



凌 阳 大 学 计 划
Sunplus University Program

内藏 SPLC501 控制器图形液晶显示模组 使用手册

北京北阳电子技术有限公司



目录

摘要	4
一、基本描述	4
二、特点	4
三、尺寸图	5
四、电气参数	5
五、接口端管脚说明	6
六、驱动控制器介绍	6
6.1 SPLC501 特性	7
特点:	7
功能介绍	7
6.2 SPLC501 电气特性	9
6.3 SPLC501 读/写操作时序	11
8080 系列 MPU 操作时序	11
6800 系列 MPU 操作时序	12
串行接口时序	13
显示控制输出时序	14
复位时序	14
6.4 微控制器接口举例	15
8080 系列 MPU 连接	15
七、液晶显示模块指令系统	17
7.1 显示开关指令	17
7.2 显示起始行设置	17
7.3 页地址设置	17
7.4 设置列地址	18
7.5 读状态	18
7.6 写显示数据	18
7.7 读显示数据	19
7.8 ADC 选择 (Segment 方向选择)	19
7.9 正向/反向显示	19
7.10 全屏点亮/变暗	19
7.11 LCD 偏压设置	20
7.12 读/改/写	20
7.13 结束	21
7.14 复位	21
7.15 正常输出模式选择	21
7.16 上电控制设置	22
7.17 V5 电压内部电阻调整设置	22



7.18 电量 (electronic Volume) 设置模式	22
7.19 静态指示器	23
7.20 页闪动	23
7.21 驱动模式设置	24
7.22 节电模式	24
7.23 空命令	25
八、液晶模块驱动程式介绍	25
九、与 61 板结合使用	32
十、 DEMO 板原理图	37

Sunplus



摘要

液晶目前在人们的生活中，应用越来越普遍，如我们较熟悉的手机的显示界面，电脑笔记本的显示器，并且现在刚刚推出液晶电视等，液晶屏的优点是显示界面清晰度高且功耗低。在此为大家介绍的是凌阳公司的一款 128*64 点阵的液晶模组，驱动芯片采用的是凌阳 SPLC501。该液晶模组接口简单，应用方便，且可以完成很多液晶特效功能。

一、基本描述

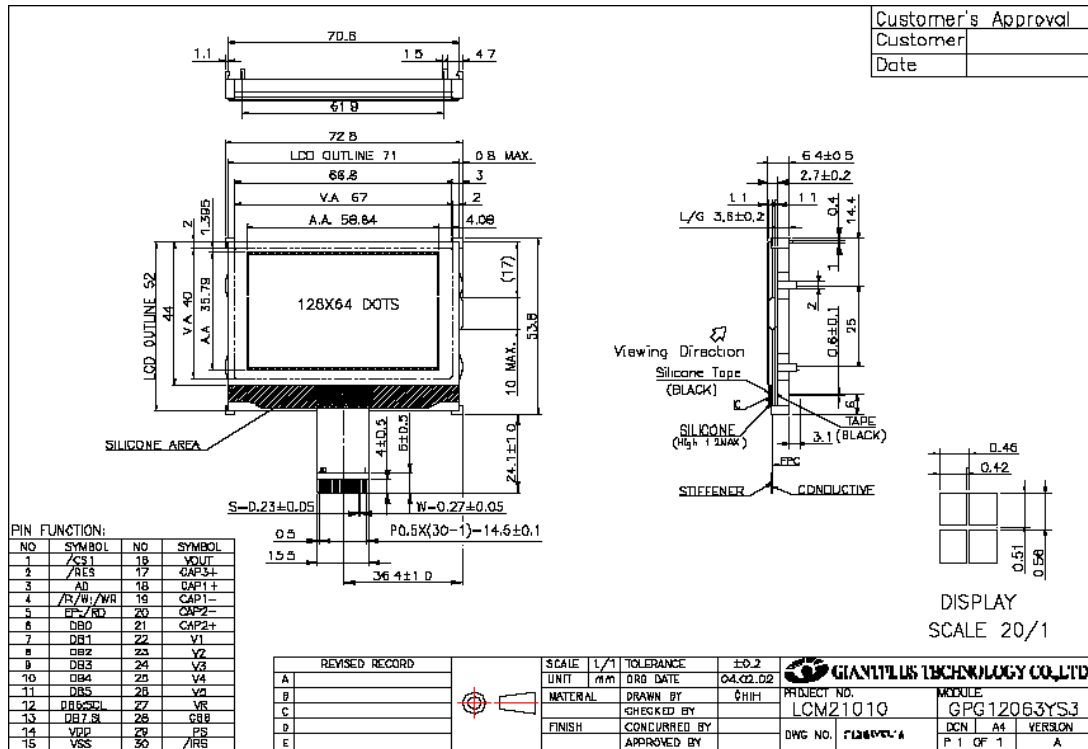
此液晶模组为 128*64 点阵，面板采用 STN（Super Twisted Nematic）超扭曲向列技术制成并且由 128 Segment 和 64 Common 组成，LCM 非常容易通过接口被访问。

二、特点

显示模式	黄色模式 STN 液晶
显示格式	128*64 点阵地图形液晶显示
输入数据	兼容 68/80 系列 MPU 数据输入
背光	黄绿色 LED
模块尺寸	72.8（长）×73.6（宽）×9.5（高）mm
视屏尺寸	58.84（宽）×35.79（长）mm
点大小	0.42（宽）×0.51（长）mm
像素尺寸	0.46（宽）×0.56（长）mm



三、尺寸图



四、电气参数

参数	符号	条件	最小	典型	最大	单位
工作电压	Vdd	—	3.3		5	V
输入电压	High Level	Vih	0.7Vdd		Vdd	V
	Low Level	Vil	Vss		0.3Vdd	V
工作耗电流	Idd	Vdd=3.3V Vlcd=9V Fsc1 = 0 Tamb = 25 °		1.5	2.0	mA
LCD 驱动电压	Vlcd	Bias=1/9	8.7	9.0	9.3	V
LED 驱动电压	Vled			2.1		V
LED 耗电流	Iled	Vled= 2.1V			84	mA

内藏 SPLC501 控制器图形液晶显示模组



五、接口端管脚说明

管脚名	说明
/CS1	片选，低有效
/RES	复位脚
A0	数据命令选择脚
R/W/WR	对于 6800 系列 MPU 的读/写信号
	对于 8080 系列 MPU 的写信号
EP/RD	对于 6800 系列 MPU 的时钟信号使能脚
	对于 8080 系列 MPU 的读信号
DB0	8 位数据总线
DB1	
DB2	
DB3	
DB4	
DB5	
DB6	
DB7	
VR	端口输出电压
C86	C86='H' 选择 6800MPU 系列 C86='L' 选择 8080MPU 系列
Vdd	逻辑电源 (3.3V~5V)
Vss	地 (0V)

