# JLX128128G-81202-BN 使用说明书

## 目　录

## 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 JLX128128G-81202 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。

JLX128128G-81202 可以显示 128×128 点阵单色或 4 灰度级的图片，或显示 8 个×8 行=64 个的 16*16 点阵的汉字，或显示 16 个×8 行=128 个的 8*16 点阵的英文、数字、符号。或显示 21 个×16 行的 5*8 点阵的英文、数字、符号。

## 2．JLX128128G-81202 图像型点阵液晶模块的特性

1.1 结构牢：带挡墙背光;
1.2 IC 采用 ST7571,功能强大，稳定性好
1.3 功耗低:1 - 100mW（不开背光 1mW <3.3V@0.3mA>,开背光不大于 100mW<3.3V@30mA>）;
1.4 显示内容:
- ●128*128 点阵单色图片或 4 灰度级的图片，
- ●或显示 8 个×8 行=64 个的 16*16 点阵的汉字，按照 12*12 点阵汉字来计算可显示 10 字/行*10 行。
- ●或显示 16 个×8 行=128 个的 8*16 点阵的英文、数字、符号。
- ●或显示 21 个×16 行的 5*8 点阵的英文、数字、符号。
- ●可选用 16*16 点阵或其他点阵的图片来自编汉字,也可配合晶联讯字库 IC(JLX-GB2312)来显示汉字。

1.5 指令功能强;
1.6 接口简单方便:可选 I²C 总线、4 线 SPI 串口、并口（6800 时序或 8080 时序可选择）。
1.7 工作温度:-20℃ - 70℃;
1.8 可靠性高。

## 3. 外形尺寸及接口引脚功能

### 3.1 外形尺寸图

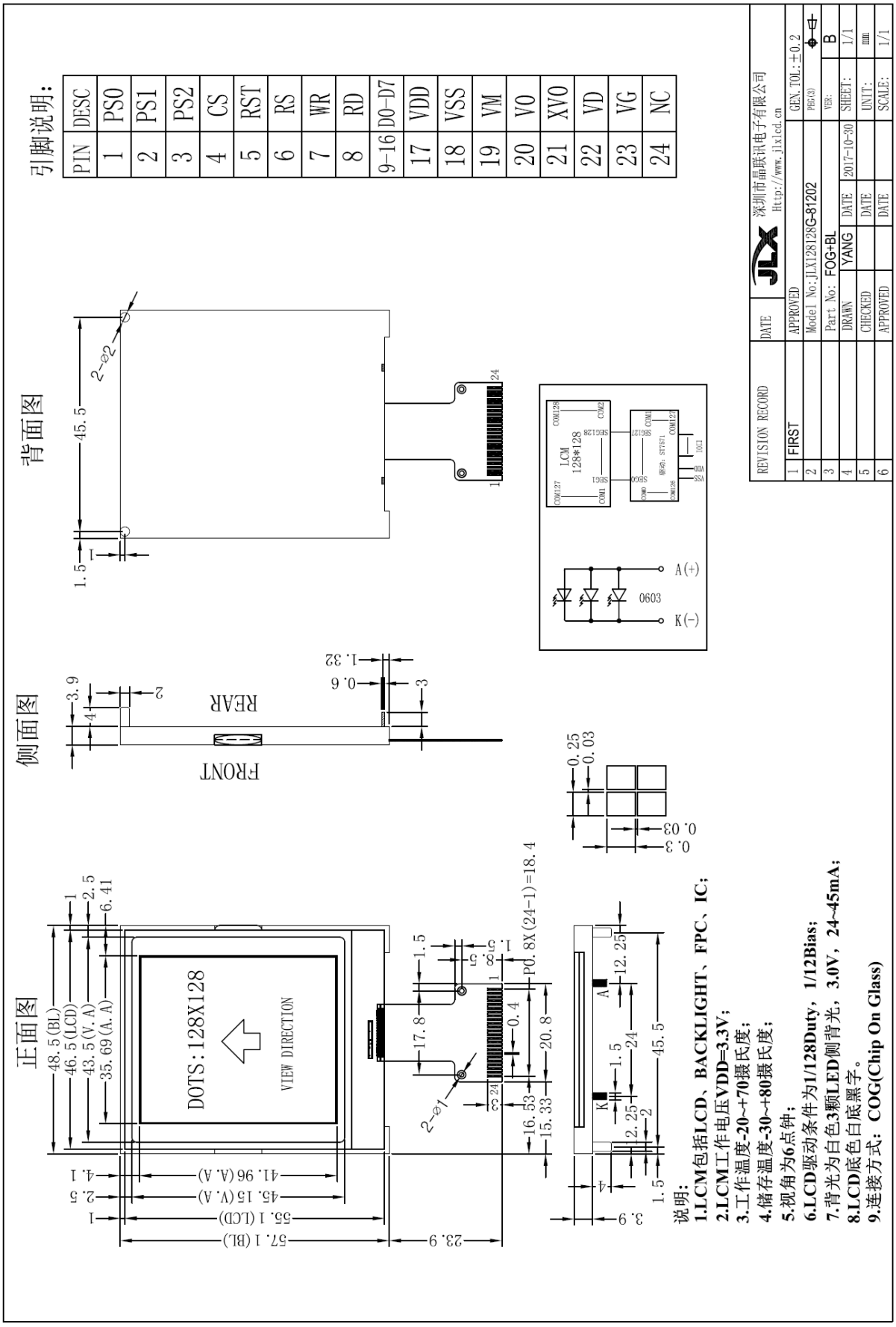



图 1. 外形尺寸

## 3.2 模块的接口引脚功能

### 3.2.1 并行时接口引脚功能

| 引 线 号 | 符 号 | 名 称 | 功 能 |
|---|---|---|---|
| 1 | PS0 | I | H:接高电平 |
| 2 | PS1 | I | H:接高电平 |
| 3 | PS2 | I | L:接低电平 |
| 4 | CS | 片选 | 低电平片选 |
| 5 | 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 |
| 6 | A0(RS) | 寄存器选择信号 | H:数据寄存器 0:指令寄存器（IC 资料上所写为“CD”） |
| 7 | RW(WR) | 读/写 | 6800 时序：H:读数据 L:写数据 |
| 8 | E（RD） | 使能信号 | 6800 时序：使能信号 |
| 9～16 | D0～D7 | I/O | 并行接口时，数据总线 DB0～DB7 |
| 17 | VDD | 电路电源 | 供电电源正极 |
| 18 | VSS | 接地 | 0V |
| 19 | VM | | VM 与 VSS 之间接一个电容 |
| 20 | V0 | 倍压电路 | V0 与 XV0 之间接一个电容 |
| 21 | XV0 | 倍压电路 | |
| 22 | VD | | VD 与 VSS 之间接一个电容，并且 VD 与 VDD 之间接一个电阻 |
| 23 | VG | 偏置电压 | LCD 偏置驱动电压，VG 与 VSS 之间接一个电容 |
| 24 | NC | | |

表 1：模块并行接口引脚功能

### 3.2.2 四线串行时接口引脚功能

| 引 线 号 | 符 号 | 名 称 | 功 能 |
|---|---|---|---|
| 1 | PS0 | I | L:接低电平 |
| 2 | PS1 | I | H:接高电平 |
| 3 | PS2 | I | L:接低电平 |
| 4 | CS | 片选 | 低电平片选 |
| 5 | 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 |
| 6 | A0(RS) | 寄存器选择信号 | H:数据寄存器 0:指令寄存器（IC 资料上所写为“CD”） |
| 7 | RW(WR) | 读/写 | 串行接口，RW 接高电平 |
| 8 | E（RD） | 使能信号 | 串行接口，RD 接高电平 |
| 9～14 | D0～D5 | I/O | 串行接口，D0-D5 接高电平 |
| 15 | D6(SCK) | I/O | 串行时钟 |
| 16 | D7(SDA) | I/O | 串行数据 |
| 17 | VDD | 电路电源 | 供电电源正极 |
| 18 | VSS | 接地 | 0V |
| 19 | VM | | VM 与 VSS 之间接一个电容 |
| 20 | V0 | 倍压电路 | V0 与 XV0 之间接一个电容 |
| 21 | XV0 | 倍压电路 | |
| 22 | VD | | VD 与 VSS 之间接一个电容，并且 VD 与 VDD 之间接一个电阻 |
| 23 | VG | 偏置电压 | LCD 偏置驱动电压，VG 与 VSS 之间接一个电容 |
| 24 | NC | | |

表 2：4 线 SPI 串行接口引脚功能

3.2.3 I²C 总线时接口引脚功能

| 引 线 号 | 符 号 | 名 称 | 功 能 |
|---|---|---|---|
| 1 | PS0 | I | L:接低电平 |
| 2 | PS1 | I | L:接低电平 |
| 3 | PS2 | I | H:接高电平 |
| 4 | CS | 片选 | I²C 接口，CS 不用，此引脚建议接高电平 |
| 5 | 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 |
| 6 | A0(RS) | 寄存器选择信号 | I²C 接口，RS 不用，此引脚建议接高电平 |
| 7 | RW(WR) | 读/写 | I²C 接口，WR 不用，此引脚建议接高电平 |
| 8 | E（RD） | 使能信号 | I²C 接口，RD 不用，此引脚建议接高电平 |
| 9～10 | D0～D1 | I/O | I²C 接口， DB0～DB1 接低电平 |
| 11～15 | D2-D6(SDA) | I/O | I²C 接口，串行数据 |
| 16 | D7(SCK) | I/O | I²C 接口，串行时针 |
| 17 | VDD | 电路电源 | 供电电源正极 |
| 18 | VSS | 接地 | 0V |
| 19 | VM | | VM 与 VSS 之间接一个电容 |
| 20 | V0 | 倍压电路 | V0 与 XV0 之间接一个电容 |
| 21 | XV0 | 倍压电路 | |
| 22 | VD | | 通过 1uF 电容连接 VDD，并且 VD 与 VDD 之间接一个电阻 |
| 23 | VG | 偏置电压 | 通过 1uF 电容连接 VSS |
| 24 | NC | | |

