

JLX12832G-520-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 | 技术参数 | 5 |
| 6 | 时序特性 | 6~7 |
| 7 | 指令功能及硬件接口与编程案例 | 7~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-520 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-520 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-520 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙，焊接式 FPC。

2.2 IC 采用矽创公司 ST7567, 功能强大，稳定性好

2.3 功耗低:1~100mW（关掉背光：[0.3mA@3.3V](#), 打开背光不大于 100mW）；

2.4 显示内容：

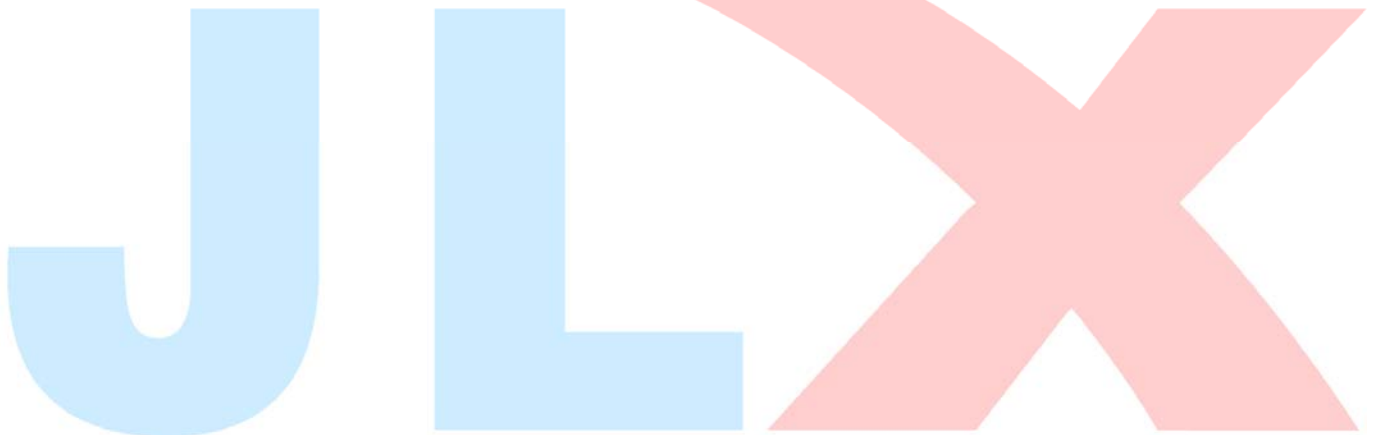
- 128*32 点阵单色图片；

- 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*2 行。按照 12*12 点阵汉字来计算可显示 10 字/行*2 行。

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:可采用 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

