，现调整为： 宽度 x 高度=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,0x30,0x00,0x00,0x00,0x00,0x00};

```

```

uchar code gong[]={
/*-- 文字： 功 --*/
/*-- 宋体 23； 此字体下对应的点阵为： 宽 x 高=31x31 --*/
/*-- 高度不是 8 的倍数，现调整为： 宽度 x 高度=32x32 --*/
0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xC0,0xC0,0xC0,0x00,
0x00,0x00,0x00,0xFE,0xFC,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x0C,0x04,0x04,
0x04,0x84,0xFF,0xFF,0x04,0x04,0x04,0x04,0x04,0xFE,0xFE,0x04,0x00,0x00,0x00,0x00,
0xC0,0xC0,0xC0,0xC0,0xE0,0x60,0x7F,0x3F,0x30,0x30,0x10,0x18,0x18,0x88,0xC0,0xF8,
0x7F,0x1F,0x01,0x00,0x00,0x00,0x00,0xF8,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x01,

```

```
0x01, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x30, 0x18, 0x0C, 0x0C, 0x07, 0x03, 0x01, 0x00, 0x04,
0x04, 0x0C, 0x0C, 0x1C, 0x38, 0x3C, 0x1F, 0x0F, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

```
uchar code zhuang[]={
/*-- 文字： 状 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=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};
```

```
uchar code tai[]={
/*-- 文字： 态 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
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};
```

```
uchar code shi[]={
/*-- 文字： 使 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
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};
```

```
uchar code yong[]={
/*-- 文字： 用 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
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};
```

```
uchar code mon[]={
/*-- 文字： ： --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=8x16 --*/
0x00, 0x00, 0x00, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00,
};
```

```
uchar code ascii_table_8x16[95][16]={
//粗体 8x16 点阵的 ASCII 码的点阵数据，从“JLX-GB2312”型号的字库 IC 中读出来的国标的。
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // - (即“空格”) ASCII 码:
0X20
0x00, 0x00, 0x38, 0xFC, 0xFC, 0x38, 0x00, 0x00, 0x00, 0x00, 0x0D, 0x0D, 0x00, 0x00, 0x00, 0x00, // ! - ASCII
码: 0X21
0x00, 0x0E, 0x1E, 0x00, 0x00, 0x1E, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // " -
0x20, 0xF8, 0xF8, 0x20, 0xF8, 0xF8, 0x20, 0x00, 0x02, 0x0F, 0x0F, 0x02, 0x0F, 0x0F, 0x02, 0x00, // # -
0x38, 0x7C, 0x44, 0x47, 0x47, 0xCC, 0x98, 0x00, 0x06, 0x0C, 0x08, 0x38, 0x38, 0x0F, 0x07, 0x00, // $ -
0x30, 0x30, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x00, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x0C, 0x0C, 0x00, // % -