表 3：I²C 总线接口引脚功能

## 4. 基本原理

### 4.1 液晶屏（LCD）

在 LCD 上排列着 128×128 点阵,128 个列信号与驱动 IC 相连，128 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 工作电图：

图 1 是 JLX128128G-81202 图像点阵型模块的电路框图,它由驱动 IC ST7571 及几个电阻电容组成。

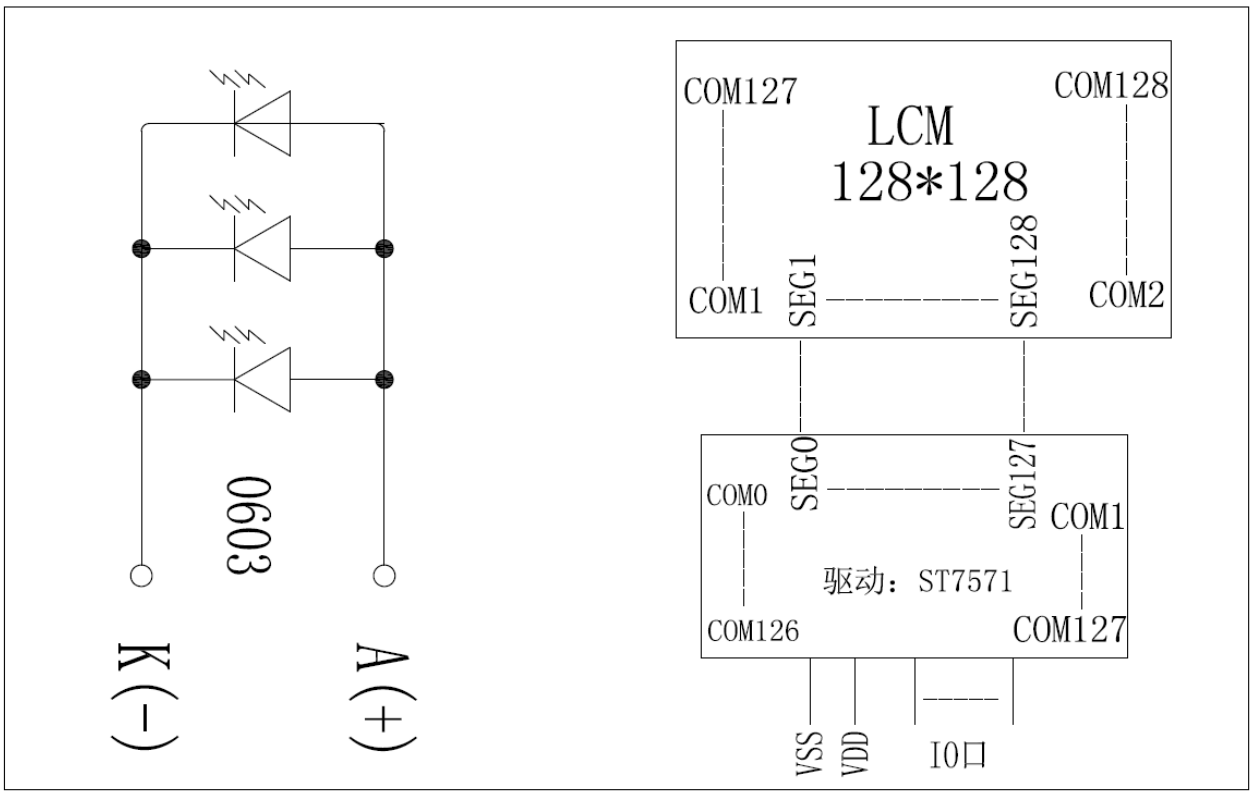



图 2:JLX128128G-81202 图像点阵型液晶模块的电路框图

## 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∽+70° C;
存储温度:-30∽+80° C;
背光板选用白色;
正常工作电流为: 24∽45mA;
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 | | -20 | | +70 | ℃ |
| 储存温度 | | -30 | | +80 | ℃ |

表 5：最大极限参数

## 5.2 直流（DC）参数

**可以选择 3.3V 供电及 5.0V 供电两种方式：**

| 名 称 | 符 号 | 测 试 条 件 | 标 准 值 | | | 单位 |
|---|---|---|---|---|---|---|
| | | | MIN | TYPE | MAX | |
| 工作电压 | VIN | 3.3V 供电 | 2.7 | 3.3 | 3.4 | 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 = 3.3V | - | | 0.3 | mA |
| 背光工作电流 | ILED | VLED=3.0V | 24 | 45 | 60 | mA |

表 6：直流（DC）参数

## 6. 读写时序特性（AC 参数）

### 6.1 4 线 SPI 串行接口写时序特性（AC 参数）

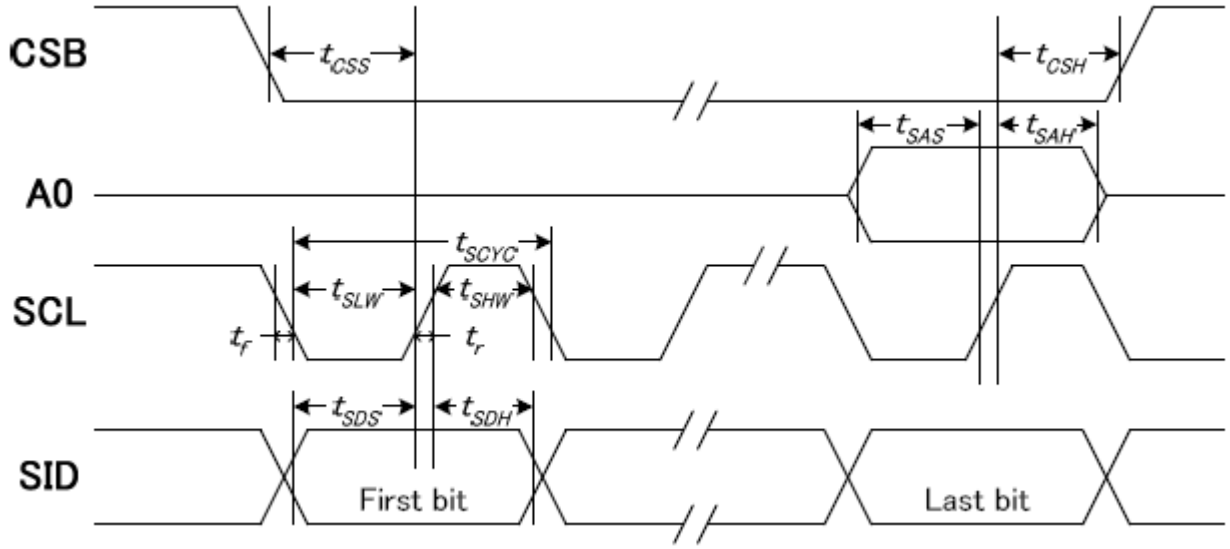



图 3. 从 CPU 写到 ST7571（Writing Data from CPU to ST7571）

**表 7. 写数据到 ST7571 的时序要求**

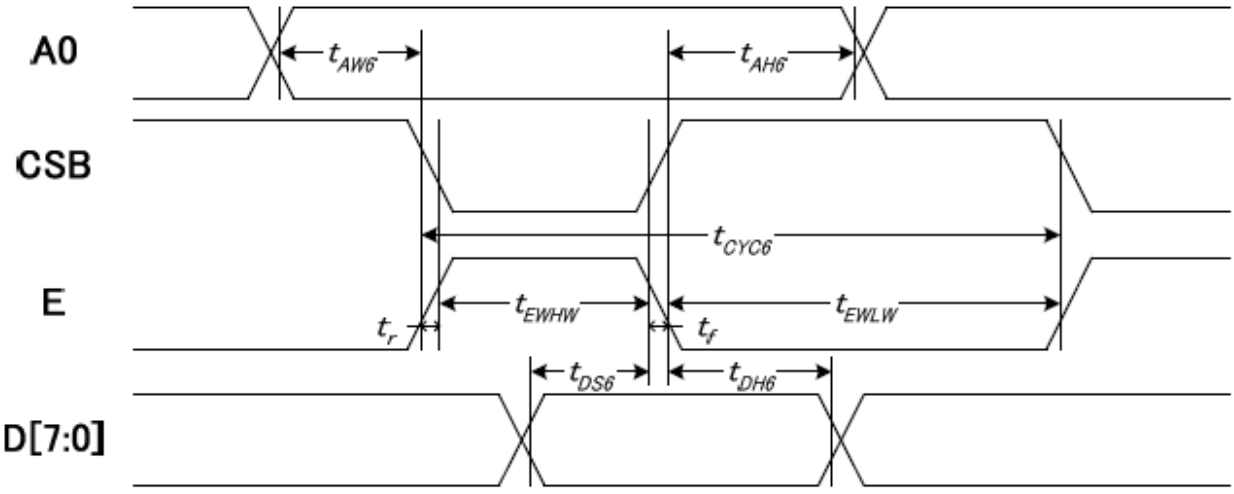
| 项 目 | 符 号 | 测试条件 | 极限值 | | | 单位 |
|---|---|---|---|---|---|---|
| | | | MIN | TYPE | MAX | |
| 4线 SPI串口时钟周期<br>（4-line SPI Clock Period） | tSCYC | 引脚：SCL | 200 | -- | -- | ns |
| 保持SCK高电平脉宽<br>（SCL “H” pulse width） | tSHW | | 80 | -- | -- | ns |
| 保持SCLK低电平脉宽<br>（SCL “L” pulse width） | tSLW | | 80 | -- | -- | ns |
| 地址建立时间<br>(Address setup time) | tSAS | 引脚：A0 | 60 | -- | -- | ns |
| 地址保持时间<br>(Address hold time) | tSAH | | 30 | -- | -- | ns |
| 数据建立时间<br>(Data setup time) | tSDS | 引脚：SID | 60 | -- | -- | ns |
| 数据保持时间<br>(Data hold time) | tSDH | | 30 | -- | -- | ns |
| 片选信号建立时间<br>（CS-SCL time） | tCSS | 引脚：CSB | 40 | -- | -- | ns |
| 片选信号保持时间<br>（CS-SCL time） | tCSH | | 100 | -- | -- | ns |