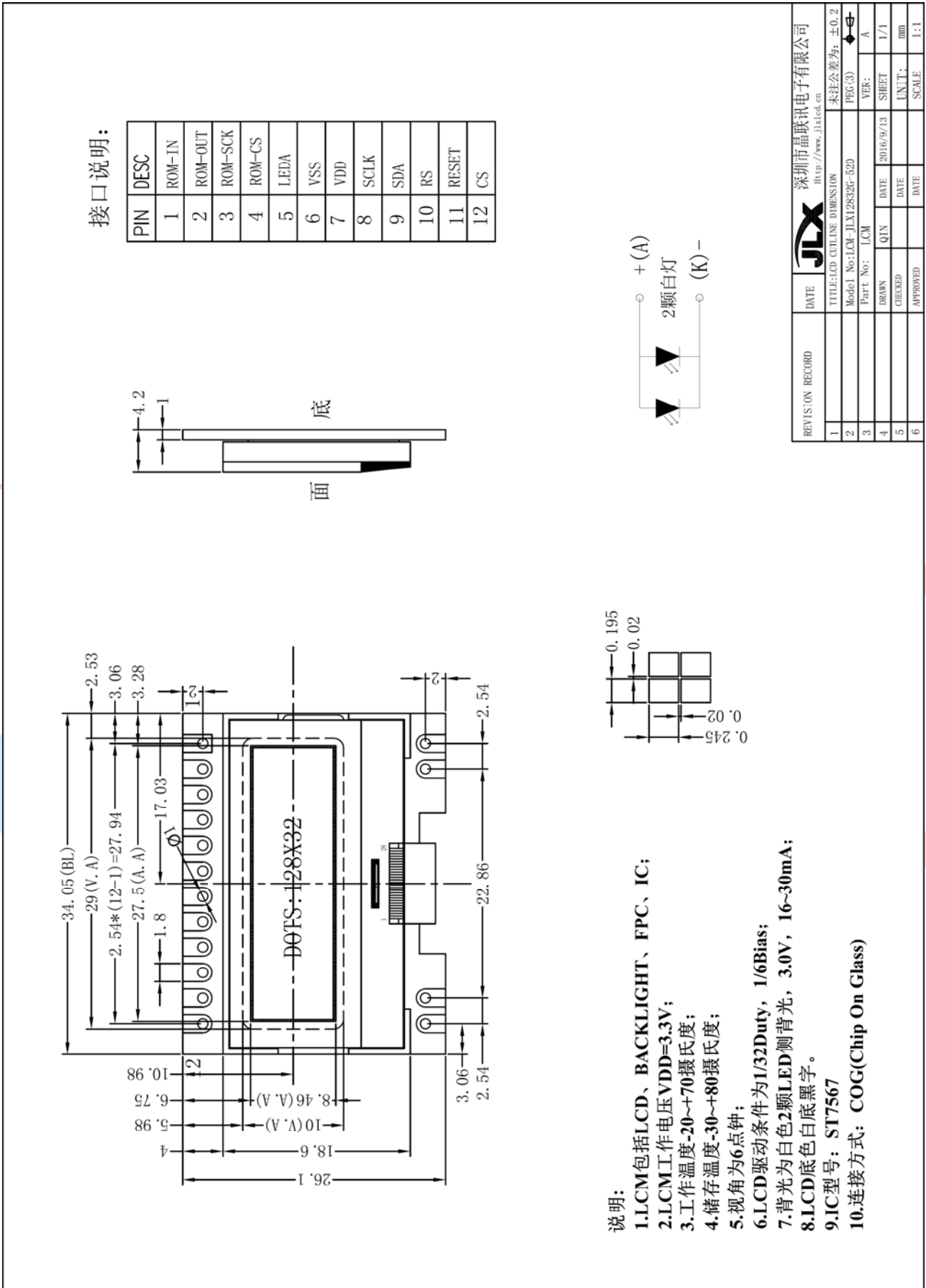


图 1. 外形尺寸

模块的接口引脚功能

| 引线号 | 符号 | 名称 | 功能 |
|-----|------|--------|----------------------------------|
| 1 | NC | NC | 空 |
| 2 | NC | NC | 空 |
| 3 | NC | NC | 空 |
| 4 | NC | NC | 空 |
| 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 | RES | 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 128×32 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 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下：
 背光板白色。
 正常工作电流为：30~40mA (LED 灯数共 2 颗)；
 工作电压：3.0V；

4.2 工作电图：

图 2 是 JLX12832G-520-PN 图像点阵型模块的电路框图, 它由驱动 IC ST7567 及几个电阻电容成。

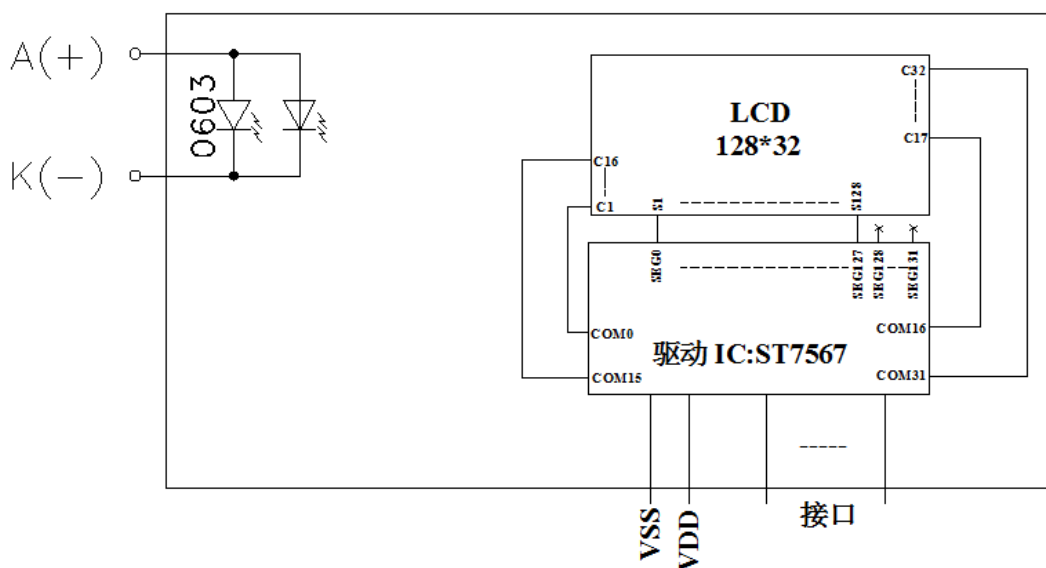


图 2: JLX12832G-520 图像点阵型液晶模块的电路框图

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 | V0、XV0 | | | 13.5 | V |
| 静电电压 | | - | - | 100 | 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 | 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 | I00 = 1.2mA | - | | 0.4 | V |
| 模块工作电流 | IDD | VDD = 3.0V | - | | 1.0 | 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)

