

液晶模块使用说明书

型号：COG12864-14LCM

版本：V1.0

版本变更历史记录

| 版本 | 修订日期 | 修改内容 | 修订人 |
|----|------------|------|-----|
| 01 | 2022-08-31 | 初版发行 | |

1. 概述

COG12864-14LCM 点阵绘图型液晶显示模块(LCM)采用 COG 工艺的 128x64 点阵液晶显示屏 (LCD)与低功耗 LED 背光组成。采用 ST7567 控制器,直接控制 128x64 点阵;支持SPI 串口与 CPU 通信,可选配带字库芯片,是一款点阵屏,广泛应用于各类仪器仪表上。

COG12864-14LCM 可以显示 128 列*64 行点阵单色图片;显示 8 个/行*4 行 16*16 点阵的汉字;显示 16个/行*8 行 8*8 点阵的英文、数字、符号。

2. 产品特点

2.1 产品薄、轻、带背光、结构牢、带PCB。

2.2 COG 工艺, IC 采用 ST7567,功能强大,稳定性好。

2.3 功耗低: 不带背光 3.3V 供电时电流在 1mA 内,带背光电流在 60mA 以内。

2.4 显示内容:

●128*64 点阵单色图片,或小于 128*64 点阵单色图片;

●可选用 16*16 点阵或其它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;

●COM 和 SEG 顺序可以改变,因此显示内容可以横向,竖向显示。

2.6 汉字和字符(可选)

●15X16 点 GB2312 标准点阵字库 6763+376 个

●8X16 点国标扩展字符 126 个

2.7 ASCII 字符

●5X7 点 ASCII 字符 96 个

●7X8 点 ASCII 字符 96 个

●8X16 点 ASCII 字符 96 个

●8X16 点 ASCII 粗体字符 96 个

●16 点阵不等宽 ASCII 方头 (Arial) 字符 96 个

●16 点阵不等宽 ASCII 白正 (TimesNewRoman) 字符 96 个

2.6 工作温度宽: -20℃ -- +70℃。

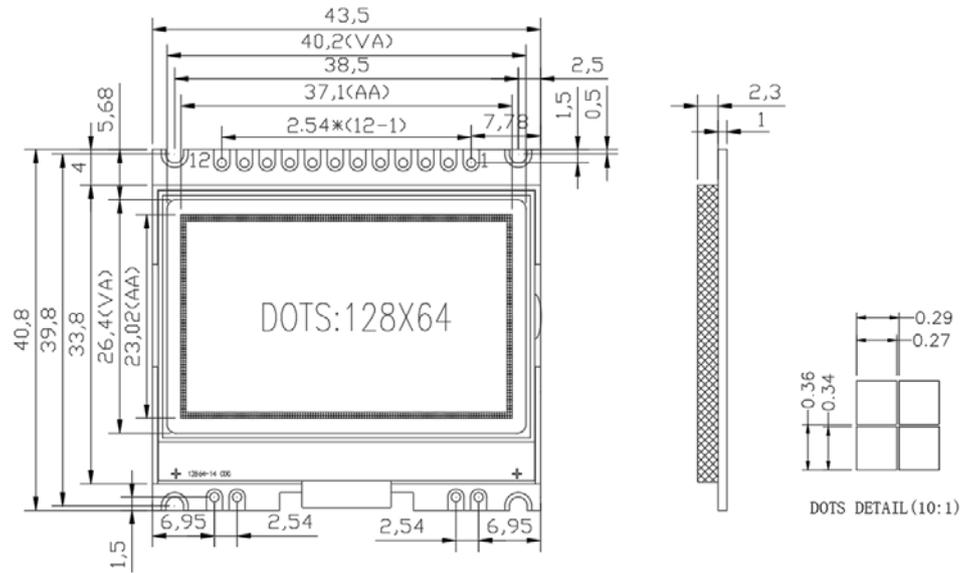
3. 规格说明

| 项目 | 规格描述 | 单位 |
|-------------|---|------|
| 驱动点阵数 | 128 列 X 64 行 | dots |
| 显示模式 | 点阵绘图型 | |
| 外形尺寸 | 43.5(L) X 40.8(W) X 3.3(T) | mm |
| 可视区域 | 40.2(L) X 26.4(W) | mm |
| 显示区域 | 37.1(L) x 23.02(W) | mm |
| 点尺寸 | 0.27(L) x 0.34(W) | mm |
| 点间距(Pitch) | 0.29(L) x 0.36(W) | mm |
| LCD 模式 | ■STN, Yellow-Green / ■ STN, Blue / ■ FSTN, Positive | |
| 偏光片类型 | ■Transmissive / ■Transflective | |
| 视角 | ■ 6 点 / ■ 12 点 | |
| 逻辑电源(VDD) | ■3.3 / ■5.0 | V |
| 控制器 IC | ST7567 | |
| LCD 偏压比 | 1/65duty, 1/9bias | |
| 时序接口类型 | ■SPI | |
| 背光类型 | ■LED | |
| 背光颜色 | ■Yellow-Green ■ White | |
| 背光结构 | ■单侧 | |
| 铁框 | ■可选 | |
| 工作温度 | -20~70 (TOPL – TOPH) | °C |
| 储存温度 | -30~80 (TSTL -- TSTH) | °C |

注：“■”表示常用规格

4. 外形机构图

4.1 外形尺寸图



5. 接口定义

5.1 串口定义

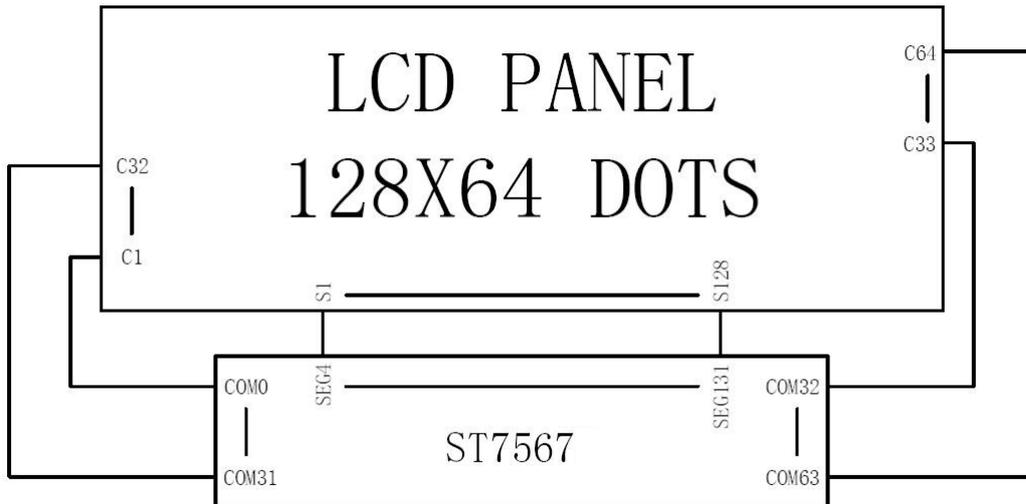
| 脚位 | 标号 | 电平 | 功能描述 |
|----|------|-----|------------------------------|
| 1 | SI | I | GB2312 串行数据输入 (字符码, 汉字码输入) |
| 2 | S0 | 0 | GB2312 串行数据输出 (字符, 汉字点阵数据输出) |
| 3 | SCLK | I | GB2312 串行时钟输入 |
| 4 | CS# | I | GB2312 片选输入 |
| 5 | LEDA | H | 背光正极 |
| 6 | VSS | 0V | 逻辑电源负极 |
| 7 | VDD | H | 逻辑电源正极 |
| 8 | SCLK | H/L | 串口时钟输入 |
| 9 | SID | H/L | 串口数据输入 |
| 10 | A0 | H/L | 寄存器选择信号 |
| 11 | /RST | L | 复位信号, 低电平有效 |
| 12 | /CS | L | 芯片使能 |

6. 基本原理

6.1 液晶屏(LCD)

在 LCD 上排列着 128*64 点阵,128 个 SEG 信号与驱动 IC 相连,64 个 COM 信号与驱动 IC 相连,IC 邦定在 LCD 玻璃上(这种加工艺叫 COG)。