VDD =1.8~3.3V±5%, Ta = -30~85℃

输入信号的上升和下降时间（TR，TF）在 15 纳秒或更少的规定。

所有的时间，用 20%和 80%作为标准规定的测定。

## 6.2 6800 时序并行接口的时序特性（AC 参数）




从 CPU 写到 ST7571（Writing Data from CPU to ST7571）

**图 4. 写数据到 ST7571 的时序要求（6800 系列 MPU）**

**表 8. 读写数据的时序要求**

| 项 目 | 符 号 | 名称 | 极限值 | | | 单位 |
|---|---|---|---|---|---|---|
| | | | MIN | TYPE | MAX | |
| 地址保持时间 | A0 | tAH6 | 0 | | -- | ns |
| 地址建立时间 | | tAW6 | 0 | | -- | ns |
| 系统循环时间 | E | tCYC6 | 500 | | -- | ns |
| 使能“低”脉冲宽度 | | tEWLW | 250 | | -- | ns |
| 使能“高”脉冲宽度 | | tEWHW | 250 | | -- | ns |
| 写数据建立时间 | DB[7：0] | tDS6 | 80 | | -- | ns |
| 写数据保持时间 | | tDH6 | 30 | | -- | ns |

VDD =1.8~3.3V±5%, Ta = -30~85℃

输入信号的上升时间和下降时间（TR，TF）是在 15 纳秒或更少的规定。当系统循环时间非常快，

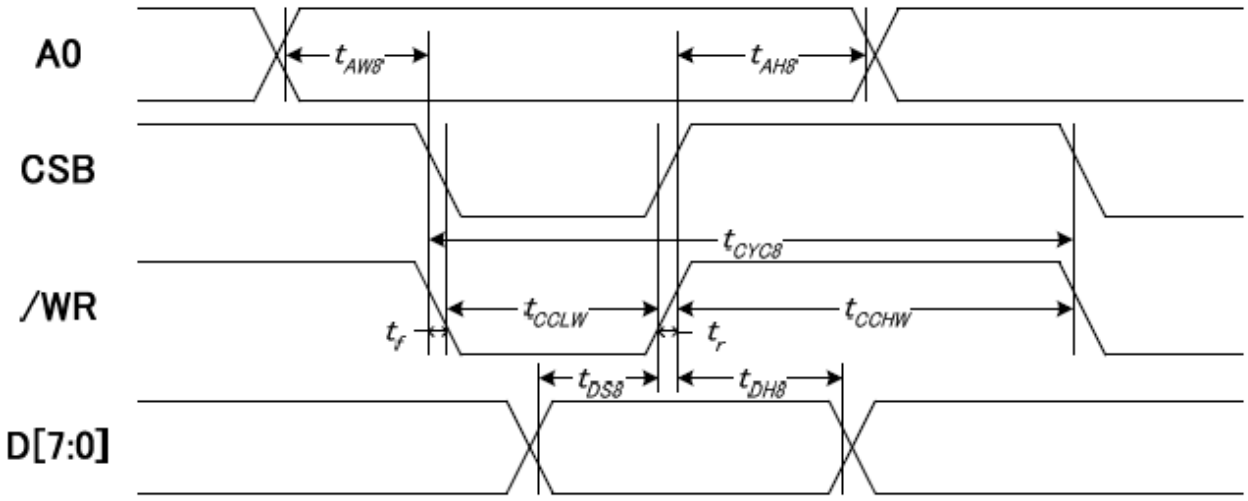
（TR + TF）≤（tcyc6－tewlw－tewhw）指定。

所有的时间，用 20%和 80%作为参考指定的测定。

tewlw 指定为重叠的 CSB“H”和“L”。

R / W 信号总是“H”

## 6.3 8080 时序并行接口的时序特性（AC 参数）




**从 CPU 写到 ST7571（Writing Data from CPU to ST7571）**

**图 4. 写数据到 ST7571 的时序要求（8080 系列 MPU）**

**表 8. 读写数据的时序要求**

| 项 目 | 符 号 | 名称 | 极限值 | | | 单位 |
|---|---|---|---|---|---|---|
| | | | MIN | TYPE | MAX | |
| 地址保持时间 | A0 | tAH8 | 0 | | -- | ns |
| 地址建立时间 | | tAW8 | 0 | | -- | ns |
| 系统循环时间 | /WR | tCYC8 | 500 | | -- | ns |
| 使能“低”脉冲宽度 | | tCCLW | 250 | | -- | ns |
| 使能“高”脉冲宽度 | | tCCHW | 250 | | -- | ns |
| 写数据建立时间 | DB | tDS8 | 80 | | -- | ns |
| 写数据保持时间 | | tDH8 | 30 | | -- | ns |

VDD =1.8~3.3V±5%, Ta = -30~85℃

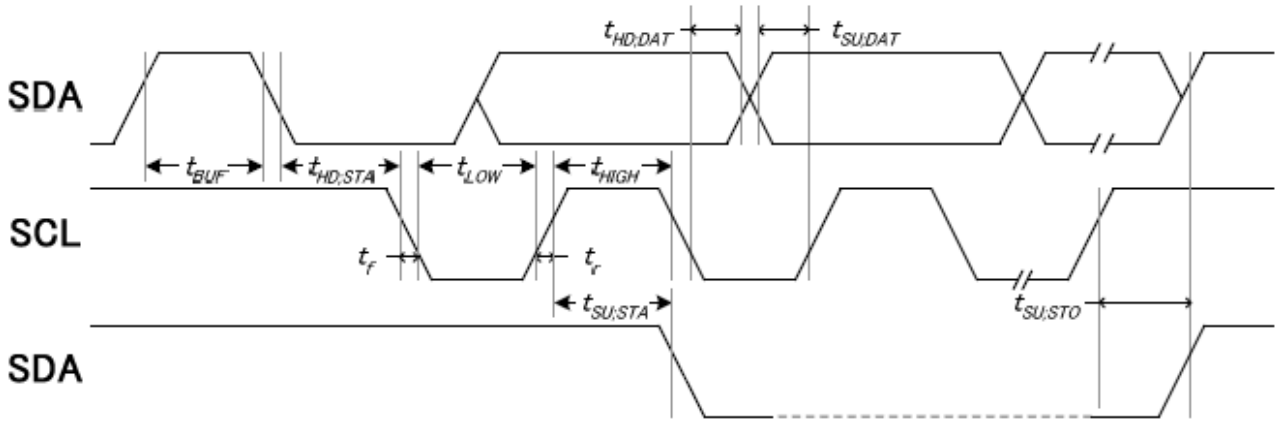
输入信号的上升时间和下降时间（TR，TF）是在 15 纳秒或更少的规定。当系统循环时间非常快，

（TR + TF）≤（tcyc8－tcclw－tcchw）指定。

所有的时间，用 20%和 80%作为参考指定的测定。

tcclw 被指定为“L”之间的重叠 CSB 和/ WR 处于“L”级

## 6.3 $I^2C$ 接口的时序特性（AC 参数）




从 CPU 写到 ST7571（Writing Data from CPU to ST7571）

图 4. 写数据到 ST7571 的时序要求（$I^2C$ 系列 MPU）

表 8. 读写数据的时序要求

| 项 目 | 符 号 | 名称 | 极限值 | | | 单位 |
|---|---|---|---|---|---|---|
| | | | MIN | TYPE | MAX | |
| SCL时钟频率 | CSL | FSCLK | -- | | 400 | kUZ |
| SCL时钟的低周期 | CSL | TLOW | 1.3 | | -- | us |
| SCL时钟周期 | CSL | THIGH | 0.6 | | -- | us |

| | | | | | | |
|---|---|---|---|---|---|---|
| 数据保持时间 | SDA | TSU;Data | 100 | | -- | ns |
| 数据建立时间 | SDA | THD;Data | 0 | | 0.9 | us |
| SCL，SDA 的上升时间 | SCL | TR | 20+0.1Cb | | 300 | ns |
| SCL，SDA 下降时间 | SCL | TF | 20+0.1Cb | | 300 | ns |
| 每个总线为代表的电容性负载 | | Cb | -- | | 400 | pF |
| 一个重复起始条件设置时间 | SDA | TSU;SUA | 0.6 | | -- | us |
| 启动条件的保持时间 | SDA | THD;STA | 0.6 | | -- | us |
| 为停止条件建立时间 | | TSU;STO | 0.6 | | -- | us |
| 容许峰值宽度总线 | | TSW | -- | | 50 | ns |
| 开始和停止条件之间的总线空闲时间 | SCL | TBUF | 1.3 | | | us |