六、驱动控制器介绍

此模组采用的驱动控制器是台湾凌阳科技的 SPLC501。SPLC501 液晶显示控制驱动器集行、列驱动器和控制器于一体，广泛应用于小规模液晶显示模块。

内藏 SPLC501 控制器图形液晶显示模组



6.1 SPLC501 特性

SPLC501 单芯片液晶驱动，可以直接与其他微控制器接口总线相连。微控制器可以将显示数据通过 8 位数据总线或者串行接口写到 SPLC501 的显存中。

特点：

- 1、内置 8580 位显示 RAM。RAM 中的一位数据控制液晶屏上的一个像素点的亮、暗状态。“1”亮“0”暗。
- 2、具有 65 行驱动输出和 132 列驱动输出
- 3、可以直接与 80 系列和 68 系列微处理器相连。
- 4、内置晶振电路，也可以外接晶振
- 5、工作温度范围为 -40 摄氏度 ~ +85 摄氏度

功能介绍

SPLC501 功能原理图和管脚图分别见图 1 和图 2。

Supplus

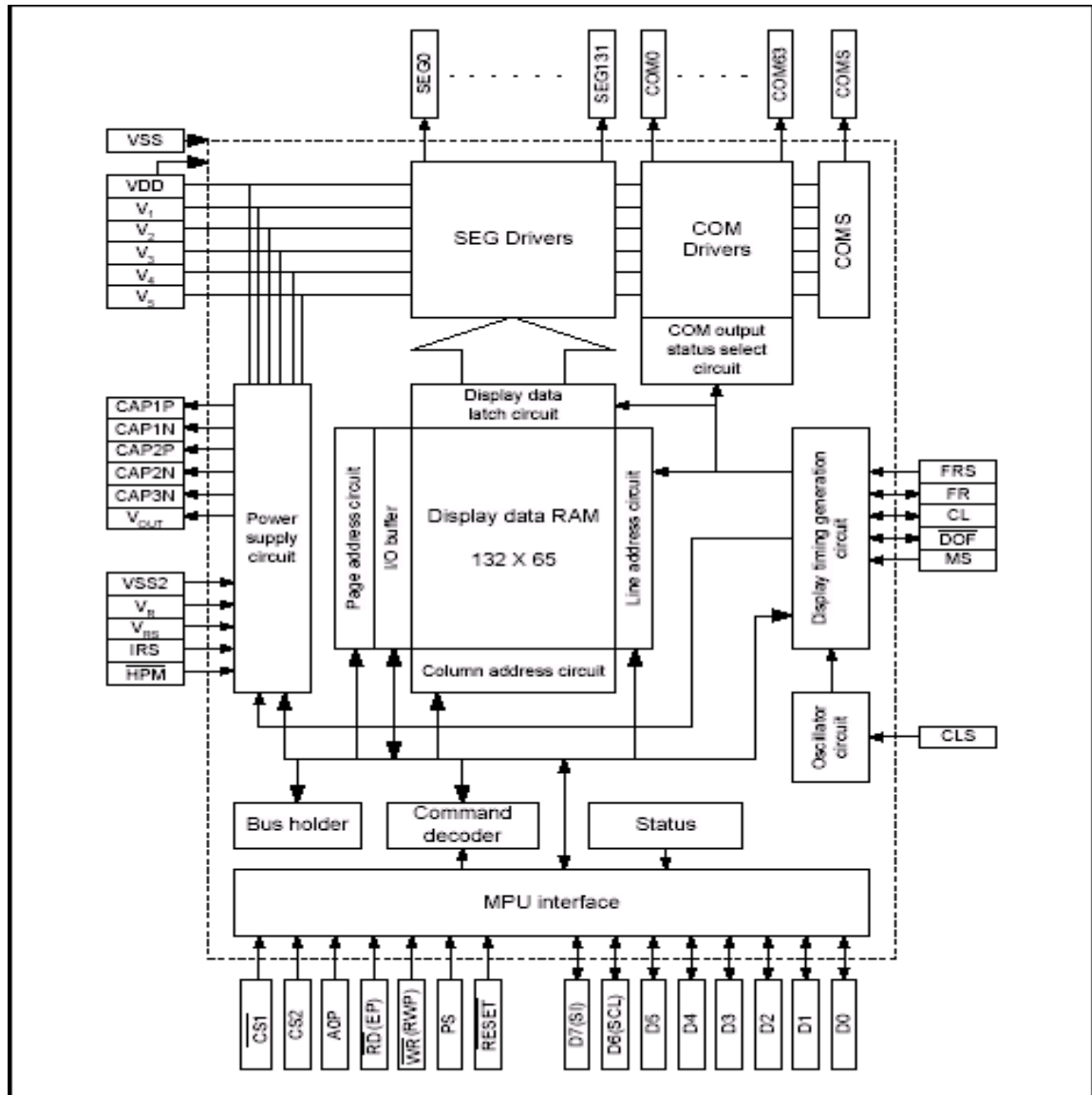


图 1 SPLC501 原理框图

内藏 SPLC501 控制器图形液晶显示模组

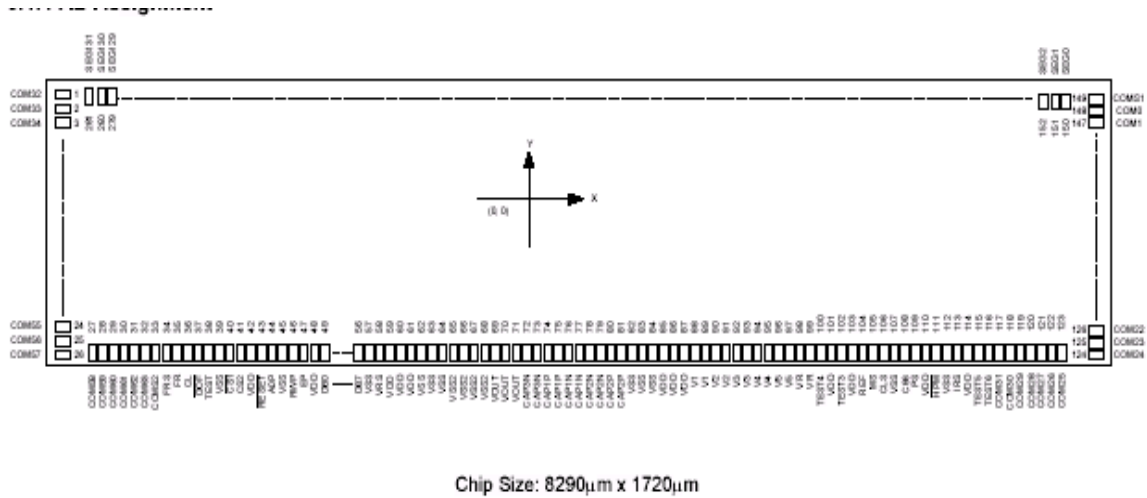


图 2 SPLC501 管脚图

6.2 SPLC501 电气特性

绝对最大范围

参数	符号	范围	单位
电源电压 1	Vdd	-0.3 ~ 7	V
电源电压 2	Vss	-7 ~ +0.3 -4 ~ +0.3 +3 ~ +0.3	V
电源电压 3	V5 ,Vout	-12 ~ +0.3	V
电源电压 4	V1,v2,V3,V4	V5 ~ +0.3	V
输入电压	Vin	-0.3 ~ Vdd+0.3	V
输出电压	Vo	-0.3 ~ Vdd+0.3	
工作温度	Topr	40 ~ +80	摄氏度
储存温度	Tstr	-55 ~ +125	摄氏度

电特性

参数	符号	条件	最小	典型	最大	单位	应用脚
----	----	----	----	----	----	----	-----

内藏 SPLC501 控制器图形液晶显示模组