System Bus Timing for 4-Line Serial Interface

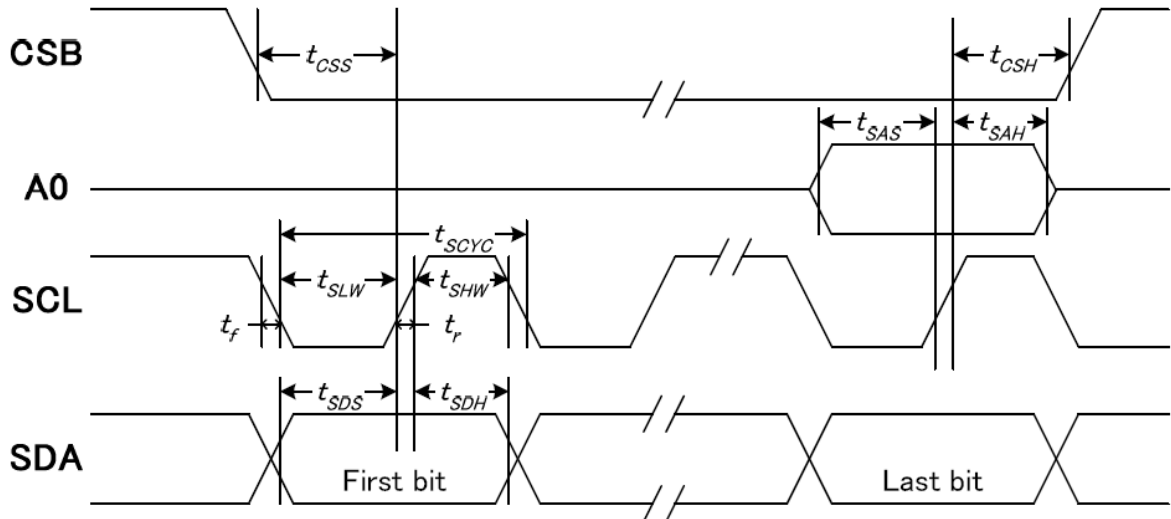


图 4. 从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

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 |

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

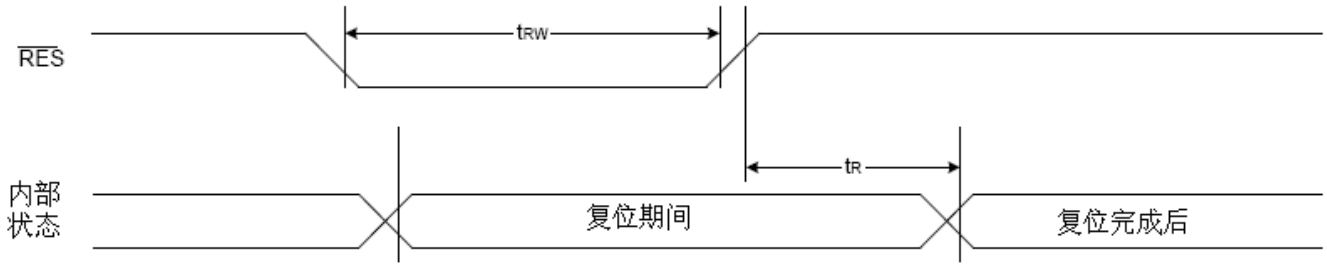


图 7：电源启动后复位的时序

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
格式:

| 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 | 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 100 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 | 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 | 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 | 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 | 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 | 设置偏压比： 0xA2 : BIAS=1/9 (常用) 0xA3 : BIAS=1/7 | |
| (12) 软件复位 (Reset) | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0xE2 :软件复位。 | |
| (13) 行扫描顺序选择 (Common output mode select) | | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | |
| (14) 电源控制 (Power control set) | | 0 | 0 | 1 | 0 | 1 | 电压操作模式选择，共3位 | | | 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。 | |
| (15) 选择内部电阻比例 | 0 | 0 | 0 | 1 | 0 | 0 | | | | 内部电压值电阻设置 | |
| (16) | 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 | 0 | 0 | 6 位电压值数据，0~63 共 64 级 | | | | | | | |
| (17)静态图标显示： 开/关 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 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 | |
| (18) 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 | |
| (19) 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” | |

| | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|--------------|
| (20)空指令 (NOP) | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 空操作 |
| (21) 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_V1.7.PDF”的第 21~28 页。