VDD =1.8~3.3V±5%,Ta = -30~85℃

注：

所有的时间，用 20%和 80%作为标准规定的测定。

这是推荐的操作 I C 接口与 VDD1 高于 2.6V。

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

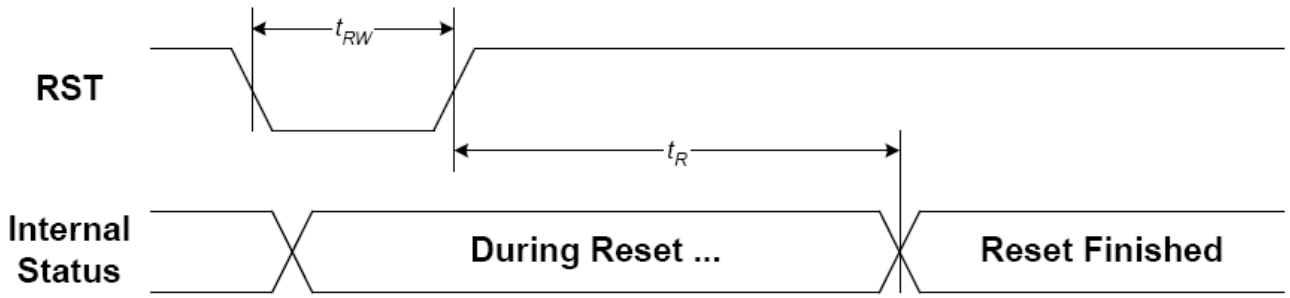



图 5：**电源启动后复位的时序**

**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 | | 120 | -- | -- | ms |
| 复位保持低电平的时间 | tRW | 引脚：RESET | 2.0 | -- | -- | us |

## 7．指令功能：

### 7. 1 指令表

| Instruction | A0 | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set Mode | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2-byte instruction | 9.1.1 |
| | 0 | 0 | FR3 | FR2 | FR1 | FR0 | BE1 | BE0 | -- | 0 | FR[3:0]: Set frame frequency<br>BE[1:0]: Set booster efficiency | |
| Write Display Data | 1 | 0 | Write data | | | | | | | | Write data into DDRAM | 9.1.2 |
| Set Icon | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | ION | ION=0: Disable Icon function<br>ION=1: Enable Icon function and set Page Address = 16 | 9.1.3 |
| Set Page Address | 0 | 0 | 1 | 0 | 1 | 1 | P3 | P2 | P1 | P0 | Set Page Address | 9.1.4 |
| Set Column Address (MSB) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X7 | X6 | X5 | Set MSB of Column Address | 9.1.5 |
| Set Column Address (LSB) | 0 | 0 | 0 | 0 | 0 | 0 | X4 | X3 | X2 | X1 | Set LSB of Column Address | 9.1.6 |
| Display ON/OFF | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | D | D=0: Display OFF<br>D=1: Display ON | 9.1.7 |
| Set Display Start Line | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -- | -- | 2-byte instruction. Specify Line Address for the 1st display line of DDRAM (vertical scrolling). | 9.1.8 |
| | 0 | 0 | -- | S6 | S5 | S4 | S3 | S2 | S1 | S0 | | |
| Set COM0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | -- | -- | 2-byte instruction. Specify a COM pin to be COM0 (moving partial display window). | 9.1.9 |
| | 0 | 0 | -- | C6 | C5 | C4 | C3 | C2 | C1 | C0 | | |
| Set Display Duty | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | -- | -- | 2-byte instruction. Set display duty | 9.1.10 |
| | 0 | 0 | L7 | L6 | L5 | L4 | L3 | L2 | L1 | L0 | | |
| Set N-line Inversion | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | -- | -- | 2-byte instruction. Set N-line inversion counter | 9.1.11 |
| | 0 | 0 | -- | -- | -- | N4 | N3 | N2 | N1 | N0 | | |
| Release N-line Inversion | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | Exit N-line inversion mode | 9.1.12 |
| Reverse Display | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | REV | REV=0: Normal display<br>REV=1: Reverse display | 9.1.13 |
| Entire Display ON | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | EON | EON=0: Normal display<br>EON=1: Entire display ON | 9.1.14 |

| Instruction | A0 | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Section |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Power Control | 0 | 0 | 0 | 0 | 1 | 0 | 1 | VC | VR | VF | Set internal power ON/OFF | 9.1.15 |
| Select Regulator Register | 0 | 0 | 0 | 0 | 1 | 0 | 0 | R2 | R1 | R0 | Select internal Regulator resistor | 9.1.16 |
| Set Contrast | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2-byte instruction. Select EV for internal Regulator's reference | 9.1.17 |
| | 0 | 0 | -- | -- | EV5 | EV4 | EV3 | EV2 | EV1 | EV0 | | |
| Select LCD bias | 0 | 0 | 0 | 1 | 0 | 1 | 0 | B2 | B1 | B0 | Select LCD bias | 9.1.18 |
| Set COM Scan Direction | 0 | 0 | 1 | 1 | 0 | 0 | MY | -- | -- | -- | Set COM scan direction:<br>MY=0: Normal direction<br>MY=1: Reverse direction | 9.1.19 |
| Set SEG Scan Direction | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | MX | Set SEG scan direction:<br>MX=0: Normal direction<br>MX=1: Reverse direction | 9.1.20 |
| Oscillator ON | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | Turn ON internal Oscillator | 9.1.21 |
| Set Power-Save Mode | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | P | P=0: Normal mode<br>P=1: Enable Power-Save mode | 9.1.22 |
| Release Power-Save Mode | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | Exit Power-Save mode | 9.1.23 |
| RESET | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | Software reset | 9.1.24 |
| Set Display Data Length | -- | -- | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2-byte instruction. Set the data counter in 3-Line SPI only | 9.1.25 |
| | -- | -- | DL7 | DL6 | DL5 | DL4 | DL3 | DL2 | DL1 | DL0 | | |
| NOP | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | No operation | 9.1.26 |
| Reserved | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Do NOT use | -- |
| Reserved | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | Do NOT use | -- |
| Reserved | 0 | 0 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | Reserved for testing | -- |
| Extension Command Set1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | TE1 | TE1=1: Enter extension Mode1 | 9.1.27 |
| Extension Command Set2 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | TE2 | TE2=1: Enter extension Mode2 | 9.1.28 |
| Extension Command Set3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | TE3 | TE3=1: Enter extension Mode3 | 9.1.29 |

| Instruction | A0 | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EXTENSION COMMAND SET 1 | | | | | | | | | | | |
| Increase Vop offset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Increase vop offset by 1step |
| Decrease Vop offset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Decrease vop offset by 1 step |
| Return normal mode | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Return normal mode |
| EXTENSION COMMAND SET 2 | | | | | | | | | | | |
| Disable autoread | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Disable autoread |
| Enter EEPROM mode | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Enter EEPROM mode |
| Enable read mode | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | Enable read mode |
| Set read pulse | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | Set read pulse width |
| Exit EEPROM mode | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Exit EEPROM mode |
| Enable erase mode | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | Enable erase mode |
| Set erase pulse | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | Set erase pulse width |
| Enable write mode | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | Enable write mode |
| Set write pulse | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | Set write pulse width |
| Return normal mode | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Return normal mode |
| EXTENSION COMMAND SET 3 | | | | | | | | | | | |
| Set Color Mode | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B/G | Select Black/White or Gray mode<br>B/G=1: Black/White mode;<br>B/G=0: Gray mode (default) |
| Return normal mode | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Return normal mode |

Note: Do NOT use non-specified instructions in any extension command mode.

**表 8. 指令表**

请详细参考 IC 资料”ST7571.PDF”。

### 7.2 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*128 点阵的屏分为 8 个“页”，从第 0“页”到第 15“页”。

DB7--DB0 的排列方向：数据是从上向下排列的。最高位 D7 是在最上面，最低位 D0 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：

| SEG Output | SEG 0 | | SEG 1 | | SEG 2 | | SEG 3 | | … | SEG 124 | | SEG 125 | | SEG 126 | | SEG 127 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Column Address X[7:1] | 00H | | 01H | | 02H | | 03H | | … | 7CH | | 7DH | | 7EH | | 7FH | |
| Internal column address X[7:0] | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | … | F8 | F9 | FA | FB | FC | FD | FE | FF |
| Display Data (MX=0) | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | … | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| LCD panel display | | | | | | | | | … | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| Display data (MX=1) | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | … | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| LCD panel display | | | | | | | | | … | | | | | | | | |

