

X16032C-1 使用说明书

目 录

序号	内 容 标 题	页码
1	概述	2
2	字符型模块的特点	2
3	外形及接口引脚功能	2~3
4	基本原理	4
5	技术参数	4
6	时序特性	5~6
7	指令功能及硬件接口与编程案例	6~15

1. 概述

中显电子专注于液晶屏及液晶模块的研发、制造。所生产X16032C-1 型液晶模块由于使用方便、带中文字库、显示清晰，广泛应用于各种人机交流面板。

X16032C-1液晶显示模块是160×32 点阵的汉字图形型液晶显示模块，可显示汉字及图形，内置8192 个中文汉字（16X16 点阵）、128 个字符（8X16 点阵）及64X256点阵显示RAM（GDRAM）。可与CPU 直接接口，提供两种界面来连接微处理器：8-位并行及串行两种连接方式。具有多种功能：光标显示、画面移位、睡眠模式等。

2. X16032C-1 图像型点阵液晶模块的特性

- 1.1 结构牢：带 PCB、背光、铁框
- 1.2 IC 采用矽创公司 ST7920, 功能强大，稳定性好
- 1.3 功耗低:10 - 100mW（不带背光 10mW, 带背光不大于 100mW）；
- 1.4 显示内容：
 - 160*32 点阵单色图片；
 - 内置8192 个中文汉字（16X16 点阵）、128 个字符（8X16 点阵）及64X256点阵显示RAM（GDRAM）。
- 1.5 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；
- 1.6 接口简单方便:采用 3 线 SPI 串行接口，可只需 3 位 MPU 的端口。也可选用 8 位并行接口。
- 1.7 工作温度宽:-20℃ - 70℃；
- 1.8 可靠性高:寿命为 50,000 小时(25℃)。