7.3 点阵与 DD RAM(显示数据存储器)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*32 点阵的屏分为 4 个“页”, 从第 0“页”到第 3“页”。

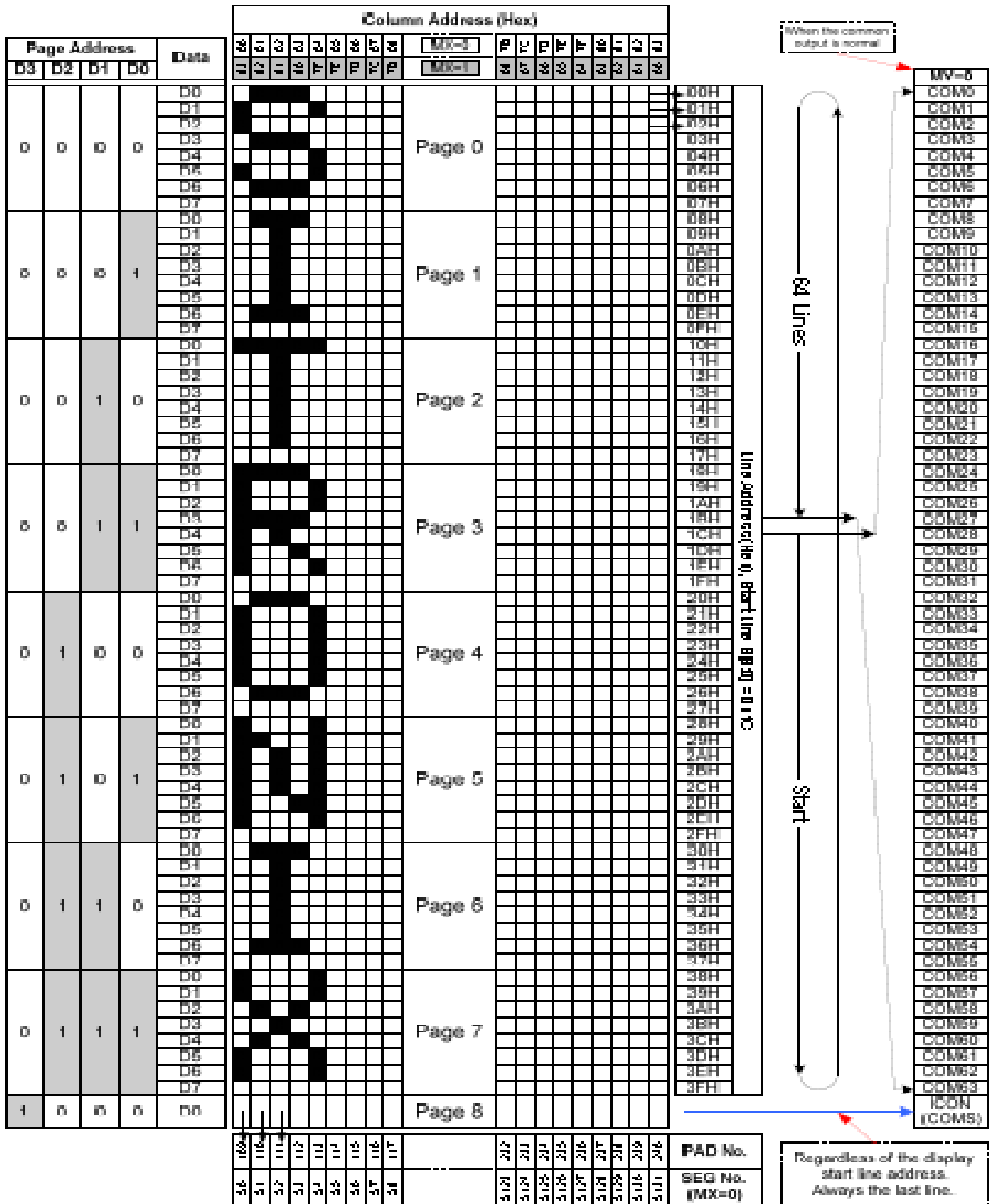
DB7--DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面, 最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵, 通常“1”代表点亮该点阵, “0”代表关掉该点阵. 如下图所示:

| | | | | | | |
|----|---|---|---|---|--|---|
| 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)



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:

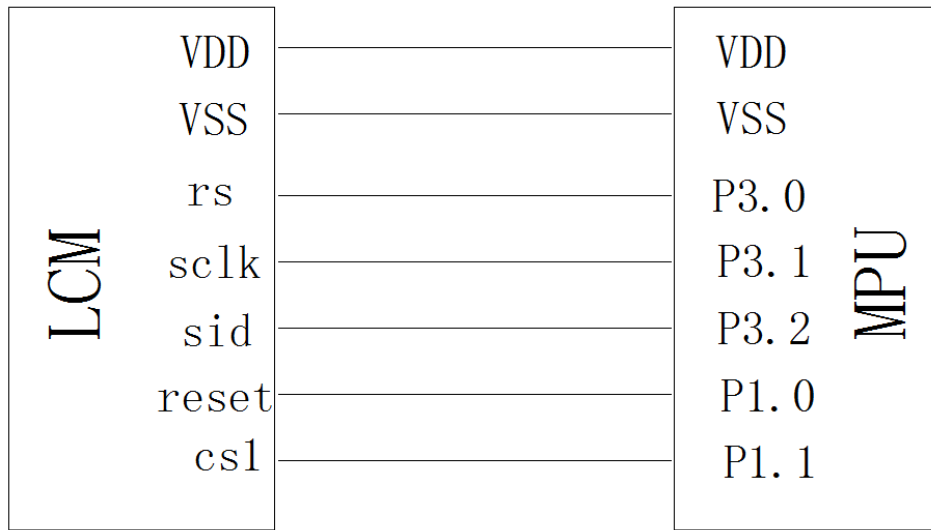


图 8. 串行接口

```

/* JLX12832G-520 测试程序****/
/* LCD 驱动 IC:ST7567*****/
/* 晶联讯电子：公司网址：http://www.jlxlcd.cn； 阿里巴巴网址：http://www.jlxlcd.com.cn*/
/* 该程序显示 2 行中文如下：*/
/* 全套液晶解决方案*/
/* 质量取胜创建口碑*/
/* 该程序显示 2 行数字如下：*/
/* 0123456789012345*/
/* 0123456789012345*/

```

```

#include <reg51.H>

sbit cs1=P1^1; //接口定义，按 51 系列单片机，P1 口接 DB0~DB7
sbit reset=P1^0;
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;

```

```

void transfer_data(int data1);
void transfer_command(int data1);
char code quan[];
char code tao[];
char code ye[];
char code jing[];
char code jie[];
char code jue[];
char code fang[];
char code an[];
char code lian[];
char code xun[];
char code mo[];
char code kuai[];
char code chang[];
char code num0[];
char code num1[];
char code num2[];
char code num3[];
char code num4[];
char code num5[];

```

```

char code num6[];
char code num7[];
char code num8[];
char code num9[];

char code num11[];
char code num12[];
char code num13[];
char code num14[];
char code num15[];
char code num16[];
char code num17[];
char code num18[];

char code graphic1[];
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];

void displaygraphic(char *dp);
void delay(int i);
void delay1(int i);
void disp_grap(int line, int column, char *dp);
void disp_char(int line, int column, char *dp);
void initial_lcd();
void clear_screen();
void waitkey();

//=====initial
void initial_lcd()
{
    reset=0;           //Reset the chip when reset=0
    delay(20);
    reset=1;
    transfer_command(0xe2); //软复位*/
    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(0x21); //粗调对比度, 可设置范围 20~27*/
    transfer_command(0x81); //微调对比度*/
    transfer_command(0x0a); //微调对比度的值, 可设置范围 0~63*/
    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: 从第一行开始*/
    transfer_command(0xaf); /*开显示*/
}

//=====clear all dot martrics=====
void clear_screen()
{
    unsigned char i, j;

    for(i=0; i<4; i++)
    {
        cs1=0;
        transfer_command(0xb0+i);
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0; j<132; j++)
        {

```

```

        transfer_data(0x00);
    }
}
/*在指定行和列位置显示指定的汉字（16*16 点阵的汉字）*/
void disp_grap(int line,int column,char *dp)
{
    int i,j,k,col_l,col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x10;
        col_h=k&0xf0;
        col_h=col_h>>4;
        col_l=k&0x0f;
        transfer_command(0x10+col_h);
        transfer_command(0x00+col_l);
        for(j=0;j<16;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

```

/*在指定行和列位置显示指定的字母、数字（8*16 点阵的）*/
void disp_char(int line,int column,char *dp)
{
    int i,j,k,col_l,col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x08;
        col_h=k&0xf0;
        col_h=col_h>>4;
        col_l=k&0x0f;
        transfer_command(0x10+col_h);
        transfer_command(0x00+col_l);
        for(j=0;j<8;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

```

//=====display a picture of 128*64 dots=====
void displaygraphic(char *dp)
{
    int i,j;
    for(i=0;i<4;i++)
    {
        cs1=0;
        transfer_command(0xb0+i); //set page address,
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0;j<128;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