**Fig. 12　The Relationship between the Column Address and The Segment Outputs**

下图摘自 ST7571 IC 资料，可通过“ST7571.PDF”之第 29 页获取最佳效果。

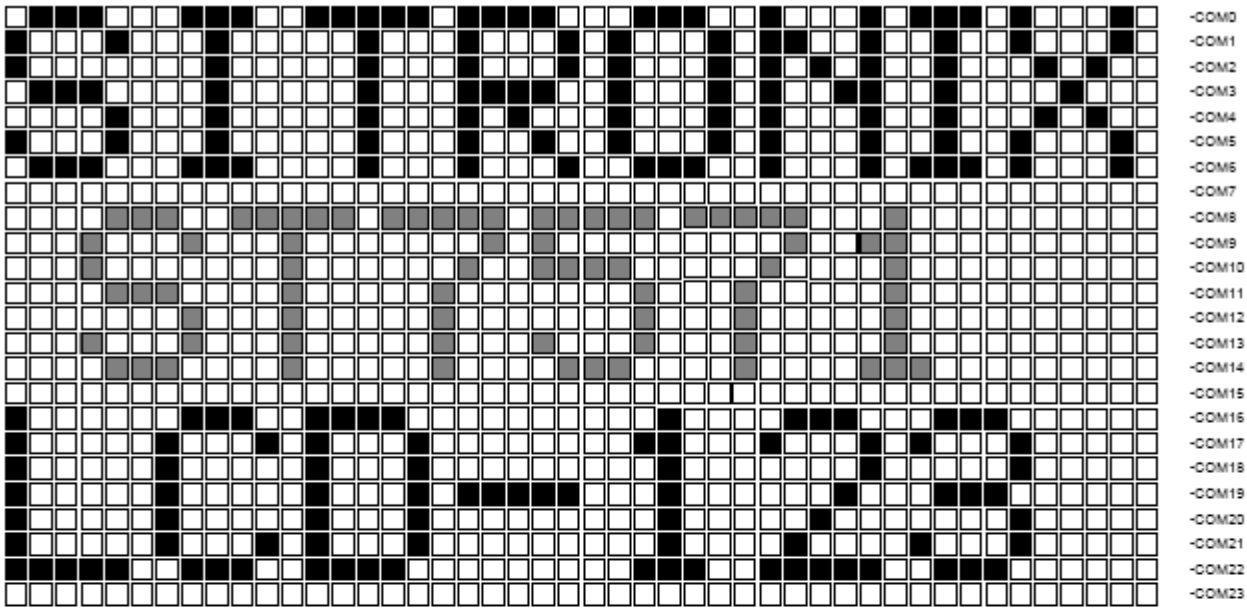



Fig. 15 Reference Example for Partial Display

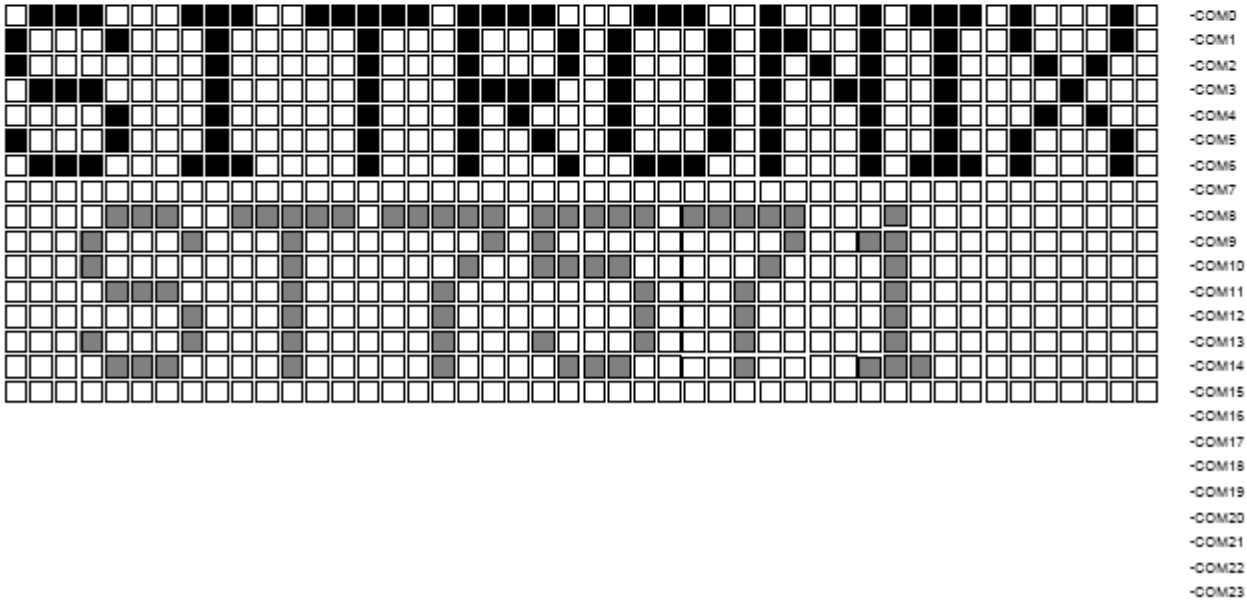



Fig. 16 Partial Display (Partial Display Duty=16, initial COM0=0)

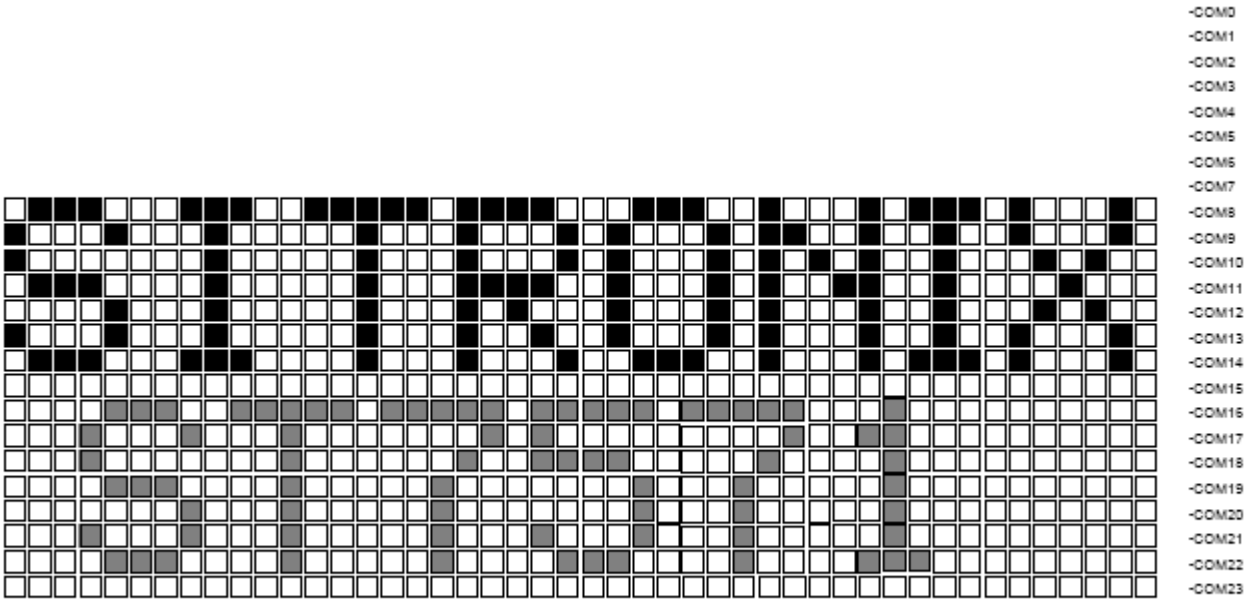
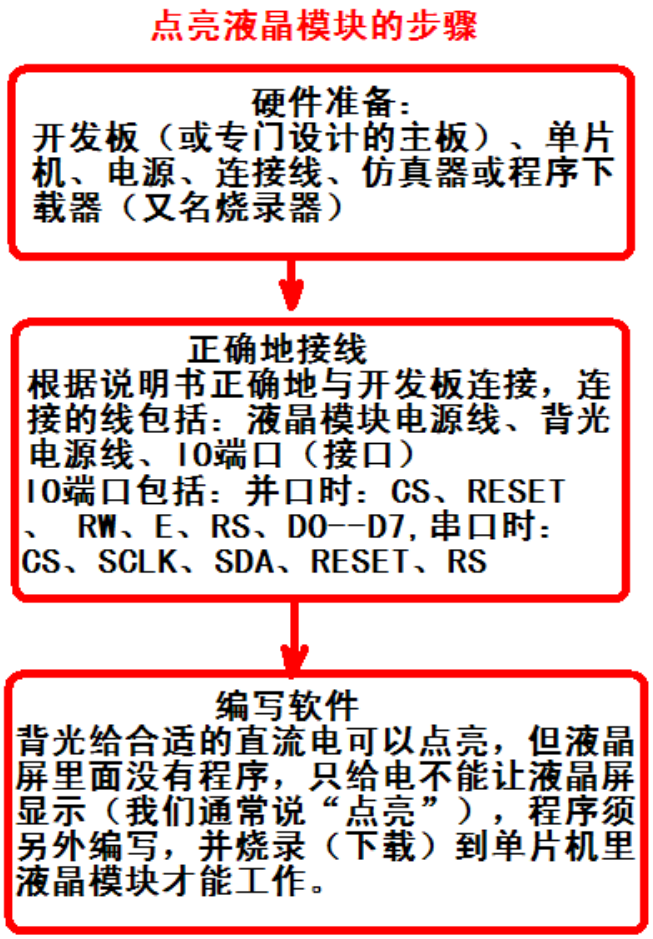