0x80, 0xD8, 0x7C, 0xE4, 0xBC, 0xD8, 0x40, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00, // & -
```

0x00, 0x10, 0x1E, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //-' -	
0x00, 0x00, 0xF0, 0xF8, 0x0C, 0x04, 0x00, 0x00, 0x00, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x00, 0x00, //-( -	
0x00, 0x00, 0x04, 0x0C, 0xF8, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0C, 0x07, 0x03, 0x00, 0x00, //)- -	
0x80, 0xA0, 0xE0, 0xC0, 0xC0, 0xE0, 0xA0, 0x80, 0x00, 0x02, 0x03, 0x01, 0x01, 0x03, 0x02, 0x00, //*- -	ASCII
码: 0X2A	
0x00, 0x80, 0x80, 0xE0, 0xE0, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, //+- -	
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x1E, 0x0E, 0x00, 0x00, 0x00, //- , -	
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //--- -	
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, //-. -	
0x00, 0x00, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x00, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, //-/ -	
0xF8, 0xF8, 0x0C, 0xC4, 0x0C, 0xF8, 0xF0, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x0C, 0x07, 0x03, 0x00, //-0 -	ASCII
码: 0X30	
0x00, 0x10, 0x18, 0xFC, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x00, //-1 -	
0x08, 0x0C, 0x84, 0xC4, 0x64, 0x3C, 0x18, 0x00, 0x0E, 0x0F, 0x09, 0x08, 0x08, 0x0C, 0x0C, 0x00, //-2 -	
0x08, 0x0C, 0x44, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x04, 0x0C, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, //-3 -	
0xC0, 0xE0, 0xB0, 0x98, 0xFC, 0xFC, 0x80, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, //-4 -	ASCII
码: 0X34	
0x7C, 0x7C, 0x44, 0x44, 0x44, 0xC4, 0x84, 0x00, 0x04, 0x0C, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, //-5 -	
0xF0, 0xF8, 0x4C, 0x44, 0x44, 0xC0, 0x80, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, //-6 -	
0x0C, 0x0C, 0x04, 0x84, 0xC4, 0x7C, 0x3C, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x00, //-7 -	
0xB8, 0xFC, 0x44, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, //-8 -	
0x38, 0x7C, 0x44, 0x44, 0x44, 0xFC, 0xF8, 0x00, 0x00, 0x08, 0x08, 0x08, 0x0C, 0x07, 0x03, 0x00, //-9 -	
0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00, //-: -	
0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0E, 0x06, 0x00, 0x00, 0x00, //-; -	
0x00, 0x80, 0xC0, 0x60, 0x30, 0x18, 0x08, 0x00, 0x00, 0x00, 0x01, 0x03, 0x06, 0x0C, 0x08, 0x00, //-< -	
0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, //-= -	