6.2 工作电路图



6.3 背光参数

COG12864-14LCM 带 LED 背光源,3个白色灯,Idd 控制在 50mA,模块已经在 LED 电路中加了限流电阻, 因此直接从 LEDA 供电。

7. 技术参数

7.1 最大典型值(超过极限值会损坏液晶模块)

| 项目 | 符号 | 标准值 | | | 单位 |
|------|----------------------------------|------|-----|------|----|
| | | 最小值 | 典型值 | 最大值 | |
| 逻辑电压 | V _{DD} -V _{SS} | -0.3 | - | +3.6 | V |
| 工作温度 | | -20 | - | 70.0 | °C |
| 储存温度 | | -30 | - | 80.0 | °C |

7.2 直流 DC 参数

测试条件 V_{SS}=0V, V_{DD}=3.3V, T_a=25°C

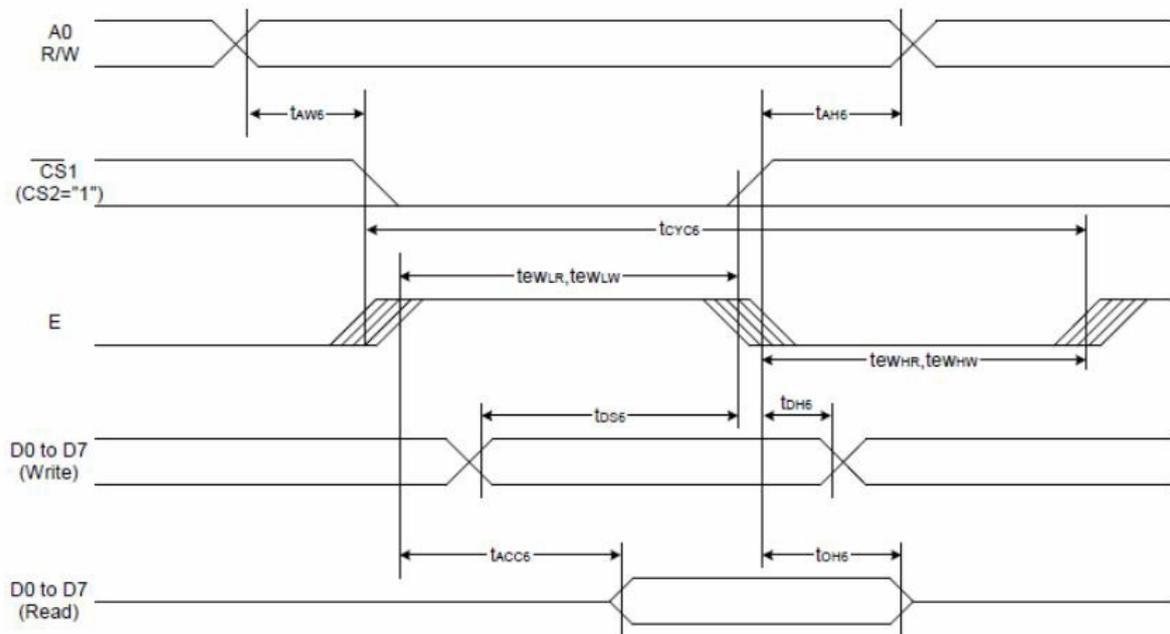
| 项目 | 符号 | 测试条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------|-------------------|-----------------------|--------------------|-----|--------------------|----|
| 逻辑电压 | V _{DD} | -- | 2.9 | 3.3 | 3.4 | V |
| 背光工作电压 | V _{Iled} | -- | -- | 3.3 | 3.4 | V |
| 模块电流 | I _{DD} | V _{DD} =3.3V | -- | 90 | 105 | mA |
| 输入高电平 | V _{IH} | -- | 0.8V _{DD} | -- | V _{DD} | V |
| 输入低电平 | V _{IL} | -- | V _{SS} | -- | 0.2V _{DD} | V |
| 输出高电平 | V _{OH} | -- | 0.8V _{DD} | -- | V _{DD} | V |
| 输出低电平 | V _{OL} | -- | V _{SS} | -- | 0.2V _{DD} | V |
| 频率 | FELM | V _{DD} =3.3V | 65 | 80 | 85 | Hz |

8. 读写时序特性(具体详细介绍参考 ST7567 IC 资料)

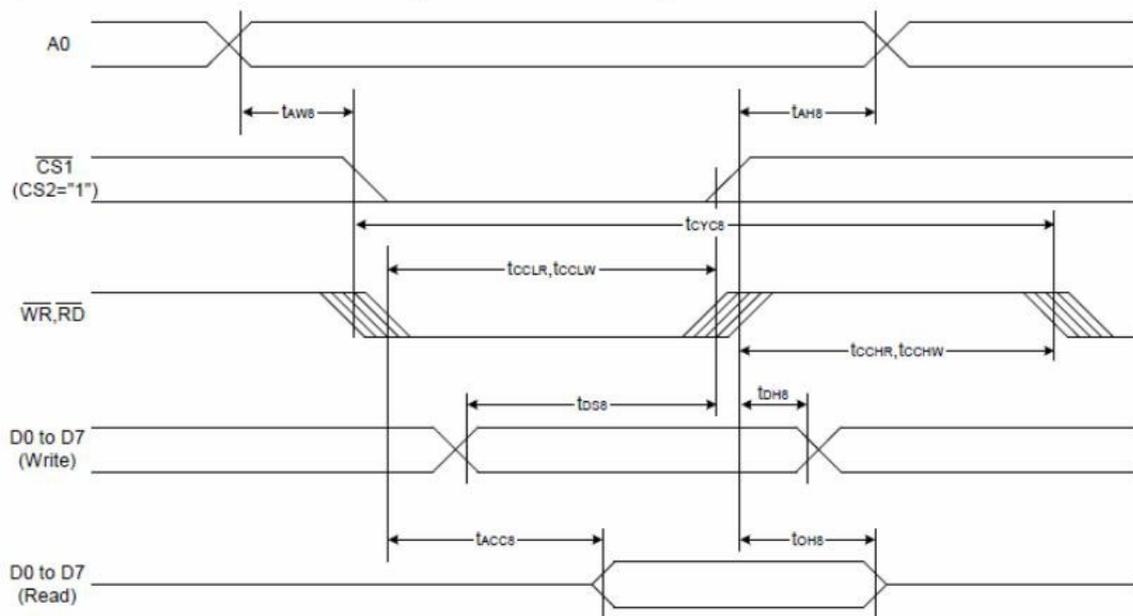
8.1 并行接口

从 CPU 写到 ST7567

System Bus Read/Write Characteristics 2 (For the 6800 Series MPU)



System Bus Read/Write Characteristics 1 (For the 8080 Series MPU)



8.2 并行接口时序要求(AC 参数)

6800系列MPU总线接口参数

(VDD1 = 3.3V , Ta =25°C)

| 参数名称 | 对应端口 | 符号 | 测试条件 | 规范值 | | 单位 |
|---------------|-------|--------|------------|-----|----|----|
| | | | | 最小 | 最大 | |
| 地址建立时间 | A0 | tAW6 | | 0 | — | ns |
| 地址保持时间 | | tAH6 | | 10 | — | |
| 系统周期时间 | E | tCYC6 | | 240 | — | |
| 使能 L 脉冲宽度 (写) | | tEWLW | | 80 | — | |
| 使能 H 脉冲宽度 (写) | | tEWHW | | 80 | — | |
| 使能 L 脉冲宽度 (读) | | tEWLR | | 80 | — | |
| 使能 H 脉冲宽度 (读) | | tEWHR | | 140 | — | |
| 写数据建立时间 | | D[7:0] | tDS6 | | 40 | |
| 写数据保持时间 | tDH6 | | | 10 | — | |
| 读数据允许时间 | tACC6 | | CL = 16 pF | — | 70 | |
| 读数据输出禁止时间 | tOH6 | | CL = 16 pF | 5 | 50 | |