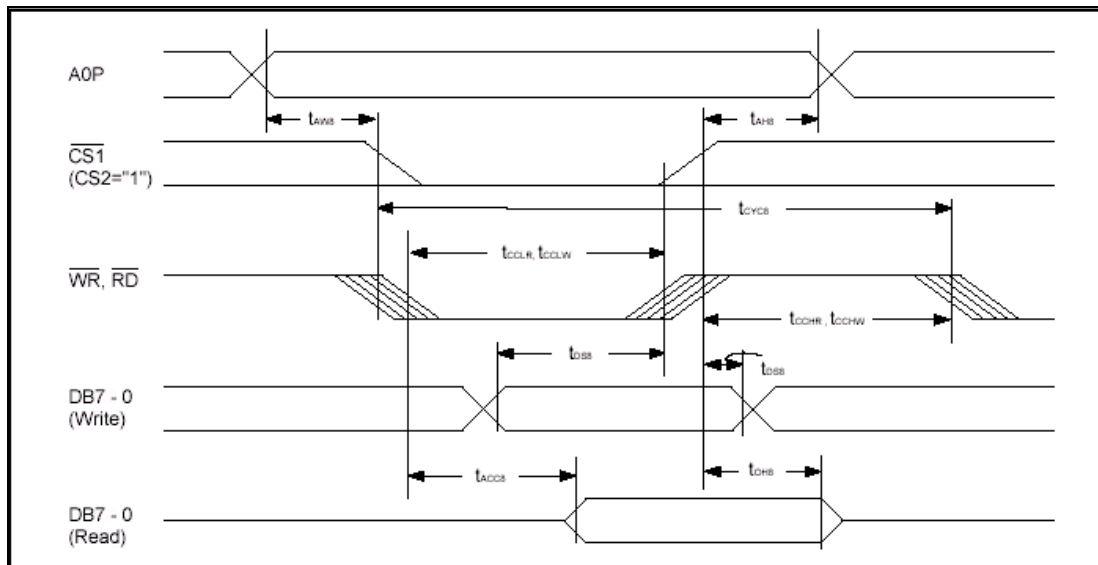


电源电压 1	参考电压	Vdd		2.7		3.3	V	Vdd
	工作电压			2.4		5.5		Vdd
电源电压 2	参考电压	Vss2		-3.3		-2.7	V	Vss2
	工作电压	Vss2		-6.0		-1.8		Vss2
电源电压 3	参考电压	V5		-12		-4.5	V	V5
	工作电压	V1, V2		4×V5		Vdd		V1, V2
	工作电压	V3, V4		V5		0.6×V5		V3, V4
高电压输入	Vi _{hc}			0.8×Vdd		Vdd	V	
低电压输入	Vi _{lc}			Vss		0.2×Vdd		
高电压输入	Vch _c	I _{ch} =0.5mA		0.8×Vdd		Vdd	V	
低电压输入	Vcl _c	I _{cl} =0.5mA		Vss		0.2×Vdd		
输入漏电流	I _{li}	V _{in} =Vdd		-1.0		1.0	uA	
输出漏电流	I _{lo}	Or Vss		-3.0		3.0		
无驱动耗电	I _{ssq}			0.01		5	uA	
晶体振荡频率	F _{osc}			18	22	26	kHz	



6.3 SPLC501 读/写操作时序

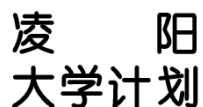
8080 系列 MPU 操作时序



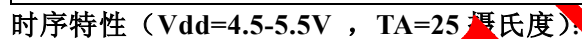
时序特性 ($V_{dd}=4.5-5.5V$, $T_A=25$ 摄氏度):

参数	管脚	符号	条件	最小值	最大值	单位
地址保持时间	AOP	tAHS		0		ns
地址建立时间	AOP	tAHD		0		ns
系统周期时间	AOP	tCYC8		166		ns
控制低脉冲宽度 (WR)	WR	tccLW		30		ns
控制低脉冲宽度 (RD)	RD	tccLR		70		ns
控制高脉冲宽度 (WR)	WR	tccHW		30		ns
控制高脉冲宽度 (RD)	RD	tccHR		30		ns