3. 外形尺寸及接口引脚功能

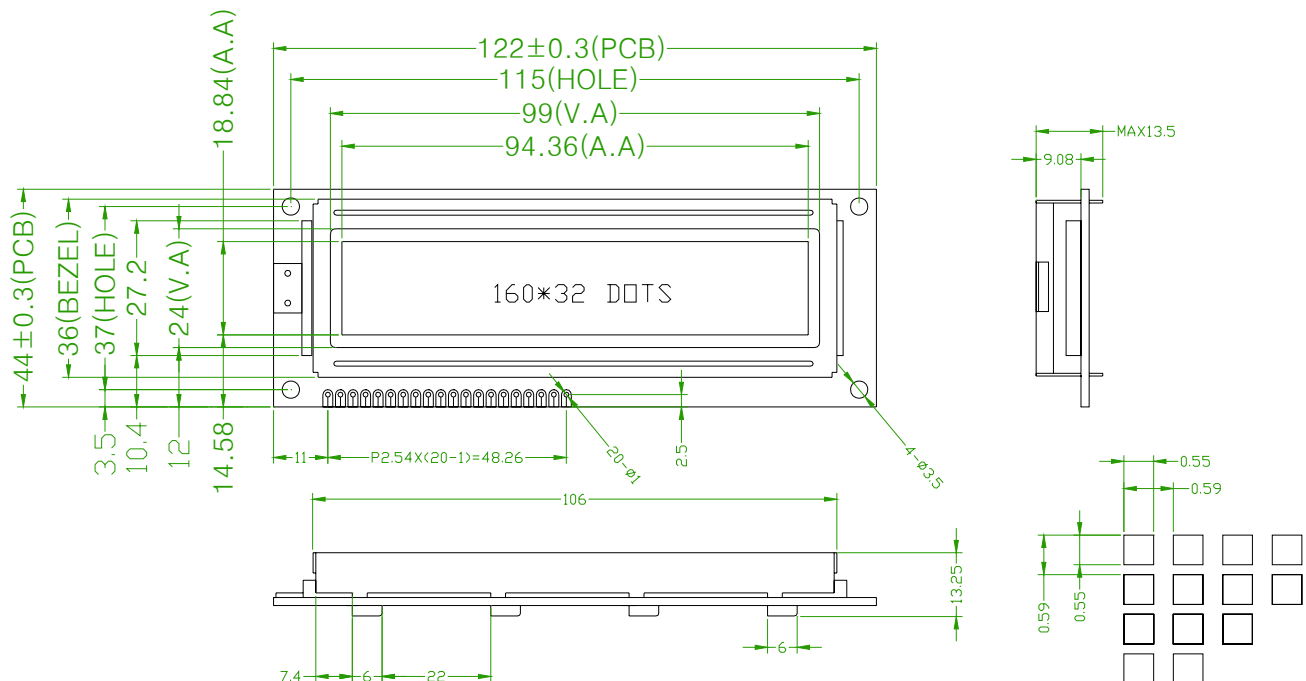


图 1. 外形尺寸

模块的接口引脚功能

引脚	符 号	名 称	功 能
1	VSS	接地	0V
2	VDD	电路电源	5V, 或 3.3V 可选
3	V0	LCD V0 电压输入	可以通过此脚对 LCD 驱动电压进行调整
4	RS(CS*)	寄存器选择信号 (串行时为片选: CS)	1. 并行接口时: 1:数据寄存器 0:指令寄存器 2. 串行接口时: 片选信号, 低电平有效
5	R/W(SID*)	读写选择(串行时 为串行数据:SID)	1. 并行接口时: 0: 写 1:读 2. 串行时为串行数据输入: SID
6	E(SCLK*)	读写使能信号(串 行时为串行时钟: SCLK)	1. 并行接口时: 读写使能信号 2. 串行时为串行时钟: SCLK
7~14	D0~D7	数据 DB0~DB7	并行接口时: 数据总线 DB0~DB7 串行接口时: 无效, 空脚 4 位并行接口时, DB4~DB7 作为数据总线, DB0~DB3 不起作用
15	PSB	并行/串行选择	1: 选择并行, 0: 选择串行, 也可在 PCB 上与 VDD(1)或 VSS(0) 连接达到选择并/串接口。
16	NC	空脚	
17	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
18	VEE	升压输出	一般不接
19	BLA	LED 背光正极	LED 背光正极, 5V
20	BLK	LED 背光负极	LED 背光负极, 0V

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 160×32 点阵。

4.2 工作电图:

图 1 是X16032C-1 图像点阵型模块的电路框图, 它由驱动 IC ST7920\ST7921 及几个电阻电容组成。

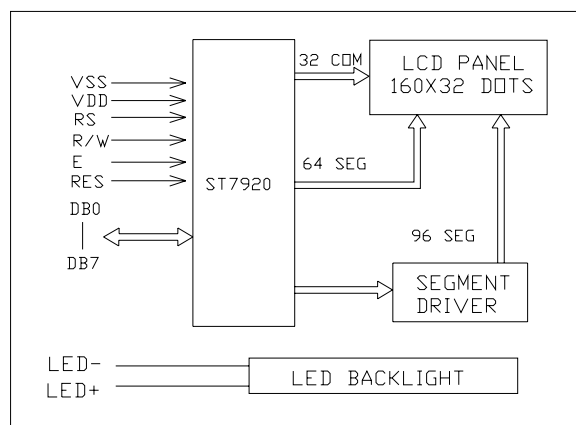


图 2: X16032C-1 图像点阵型液晶模块的电路框图

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°C ~ $+70^{\circ}\text{C}$ ；

存储温度： -30°C ~ $+80^{\circ}\text{C}$ ；

背光板可显示绿色, 黄绿色, 兰色和白色。背光一般为绿色, 也可为客户设计为其他颜色, 但价格较绿色贵一点。

正常工作电流为： $10\sim 20\text{mA}$ （若 LED 灯数不止一颗, 则乘以相应数量）；

工作电压： 5.0V ；

正常工作条件下, LED 可连续点亮 5 万小时；

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	$^{\circ}\text{C}$
储存温度		-30		+80	$^{\circ}\text{C}$

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	5.0V 供电	4.0	5.0	5.2	V
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIL	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOL	IOL = 1.2mA	-		0.4	V
工作电流	IDD	VDD = 5.0V		2.0		mA

表 3：直流（DC）参数

6. 读写时序特性

6.1 并行接口时：

从 CPU 写到 LCD 驱动 IC:ST7920 (Writing Data from CPU to ST7920)