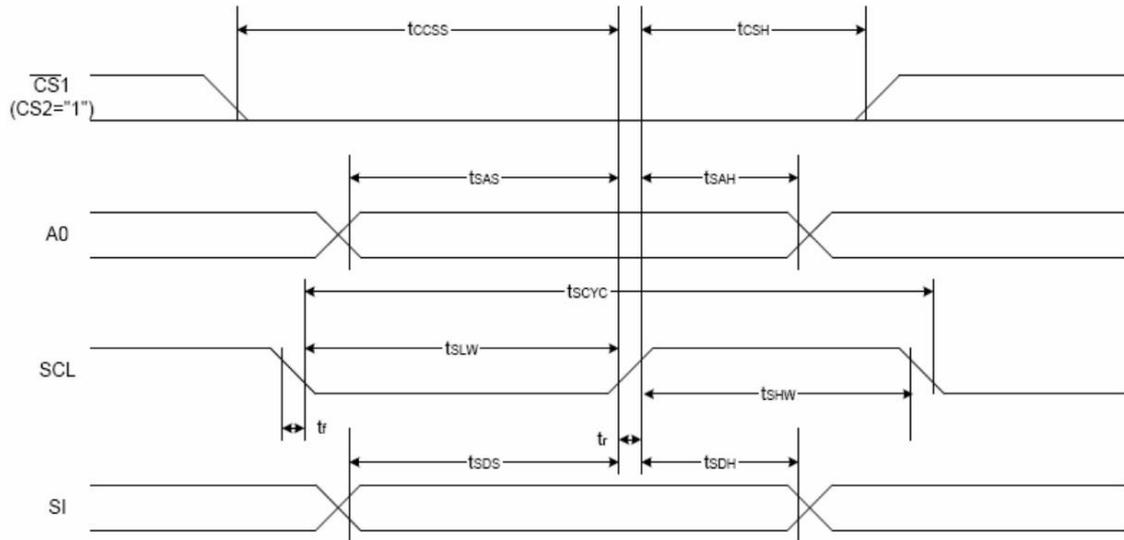
8080系列MPU总线接口参数

(VDD1 = 3.3V , Ta =25°C)

| 参数名称 | 对应端口 | 符号 | 测试条件 | 规范值 | | 单位 |
|----------------|--------|-------|------------|-----|-----|----|
| | | | | 最小 | 最大 | |
| 地址建立时间 | A0 | tAW8 | | 0 | — | ns |
| 地址保持时间 | | tAH8 | | 10 | — | |
| 系统周期时间 | /WR | tCYC8 | | 240 | — | |
| /WR L 脉冲宽度 (写) | | tCCLW | | 80 | — | |
| /WR H 脉冲宽度 (写) | | tCCHW | | 80 | — | |
| /RD L 脉冲宽度 (读) | | RD | tCCLR | | 140 | |
| /RD H 脉冲宽度 (读) | tCCHR | | | 80 | — | |
| 写数据建立时间 | D[7:0] | tDS8 | | 40 | — | |
| 写数据保持时间 | | tDH8 | | 20 | — | |
| 读数据允许时间 | | tACC8 | CL = 16 pF | — | 70 | |
| 读数据输出禁止时间 | | tOH8 | CL = 16 pF | 5 | 50 | |

8.3 串行接口 从 CPU 写到 ST7567

The 4-line SPI Interface

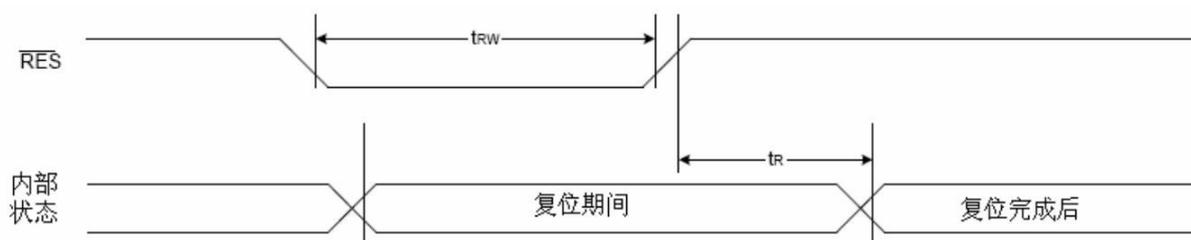


8.4 串行接口时序要求(AC 参数) 写数据到 ST7567 的时序要求

| 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 (4-line SPI Clock Period) | T_{scyc} | 引脚: SCK | 50 | -- | 25 | ns |
| 保持SCK高电平脉宽 (SCK "H" pulse width) | T_{shw} | 引脚: SCK | 25 | | | ns |
| 保持SCK低电平脉宽 (SCK "L" pulse width) | T_{slw} | 引脚: SCK | 25 | | | ns |
| 地址建立时间 (Address setup time) | T_{sas} | 引脚: RS | 20 | -- | -- | ns |
| 地址保持时间 (Address hold time) | T_{sah} | 引脚: RS | 10 | -- | -- | ns |
| 数据建立时间 (Data setup time) | T_{sds} | 引脚: SI | 20 | -- | -- | ns |
| 数据保持时间 (Data hold time) | T_{sdh} | 引脚: SI | 10 | -- | -- | ns |
| 片选信号建立时间 (CS-SCL time) | T_{css} | 引脚: CS | 20 | | | ns |
| 片选信号保持时间 (CS-SCL time) | T_{csh} | 引脚: CS | 40 | | | ns |

VDD = 3.0V ± 5%, Ta = 25°C

8.5 电源启动后复位时序



| 项目 | 符号 | 测试条件 | 极限值 | | | 单位 |
|------------|----------|---------|-----|------|-----|----|
| | | | MIN | TYPE | MAX | |
| 复位时间 | t_r | | -- | -- | 1.0 | us |
| 复位保持低电平的时间 | t_{rw} | 引脚: RES | 1.0 | -- | -- | us |

9. 指令功能

9.1 指令表

| 指令名称 | | 指令码 | | | | | | | | 说明 | |
|------------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| | | RS | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | | DB0 |
| (1)显示开/关 | | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | D | D=1 开; D=0 关 |
| (2)显示初始行 | | 0 | 0 | 1 | S5 | S4 | S3 | S2 | S1 | S0 | 设置显示存储器的显示初始行,可设置为0x40~0x7F,分别代表第0~63行 |
| (3)页地址设置 | | 0 | 1 | 0 | 1 | 1 | Y3 | Y2 | Y1 | Y0 | 设置显示页地址。每8行为一页,64行分为8个页 |
| (4) | 列地址高4位 | 0 | 0 | 0 | 0 | 1 | X7 | X6 | X5 | X4 | 高4位与低4位共同组成列地址,寻址128列,第一列的地址为0x00;指令表达为:0x10,0x00 |
| | 列地址低4位 | 0 | 0 | 0 | 0 | 0 | X3 | X2 | X1 | X0 | |
| (5)读状态 | | 0 | 0 | MX | D | RST | 0 | 0 | 0 | 0 | 本型号没用到 |
| (6)写数据 | | 1 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | 从CPU写数据到LCM |
| (7)读数据 | | 1 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | 本型号没用到 |
| (8)显示列地址增减 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | MX | MX=0: 常规, 从左到右 MX=1: 反转, 从右到左 |
| (9)显示正显/反显 | | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | INV | INV=0: 常规, 正显 INV=1: 反显 |
| (10)显示全部点阵 | | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | AP | AP=0: 常规 AP=1: 显示全部点阵 |
| (11)LCD 偏压比设置 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | BS | BS=0:1/9 Bias BS=1:1/7 Bias |
| (12)读-修改-写 | | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 写的时候地址自加1, 读的时候地址不变 |
| (13)退出上述指令 | | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 退出“读-修改-写”模式 |
| (14)软件复位 | | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 软件复位 |
| (15)行扫描顺序选择(COM) | | 0 | 1 | 1 | 0 | 0 | MY | 0 | 0 | 0 | MY=0: 普通顺序 MY=1: 反向扫描 |
| (16)电源控制 | | 0 | 0 | 0 | 1 | 0 | 1 | VB | VR | VF | 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 | RR2 | RR1 | RR0 | 选择内部电阻比例 |
| (18) | 设置内部电压模式 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 设置内部电阻微调, 以设置液晶电压, 这两个指令需紧接着使用 |
| | 设置的电压值 | 0 | 0 | 0 | EV5 | EV4 | EV3 | EV2 | EV1 | EV0 | |

| | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|-----|-----|---------|---------|
| (19) 升压倍数选择 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 00: 4 倍 | 01: 5 倍 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | BL1 | BL0 | 10: 6 倍 | |
| (20) 空指令 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 空操作 | |