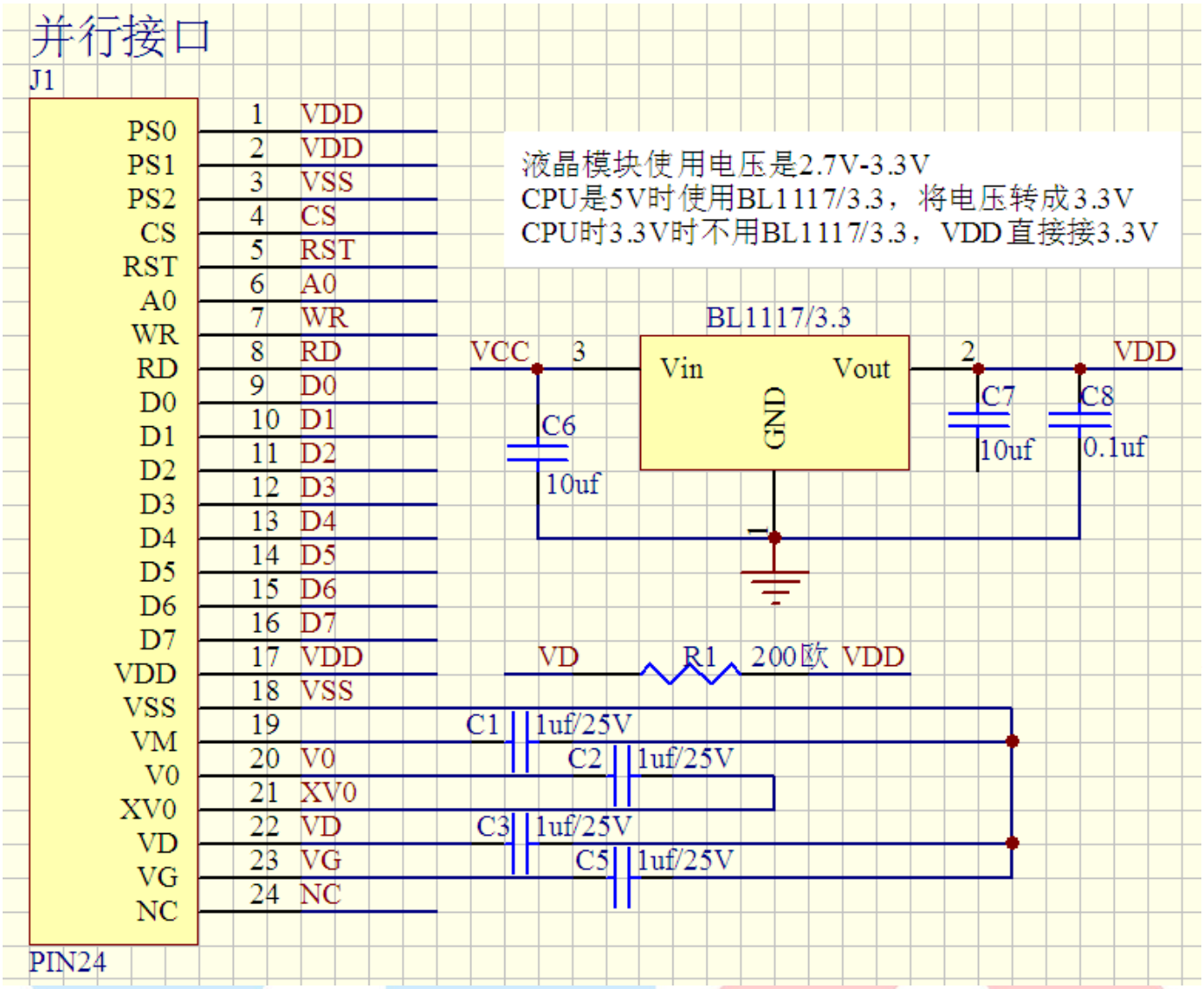

Fig. 17 Moving Display (Partial Display Duty=16, initial COM0=8)

## 7.3 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



## 并行接口电路图




7.4 程序举例:

7.4.1 并行接口

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:

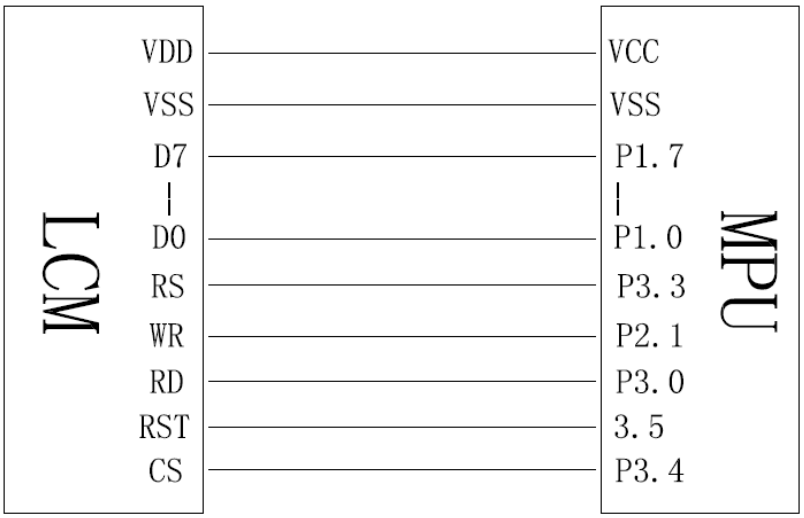


图 8. 并行接口

### 7.5.2 以下是并行接口例程序

```
#include <reg51.H>
#include <intrins.h>
#include <chinese_code.h>
#include <image.h>

sbit cs1=P3^4;       /*3.4 接口定义*/
sbit reset=P3^5;     /*3.3 接口定义*/
sbit rs=P3^3;        /*接口定义*/
sbit rd=P3^0;        /*接口定义*/
sbit wr=P2^1;        /*接口定义。另外 P1.0~1.7 对应 DB0~DB7*/
sbit key=P2^0;       /*按键接口，P2.0 口与 GND 之间接一个按键*/

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

void delay_us(int i);


//=============transfer command to LCM===============
void transfer_command(int data1)
{
    cs1=0;
    rs=0;
    rd=0;
    delay_us(1);
    wr=0;
    P1=data1;
    rd=1;
    delay_us(1);
    cs1=1;
    rd=0;
}

//-----------transfer data to LCM---------------
void transfer_data(int data1)
{
    cs1=0;
    rs=1;
    rd=0;
    delay_us(1);
    wr=0;
    P1=data1;
```

```
    rd=1;
    delay_us(1);
    cs1=1;
    rd=0;
}

void delay_us(int i)
{
    int j,k;
    for(j=0;j<i;j++)
    for(k=0;k<10;k++);
}

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

//等待一个按键
void waitkey()
{
 repeat:
    if (key==1) goto repeat;
    else;
        delay(1500);
}

void initial_lcd()
{
    reset=0;
    delay(500);
    reset=1;
    delay(100);

    transfer_command(0x2c);
    delay(200);
    transfer_command(0x2e);
    delay(200);
    transfer_command(0x2f);
    delay(10);

    transfer_command(0xae);     //显示关
    transfer_command(0x38);     //模式设置
    transfer_command(0xb8);     //85HZ
```

```
    transfer_command(0xc8);      //行扫描顺序
    transfer_command(0xa0);      //列扫描顺序

    transfer_command(0x44);      //Set initial COM0 register
    transfer_command(0x00);
    transfer_command(0x40);      //Set initial display line register
    transfer_command(0x00);

    transfer_command(0xab);
    transfer_command(0x67);
    transfer_command(0x26);      //0x24 粗调对比度，可设置范围 0x20～0x27
    transfer_command(0x81);      //微调对比度
    transfer_command(0x36);      //36 微调对比度的值，可设置范围 0x00～0x3f

    transfer_command(0x54);      //0x54  1/9 bias
    transfer_command(0xf3);
    transfer_command(0x04);
    transfer_command(0x93);

//  transfer_command(0x7b);      //Extension Command Set3
//  transfer_command(0x11); //Gray mode
//  transfer_command(0x10);      //Gray mode
//  transfer_command(0x00);
    transfer_command(0xaf);      //显示开

}


void lcd_address(uchar page,uchar column)
{
    cs1=0;
    column=column;
    page=page-1;
    transfer_command(0xb0+page);
    transfer_command(((column>>4)&0x0f)+0x10);
    transfer_command(column&0x0f);
}

void clear_screen()
{
    uchar i,j;
    for(j=0;j<16;j++)
    {
        lcd_address(j+1,0);
        for(i=0;i<128;i++)
```

```
        {
            transfer_data(0x00);
            transfer_data(0x00);
        }
    }
}


void test_screen()
{
    uchar i,j;
    for(j=0;j<16;j++)
    {
        lcd_address(j+1,0);
        for(i=0;i<256;i++)
        {
            transfer_data(0xaa);
            waitkey();
        }
    }
}

//显示 8x16 的点阵的字符串，括号里的参数分别为（页,列，字符串指针）
void display_string_8x16(uchar page,uchar column,uchar *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]);
                    transfer_data(ascii_table_8x16[j][k+8*n]);
                }
            }
            i++;
            column+=8;
        }
        else
        i++;
```

```
        if(column>127)
        {
            column=0;
            page+=2;
        }
    }
}

//写入一组 16x16 点阵的汉字字符串（字符串表格中需含有此字）
//括号里的参数：(页，列，汉字字符串)
void display_string_16x16(uchar page,uchar column,uchar *text)
{
    uchar i,j,k;
    uint address;
    j = 0;
    while(text[j] != '\0')
    {
        i=0;
        address=1;
        while(Chinese_text_16x16[i]> 0x7e )
        {
            if(Chinese_text_16x16[i] == text[j])
            {
                if(Chinese_text_16x16[i+1] == text[j+1])
                {
                    address = i*16;
                    break;
                }
            }
            i +=2;
        }
        if(column>127)
        {
            column =0;
            page +=2;
        }
        if(address !=1)
        {
            for(k=0;k<2;k++)
            {
                lcd_address(page+k,column);
                for(i=0;i<16;i++)
                {
                    transfer_data(Chinese_code_16x16[address]);
                    transfer_data(Chinese_code_16x16[address]);
                    address++;
```

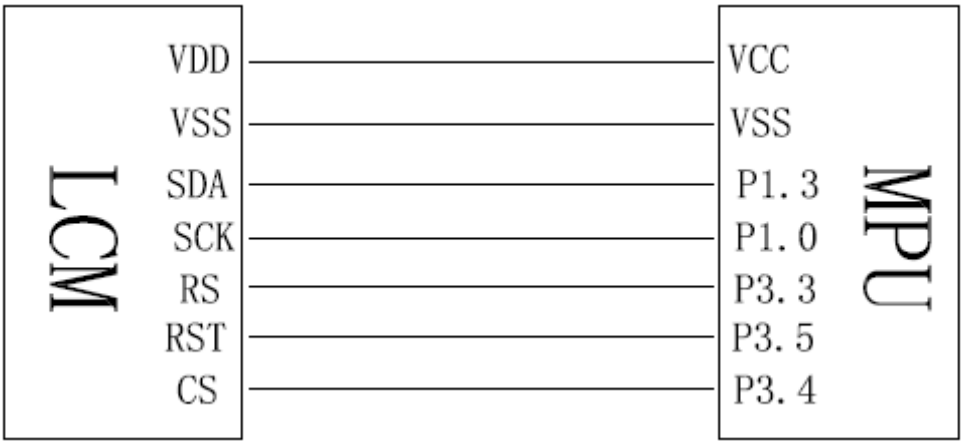
```
                }
            }
            j +=2;
        }
        else
        {
            for(k=0;k<2;k++)
            {
                lcd_address(page+k,column);
                for(i=0;i<16;i++)
                {
                    transfer_data(0x00);
                    transfer_data(0x00);
                }
            }
            j++;
        }
        column +=16;
    }
}

//显示 16x16 点阵的汉字或者 ASCII 码 8x16 点阵的字符混合字符串
//括号里的参数：(页，列，字符串)
void display_string_8x16_16x16(uchar page,uchar column,uchar *text)
{
    uchar temp[3];
    uchar i=0;
    while(text[i] !='\0')
    {
        if(text[i]>0x7e)
        {
            temp[0]=text[i];
            temp[1]=text[i+1];
            temp[2]='\0';                                    //汉字为两个字节
            display_string_16x16(page,column,temp); //显示汉字
            column +=16;
            i +=2;
            if(column>127)
            {
                column =0;
                page +=2;
            }
        }
        else
        {
            temp[0]=text[i];
```

```
            temp[1]='\0';                                                    //字母占一个字节
            display_string_8x16(page,column,temp);  //显示字母
            column +=8;
            i++;
            if(column>127)
            {
                column =0;
                page +=2;
            }
        }
    }
}


void display_32x32(uchar page,uchar column,uchar *dp)
{
    int i,j;
    for(j=0;j<4;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<32;i++)
        {
            transfer_data(*dp);
            transfer_data(*dp);
            dp++;
        }
    }

}
void display_graphic(uchar *dp)
{
    int i,j;
    for(j=0;j<16;j++)
    {
        lcd_address(j+1,0);
        for(i=0;i<128;i++)
        {
            transfer_data(*dp);
            transfer_data(*dp);
            dp++;
        }
    }
}


void main(void)
```