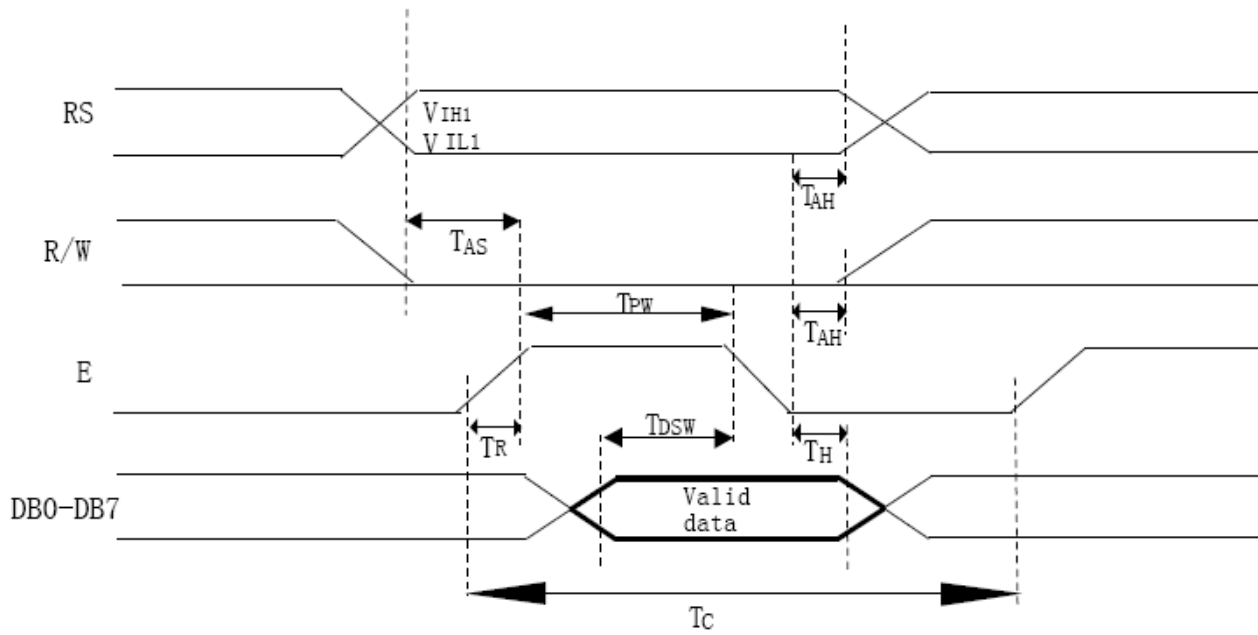
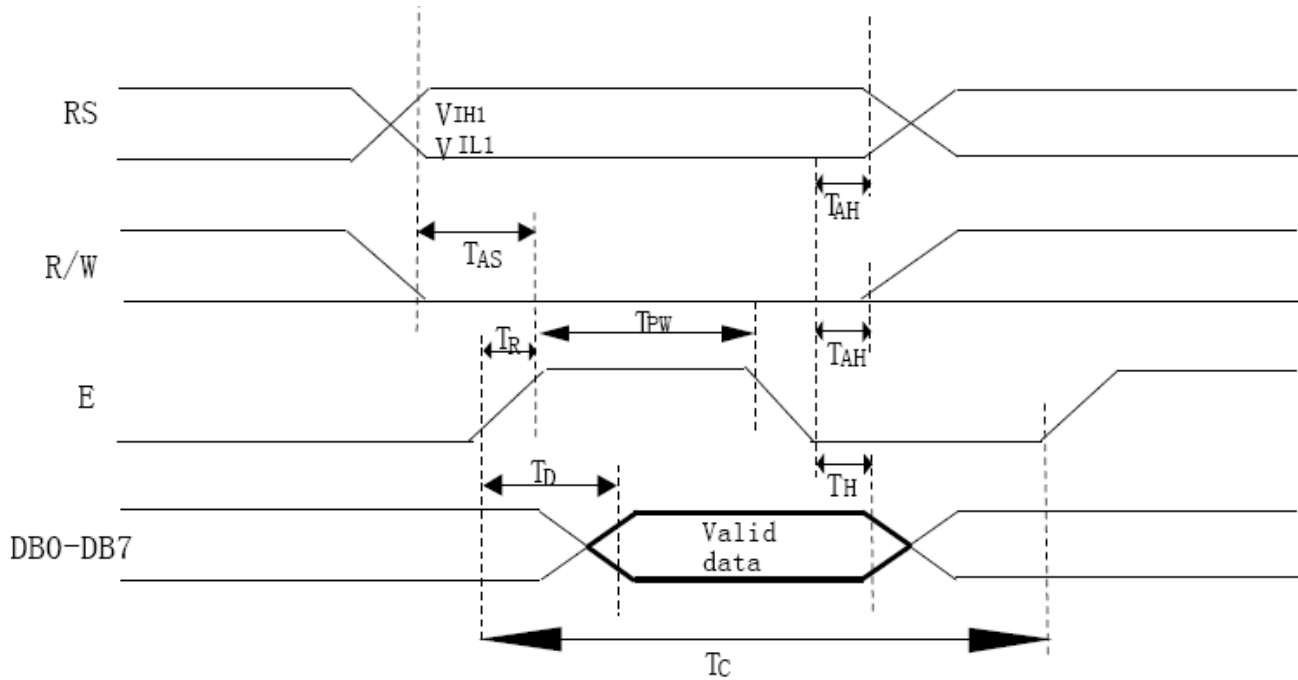