9.2 地址映射

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

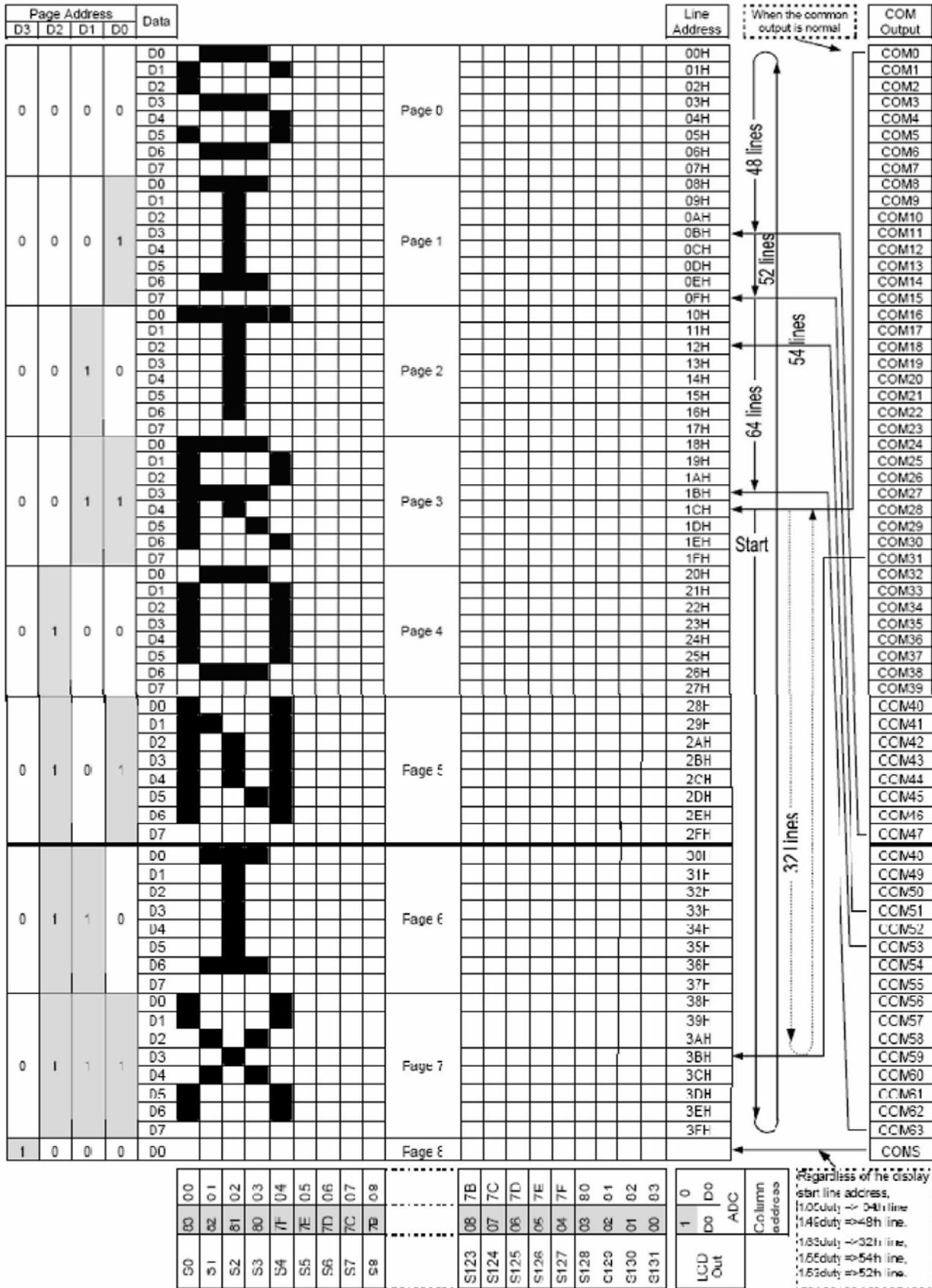
DB7—DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵，通常“1”代表点亮该点阵，“0”代表关掉该点阵。 如下图所示：

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

Display data RAM
(显示数据存储器)

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

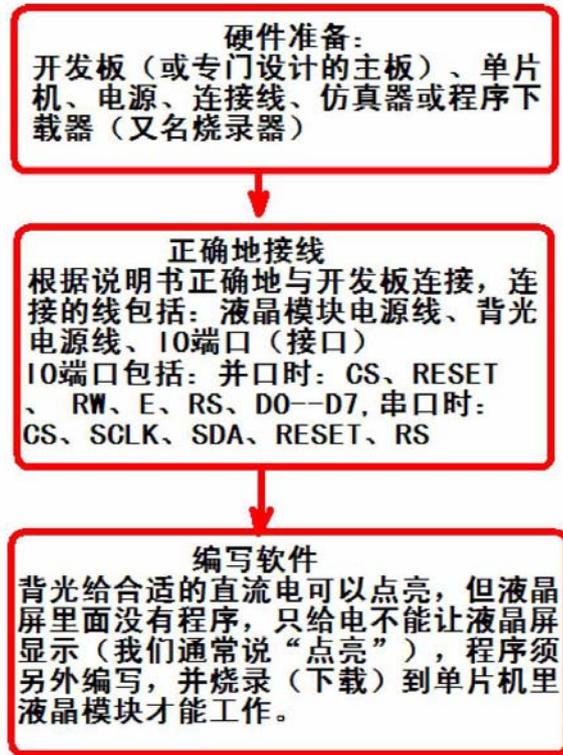
Liquid crystal display
(液晶屏)