内藏 SPLC501 控制器图形液晶显示模组



6800 系列 MPU 操作时序

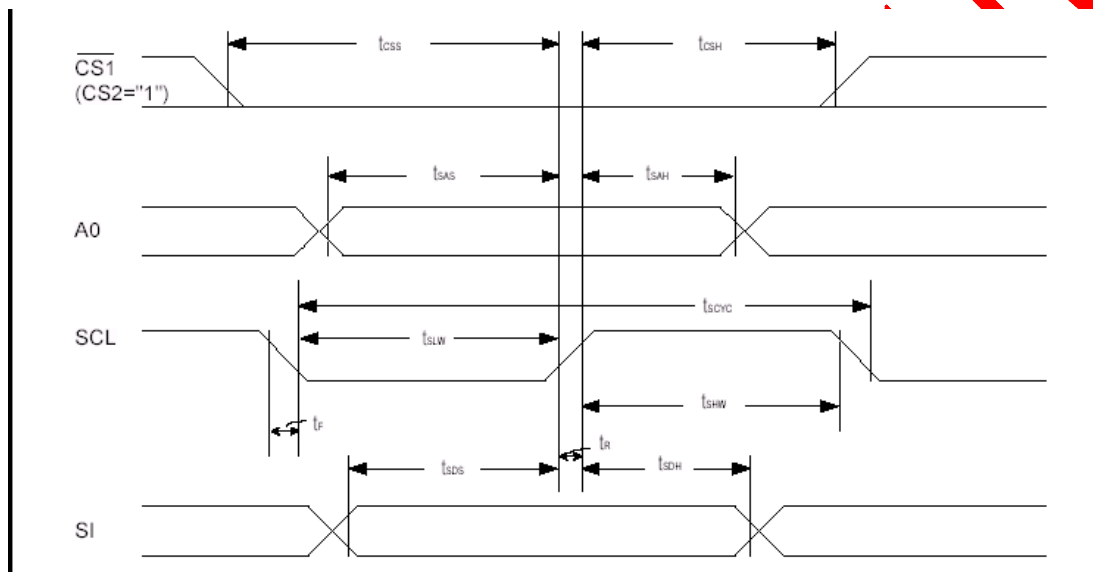


内藏 SPLC501 控制器图形液晶显示模组



系统周期 时间	AOP	tCYC8		166		ns
使能低脉 冲 宽 度 (WR)	WR	tewlw		30		ns
使能低脉 冲 宽 度 (RD)	RD	tewlr		30		ns
使能高脉 冲 宽 度 (WR)	WR	tewhw		30		ns
使能高脉 冲 宽 度 (RD)	RD	tewhr		70		ns

串行接口时序



时序特性 (V_{dd}=4.5-5.5V , TA=25 摄氏度):

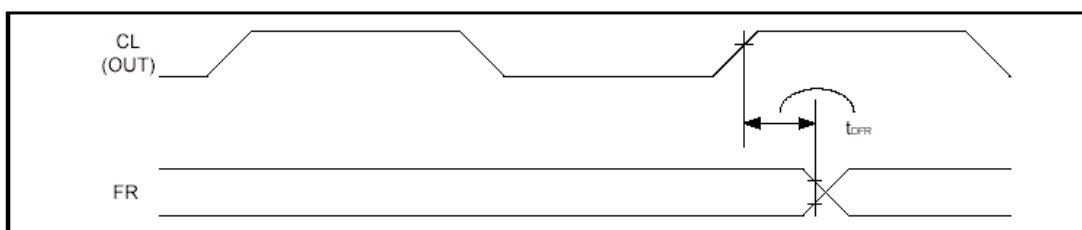
参数	管脚	符号	条件	最小值	最大值	单位
串行时钟 周期	SCL	tSCYC		200		ns
SCL 高脉 冲宽度	SCL	tSHW		75		ns
SCL 低脉 冲宽度	SCL	tSLW		75		ns
地址建立 时间	AOP	tSAS		50		ns

内藏 SPLC501 控制器图形液晶显示模组



地址保持时间	AOP	tSAH		100		ns
数据建立时间	SI	tSDS		50		ns
数据保持时间	SI	tSDH		50		ns
CS-SCL时间	CS	tCSS tCSH		100 100		ns

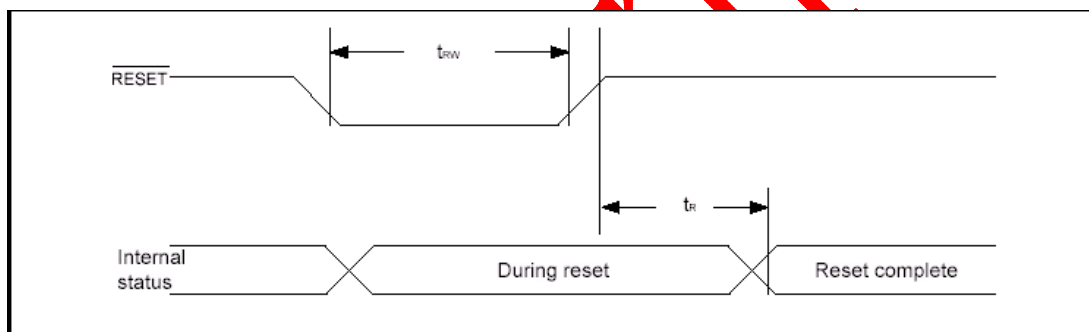
显示控制输出时序



时序特性 (V_{dd}=4.5-5.5V , TA=25 摄氏度) :

参数	管脚	符号	条件	典型	最大值	单位
FR 延迟时间	FR	tDFR	Cl=50pf	10	40	ns

复位时序



时序特性 (V_{dd}=4.5-5.5V , TA=25 摄氏度) :

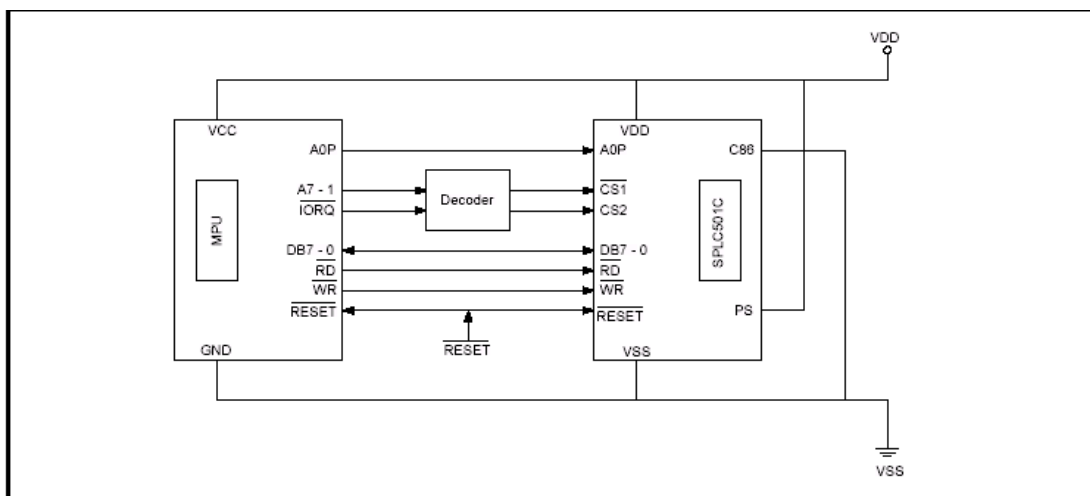
参数	管脚	符号	条件	最小	最大值	单位
复位时间		tR			0.5	us
复位低脉冲宽度	RES	tRW		0.5		us