图 3. 从 CPU 写到 IC:ST7920 (Writing Data from CPU to ST7920)

从 LCD 驱动 IC:ST7920 读到 CPU



7. 指令功能:

7.1 指令表

指令表 1: (RE=0: 基本指令集)

指令	指令碼										說明	執行時間 (540KHZ)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
清除顯示	0	0	0	0	0	0	0	0	0	1	將 DDRAM 填滿 "20H", 並且設定 DDRAM 的位址計數器 (AC) 到 "00H"	4.6 ms
位址歸位	0	0	0	0	0	0	0	0	1	X	設定 DDRAM 的位址計數器 (AC) 到 "00H", 並且將游標移到開頭原點位置; 這個指令並不改變 DDRAM 的內容	72us
進入點設定	0	0	0	0	0	0	0	1	I/D	S	指定在資料的讀取與寫入時, 設定游標的移動方向及指定顯示的移位	72us
顯示狀態 開/關	0	0	0	0	0	0	1	D	C	B	D=1: 整體顯示 ON C=1: 游標 ON B=1: 游標位置反白 ON	72 us
游標或顯示 移位控制	0	0	0	0	0	1	S/C	R/L	X	X	設定游標的移動與顯示的移位控制位元; 這個指令並不改變 DDRAM 的內容	72 us
功能設定	0	0	0	0	1	DL	X	0 RE	X	X	DL=1 8-BIT 控制介面 DL=0 4-BIT 控制介面 RE=1: 擴充指令集動作 RE=0: 基本指令集動作	72 us
設定 CGRAM 位址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	設定 CGRAM 位址到位址計數器 (AC) 需確認擴充指令中 SR=0 (捲動位址或 RAM 位址選擇)	72 us
設定 DDRAM 位址	0	0	1	0 AC6	AC5	AC4	AC3	AC2	AC1	AC0	設定 DDRAM 位址到位址計數器 (AC) AC6 固定為 0	72 us
讀取忙碌旗 標 (BF) 和 位址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	讀取忙碌旗標 (BF) 可以確認內部動作是否完成, 同時可以讀出位址計數器 (AC) 的值	0 us
寫資料到 RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	寫入資料到內部的 RAM (DDRAM/CGRAM/IRAM/GDRAM)	72 us
讀出 RAM 的值	1	1	D7	D6	D5	D4	D3	D2	D1	D0	從內部 RAM 讀取資料 (DDRAM/CGRAM/IRAM/GDRAM)	72 us

指令表 2: (RE=1: 擴充指令集)

指令	指令碼										說明	執行時間 (540KHZ)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
待命模式	0	0	0	0	0	0	0	0	0	1	進入待命模式，執行任何其他指令都可終止待命模式	72 us
捲動位址或 RAM 位址 選擇	0	0	0	0	0	0	0	0	1	SR	SR=1: 允許輸入垂直捲動位址 SR=0: 允許輸入 IRAM 位址(擴充指令) SR=0: 允許設定 CGRAM 位址(基本指令)	72 us
反白選擇	0	0	0	0	0	0	0	1	R1	R0	選擇 4 行中的任一 行作反白顯示，並可決定反白與否 R1,R0 初值為 00 當第一次設定時為反白顯示在一次設定時為正常顯示	72 us
睡眠模式	0	0	0	0	0	0	1	SL	X	X	SL=1: 脫離睡眠模式 SL=0: 進入睡眠模式	72 us
擴充 功能設定	0	0	0	0	1	DL	X	1 RE	G	0	DL=1 8-BIT 控制介面 DL=0 4-BIT 控制介面 <u>RE=1: 擴充指令集動作</u> <u>RE=0: 基本指令集動作</u> G=1 :繪圖顯示 ON G=0 :繪圖顯示 OFF	72 us
設定 IRAM 位址 或捲動位址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	SR=1: AC5~AC0 為垂直捲動位址 SR=0: AC3~AC0 為 ICON RAM 位址	72 us
設定繪圖 RAM 位址	0	0	1	0	0	0	AC3	AC2	AC1	AC0	設定 GDRAM 位址到位址計數器 (AC) 先設垂直位址再設水平位址(連續寫入兩個位元組的資料來完成垂直與水平的座標位址) 垂直位址範圍 AC6...AC0 水平位址範圍 AC3...AC0	72 us