9.3 初始化

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

点亮液晶模块的步骤

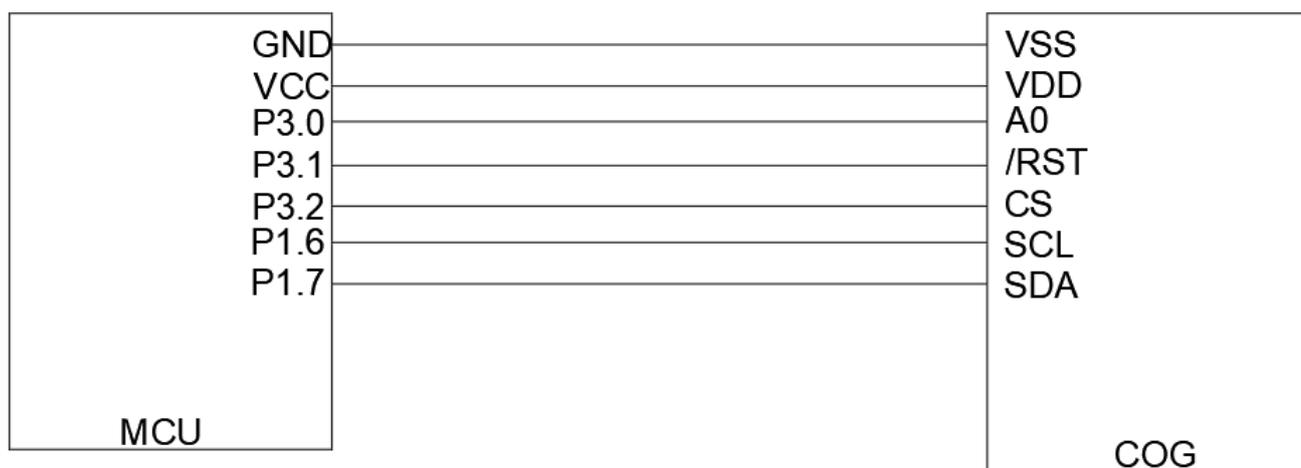


9.4 液晶模块与 MCU(以 51 单片机为例) 并口接线图如下



COG7567 并行传输接线方法

9.5 液晶模块与 MCU(以 51 单片机为例) 串口接线图如下



COG7567 串行传输接线方法

10. 字符汉字库使用说明

10.1 引脚描述

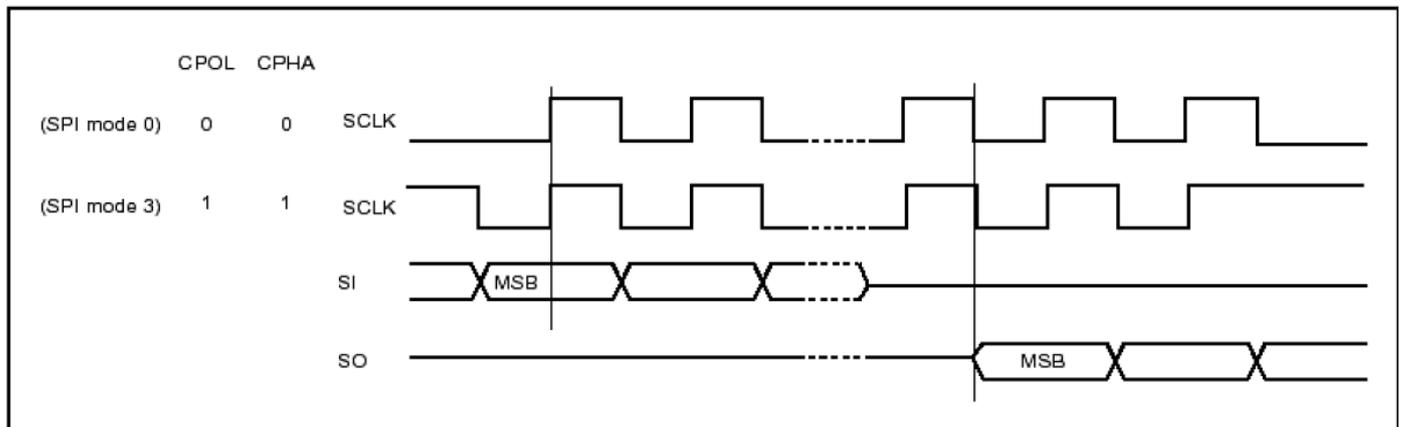
| 名称 | I/O | 描述 |
|-------|-----|-----------------------------|
| SCLK | I | 串行时钟输入 (Serial clock input) |
| GND | | 地(Ground) |
| CS# I | | 片选输入 (Chip enable input) |
| VCC | | 电源(+ 3.3V Power Supply) |
| SO | | 串行数据输出 (Serial data output) |
| SI | | 串行数据输入 (Serial data input) |

串行数据输出 (SO)：该信号用来把数据从芯片串行输出，数据在时钟的下降沿移出。

串行数据输入 (SI)：该信号用来把数据从串行输入芯片，数据在时钟的上升沿移入。

串行时钟输入 (SCLK)：数据在时钟上升沿移入，在下降沿移出。

片选输入 (CS#)：所有串行数据传输开始于CS#下降沿，CS#在传输期间必须保持为低电平，在两条指令之间保持为高电平。



10.2 指令参数

Instruction Set

| Instruction | Description | Instruction Code(One-Byte) | | Address Bytes | Dummy Bytes | Data Bytes |
|-------------|---------------------------------|----------------------------|------|---------------|-------------|---------------|
| READ | Read Data Bytes | 0000 0011 | 03 h | 3 | - | 1 to ∞ |
| FAST_READ | Read Data Bytes at Higher Speed | 0000 1011 | 0B h | 3 | 1 | 1 to ∞ |

所以对字库的操作只有2个，那就是 Read Data Bytes (READ “一般读取”) 和 Read Data Bytes at Higher Speed (FAST_READ “快速读取点阵数据”)。

10.2.1 Read Data Bytes (一般读取)

Read Data Bytes 需要用指令码来执行每一次操作。READ 指令的时序如下(图):

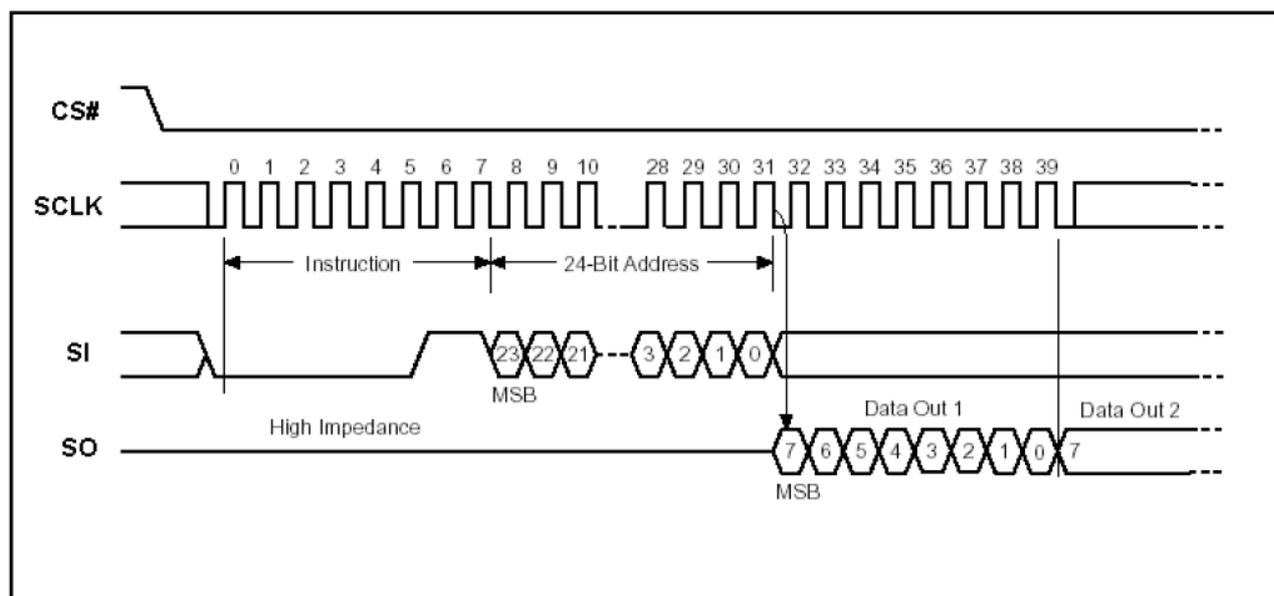
■首先把片选信号 (CS#) 变为低，紧跟着的是 1 个字节的命令字 (03 h) 和 3 个字节的地址和通过串行数据输入引脚 (SI) 移位输入，每一位在串行时钟 (SCLK) 上升沿被锁存。

■然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出，每一位在串行时钟 (SCLK) 下降沿被移出。

■读取字节数据后，则把片选信号 (CS#) 变为高，结束本次操作。

如果片选信号 (CS#) 继续保持为低，则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。

图：Read Data Bytes (READ) Instruction Sequence and Data-out sequence:



10.2.2 Read Data Bytes at Higher Speed(快速读取点阵数据)

Read Data Bytes at Higher Speed 需要用指令码来执行操作。READ_FAST 指令的时序如下(图):