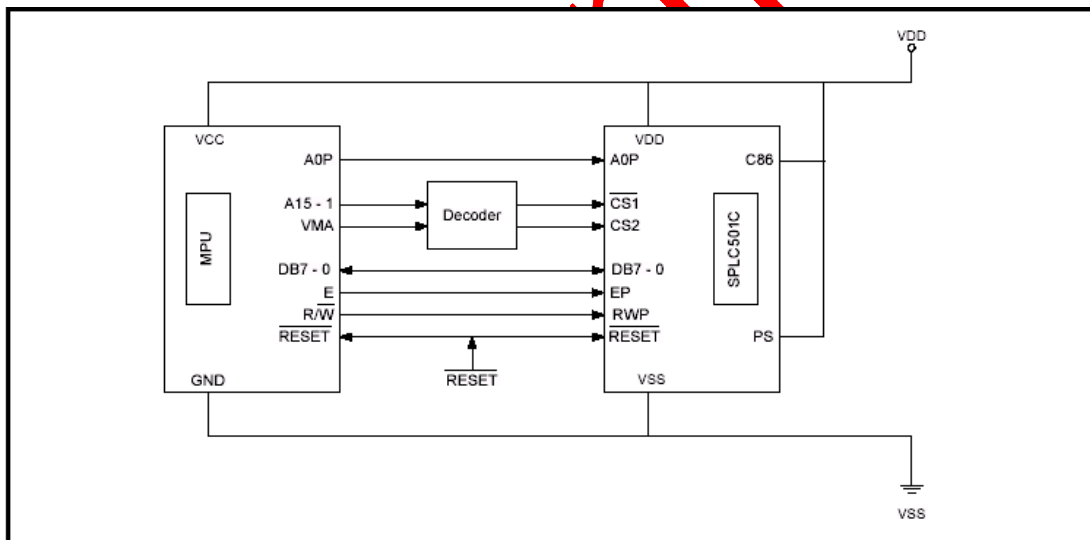
6.4 微控制器接口举例

SPLC501 可以连接 80×86 系列 MPU 和 68000 系列 MPU，如果采用串行连接方式可以使用更少的管脚完成。另外，通过使用更多的 SPLC501 可以加大液晶显示面积，这时就可以利用片选脚决定访问那颗 SPLC501。

8080 系列 MPU 连接

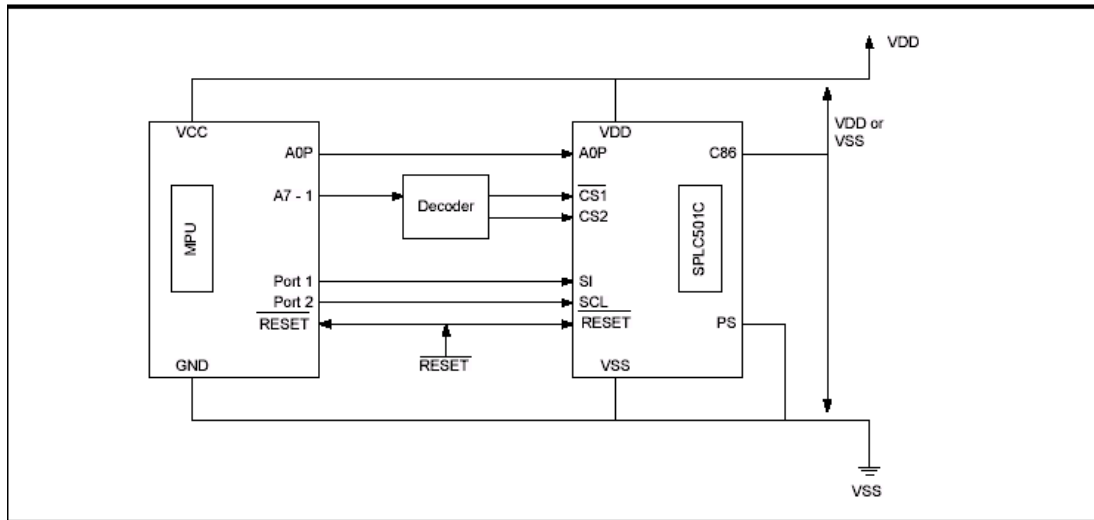


6800 系列 MPU 连接

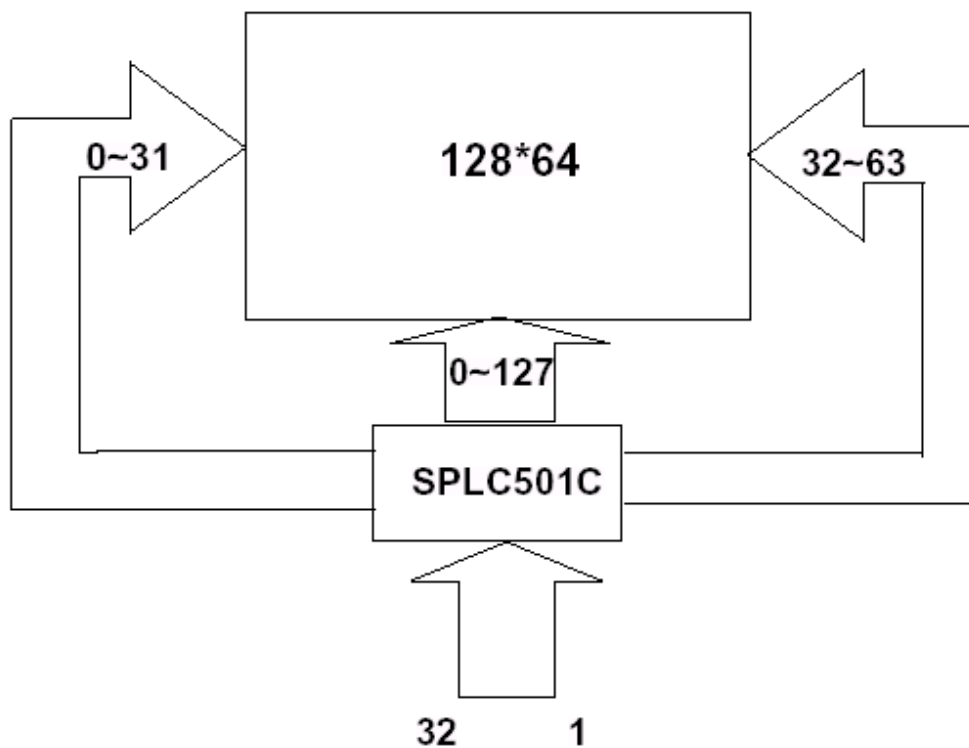


利用串行接口连接

内藏 SPLC501 控制器图形液晶显示模组



LCM 结构图



内藏 SPLC501 控制器图形液晶显示模组



七、液晶显示模块指令系统

该类液晶显示模块共有 23 种显示指令，下面以与 68 系列 MPU 接口为例，分别介绍以下 23 种指令

7.1 显示开关指令

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Setting
0	1	0	1	0	1	0	1	1	1	1	Display ON
										0	Display OFF