0x00, 0x08, 0x18, 0x30, 0x60, 0xC0, 0x80, 0x00, 0x00, 0x08, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x00, //> -	ASCII
码: 0X3E	
0x18, 0x1C, 0x04, 0xC4, 0xE4, 0x3C, 0x18, 0x00, 0x00, 0x00, 0x00, 0x0D, 0x0D, 0x00, 0x00, 0x00, //-? -	
0xF0, 0xF0, 0x08, 0xC8, 0xC8, 0xF8, 0xF0, 0x00, 0x07, 0x0F, 0x08, 0x0B, 0x0B, 0x0B, 0x01, 0x00, //-@ -	
0xE0, 0xF0, 0x98, 0x8C, 0x98, 0xF0, 0xE0, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, //-A -	ASCII
码: 0X41	
0x04, 0xFC, 0xFC, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x0F, 0x07, 0x00, //-B -	
0xF0, 0xF8, 0x0C, 0x04, 0x04, 0x0C, 0x18, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x08, 0x0C, 0x06, 0x00, //-C -	
0x04, 0xFC, 0xFC, 0x04, 0x0C, 0xF8, 0xF0, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x0C, 0x07, 0x03, 0x00, //-D -	
0x04, 0xFC, 0xFC, 0x44, 0xE4, 0x0C, 0x1C, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x0C, 0x0E, 0x00, //-E -	
0x04, 0xFC, 0xFC, 0x44, 0xE4, 0x0C, 0x1C, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, 0x00, 0x00, //-F -	
0xF0, 0xF8, 0x0C, 0x84, 0x84, 0x8C, 0x98, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x08, 0x07, 0x0F, 0x00, //-G -	
0xFC, 0xFC, 0x40, 0x40, 0x40, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, //-H -	ASCII
码: 0X48	
0x00, 0x00, 0x04, 0xFC, 0xFC, 0x04, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, //-I -	
0x00, 0x00, 0x00, 0x04, 0xFC, 0xFC, 0x04, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x0F, 0x07, 0x00, 0x00, //-J -	

0x04, 0xFC, 0xFC, 0xC0,	0xE0, 0x3C, 0x1C, 0x00,	0x08, 0x0F, 0x0F, 0x00,	0x01, 0x0F, 0x0E, 0x00,	//-k-
0x04, 0xFC, 0xFC, 0x04,	0x00, 0x00, 0x00, 0x00,	0x08, 0x0F, 0x0F, 0x08,	0x08, 0x0C, 0x0E, 0x00,	//-l-
0xFC, 0xFC, 0x38, 0x70,	0x38, 0xFC, 0xFC, 0x00,	0x0F, 0x0F, 0x00, 0x00,	0x00, 0x0F, 0x0F, 0x00,	//-m-
0xFC, 0xFC, 0x38, 0x70,	0xE0, 0xFC, 0xFC, 0x00,	0x0F, 0x0F, 0x00, 0x00,	0x00, 0x0F, 0x0F, 0x00,	//-n-
0xF8, 0xFC, 0x04, 0x04,	0x04, 0xFC, 0xF8, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x08, 0x0F, 0x07, 0x00,	//-o-
0x04, 0xFC, 0xFC, 0x44,	0x44, 0x7C, 0x38, 0x00,	0x08, 0x0F, 0x0F, 0x08,	0x00, 0x00, 0x00, 0x00,	//-p-
0xF8, 0xFC, 0x04, 0x04,	0x04, 0xFC, 0xF8, 0x00,	0x07, 0x0F, 0x08, 0x0E,	0x3C, 0x3F, 0x27, 0x00,	//-q-
0x04, 0xFC, 0xFC, 0x44,	0xC4, 0xFC, 0x38, 0x00,	0x08, 0x0F, 0x0F, 0x00,	0x00, 0x0F, 0x0F, 0x00,	//-r-
0x18, 0x3C, 0x64, 0x44,	0xC4, 0x9C, 0x18, 0x00,	0x06, 0x0E, 0x08, 0x08,	0x08, 0x0F, 0x07, 0x00,	//-s-
0x00, 0x1C, 0x0C, 0xFC,	0xFC, 0x0C, 0x1C, 0x00,	0x00, 0x00, 0x08, 0x0F,	0x0F, 0x08, 0x00, 0x00,	//-t-