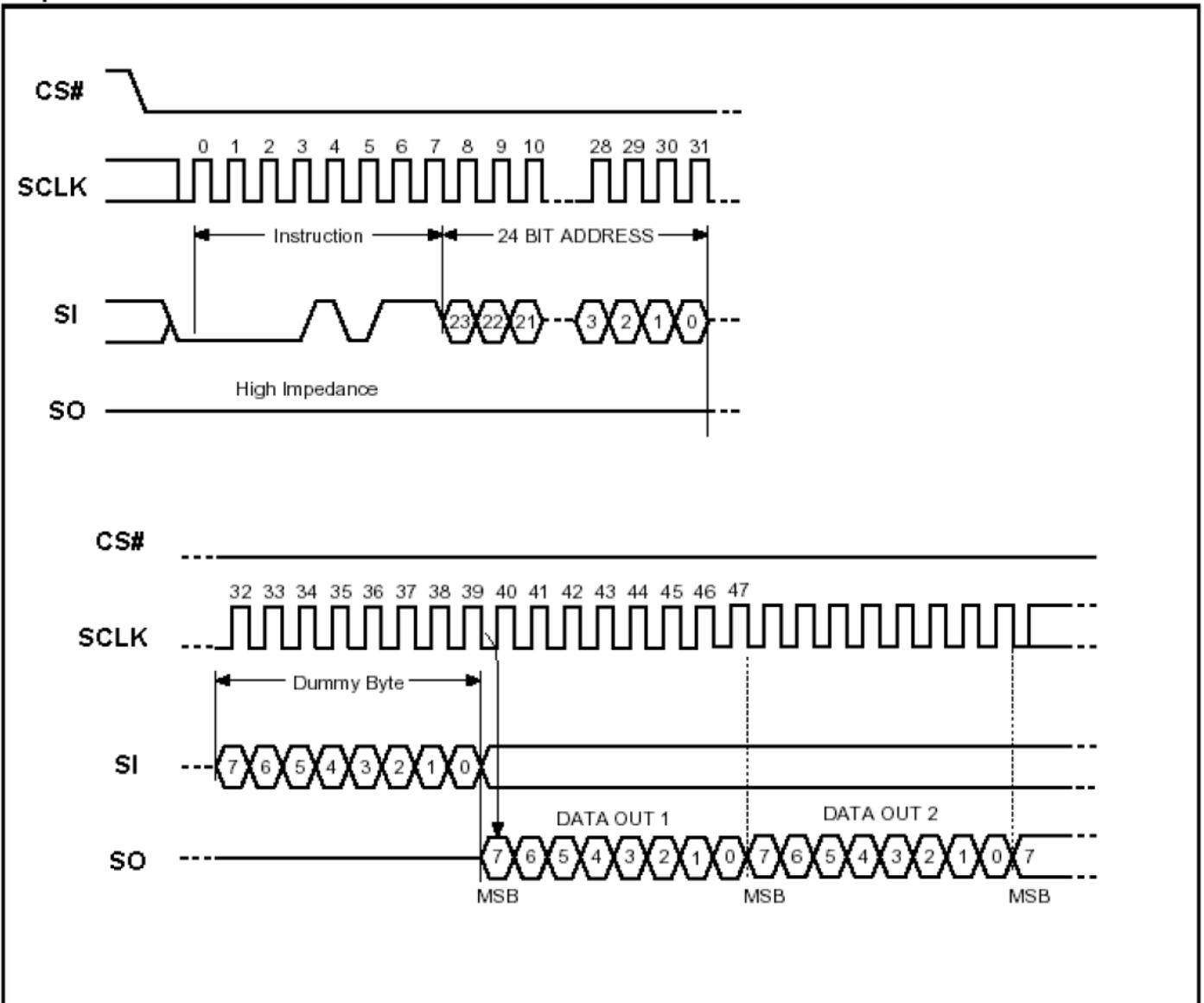
■首先把片选信号 (CS#) 变为低, 紧跟着的是 1 个字节的命令字 (0B h) 和 3 个字节的地址以及一个字节 Dummy Byte 通过串行数据输入引脚 (SI) 移位输入, 每一位在串行时钟 (SCLK) 上升沿被锁存。

■然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出, 每一位在串行时钟 (SCLK) 下降沿被移出。

■如果片选信号 (CS#) 继续保持为低, 则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。例: 读取一个 15x16 点阵汉字需要 32Byte, 则连续 32 个字节读取后结束一个汉字的点阵数据读取操作。

如果不需要继续读取数据, 则把片选信号 (CS#) 变为高, 结束本次操作。

图: Read Data Bytes at Higher Speed (READ_FAST) Instruction Sequence and Data-out



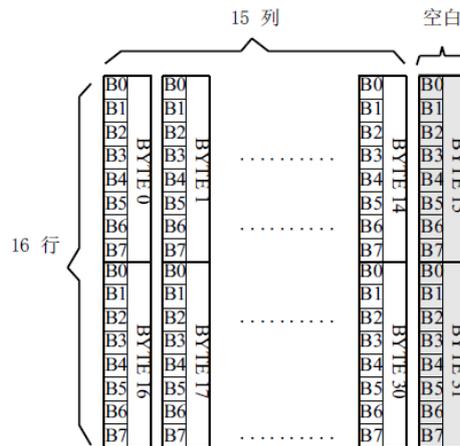
10.3 字库调用方法

10.3.1 汉字点阵排列格式

每个汉字在字库中是以汉字点阵字模的形式存储的，每个点用一个二进制位表示，存1的点为点亮，存0的点为不显示。点阵排列格式为竖置横排：即一个字节的低位表示下面的点，高位表示上面的点，排满一行后再排下一行。这样把点阵信息用来直接显示，则将显示对应的汉字。