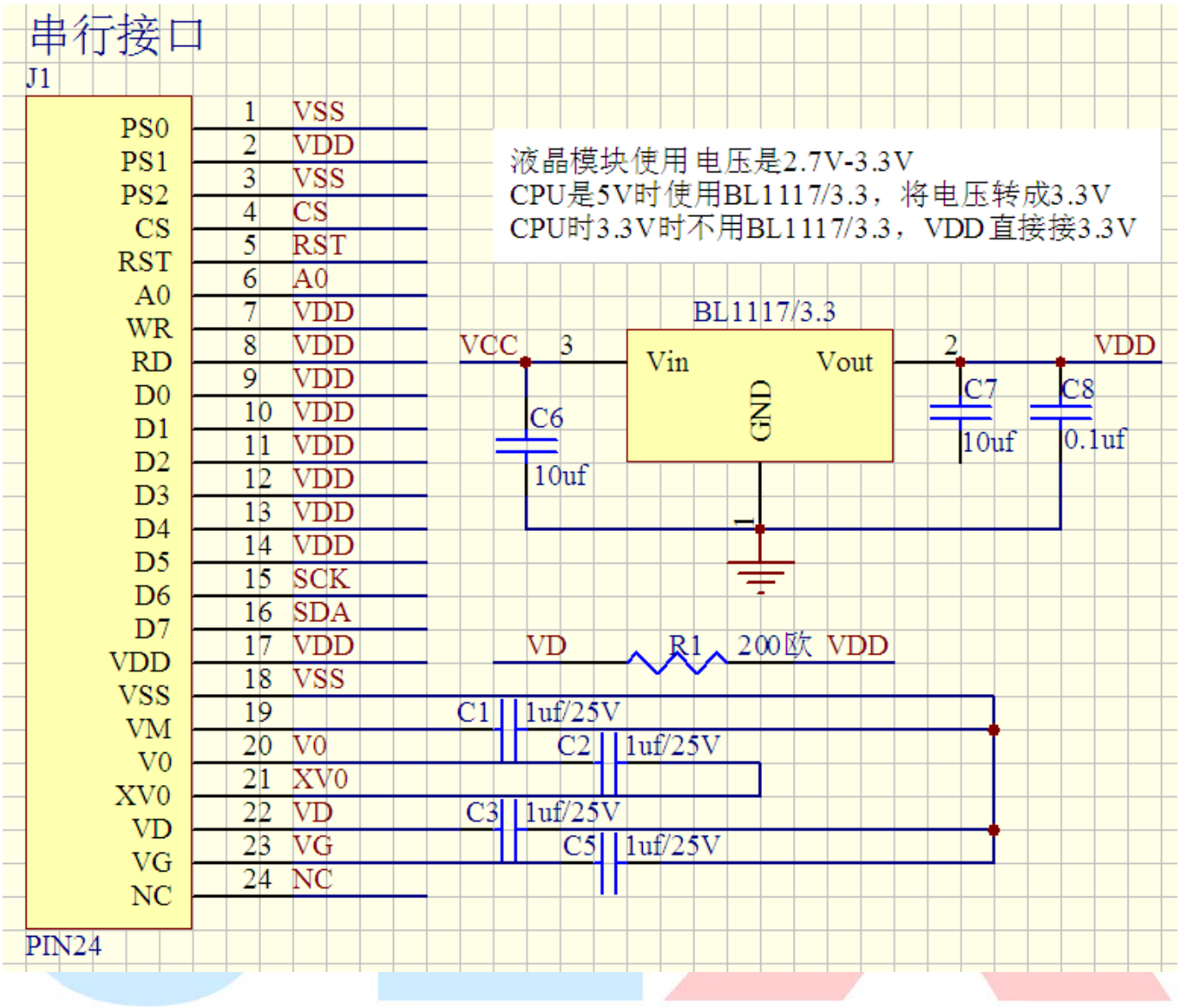
```
{
    initial_lcd();
    while(1)
    {
        clear_screen();
        display_graphic(bmp2);
        waitkey();
        clear_screen();
        display_32x32(1,16,jing32);
        display_32x32(1,48,lian32);
        display_32x32(1,80,xun32);
        display_string_16x16(5,1,"深圳市晶联讯电子有限公司是集研发、生产、销售于一体的从事液晶显示屏及液晶显示模块的高科技公司。");
        waitkey();
        clear_screen();

    display_string_8x16(1,1,"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&()*+-,-./:;<=>?@[\]^_`{|}~0123456789ABCDFGHIJKLMNOPQRSTUVWXYZ");
        waitkey();
        clear_screen();
        display_string_8x16_16x16(1,1,"深圳市晶联讯电子 JLX128128G-81202 128x128 点阵 视区:43.5x45.1mm 带 16x16 点阵中文 字库,或 8x16 或 5x7 点阵 ASCII 码,四灰度级显示功能。");
        waitkey();

    }
}
```

### 7.5.4 串行接口

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:



## 串行电路图



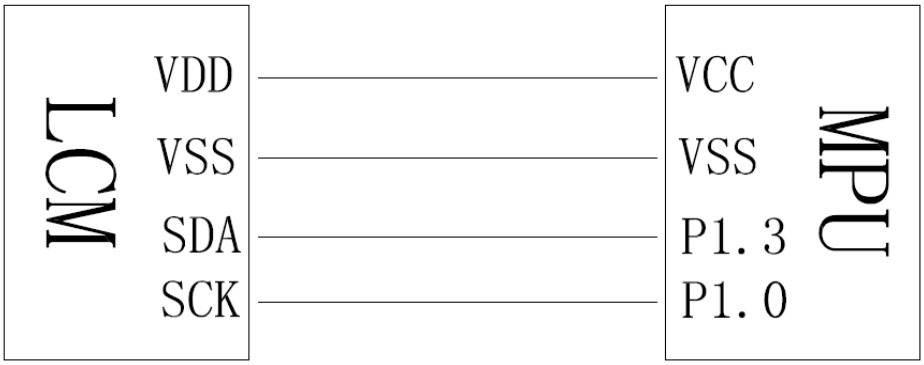

**7.5.5 以下是串行接口例程序**
**与并行程序相比，只需改变接口顺序及传送数据和命令子程序即可**
**//传送指令**

```
sbit rs=P3^3;     /*接口定义:LCD 的 rs*/
sbit sclk=P1^6;   /*接口定义:LCD 的 sclk*/
sbit sid=P1^7;    /*接口定义:LCD 的 sid*/
sbit reset=P3^5; /*接口定义:LCD 的 reset*/
sbit cs1=P3^4;    /*接口定义:LCD 的 cs1*/
sbit key=P2^0;    //P2.0 口与 GND 之间接一个按键


/*写指令到 LCD 模块*/
void transfer_command(int data1)
{
    char i;
    cs1=0;
```