请详细参考 IC 资料“ST7920C17.PDF”的第 13~14 页。

备注;当IC1在接受指令前,微处理器必须先确认其内部处于非忙碌状态,即读取BF标志时,BF 需为零,方可接受新的指令;如果在送出一个指令前并不检查BF 标志,那么在前一个指令和这个指令中间必须延长一段较长的时间,即是等待前一个指令确实执行完成。

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

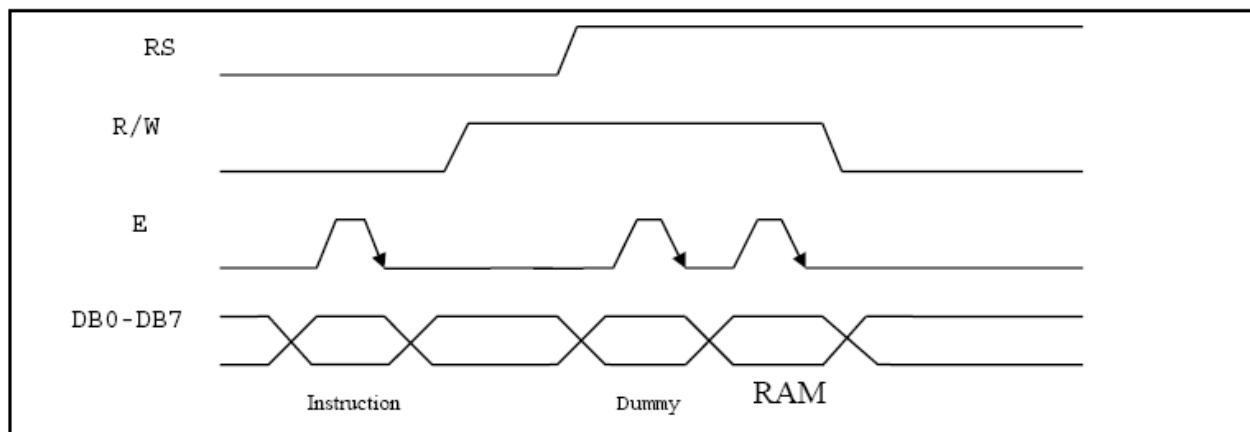
並列介面資料傳輸訊號

當PSB腳接高電位時，ST7920將進入並列模式，在並列模式下可由指令 **DL FLAG** 來選擇8-位元或4-位元介面，主控制系統將配合(RS, RW, E, DB0..DB7)來達成傳輸動作。

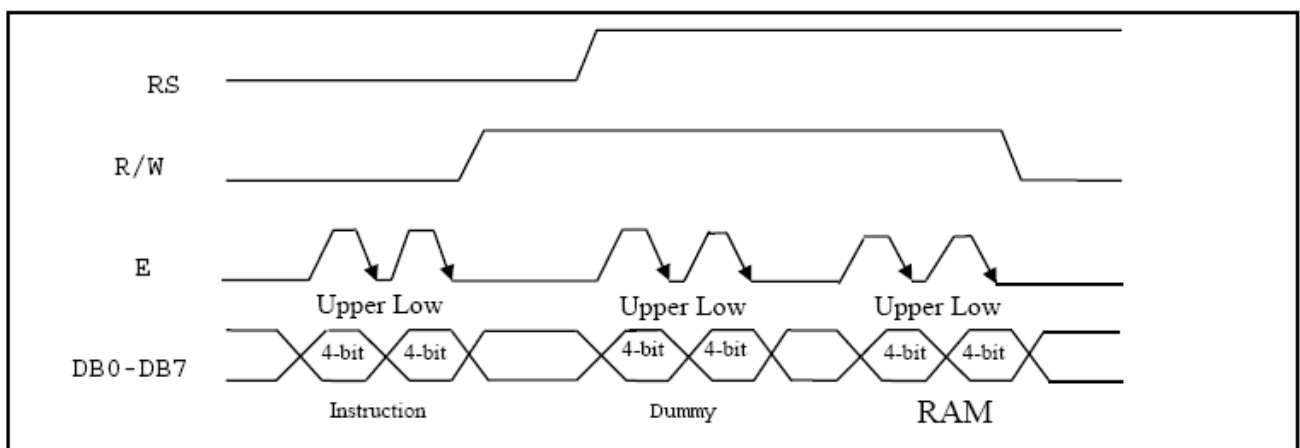
從一個完整的流程來看，當下設定位址指令後(CGRAM, DDRAM, IRAM.....)若要讀取資料時需先 DUMMY READ 一次才會讀取到正確資料第二次讀取時則不需 DUMMY READ 除非又下設定位址指令才需再次 DUMMY READ。