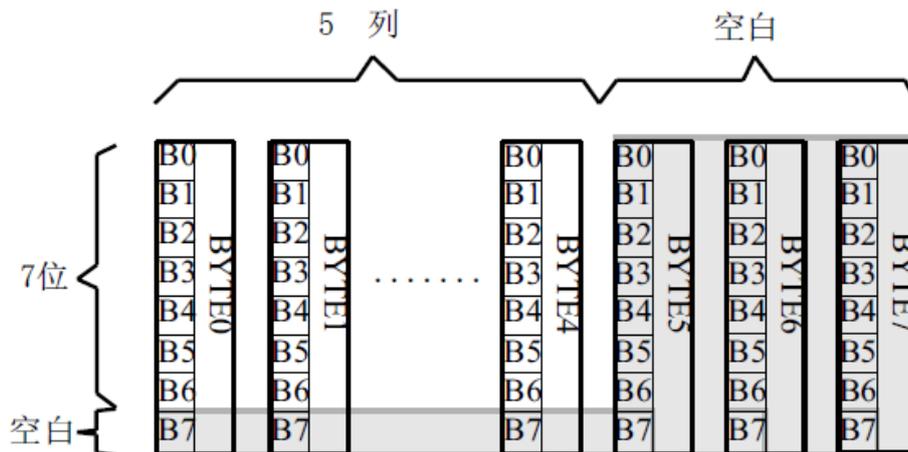
15X16点汉字排列格式

15X16点汉字的字模需要32个字节（BYTE0-BYTE31）来表示。该15X16点汉字的点阵数据是竖置横排的，其具体排列结构如下图：



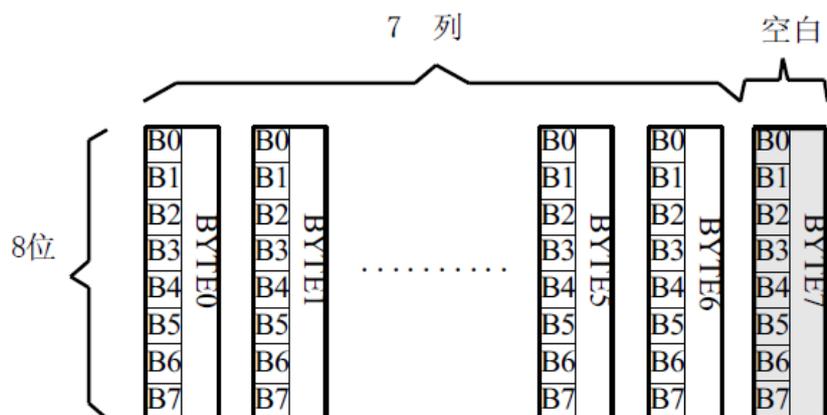
5X7点ASCII字符排列格式

5X7点ASCII的字模需要8个字节（BYTE0-BYTE7）来表示。该ASCII点阵数据是竖置横排的，其具体排列结构如下图：



7X8点ASCII字符排列格式

7X8点ASCII的字模需要8个字节（BYTE0-BYTE7）来表示。该ASCII点阵数据是竖置横排的，其具体排列结构如下图：



8X16点字符排列格式

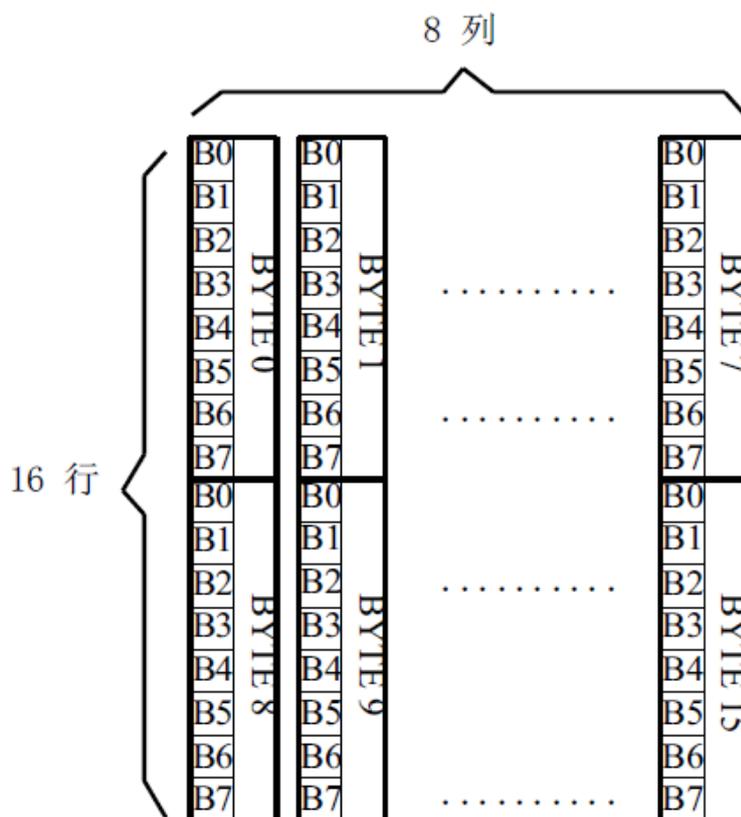
适用于此种排列格式的字体有：

8X16点ASCII字符

8X16点ASCII粗体字符

8X16点国标扩展字符

8X16点字符字模需要16个字节（BYTE0-BYTE15）来表示。该点阵数据是竖置横排的，其具体排列结构如下图：

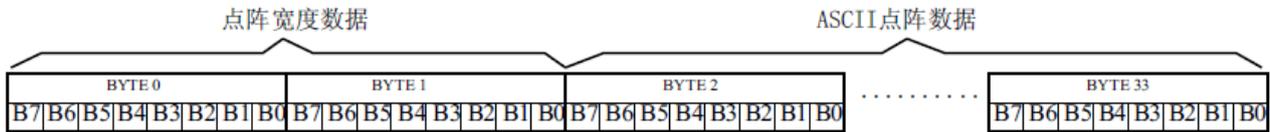


16点阵不等宽ASCII方头 (Arial)、白正 (Times New Roman) 字符排列格式

16点阵不等宽字符的字模需要34个字节 (BYTE0-BYTE33) 来表示。

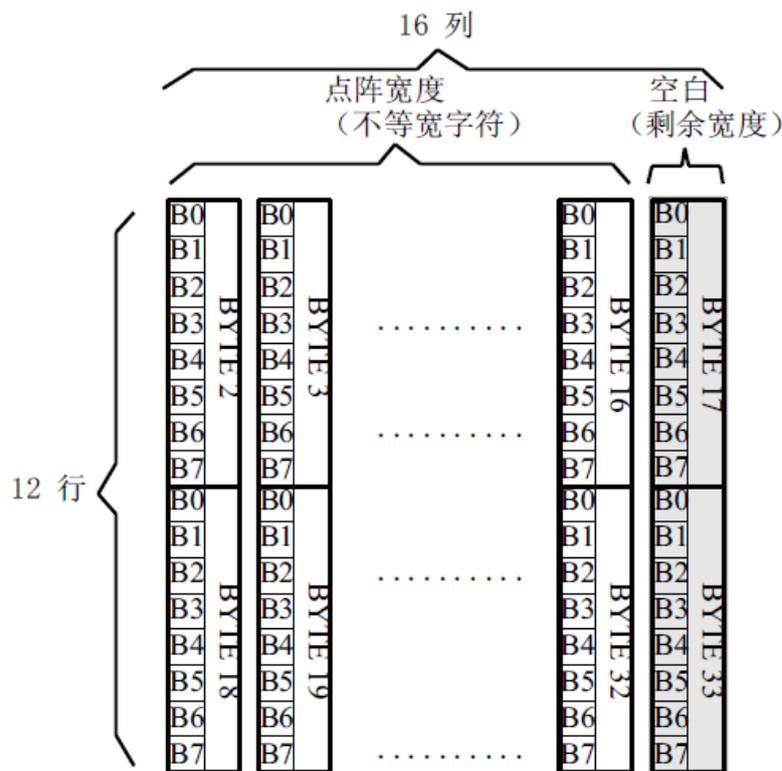
■ 存储格式

由于字符是不等宽的，因此在存储格式中BYTE0-BYTE1存放点阵宽度数据，BYTE2-33存放竖置横排点阵数据。具体格式见下图：



■ 存储结构

不等宽字符的点阵存储宽度是以BYTE为单位取整的，根据不同字符宽度会不现相应的空白区。根据BYTE0-BYTE1所存放点阵的实际宽度数据，可以对还原下一个字的显示或排版留作参考。



例如：ASCII方头字符B

0-33BYTE的点阵数据是：00 0C 00 F8 F8 18 18 18 18 18 F8 F0 00 00 00 00 00 00 7F 7F 63
63 63 63 63 67 3E 1C 00 00 00 00 00

其中：

BYTE0-BYTE1：00 0C为ASCII方头字符B的点阵宽度数据，即：12位宽度。字符后面有4位空白区，可以在排版下一个字时考虑到这一点，将下一个字的起始位置前移。

BYTE2-33：00 F8 F8 18 18 18 18 18 F8 F0 00 00 00 00 00 00 00 7F 7F 63 63 63 63 67
3E 1C 00 00 00 00 00 为ASCII方头字符B的点阵数据。

10.3.2 汉字点阵字库地址表

| | 字库内容 | 编码体系 | 码位范围 | 字符数 | 起始地址 | 结束地址 | 参考算法 |
|---|-------------------------------------|--------|-----------|----------|-------|-------|---------|
| 1 | 15X16点 GB2312 标准点阵字库 | GB2312 | A1A1-F7FE | 6763+376 | 00000 | 3B7BF | 6.3.1.1 |
| 2 | 7X8点 ASCII 字符 | ASCII | 20~7F 96 | | 66C0 | 69BF | 6.3.2.2 |
| 3 | 8X16点国标扩展字符 | GB2312 | AAA1-ABC0 | 126 | 3B7D0 | 3BFBF | 6.3.1.2 |
| 4 | 8X16点 ASCII 字符 | ASCII | 20~7F | 96 | 3B7C0 | 3BFBF | 6.3.2.3 |
| 5 | 5X7点 ASCII 字符 ASCII | | 20~7F | 96 | 3BFC0 | 3C2BF | 6.3.2.1 |
| 6 | 16点阵不等宽 ASCII 方头 (Arial) 字符 | ASCII | 20~7F | 96 | 3C2C0 | 3CF7F | 6.3.2.4 |
| 7 | 8X16点 ASCII 粗体字符 ASCII | | 20~7F | 96 | 3CF80 | 3D57F | 6.3.2.5 |
| 8 | 16点阵不等宽 ASCII 白正 (TimesNewRoman) 字符 | ASCII | 20~7F | 96 | 3D580 | 3E23F | 6.3.2.6 |

10.4 字符在字库中的地址计算方法

用户只要知道字符的内码，就可以计算出该字符点阵在字库中的地址，然后就可以从该地址连续读出点阵字模用于显示。

10.4.1 汉字字符的地址计算

15X16点GB2312标准点阵字库

参数说明：

GBCode表示汉字内码。

MSB表示汉字内码GBCode的高8bits。

LSB表示汉字内码GBCode的低8bits。

Address 表示汉字或ASCII字符点阵在字库中的字节地址。

BaseAdd 说明点阵数据在字库中的起始地址。

计算方法：

BaseAdd=0;

if(MSB ==0xA9 && LSB >=0xA1)