0xFC, 0xFC, 0x00, 0x00,	0x00, 0xFC, 0xFC, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x08, 0x0F, 0x07, 0x00,	//-u-
0xFC, 0xFC, 0x00, 0x00,	0x00, 0xFC, 0xFC, 0x00,	0x01, 0x03, 0x06, 0x0C,	0x06, 0x03, 0x01, 0x00,	//-v-
0xFC, 0xFC, 0x00, 0x00,	0x00, 0xFC, 0xFC, 0x00,	0x07, 0x0F, 0x0E, 0x03,	0x0E, 0x0F, 0x07, 0x00,	//-w-
0x0C, 0x3C, 0xF0, 0xE0,	0xF0, 0x3C, 0x0C, 0x00,	0x0C, 0x0F, 0x03, 0x01,	0x03, 0x0F, 0x0C, 0x00,	//-x-
0x00, 0x0C, 0x7C, 0xC0,	0xC0, 0x7C, 0x3C, 0x00,	0x00, 0x00, 0x08, 0x0F,	0x0F, 0x08, 0x00, 0x00,	//-y-
0x1C, 0x0C, 0x84, 0xC4,	0x64, 0x3C, 0x1C, 0x00,	0x0E, 0x0F, 0x09, 0x08,	0x08, 0x0C, 0x0E, 0x00,	//-z-
0x00, 0x00, 0xFC, 0xFC,	0x04, 0x04, 0x00, 0x00,	0x00, 0x00, 0x0F, 0x0F,	0x08, 0x08, 0x00, 0x00,	//-[
0x38, 0x70, 0xE0, 0xC0,	0x80, 0x00, 0x00, 0x00,	0x00, 0x00, 0x00, 0x01,	0x03, 0x07, 0x0E, 0x00,	//-\-
0x00, 0x00, 0x04, 0x04,	0xFC, 0xFC, 0x00, 0x00,	0x00, 0x00, 0x08, 0x08,	0x0F, 0x0F, 0x00, 0x00,	//-]-
0x08, 0x0C, 0x06, 0x03,	0x06, 0x0C, 0x08, 0x00,	0x00, 0x00, 0x00, 0x00,	0x00, 0x00, 0x00, 0x00,	//-^-
0x00, 0x00, 0x00, 0x00,	0x00, 0x00, 0x00, 0x00,	0x20, 0x20, 0x20, 0x20,	0x20, 0x20, 0x20, 0x20,	//-_-
0x00, 0x00, 0x03, 0x07,	0x04, 0x00, 0x00, 0x00,	0x00, 0x00, 0x00, 0x00,	0x00, 0x00, 0x00, 0x00,	//-^-
0x00, 0xA0, 0xA0, 0xA0,	0xE0, 0xC0, 0x00, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x07, 0x0F, 0x08, 0x00,	//-a-
码: 0X61				ASCII
0x04, 0xFC, 0xFC, 0x20,	0x60, 0xC0, 0x80, 0x00,	0x00, 0x0F, 0x0F, 0x08,	0x08, 0x0F, 0x07, 0x00,	//-b-
0xC0, 0xE0, 0x20, 0x20,	0x20, 0x60, 0x40, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x08, 0x0C, 0x04, 0x00,	//-c-
0x80, 0xC0, 0x60, 0x24,	0xFC, 0xFC, 0x00, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x07, 0x0F, 0x08, 0x00,	//-d-
0xC0, 0xE0, 0xA0, 0xA0,	0xA0, 0xE0, 0xC0, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x08, 0x0C, 0x04, 0x00,	//-e-
0x40, 0xF8, 0xFC, 0x44,	0x0C, 0x18, 0x00, 0x00,	0x08, 0x0F, 0x0F, 0x08,	0x00, 0x00, 0x00, 0x00,	//-f-
0xC0, 0xE0, 0x20, 0x20,	0xC0, 0xE0, 0x20, 0x00,	0x27, 0x6F, 0x48, 0x48,	0x7F, 0x3F, 0x00, 0x00,	//-g-
0x04, 0xFC, 0xFC, 0x40,	0x20, 0xE0, 0xC0, 0x00,	0x08, 0x0F, 0x0F, 0x00,	0x00, 0x0F, 0x0F, 0x00,	//-h-
0x00, 0x00, 0x20, 0xEC,	0xEC, 0x00, 0x00, 0x00,	0x00, 0x00, 0x08, 0x0F,	0x0F, 0x08, 0x00, 0x00,	//-i-
0x00, 0x00, 0x00, 0x00,	0x20, 0xEC, 0xEC, 0x00,	0x00, 0x30, 0x70, 0x40,	0x40, 0x7F, 0x3F, 0x00,	//-j-