在4-位元傳輸模式中，每一個八位元的指令或資料都將被分為兩個位元組動作：較高4位元（DB7~DB4）的資料將會被放在第一個位元組的（DB7~DB4）部分，而較低4位元（DB3~DB0）的資料則會被放在第二個位元組的（DB7~DB4）部分，至於相關的另四位元則在4-位元傳輸模式中DB3~DB0介面未使用。

相關介面傳輸訊號請參考下圖說明：



Timing Diagram of 8-bit Parallel Bus Mode Data Transfer



Timing Diagram of 4-bit Parallel Bus Mode Data Transfer

在接收到同步位元及RW和RS資料的起始位元組後，每一個八位元的指令將被分為兩個位元組接收到：較高4位元（DB7~DB4）的指令資料將會被放在第一個位元組的LSB部分，而較低4位元（DB3~DB0）的指令資料則會被放在第二個位元組的LSB部分，至於相關的另四位元則都為0。

串列傳輸訊號請參考下圖說明：

串列介面與串列傳輸資料

當PSB腳接低電位時，ST7920將進入串列模式，在串列模式下將使用兩條資料傳輸線作串列資料的傳送，主控制系統將配合傳輸同步時脈線（SCLK）與接收串列資料線（SID），來達成串列傳輸的動作。

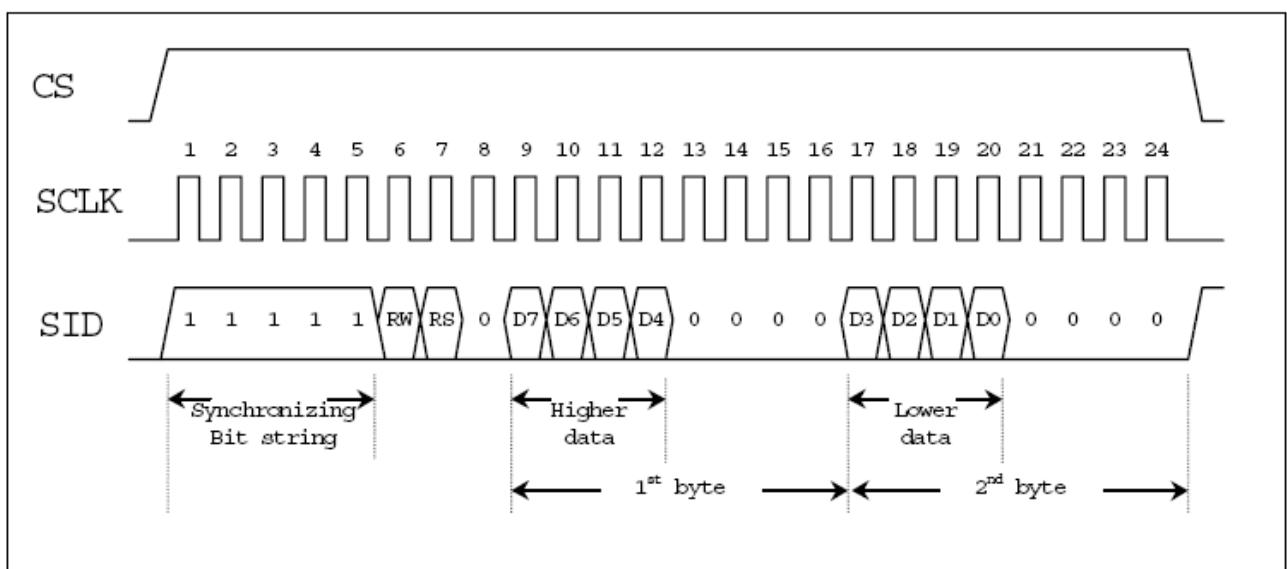
當需要同時連接數顆ST7920晶片時，晶片選擇腳（CS）將要被配合使用，在晶片選擇腳（CS）設為高電位時，同步時脈線（SCLK）輸入的訊號才會被接收，另一方面，當晶片選擇腳（CS）設為低電位時，ST7920的內部串列傳輸計數與串列資料將會被重置，也就是說在此狀態下，傳輸中的資料將被終止清除，並且將待傳輸的串列資料計數重設回第一位元；在一個最小的系統架構下，由一個微處理器連接控制單一個ST7920晶片時，相關的連接介面只需要使用同步時脈線（SCLK）與接收串列資料線（SID）兩隻腳，在這個模式下晶片選擇腳（CS）將被固定接到高電位。

ST7920的同步時脈線（SCLK）具有獨立的操作時脈，但是當有連續多個指令需要被傳送時，指令執行的時間將需要被考慮，必須確實等到前一個指令完全執行完成才能傳送下一筆資料，因為ST7920內部並沒有傳送/接收緩衝區。