Address= (282+(LSB-0xA1))*32+BaseAdd;

else if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)

Address= ((MSB-0xA1)*94+(LSB-0xA1))*32+BaseAdd;

else if(MSB >=0xB0 && MSB <=0xF7 && LSB >=0xA1)

Address= ((MSB-0xB0)*94+(LSB-0xA1)+846)*32+BaseAdd;

8X16点国标扩展字符

说明：

BaseAdd:说明本套字库在字库芯片中的起始字节地址。

FontCode:表示字符内码 (16bits)

ByteAddress:表示字符点阵在字库中的字节地址。

计算方法：

BaseAdd=0x3b7d0

if(FontCode>=0xAAA1)and (FontCode<=0xAAFE)then

ByteAddress=(FontCode-0xAAA1)*16+BaseAdd

Else if(FontCode>=0xABA1)and (FontCode<=0xABC0)then

ByteAddress=(FontCode-0xABA1+95)*16+BaseAdd

10.4.2 ASCII字符的地址计算

5X7点ASCII字符

参数说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

计算方法:

BaseAdd=0x3bfc0

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*8+BaseAdd

7X8点ASCII字符

说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

Address:ASCII字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x66c0

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*8+BaseAdd

8X16点ASCII字符

说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

Address:ASCII字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3b7c0

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*16+BaseAdd

16点阵不等宽ASCII方头(Arial)字符

说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

Address:ASCII字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3c2c0

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*34+BaseAdd

8X16点ASCII粗体字符

参数说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

Address:ASCII字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3cf80

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*16+BaseAdd

16点阵不等宽ASCII白正(Times New Roman)字符

说明:

ASCIICode:表示ASCII码(8bits)

BaseAdd:说明该套字库在芯片中的起始地址。

Address:ASCII字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3d580

if (ASCIICode >=0x20) and (ASCIICode <=0x7E) then

Address=(ASCIICode-0x20)*34+BaseAdd

10.5 GB2312 1区(376字符)

GB2312标准点阵字符1区对应码位的A1A1-A9EF共计376个字符;

| A1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|---|---|---|---|---|---|---|----|---|---|---|----|---|----|----|----|
| A | | | 、 | 。 | · | - | ˇ | ¨ | ” | 々 | — | ~ | | … | ‘ | ’ |
| B | “ | ” | { | } | < | > | < | > | 「 | 」 | 『 | 』 | 【 | 】 | 【 | 】 |
| C | ± | × | ÷ | : | ∧ | ∨ | Σ | Π | U | ∩ | € | :: | √ | ⊥ | // | ∠ |
| D | ^ | ⊙ | ∫ | φ | = | ∞ | ≈ | ∞ | ∞ | ≠ | < | > | < | > | ∞ | :: |
| E | ∴ | ↑ | ♀ | ° | ' | ” | ℃ | \$ | ¤ | ¢ | £ | % | § | No | ☆ | ★ |
| F | ○ | ● | ◎ | ◇ | ◆ | □ | ■ | △ | ▲ | ※ | → | ← | ↑ | ↓ | = | |

| A2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|------|------|
| A | | | | | | | | | | | | | | | | |
| B | | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. | 13. | 14. | 15. |
| C | 16. | 17. | 18. | 19. | 20. | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) |
| D | (12) | (13) | (14) | (15) | (16) | (17) | (18) | (19) | (20) | ① | ② | ③ | ④ | ⑤ | ⑥ | ⑦ |
| E | ⑧ | ⑨ | ⑩ | € | (一) | (二) | (三) | (四) | (五) | (六) | (七) | (八) | (九) | (十) | | |
| F | | I | II | III | IV | V | VI | VII | VIII | IX | X | XI | XII | | | |

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| A | | ! | " | # | ¥ | % | & | ' | (|) | * | + | , | - | . | / |
| B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| C | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| D | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| E | ' | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| F | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | |

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|-----|-----|---|---|-----|-----|---|---|
| A9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| A | | | | | — | — | | | --- | --- | ! | ! | --- | --- | ! | ! |
| B | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| C | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ | ┌ |
| D | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ | └ |
| E | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |
| F | | | | | | | | | | | | | | | | |

10.6 8X16点国标扩展字符（126字符）

内码组成为AAA1-ABCO共计126个字符

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AA | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| A | | ! | " | # | ¥ | % | & | ' | (|) | * | + | , | - | . | / |
| B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| C | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| D | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| E | ' | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| F | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | |

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AB | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| A | | ā | á | ǎ | à | ē | é | ě | è | ī | í | ǐ | ì | ō | ó | ǒ |
| B | ò | ū | ú | ǔ | ù | ǘ | ú | ǚ | ù | ü | ê | á | ń | ň | ñ | ñ |
| C | g | | | | | | | | | | | | | | | |

10. 使用过程中注意事项

尊敬的顾客：感谢您使用我司生产的液晶显示模块，在使用前请仔细阅读以下内容，主要讲

如何正确使用液晶显示模块，防止为您带来损失，如果您在使用过程中遇到困难，请拨打我司技术服务部电话，我们在最短的时间内为您提供服务。

一：保护膜

液晶显示模块表面贴有一层保护膜，是防止在装配时沾污显示表面，在整机装配结束前不得揭去。

二：防止静电

液晶显示模块中的液晶显示驱动芯片是大规模的 COMS 电路，极易被静电击穿，而静电击穿是不可修复，所以在操作、装配以及使用中都应严防静电，因此必须注意：

1：不能用手随意接触液晶显示模块的驱动芯片与电路板上的金手指，若必须接触时，一定带上抗静电手环，防静电手指套/手套，最好穿上防静电工作服，使人体良好接地。

2：工作台，烙铁及工具良好接地。

3：工作间保持湿度在 RH60%以上。

4：包装材料都是防静电材料。 三：焊

接

1：烙铁头温度小于 280℃。

2：焊接时间小于 3-4S。

3：不使用酸性助焊剂。

4：重复焊接不超过 3 次，且每次重复需间隔 5 分钟。

5：焊接完后彻底清洗，防止焊接短路和产生信号干扰，注意不能把清洗液洒到背光上。

四：装机

不得私自更改液晶显示模块。如：更改零件、更改电路、扭动铁框脚、拆卸、对 PCB 板加工。以上行为造成液晶显示模块不显示一切不给予维修，除经我司人员同意后方可。

五：测试

1：液晶显示模块与单片机接法请参照《接线图》。注意正、负电源的极性不能

搞错, 否则造成过流、过压、烧毁电路上的芯片。 2: 正确接线后, 开始测试液晶显示模块时, 先打开逻辑电源, 再打开液晶显示模块的驱动电源; 断电时, 先关掉液晶显示模块的驱动电源, 再关掉逻辑电源. 在正电源稳定接入以后, 才能输入信号电平, 如在电源稳定前或断开后输入信号电平, 可能损坏液晶显示模块中的 IC。

3: 液晶显示模块的显示效果会随着温度的变化而相应变化。外界环境温度超过液晶显示模块的工作温度时, 液晶态消失, 变成液态, 显示画面呈黑色, 此时一定不可通电, 等温度降低后可自行恢复; 反之, 温度过低, 液晶开始结成冰花, 造成永久的损伤。另外长时间储存在极限温度下或受振动、冲击, LCD 还会产生气泡。

这就是在液晶显示模块上加温度补偿电路的原因, 或者外部加电位器来调节对比度。

4: 液晶显示模块 VOP 不能输出过高, 会影响显示效果, 还会缩短模块的使用寿命。请参考每款产品说明书的正确 VOP 电压值。

五: 存储

1: 保持原包装。

2: 宽温型在 $-30^{\circ}\text{C} \sim +80^{\circ}\text{C}$ 范围内保存; 普温型在 $-10^{\circ}\text{C} \sim +60^{\circ}\text{C}$ 范围内保存。

3: 放暗处, 避强光。

4: 不能在表面压放任何物品。

5: 避免在极限温/湿度条件下存放。