0x04, 0xFC, 0xFC, 0x80,	0xC0, 0x60, 0x20, 0x00,	0x08, 0x0F, 0x0F, 0x01,	0x03, 0x0E, 0x0C, 0x00,	//-k-
0x00, 0x00, 0x04, 0xFC,	0xFC, 0x00, 0x00, 0x00,	0x00, 0x00, 0x08, 0x0F,	0x0F, 0x08, 0x00, 0x00,	//-l-
0xE0, 0xE0, 0x60, 0xC0,	0x60, 0xE0, 0xC0, 0x00,	0x0F, 0x0F, 0x00, 0x07,	0x00, 0x0F, 0x0F, 0x00,	//-m-
0x20, 0xE0, 0xC0, 0x20,	0x20, 0xE0, 0xC0, 0x00,	0x00, 0x0F, 0x0F, 0x00,	0x00, 0x0F, 0x0F, 0x00,	//-n-
0xC0, 0xE0, 0x20, 0x20,	0x20, 0xE0, 0xC0, 0x00,	0x07, 0x0F, 0x08, 0x08,	0x08, 0x0F, 0x07, 0x00,	//-o-
0x20, 0xE0, 0xC0, 0x20,	0x20, 0xE0, 0xC0, 0x00,	0x40, 0x7F, 0x7F, 0x48,	0x08, 0x0F, 0x07, 0x00,	//-p-
0xC0, 0xE0, 0x20, 0x20,	0xC0, 0xE0, 0x20, 0x00,	0x07, 0x0F, 0x08, 0x48,	0x7F, 0x7F, 0x40, 0x00,	//-q-
0x20, 0xE0, 0xC0, 0x60,	0x20, 0xE0, 0xC0, 0x00,	0x08, 0x0F, 0x0F, 0x08,	0x00, 0x00, 0x00, 0x00,	//-r-
0x40, 0xE0, 0xA0, 0x20,	0x20, 0x60, 0x40, 0x00,	0x04, 0x0C, 0x09, 0x09,	0x0B, 0x0E, 0x04, 0x00,	//-s-

```

0x20, 0x20, 0xF8, 0xFC, 0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x0C, 0x04, 0x00, //~t-
0xE0, 0xE0, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00, //-u-
0x00, 0xE0, 0xE0, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x00, 0x03, 0x07, 0x0C, 0x0C, 0x07, 0x03, 0x00, //-v-
0xE0, 0xE0, 0x00, 0x80, 0x00, 0xE0, 0xE0, 0x00, 0x07, 0x0F, 0x0C, 0x07, 0x0C, 0x0F, 0x07, 0x00, //-w-
0x20, 0x60, 0xC0, 0x80, 0xC0, 0x60, 0x20, 0x00, 0x08, 0x0C, 0x07, 0x03, 0x07, 0x0C, 0x08, 0x00, //-x-
0xE0, 0xE0, 0x00, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x47, 0x4F, 0x48, 0x48, 0x68, 0x3F, 0x1F, 0x00, //-y-

0x60, 0x60, 0x20, 0xA0, 0xE0, 0x60, 0x20, 0x00, 0x0C, 0x0E, 0x0B, 0x09, 0x08, 0x0C, 0x0C, 0x00, //~z- //
0x00, 0x40, 0x40, 0xF8, 0xBC, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x00, //-{-
0x00, 0x00, 0x00, 0xBC, 0xBC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, //-|-
0x00, 0x04, 0x04, 0xBC, 0xF8, 0x40, 0x40, 0x00, 0x00, 0x08, 0x08, 0x0F, 0x07, 0x00, 0x00, 0x00, //-}-
0x08, 0x0C, 0x04, 0x0C, 0x08, 0x0C, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //-~- ASCII 码:
0X7E

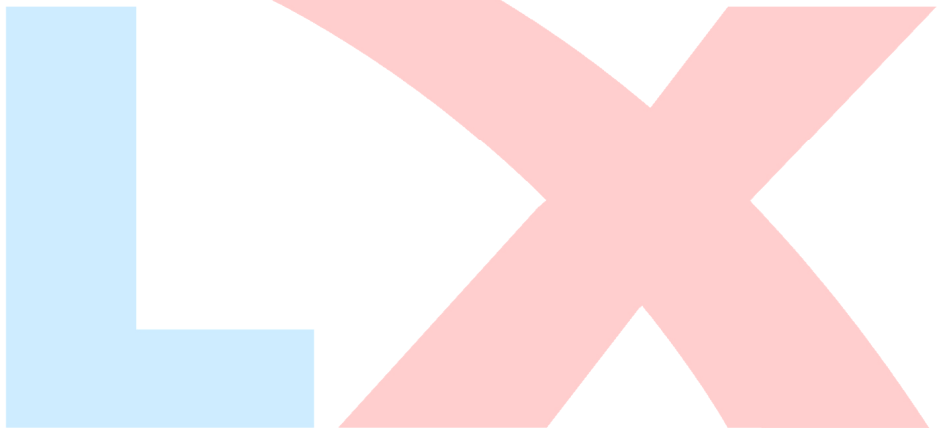
```

};

```

uchar code ascii_table_5x8[95][5]={
/*全体 ASCII 列表:5x8 点阵*/
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, 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, //~
};
    
```

```

char code bmp1[]={
/*-- 调入了一幅图像: E:\work\图片收藏夹\黑白屏图片\JLX12864G-573.bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0xFF, 0x61, 0x07, 0xE1, 0x01, 0x27, 0x23, 0x1B, 0x03, 0xC3, 0x0B, 0x13, 0x33, 0x07, 0x03, 0x01,