從一個完整的串列傳輸流程來看，一開始先傳輸起始位元組，它需先接收到五個連續的“1”（同步位元字串）在起始位元組，此時傳輸計數將被重置並且串列傳輸將被同步，再跟隨的兩個位元字串分別指定傳輸方向位元（RW）及暫存器選擇位元（RS），最後第八的位元則為“0”。

在接收到同步位元及RW和RS資料的起始位元組後，每一個八位元的指令將被分為兩個位元組接收到：較高4位元（DB7~DB4）的指令資料將會被放在第一個位元組的LSB部分，而較低4位元（DB3~DB0）的指令資料則會被放在第二個位元組的LSB部分，至於相關的另四位元則都為0。

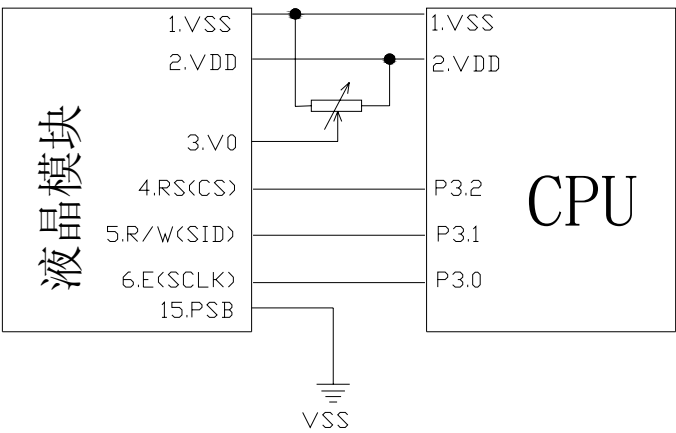
串列傳輸訊號請參考下圖說明：



Timing Diagram of Serial Mode Data Transfer

7.5 程序举例：

液晶模块与 MPU 串行接口图如下 (以 8051 系列单片机为例)：



串行程序案例（汇编语言）：
;写指令到指令寄存器的方法：

```

;-----
; Write data from A into INSTRUCTION Register
;-----
WRINS:
    SETB    CS
    SETB    SID    ; SID = 1
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    CLR     SID    ; SID = 0
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    MOVBIT  SID, A.7 ; SID = A.7
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    MOVBIT  SID, A.6 ; SID = A.6
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    MOVBIT  SID, A.5 ; SID = A.5
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    MOVBIT  SID, A.4 ; SID = A.4
    SETB    SCLK    ; READ DATA FROM SID
    CLR     SCLK
    CLR     SID    ; SID = 0
    SETB    SCLK    ; READ DATA FROM SID
```

```

CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
MOVBIT   SID, A.3   ; SID = A.3
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
MOVBIT   SID, A.2   ; SID = A.2
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
MOVBIT   SID, A.1   ; SID = A.1
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
MOVBIT   SID, A.0   ; SID = A.0
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
CLR      SID        ; SID = 0
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      SCLK
SETB     SCLK      ; READ DATA FROM SID
CLR      CS
CALL     DLY8
RET

```

;写数据到数据寄存器的方法:

```

;-----
;  Write data from A into DATA Register
;-----
WRDATA:
    SETB     CS
    SETB     SID      ; SID = 1
    CLR      SCLK
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SID      ; SID = 0
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    SETB     SID      ; SID = 1
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    CLR      SID      ; SID = 0
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    MOVBIT   SID, A.7   ; SID = A.7
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    MOVBIT   SID, A.6   ; SID = A.6
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    MOVBIT   SID, A.5   ; SID = A.5
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    MOVBIT   SID, A.4   ; SID = A.4
    SETB     SCLK      ; READ DATA FROM SID
    CLR      SCLK
    CLR      SID      ; SID = 0
    SETB     SCLK      ; READ DATA FROM SID

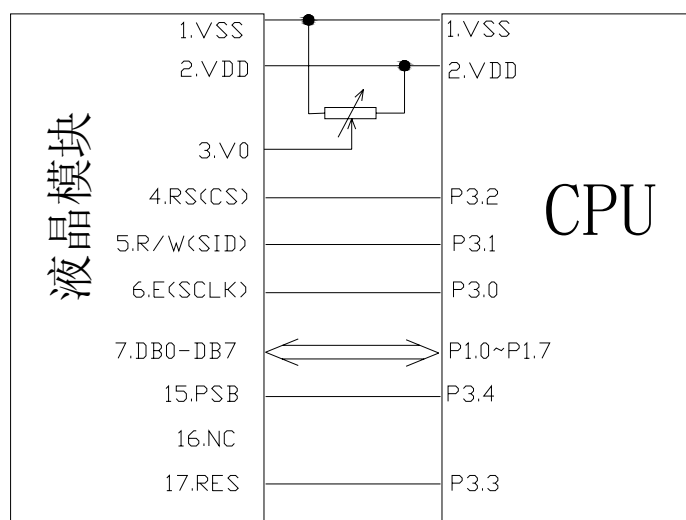
```

```

CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
MOVBIT SID, A.3 ; SID = A.3
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
MOVBIT SID, A.2 ; SID = A.2
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
MOVBIT SID, A.1 ; SID = A.1
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
MOVBIT SID, A.0 ; SID = A.0
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
CLR    SID      ; SID = 0
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
SETB   SCLK    ; READ DATA FROM SID
CLR    SCLK
CLR    CS
CALL   DLY8
RET