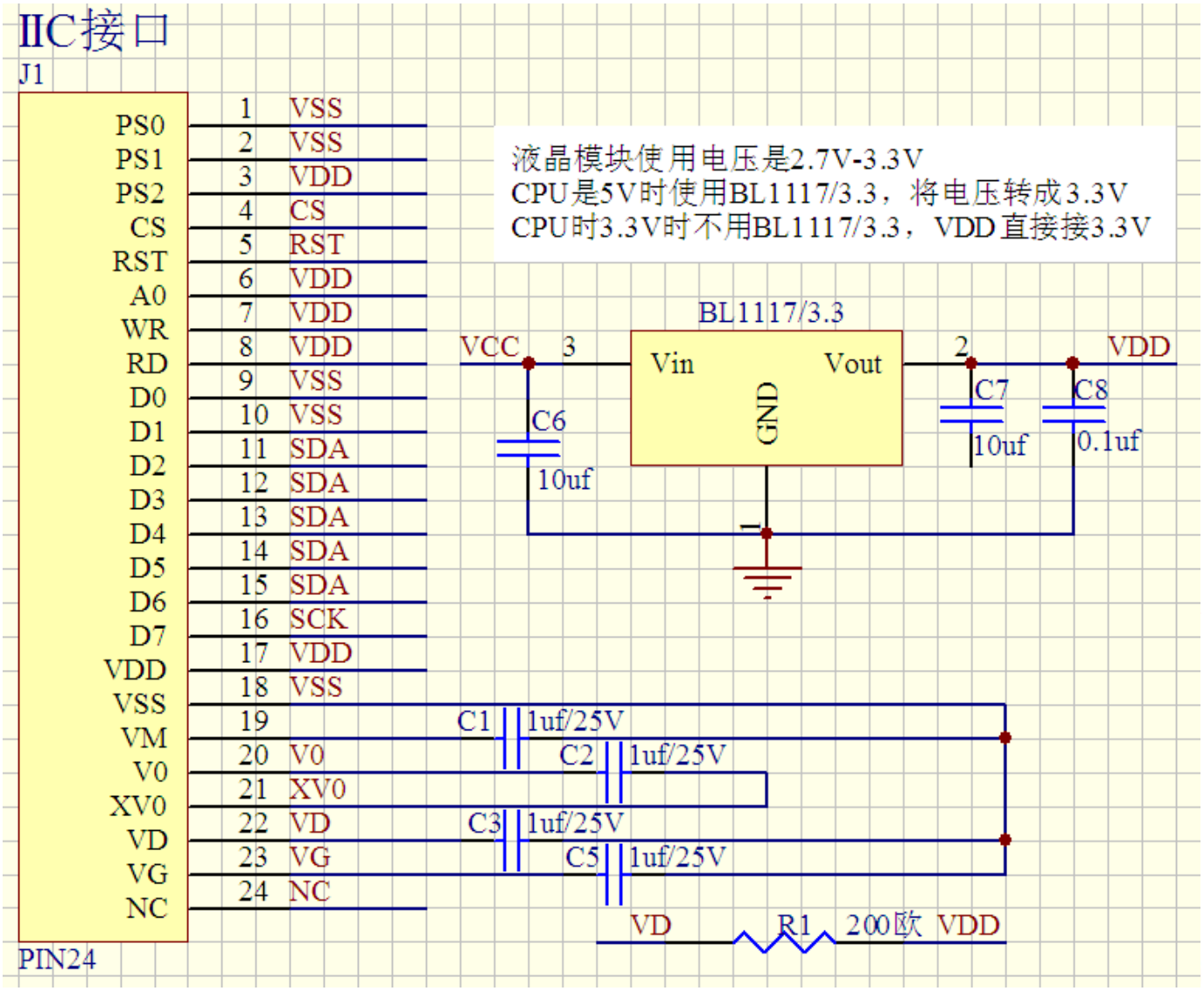
```
    rs=0;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        delay_us(1);
        data1<<=1;
    }
    cs1=1;
}
```

```
/*写数据到 LCD 模块*/
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<<=1;
    }
    cs1=1;
}
```

### 7.5.4 I²C 接口



**IIC 接口电路图**




7.5.5 以下是 I²C 接口例程序

7.5.5 以下是 I²C 接口例程序

```
#include <reg51.H>
#include <intrins.h>
#include <chinese_code.h>
#include <image.h>

sbit    reset=P1^1;
sbit  scl=P1^2;
sbit  sda=P1^3;
sbit    key=P2^0;

#define uchar unsigned char
#define uint unsigned int
```

```
#define ulong unsigned long

void delay_us(int i);

void transfer(int data1)
{
      int i;
      for(i=0;i<8;i++)
     {
            scl=0;
            if(data1&0x80) sda=1;
            else sda=0;
            scl=1;
            scl=0;
            data1=data1<<1;
     }
            sda=0;
            scl=1;
            scl=0;
}

void start_flag()
{
      scl=1;          /*START FLAG*/
      sda=1;          /*START FLAG*/
      sda=0;          /*START FLAG*/
}

void stop_flag()
{
      scl=1;          /*STOP FLAG*/
      sda=0;          /*STOP FLAG*/
      sda=1;          /*STOP FLAG*/
}

void delay_us(int i)
{
      int j,k;
      for(j=0;j<i;j++)
      for(k=0;k<10;k++);
}

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

```
    for(k=0;k<110;k++);
}

//等待一个按键
void waitkey()
{
 repeat:
    if (key==1) goto repeat;
    else;
        delay(1500);
}

void initial_lcd()
{
    reset=0;
    delay(500);
    reset=1;
    delay(100);

    start_flag();
    transfer(0x7e);
    transfer(0x00);

    transfer(0x2c);
    delay(200);
    transfer(0x2e);
    delay(200);
    transfer(0x2f);
    delay(10);
    transfer(0xae);
    transfer(0x38);
    transfer(0xb8);
    transfer(0xc8);      //行扫描顺序
    transfer(0xa0);      //列扫描顺序
    transfer(0x44);
    transfer(0x00);
    transfer(0x40);
    transfer(0x00);
    transfer(0xab);
    transfer(0x67);
    transfer(0x26);//0x24 粗调对比度，可设置范围 0x20～0x27
    transfer(0x81);//微调对比度
    transfer(0x36);//36  微调对比度的值，可设置范围 0x00～0x3f
    transfer(0x54);//0x54    1/9 bias
    transfer(0xf3);
    transfer(0x04);
```

```
    transfer(0x93);
    transfer(0xaf);
    stop_flag();
}

void lcd_address(uchar page,uchar column)
{
    column=column;
    page=page-1;
    start_flag();
    transfer(0x7e);
    transfer(0x00);
    transfer(0xb0+page);
    transfer(((column>>4)&0x0f)+0x10);
    transfer(column&0x0f);
    stop_flag();
}

void clear_screen()
{
    uchar i,j;
    for(j=0;j<16;j++)
    {
        lcd_address(j+1,0);
        start_flag();
        transfer(0x7e);
        transfer(0x40);
        for(i=0;i<128;i++)
        {
            transfer(0x00);
            transfer(0x00);
        }
        stop_flag();
    }
}

//显示 8x16 的点阵的字符串，括号里的参数分别为（页,列，字符串指针）
void display_string_8x16(uchar page,uchar column,uchar *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(ascii_table_8x16[j][k+8*n]);
                    transfer(ascii_table_8x16[j][k+8*n]);
                }
            }
            i++;
            column+=8;
        }
        else
        i++;

        if(column>127)
        {
            column=0;
            page+=2;
        }
    }
}

//写入一组 16x16 点阵的汉字字符串（字符串表格中需含有此字）
//括号里的参数：(页，列，汉字字符串）
void display_string_16x16(uchar page,uchar column,uchar *text)
{
    uchar i,j,k;
    uint address;
    j = 0;
    while(text[j] != '\0')
    {
        i=0;
        address=1;
        while(Chinese_text_16x16[i]> 0x7e )
        {
            if(Chinese_text_16x16[i] == text[j])
            {
                if(Chinese_text_16x16[i+1] == text[j+1])
                {
                    address = i*16;
                    break;
                }
            }
            i +=2;
        }
```

```
        if(column>127)
        {
            column =0;
            page +=2;
        }
        if(address !=1)
        {
            for(k=0;k<2;k++)
            {
                lcd_address(page+k,column);
                for(i=0;i<16;i++)
                {
                    transfer(Chinese_code_16x16[address]);
                    transfer(Chinese_code_16x16[address]);
                    address++;

                }
            }
            j +=2;
        }
        else
        {
            for(k=0;k<2;k++)
            {
                lcd_address(page+k,column);
                for(i=0;i<16;i++)
                {
                    transfer(0x00);
                    transfer(0x00);
                }
            }
            j++;
        }
        column +=16;
    }
}

//显示 16x16 点阵的汉字或者 ASCII 码 8x16 点阵的字符混合字符串
//括号里的参数：(页，列，字符串）
void display_string_8x16_16x16(uchar page,uchar column,uchar *text)
{
    uchar temp[3];
    uchar i=0;
    while(text[i] !='\0')
    {
        if(text[i]>0x7e)
```

```
        {
            temp[0]=text[i];
            temp[1]=text[i+1];
            temp[2]='\0';                                   //汉字为两个字节
            display_string_16x16(page,column,temp); //显示汉字
            column +=16;
            i +=2;
            if(column>127)
            {
                column =0;
                page +=2;
            }
        }
        else
        {
            temp[0]=text[i];
            temp[1]='\0';                                   //字母占一个字节
            display_string_8x16(page,column,temp);    //显示字母
            column +=8;
            i++;
            if(column>127)
            {
                column =0;
                page +=2;
            }
        }
    }
}


void display_32x32(uchar page,uchar column,uchar *dp)
{
    int i,j;
    for(j=0;j<4;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<32;i++)
        {
            transfer(*dp);
            transfer(*dp);
            dp++;
        }
    }

}
```

```
void display_graphic(uchar *dp)
{
    int i,j;
    for(j=0;j<16;j++)
    {
        lcd_address(j+1,0);
        start_flag();
        transfer(0x7e);
        transfer(0x40);
        for(i=0;i<128;i++)
        {
            transfer(*dp);
            transfer(*dp);
            dp++;
        }
        trop_flag();
    }
}



void main(void)
{
    initial_lcd();
    while(1)
    {
        clear_screen();
        display_graphic(bmp2);
        waitkey();
        clear_screen();
        display_32x32(1,16,jing32);
        display_32x32(1,48,lian32);
        display_32x32(1,80,xun32);
        display_string_16x16(5,1,"深圳市晶联讯电子有限公司是集研发、生产、销售于一体的从事液晶显示屏及液晶显示模块的高科技公司。");
        waitkey();
        clear_screen();

    display_string_8x16(1,1,"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&()*+-,-./:;<=>?@[\]^_^{|}~0123456789ABCDFGHIJKLMNOPQRSTUVWXYZ");
        waitkey();
        clear_screen();
        display_string_8x16_16x16(1,1," 深 圳 市 晶 联 讯 电 子  JLX128128G-81202  128x128 点 阵  视区:43.5x45.1mm 带 16x16 点阵中文 字库,或 8x16 或 5x7 点阵 ASCII 码,四灰度级显示功能。");
        waitkey();
```

```
    }
}
```