```
}

```

```
/*=====写指令=====*/

```

```
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;
        delay1(5);
        sclk=1;
        delay1(5);
        data1=data1<<=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;

    }
}

```

```
//=====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(5);
if (P2&0x01) goto repeat;
else;
delay(40);
}

//=====main program=====
void main(void)
{

while(1)
{
initial_lcd();
clear_screen(); //clear all dots
displaygraphic(graphic1); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic2); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic3); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic4); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
disp_grap(1,1,quan); /*在第1行第1列显示一个 16*16 点阵汉字:全*/
//delay(20);
disp_grap(1,2,tao); /*在第1行第2列显示一个 16*16 点阵汉字:套*/
//delay(20);
disp_grap(1,3,ye); /*显示一个 16*16 点阵汉字:液*/
//delay(20);
disp_grap(1,4,jing); /*显示一个 16*16 点阵汉字:晶*/
//delay(20);
disp_grap(1,5,jie); /*显示一个 16*16 点阵汉字:解*/
//delay(20);
disp_grap(1,6,jue); /*显示一个 16*16 点阵汉字:决*/
//delay(20);
disp_grap(1,7,fang); /*显示一个 16*16 点阵汉字:方*/
//delay(20);
disp_grap(1,8,an); /*显示一个 16*16 点阵汉字:案*/
//delay(20);
disp_grap(2,1,jing); /*显示一个 16*16 点阵汉字:晶*/
//delay(20);
disp_grap(2,2,lian); /*显示一个 16*16 点阵汉字:联*/
//delay(20);
disp_grap(2,3,xun); /*显示一个 16*16 点阵汉字:讯*/
//delay(20);
disp_grap(2,4,ye); /*显示一个 16*16 点阵汉字:液*/
//delay(20);
disp_grap(2,5,jing); /*显示一个 16*16 点阵汉字:晶*/
//delay(20);
disp_grap(2,6,mo); /*显示一个 16*16 点阵汉字:模*/
//delay(20);
disp_grap(2,7,kuai); /*显示一个 16*16 点阵汉字:块*/
//delay(20);
disp_grap(2,8,chang); /*显示一个 16*16 点阵汉字:厂*/
waitkey();
clear_screen(); //clear all dots
disp_char(1,1,num0); /*在第1行第1列显示一个 8*16 点阵数字:0*/
disp_char(1,2,num1); /*在第1行第2列显示一个 8*16 点阵数字:1*/
disp_char(1,3,num2); /*在第1行第3列显示一个 8*16 点阵数字:2*/
disp_char(1,4,num3); /*在第1行第4列显示一个 8*16 点阵数字:3*/
disp_char(1,5,num4); /*在第1行第5列显示一个 8*16 点阵数字:4*/
disp_char(1,6,num5); /*以下类同*/
```

```

disp_char(1,7,num6);
disp_char(1,8,num7);
disp_char(1,9,num8);
disp_char(1,10,num9);
disp_char(1,11,num0);
disp_char(1,12,num1);
disp_char(1,13,num2);
disp_char(1,14,num3);
disp_char(1,15,num4);
disp_char(1,16,num5);
disp_char(2,1,num0); /*在第2行第1列显示一个 8*16 点阵数字:0*/
disp_char(2,2,num1); /*在第2行第2列显示一个 8*16 点阵数字:1*/
disp_char(2,3,num2); /*在第2行第3列显示一个 8*16 点阵数字:2*/
disp_char(2,4,num3); /*在第2行第4列显示一个 8*16 点阵数字:3*/
disp_char(2,5,num4); /*在第2行第5列显示一个 8*16 点阵数字:4*/
disp_char(2,6,num5); /*以下类同*/
disp_char(2,7,num6);
disp_char(2,8,num7);
disp_char(2,9,num8);
disp_char(2,10,num9);
disp_char(2,11,num0);
disp_char(2,12,num1);
disp_char(2,13,num2);
disp_char(2,14,num3);
disp_char(2,15,num4);
disp_char(2,16,num5);
waitkey();
disp_char(1,1,num11); /*在第1行第1列显示一个 8*16 点阵数字:0*/
disp_char(1,2,num12); /*在第1行第2列显示一个 8*16 点阵数字:1*/
disp_char(1,3,num13); /*在第1行第3列显示一个 8*16 点阵数字:2*/
disp_char(1,4,num14); /*在第1行第4列显示一个 8*16 点阵数字:3*/
disp_char(1,5,num15); /*在第1行第5列显示一个 8*16 点阵数字:4*/
disp_char(1,6,num16); /*以下类同*/
disp_char(1,7,num17);
disp_char(1,8,num18);
waitkey();
}
}
//-----
char code quan[]={
/*-- 文字: 全 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00,0x80,0x40,0x60,0x50,0x48,0x44,0xC3,0x44,0x48,0x50,0x70,0x60,0x20,0x00,0x00,
0x00,0x40,0x40,0x44,0x44,0x44,0x44,0x7F,0x44,0x44,0x44,0x46,0x44,0x60,0x40,0x00
};