当显示被打开或关闭时，系统进入节电模式。

7.2 显示起始行设置

这个指令设置了对应显示屏上首行的显示 RAM 行号。有规律的修改该行号，可以实现滚屏功能。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Line Address
0	1	0	0	1	0	0	0	0	0	0	0
					0	0	0	0	0	1	1
					0	0	0	0	1	0	2
											↓
					1	1	1	1	1	0	62
					1	1	1	1	1	1	63

7.3 页地址设置

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Page Address
0	1	0	1	0	1	1	0	0	0	0	0
							0	0	0	1	1
							0	0	1	0	2
											↓
							0	1	1	1	7
							1	0	0	0	8



7.4 设置列地址

由上图可以看出显示 RAM 被分成 9 页每页 132 个字节，当设置了页地址和列地址后，就确定了显示 RAM 中的唯一单元，该单元由低到高各个数据位对应于显示屏上的某一列的 8 行数据位。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	A7	A6	A5	A4	A3	A2	A1	A0	Column Address
0	1	0	0	0	0	1	A7	A6	A5	A4	0	0	0	0	0	0	0	0	0
						0	A3	A2	A1	A0	0	0	0	0	0	0	0	1	1
											0	0	0	0	0	0	1	0	2
																			↓
											1	0	0	0	0	0	0	0	130
											1	0	0	0	0	0	1	1	131

7.5 读状态

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	BUSY	ADC	ON/OFF	RESET	0	0	0	0

BUSY	当 BUSY 为 1 时，忙状态；当 BUSY 为 0 时，准备好状态，
ADC	表示行和列的关系 ADC:1 正常输出 ($n-131==SEGn$)，ADC:0 为反向输出 ($131-n==SEG n$)
ON/OFF	表示液晶显示开和关 0: 显示打开，1: 显示关闭
RESET	0: 正常工作状态，1: 复位

7.6 写显示数据

这条指令可以将显示数据（8 位）写到 RAM 中，显示地址自动加一。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	0	Write data							



7.7 读显示数据

这条指令从指定地址中读取显示数据，读取显示数据后，列地址自动加一。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	1	Read Data							

7.8 ADC 选择（Segment 方向选择）

这条命令用于将 Segment 驱动输出反向。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Setting
0	1	0	1	0	1	0	0	0	0	0	Normal
										1	Reverse

7.9 正向/反向显示

这条命令用于设置显示正向和反向。当执行该指令后，显示 RAM 中的内容不变。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Setting
0	1	0	1	0	1	0	0	1	1	0	RAM Data 'H' LCD ON voltage (normal)
										1	RAM Data 'L' LCD ON voltage (reverse)

7.10 全屏点亮/变暗

这条命令使所有的液晶点被点亮/变暗，无论显示 RAM 中有任何数据。此命令优先于正向/反向显示。当液晶处于显示关闭状态时，执行此命令将会自动进入节电状态。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Setting
0	1	0	1	0	1	0	0	1	0	0	Normal display mode
										1	Display all points ON



7.11 LCD 偏压设置

这条命令用于液晶显示的偏压设置。

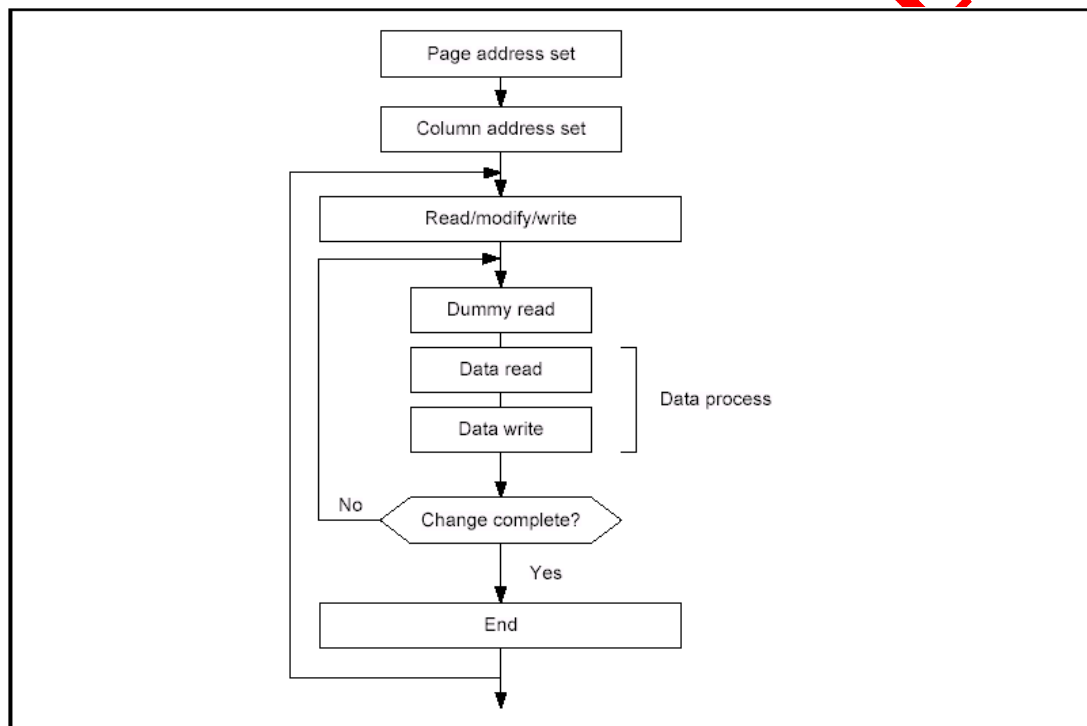
A0P	EP	RWP									Select Status
	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	SPLC501C
0	1	0	1	0	1	0	0	0	1	0	1/9 bias
										1	1/7 bias

7.12 读/改/写

这条指令用到两次结束命令，一旦写入此命令后，读显示数据命令不再修改列地址，但是写显示数据命令还可以使列地址自动加一。当有结束命令输入时，列地址恢复到读/改/写时的列地址。这个命令可用于光标显示。