0x11, 0x11, 0x11, 0xFF, 0x11, 0x11, 0xFF, 0x01, 0x01, 0xFD, 0x01, 0x01, 0x01, 0xFF, 0x01, 0x01,
0x05, 0x05, 0x05, 0xE5, 0x25, 0x25, 0x25, 0xFF, 0x25, 0x25, 0x25, 0x25, 0xE5, 0x05, 0x05, 0x01,
0x01, 0x01, 0x01, 0x01, 0x7F, 0x2B, 0x2B, 0x2B, 0x2B, 0x2B, 0x2B, 0x7F, 0x01, 0x01, 0x01, 0x01,
0x03, 0xFF, 0x93, 0x93, 0x93, 0xFF, 0x13, 0x11, 0x13, 0x1D, 0xF1, 0x19, 0x17, 0x13, 0x11, 0x01,
0x21, 0x21, 0x2F, 0xE5, 0x01, 0x43, 0x43, 0xFF, 0x43, 0x43, 0x43, 0x03, 0xFF, 0x01, 0x01, 0x01,
0x01, 0x01, 0xF9, 0x49, 0x49, 0x49, 0x49, 0xFF, 0x49, 0x49, 0x49, 0x49, 0xF9, 0x01, 0x01, 0x01,
0x01, 0x01, 0x03, 0x03, 0x03, 0x03, 0x03, 0xE3, 0x13, 0x0B, 0x07, 0x03, 0x01, 0x81, 0x01, 0xFF,
0xFF, 0xFC, 0x03, 0x20, 0x20, 0x11, 0x11, 0x09, 0x05, 0xFF, 0x05, 0x09, 0x19, 0x31, 0x10, 0x00,
0x08, 0x08, 0x04, 0x47, 0x24, 0x18, 0x07, 0x00, 0x00, 0x1F, 0x00, 0x00, 0x00, 0x7F, 0x00, 0x00,
0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x10, 0x20, 0x1F, 0x00, 0x00, 0x00,
0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00,
0x08, 0x1F, 0x08, 0x08, 0x04, 0xFF, 0x05, 0x81, 0x41, 0x31, 0x0F, 0x11, 0x21, 0xC1, 0x41, 0x00,
0x00, 0x00, 0x00, 0x7F, 0x20, 0x10, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x40, 0x38, 0x00,
0x00, 0x00, 0x0F, 0x04, 0x04, 0x04, 0x04, 0x3F, 0x44, 0x44, 0x44, 0x44, 0x4F, 0x40, 0x70, 0x00,
0x01, 0x01, 0x01, 0x01, 0x01, 0x41, 0x81, 0x7F, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08,
    
```



0xF8, 0x08, 0x08, 0x00, 0x08, 0xF8, 0x08, 0x00, 0x00, 0x00, 0x00, 0x08, 0x18, 0x68, 0x80,  
0x80, 0x68, 0x18, 0x08, 0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x70, 0x08, 0x08,  
0x08, 0x88, 0x70, 0x00, 0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0xE0, 0x10, 0x88,  
0x88, 0x18, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0xC0, 0x30, 0x08, 0x08,  
0x08, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x08, 0x88,  
0x88, 0x08, 0x08, 0x00, 0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x30, 0x08, 0x88,  
0x88, 0x48, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x80, 0x80, 0x80,  
0x7F, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x20, 0x30, 0x00, 0x20, 0x30, 0x2C, 0x03,  
0x03, 0x2C, 0x30, 0x20, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00, 0x00, 0x30, 0x28, 0x24,  
0x22, 0x21, 0x30, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00, 0x00, 0x0F, 0x11, 0x20,  
0x20, 0x11, 0x0E, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00, 0x07, 0x18, 0x20, 0x20,  