char code tao[]={
/*-- 文字: 套 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x40,0x44,0x24,0x24,0xF4,0x5C,0x57,0x54,0x54,0x5C,0x54,0x24,0x64,0xC4,0x44,0x00,
0x04,0x04,0x44,0x64,0x57,0x4D,0x45,0x65,0x25,0x25,0x35,0xE4,0x44,0x04,0x04,0x00
};

char code ye[]={
/*-- 文字: 液 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x10,0x61,0x06,0xE0,0x18,0x84,0xE4,0x1C,0x84,0x65,0xBE,0x24,0xA4,0x64,0x04,0x00,
0x04,0x04,0xFF,0x00,0x01,0x00,0xFF,0x41,0x21,0x12,0x0C,0x1B,0x61,0xC0,0x40,0x00
};

char code jing[]={
/*-- 文字: 晶 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00,0x00,0x00,0x00,0x7E,0x2A,0x2A,0x2A,0x2A,0x2A,0x2A,0x7E,0x00,0x00,0x00,0x00,

```



```
0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00
};
```

```
char code jie[]={
/*-- 文字： 解 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x08, 0xF4, 0x57, 0x54, 0xFC, 0x54, 0xF0, 0x42, 0xA2, 0x1E, 0x02, 0xA2, 0x62, 0x3E, 0x00, 0x00,
0x80, 0x7F, 0x02, 0x02, 0x5F, 0x82, 0x7F, 0x0A, 0x09, 0x09, 0x09, 0xFF, 0x09, 0x09, 0x09, 0x00
};
```

```
char code jue[]={
/*-- 文字： 决 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x02, 0x04, 0xD8, 0x08, 0x00, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,
0x02, 0xFE, 0x01, 0x80, 0x41, 0x21, 0x11, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00
};
```

```
char code fang[]={
/*-- 文字： 方 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0xF9, 0x4A, 0x4C, 0x48, 0x48, 0xC8, 0x08, 0x08, 0x08, 0x00,
0x40, 0x40, 0x20, 0x10, 0x0C, 0x03, 0x00, 0x00, 0x20, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00
};
```

```
char code an[]={
/*-- 文字： 案 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x00, 0x20, 0x2C, 0x24, 0x64, 0x74, 0xAD, 0xA6, 0xE4, 0x34, 0x24, 0x24, 0x2C, 0x24, 0x00, 0x00,
0x00, 0x24, 0x24, 0x25, 0x15, 0x15, 0x0D, 0xFE, 0x04, 0x0D, 0x17, 0x14, 0x24, 0x64, 0x24, 0x00
};
```

```
char code lian[]={
/*-- 文字： 联 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x02, 0xFE, 0x92, 0x92, 0x92, 0xFE, 0x12, 0x11, 0x12, 0x1C, 0xF0, 0x18, 0x17, 0x12, 0x10, 0x00,
0x08, 0x1F, 0x08, 0x08, 0x04, 0xFF, 0x05, 0x81, 0x41, 0x31, 0x0F, 0x11, 0x21, 0xC1, 0x41, 0x00
};
```

```
char code xun[]={
/*-- 文字： 讯 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x20, 0x21, 0x2E, 0xE4, 0x00, 0x42, 0x42, 0xFE, 0x42, 0x42, 0x42, 0x02, 0xFE, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x7F, 0x20, 0x10, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x40, 0x38, 0x00
};
```

```
char code mo[]={
/*-- 文字： 模 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x10, 0xD0, 0xFF, 0x50, 0x90, 0x04, 0xF4, 0x54, 0x5F, 0x54, 0x54, 0x5F, 0xF4, 0x04, 0x00, 0x00,
0x03, 0x00, 0xFF, 0x00, 0x00, 0x84, 0x85, 0x45, 0x35, 0x0F, 0x15, 0x25, 0x65, 0xC4, 0x44, 0x00
};
```