A0P	EP	RWP								
	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	1	1	0	0	0	0	0

光标显示流程图：

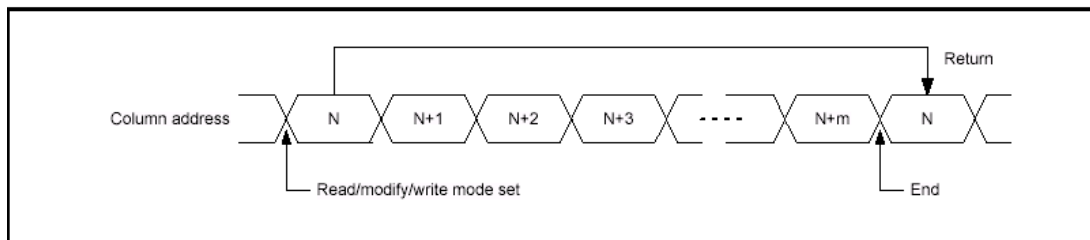




7.13 结束

这条指令用于结束读/改/写模式。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	1	1	0	1	1	1	0



7.14 复位

这条指令初始化显示起始行、起始列地址、起始页地址、正常输出模式。结束读/改/写模式和测试模式。此命令不影响显示 RAM 中的数据。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	1	1	0	0	0	1	0

7.15 正常输出模式选择

这条指令用于确定 COM 口扫描的方向。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Select Status SPLC501C	
0	1	0	1	1	0	0	0	*	*	*	Normal	COM0 --> COM63
							1				Reverse	COM63 --> COM0



7.16 上电控制设置

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Selected Mode
0	1	0	0	0	1	0	1	0			Booster circuit: OFF Booster circuit: ON
									0		Voltage regulator circuit :OFF Voltage regulator circuit: ON
									1		
										0	Voltage follower circuit: OFF Voltage follower circuit: ON
										1	

7.17 V5 电压内部电阻调整设置

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Setting
0	1	0	0	0	1	0	0	0	0	0	Small
								0	0	1	
								0	1	0	
									↓		
								1	1	0	
								1	1	1	Large

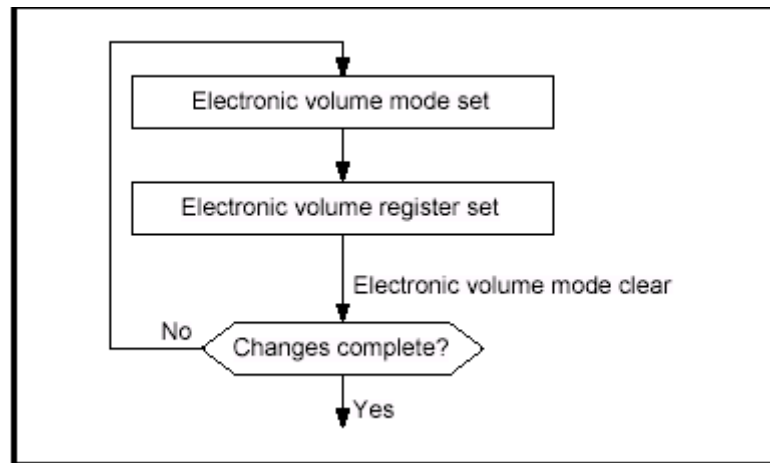
7.18 电量（electronic Volume）设置模式

这条命令用于调整显示屏的亮度。此命令用到双字节：一个是设置为电量设置模式，另一个是设置电量寄存器设置模式。

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	0	0	0	0	0	0	1

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Vs
0	1	0	*	*	0	0	0	0	0	1	Small
0	1	0	*	*	0	0	0	0	1	0	
0	1	0	*	*	0	0	0	0	1	1	
							↓				
0	1	0	*	*	1	1	1	1	1	0	
0	1	0	*	*	1	1	1	1	1	1	Large

流程如下：



7.19 静态指示器

这条命令用于控制静态驱动指示器显示。为双字节命令。

静态指示器开/关

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Static Indicator
0	1	0	1	0	1	0	1	1	0	0	OFF
										1	ON

静态指示器寄存器设置状态

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Static Indicator
0	1	0	*	*	*	*	*	0	0		OFF
			*	*	*	*	*	0	1		ON (blinking at approximately 0.5 second intervals)
			*	*	*	*	*	1	0		ON (blinking at approximately one second intervals)
			*	*	*	*	*	1	1		ON (constantly on)

7.20 页闪动

页闪动模式设置

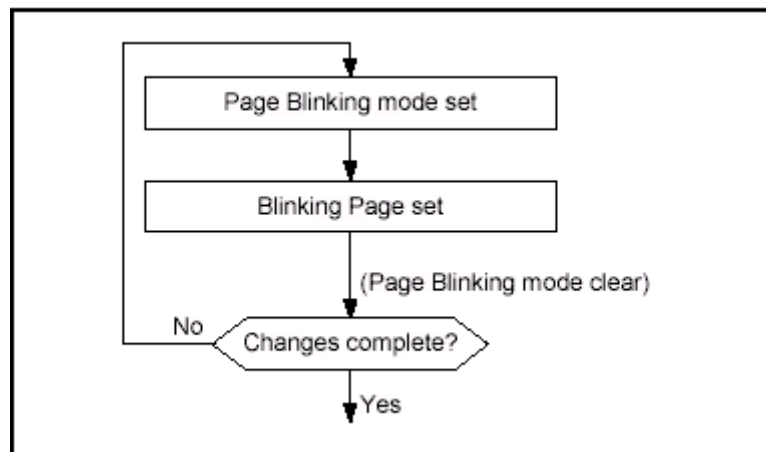
A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	0	1	0	0	1	0	1

页闪动寄存器设置



A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Blinking Page
0	1	0	1	0	0	0	0	0	0	0	PAGE 7 blink
			0	1	0	0	0	0	0	0	PAGE 6 blink
			0	0	1	0	0	0	0	0	PAGE 5 blink
						↓					
			0	0	0	0	0	0	0	1	PAGE 0 blink