0x22, 0x1E, 0x02, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x19, 0x21, 0x20,  
0x20, 0x11, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x20, 0x20,  
0x20, 0x11, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00,  
0x00, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00,  
0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x20, 0x20, 0x20, 0x3F,  
0x24, 0x24, 0x24, 0xF4, 0x24, 0x00, 0x00, 0x00, 0xFE, 0x02, 0x12, 0x2A, 0xC6, 0x88, 0xC8, 0xB8,  
0x8F, 0xE8, 0x88, 0x88, 0x88, 0x88, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,  
0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00,  
0x00, 0x20, 0x31, 0x2E, 0x0E, 0x31, 0x20, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,  
0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00, 0x00, 0x40, 0x30, 0x07, 0x12, 0x62, 0x02, 0x0A,  
0x12, 0x62, 0x02, 0x0F, 0x10, 0x60, 0x00, 0xFF, 0x00, 0x02, 0x04, 0x03, 0x04, 0x04, 0x04,  
0x04, 0xFF, 0x04, 0x04, 0x04, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08,  
0x89, 0x4E, 0xAA, 0x18, 0x00, 0xFE, 0x02, 0x02, 0xFA, 0x02, 0x02, 0xFE, 0x00, 0x00, 0x00, 0xFE,  
0x02, 0x02, 0x12, 0x22, 0x22, 0x42, 0x82, 0x62, 0x1E, 0x0A, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00,  
0x00, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0xE0,  
0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x70,  
0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00,  
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, 0x80, 0x80, 0x82, 0x81,  
0x80, 0xFF, 0x80, 0x87, 0xC2, 0xA1, 0x98, 0x86, 0x81, 0xBE, 0xC0, 0xC3, 0xF0, 0x80, 0x80, 0xBF,  
0xA0, 0xB0, 0xA8, 0xA4, 0xA2, 0xA1, 0xA0, 0xA1, 0xA6, 0xBC, 0xA8, 0xA0, 0x80, 0x80, 0x80, 0x80,  
0x80, 0xB0, 0xB0, 0x80, 0x80, 0x80, 0x80, 0x98, 0xA0, 0xA0, 0xA0, 0x91, 0x8E, 0x80, 0x80, 0x8F,  
0x91, 0xA0, 0xA0, 0x91, 0x8E, 0x80, 0x80, 0xA0, 0xB1, 0xAE, 0x8E, 0xB1, 0xA0, 0x80, 0x80, 0xB0,  
0xA8, 0xA4, 0xA2, 0xA1, 0xB0, 0x80, 0x80, 0x8F, 0x90, 0xA0, 0xA0, 0x90, 0x8F, 0x80, 0xA0, 0xBF,  
0xA0, 0x80, 0xBF, 0xA0, 0x80, 0xBF, 0xA0, 0xBF, 0xA0, 0x80, 0xBF, 0xA0, 0x80, 0xBF, 0x80, 0x80,

0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xFF,

};

```
char code bmp2[]={
/*-- 调入了一幅图像: E:\work\图片收藏夹\黑白屏图片\12864 菜单. bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0x10, 0x10, 0x11, 0xF6,
0x12, 0x10, 0x10, 0x10, 0x98, 0x10, 0x00, 0x00, 0x02, 0x02, 0x32, 0x52, 0x92, 0x1F, 0x32, 0x52,