```
char code kuai[]={
/*-- 文字： 块 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x10, 0x10, 0xFF, 0x10, 0x10, 0x00, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,
0x08, 0x18, 0x0F, 0x04, 0x85, 0x41, 0x31, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00
};
```

```
char code chang[]={
/*-- 文字： 厂 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x00, 0x00, 0xFE, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00,
0x40, 0x30, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

```
char code num0[]={
/*-- 文字： 0 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=8x16 --*/
```

```
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x10, 0x0F, 0x00
};
char code num1[]={
/*-- 文字: 1 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00
};
char code num2[]={
/*-- 文字: 2 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00
};
char code num3[]={
/*-- 文字: 3 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00
};
char code num4[]={
/*-- 文字: 4 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00
};

char code num5[]={
/*-- 文字: 5 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0xF8, 0x08, 0x88, 0x88, 0x08, 0x08, 0x00, 0x00, 0x19, 0x21, 0x20, 0x20, 0x11, 0x0E, 0x00
};

char code num6[]={
/*-- 文字: 6 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00
};
char code num7[]={
/*-- 文字: 7 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00
};
char code num8[]={
/*-- 文字: 8 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00
};

char code num9[]={
/*-- 文字: 9 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00
};

char code num11[]={
/*-- 文字: a --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x19, 0x24, 0x22, 0x22, 0x22, 0x3F, 0x20,
};

char code num12[]={
/*-- 文字: b --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x08, 0xF8, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,
};

char code num13[]={
/*-- 文字: c --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
```

```
0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x20, 0x11, 0x00,
};
```

```
char code num14[]={
/*-- 文字: d --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x00, 0x00, 0x80, 0x80, 0x88, 0xF8, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x10, 0x3F, 0x20,
};
```

```
char code num15[]={
/*-- 文字: A --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x00, 0x00, 0xC0, 0x38, 0xE0, 0x00, 0x00, 0x00, 0x20, 0x3C, 0x23, 0x02, 0x02, 0x27, 0x38, 0x20,
};
```

```
char code num16[]={
/*-- 文字: B --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x08, 0xF8, 0x88, 0x88, 0x88, 0x70, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,
};
```

```
char code num17[]={
/*-- 文字: C --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0xC0, 0x30, 0x08, 0x08, 0x08, 0x08, 0x38, 0x00, 0x07, 0x18, 0x20, 0x20, 0x20, 0x10, 0x08, 0x00,
};
```

```
char code num18[]={
/*-- 文字: D --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=8x16 --*/
0x08, 0xF8, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x10, 0x0F, 0x00,
};
```

```
char code graphic1[]={
/*-- 调入了一幅图像: E:\显示图案收藏\JLX12832G-520 图片.bmp --*/
/*-- 宽度 x 高度=128x32 --*/
0x00, 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, 0x00, 0x08, 0x18, 0x68, 0x80,
0x80, 0x68, 0x18, 0x08, 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, 0x30, 0x08, 0x88,
0x88, 0x48, 0x30, 0x00, 0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 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, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0xE0, 0x10, 0x08,
0x08, 0x10, 0xE0, 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,
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, 0x18, 0x20, 0x20,
0x20, 0x11, 0x0E, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00, 0x07, 0x18, 0x20, 0x20,
0x22, 0x1E, 0x02, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x19, 0x21, 0x20,
0x20, 0x11, 0x0E, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00, 0x00, 0x0F, 0x10, 0x20,
0x20, 0x10, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x10, 0x61, 0x06, 0xE0, 0x00, 0x26, 0x22, 0x1A, 0x02, 0xC2, 0x0A, 0x12, 0x32, 0x06, 0x02, 0x00,
0x10, 0x10, 0x10, 0xFE, 0x10, 0x10, 0xFE, 0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0xFE, 0x00, 0x00,
0x04, 0x04, 0x04, 0xE4, 0x24, 0x24, 0x25, 0xFE, 0x24, 0x24, 0x24, 0x24, 0xE4, 0x04, 0x04, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7E, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x7E, 0x00, 0x00, 0x00, 0x00,
0x02, 0xFE, 0x92, 0x92, 0x92, 0xFE, 0x12, 0x11, 0x12, 0x1C, 0xF0, 0x18, 0x17, 0x12, 0x10, 0x00,
0x20, 0x21, 0x2E, 0xE4, 0x00, 0x42, 0x42, 0xFE, 0x42, 0x42, 0x42, 0x02, 0xFE, 0x00, 0x00, 0x00,
0x00, 0x00, 0xF8, 0x48, 0x48, 0x48, 0xFF, 0x48, 0x48, 0x48, 0x48, 0xF8, 0x00, 0x00, 0x00,
```