```

液晶模块与 MPU 并行接口图如下(以 8051 系列单片机为例)：



并程序案例：（C 语言）

/* Test program for JLX16032C-1

Driver IC is:ST7920(or competible)

Programmed by Ken, Feb.15,2011

JLX electronic Co.,ltd, <http://www.jlxlcd.cn>; <http://www.jlxlcd.com.cn>

本程序显示内容是在第一行第一位置显示中文字“深圳晶联讯”

*/

#include <reg51.h>

sbit r_w=P3^1;

sbit reset=P3^3;

sbit rs=P3^2;

```
sbit e_1=P3^0;
sbit psb=P3^4;
#define data_bus P1

void Transfer_data_ic1(int data1);
void Transfer_command_ic1(int data1);
void Delay(int i);
void Delay1(int i);
void display_char();
void Initial_ic1();
void Switch();

/*主程序*/
void main(void)
{
    int i, j, k;
    psb=1;      /*选择并口*/
    Initial_ic1(); /*初始化*/
    while(1)
    {
        display_char(); /*调用显示汉字子程序*/
        Switch(); /*等待按键 P2.0 口*/
    }
}

/*初始化*/
void Initial_ic1()
{
    Transfer_command_ic1(0x30); /*选用基本指令集*/
    Delay(10);
    Transfer_command_ic1(0x01); /*清屏*/
    Delay(10);
    Transfer_command_ic1(0x06); //
    Delay(10);
    Transfer_command_ic1(0x0c); /*开显示，关光标*/
    Delay(10);
}

/*=====显示汉字=====*/
void display_char()
{
    int i, j;
    Transfer_command_ic1(0x80);
    for(i=0; i<1; i++)
    {
        Transfer_data_ic1(0xc9);
```

```
Transfer_data_ic1(0xee);/* 以上两行数据合起来显示汉字“深”,*/
Transfer_data_ic1(0xdb);
Transfer_data_ic1(0xda);/* 以上两行数据合起来显示汉字“圳”,*/
Transfer_data_ic1(0xbe);
Transfer_data_ic1(0xa7);/* 以上两行数据合起来显示汉字“晶”,*/
Transfer_data_ic1(0xc1);
Transfer_data_ic1(0xaa);/* 以上两行数据合起来显示汉字“联”,*/
Transfer_data_ic1(0xd1);
Transfer_data_ic1(0xb6);/* 以上两行数据合起来显示汉字“讯”,*/
}
}
```

```
/*===检查忙标志位 BF=====*/
```

```
void check_busy1()
```

```
{
    rs=0;
    data_bus=0xff;
    if(P1^7==1)
    {
        r_w=1;
        e_1=1;
        Delay(5);
        e_1=0;
    }
    else;
}
```

```
/*===送指令到液晶模块驱动 IC=====*/
```

```
void Transfer_command_ic1(int data1)
```

```
{
    check_busy1();
    rs=0;
    r_w=0;
    data_bus=data1;
    e_1=1;
    Delay(5);
    e_1=0;
    Delay(5);
}
```

```
/*=====传送数据到液晶模块驱动 IC=====*/
```

```
void Transfer_data_ic1(int data1)
```

```
{
    check_busy1();
    rs=1;
    r_w=0;
    data_bus=data1;
}
```

```
e_1=1;
Delay(5);
e_1=0;
Delay(5);
}
/*=====延时=====*/
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++);
}

/*====等待按键：P2.0 与地相接时按键生效=====*/
void Switch()
{
    repeat:
        if (P2&0x01) goto repeat;
        else Delay(1);
        if (P2&0x01) goto repeat;
        else;
}
```