0x12, 0x1F, 0x92, 0x52, 0x32, 0x02, 0x02, 0x00, 0x00, 0x00, 0xF8, 0x28, 0x29, 0x2E, 0x2A, 0xF8,
0x28, 0x2C, 0x2B, 0x2A, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x20, 0x21, 0x21, 0x21, 0x21, 0x21, 0x3F,
0x21, 0x21, 0x21, 0x21, 0x21, 0x31, 0x20, 0x00, 0x42, 0x42, 0x22, 0x12, 0x0A, 0x06, 0xFF,
0x06, 0x0A, 0x12, 0x32, 0x62, 0x22, 0x02, 0x00, 0x08, 0x08, 0x0B, 0x09, 0x09, 0x09, 0x09, 0xFF,
0x09, 0x09, 0x09, 0x09, 0x0B, 0x08, 0x08, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0xBF, 0xBE, 0x31, 0xFB, 0xFF, 0x7F, 0xBF, 0x41, 0x7D, 0x7D, 0x7D, 0x41, 0x3F, 0xBF, 0xBF, 0xFF,
0xFF, 0xDF, 0xD0, 0x56, 0x56, 0x50, 0x16, 0x46, 0x56, 0x50, 0x56, 0x56, 0xD0, 0xDF, 0xFF, 0xFF,
0xF7, 0xF7, 0xF7, 0x00, 0x77, 0xB7, 0xFF, 0x67, 0xB7, 0xD7, 0xF5, 0xD3, 0xB7, 0x27, 0xF7, 0xFF,
0xFF, 0xAF, 0xB0, 0xB5, 0xB7, 0x00, 0xB7, 0xB7, 0xB7, 0xFF, 0x03, 0xFF, 0xFF, 0x00, 0xFF, 0xFF,
0xFF, 0xF7, 0x37, 0xC6, 0xF1, 0xE7, 0x57, 0xB7, 0xBF, 0x0F, 0xE0, 0xED, 0xEF, 0x0F, 0xEF, 0xFF,
0xFF, 0xFF, 0xFF, 0xC1, 0xD5, 0xD5, 0xD5, 0x01, 0xD5, 0xD5, 0xD5, 0xC1, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0x80, 0xDF, 0x6F, 0x7F, 0xBF, 0xBC, 0xD3, 0xEF, 0xEF, 0xD3, 0xBC, 0x3F, 0xBF, 0xFF,
0x7F, 0x7F, 0x7F, 0x00, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x00, 0x7F, 0x7F, 0x7F, 0xFF,
0xFD, 0xBD, 0x7E, 0x80, 0xFF, 0xFF, 0xBF, 0xBD, 0xBD, 0xBD, 0x81, 0xBD, 0xBD, 0xBD, 0xBF, 0xFF,
0xFF, 0xFF, 0xC0, 0xFE, 0xFE, 0x00, 0xDE, 0x9E, 0xC0, 0xFF, 0xF0, 0xBF, 0x7F, 0x80, 0xFF, 0xFF,
0xBF, 0xBE, 0xDE, 0xED, 0xF3, 0xF5, 0xAE, 0xBF, 0xDF, 0xDF, 0xEC, 0xF3, 0xCC, 0x9F, 0xDF, 0xFF,
0xDE, 0xDE, 0xEE, 0xEE, 0xF6, 0xFA, 0xFC, 0x00, 0xFC, 0xFA, 0xF6, 0xF6, 0xEE, 0xCE, 0xEE, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x00, 0x00, 0x00, 0x3E, 0x2A, 0xEA, 0x2A, 0x2A, 0x2A, 0xEA, 0x2A, 0x3E, 0x00, 0x00, 0x00, 0x00,
0x00, 0x20, 0x20, 0x22, 0x22, 0x22, 0x22, 0xE2, 0x22, 0x22, 0x22, 0x22, 0x22, 0x20, 0x20, 0x00,
0x40, 0x41, 0xCE, 0x04, 0x00, 0x80, 0x40, 0xBE, 0x82, 0x82, 0x82, 0xBE, 0xC0, 0x40, 0x40, 0x00,
0x00, 0x20, 0x2F, 0xA9, 0xA9, 0xAF, 0xE9, 0xB9, 0xA9, 0xAF, 0xA9, 0xA9, 0x2F, 0x20, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```





```
0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91,
0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91,
0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91,
0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x90, 0x90, 0x90, 0x90, 0x9F, 0x80, 0x80, 0x80, 0x80, 0xFF
};
```