设置流程图：



7.21 驱动模式设置

驱动模式设置

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	1	0	1	0	0	1	0

模式选择寄存器设置

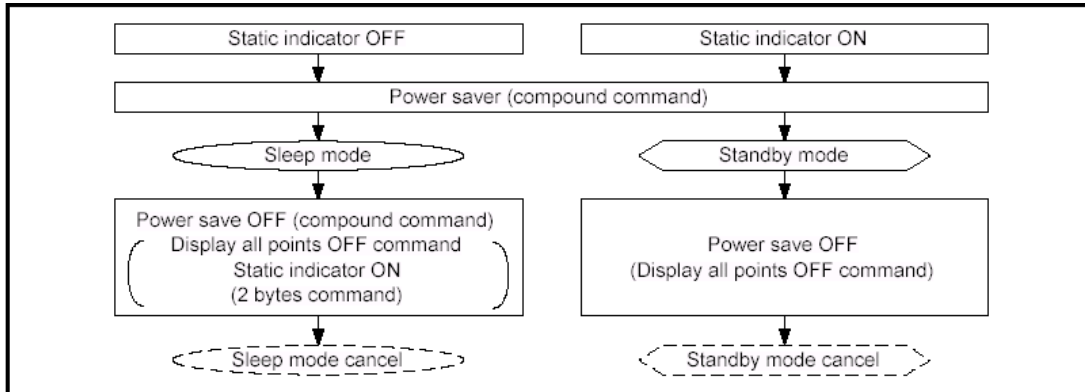
A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Driving Duty Selection
0	1	0	1	1	0	0	0	0	0	0	Mode 1
			0	0	0	0	0	0	0	0	Mode 2
			0	1	0	0	0	0	0	0	Mode 3
			1	0	0	0	0	0	0	0	Mode 4

7.22 节电模式

当在显示关闭时，设置全屏点亮，则进入节电状态。节电模式有两种状态一个是睡眠



模式另一个是备用模式。当静态指示器关闭时，进入睡眠模式。当静态指示器打开时，进入备用模式。在睡眠模式和备用模式时，显示数据保存操作模式时的数据。在这种模式时，MPU 可以访问显示 RAM。



睡眠模式：

在此模式下，除了 MPU 访问显示 RAM 外，停止所有的液晶显示操作。晶振、液晶上电和液晶驱动电路全部暂停。

备用模式：

在此模式下，液晶上电和液晶驱动电路暂停，晶振继续振荡。在备用模式下，有复位命令时，系统进入睡眠模式。

7.23 空命令

A0P	EP RD	RWP WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	0	1	1	1	0	0	0	1	1

八、液晶模块驱动程序介绍

硬件连接图，见九部分的硬件连接图。

1、液晶模组初始化

```

//=====
// Function Name: F_InitialLCD
// Description:      Initial LCD for display.
// Input:           None
// Output:          None
// Destroy:         r1,r2;
  
```



```
// Used:          r1,r2;
// Stacks:        2;
//=====
F_InitialLCD:.PROC
    M_LCD_Reset_Command;          //Reset LCD
    M_LCD_Command_Nop;            //Nop
    M_LCD_Solomon_Freq;           //Set Scan Frequency
    call F_LCD_Power_Set;         //Set Power
    M_LCD_Display_On;             //Display on
.IFDEF REVERSEDISPLAY
    M_LCD_Common_Reverse;
    M_LCD_ADC_Reverse;
.ENDIF
    r1=0;
    M_Set_LCD_Page_No;            //Page No. = 0
    r1=0;
    M_Set_LCD_Start_Line;         //Start line = 0
    R1 = 0x0000;                  //Column address = 0
    CALL    F_Set_LCD_Column_Addr;
    M_LCD_ALL_On;                 //
    M_LCD_Normal_Display;         //Normal
    RETF;
.ENDP
```

2、液晶上电设置程序

```
//=====
// Function Name:    F_LCD_Power_Set
// Description:       Set power.
// Input:            None
// Output:           None
// Destroy:          r1;
// Used:             r1;
// Stacks:           2;
//=====
F_LCD_Power_Set: .PROC

    R1=0xA2;                // 1/9 bias
    CALL F_LCD_Command_Send;
    R1=0x25;                // Contrast
                                // V5 RATIO 5 FOR 1/9 BIAS
    CALL F_LCD_Command_Send;
    R1=0x81;                // ELECTRONIC VOLUME MODE
```



```
CALL F_LCD_Command_Send;
R1=0x0A;           // MS Column : A
[LCDLighting] = r1;
CALL F_LCD_Command_Send;
R1=0xD2;           // Set driving mode register
CALL F_LCD_Command_Send;
R1=0x00;           // LS Column : 0
CALL F_LCD_Command_Send;
R1=0x2C;           // REGULATOR on
CALL F_LCD_Command_Send;
R1=0x2E;           // +FOLLOWER oFF
CALL F_LCD_Command_Send;
R1=0x2F;           // +FOLLOWER oFF
CALL F_LCD_Command_Send;

RETF;
.ENDP
```

3、清屏程序

```
//=====
// Function Name: F_Clr_LCD
// Description:      Clear all things on LCD.
// Input:           None
// Output:          None
// Destroy:         r1,r4;
// Used:            r1,r4;
// Stacks:          4;
//=====

F_Clr_LCD: .PROC
    r1=0;
L_Fill_Page_Loop:
    push r1 to [sp];
    M_Set_LCD_Page_No;           //Page No. = 0
    R1 = 0x0000;                 //Column address (00H ~ 63H)
    CALL F_Set_LCD_Column_Addr; //Column Addr. = 0
    r4 = 0x0084;                 //Max. column address +1
L_Fill_Column_Loop:
    r1 = 0x00;                   //Write data
    CALL F_LCD_Data_Send;
    R4 -= 0x0001;                //Next column address
    JNZ L_Fill_Column_Loop;      //End Page ?
```



```
pop r1 from [sp];
R1 += 0x0001;
CMP R1,0x0008;           //Next page address
JNE L_Fill_Page_Loop;
```

```
RETF
.ENDP
```

4、列地址设置

```
//=====
// Function Name: F_Set_LCD_Column_Addr
// Description:      Set Column.
// Input:           r1
// Output:          None
// Destroy:         r1,r4;
// Used:            r1,r4;
// Stacks:          2;
//=====
F_Set_LCD_Column_Addr: .PROC
.IFDEF REVERSEDISPLAY
    r1 += 4;
.ENDIF

push r1 to [sp];           //Store temporarily
R1 = R1 LSR 4;
R1 &= 0x000F;             //Isolate MS 4-bits
R1 |= 0x10;               //Set column address : Most significant 4 bits
CALL F_LCD_Command_Send;
pop r1 from [sp];
R1 =r1 & 0x000F;          //Isolate LS 4-bits
R1 |= 0x00;               //Set column address : Least significant 4 bits
CALL F_LCD_Command_Send;

RETF;
.ENDP
```

5、复位命令设置

```
M_LCD_Reset_Command: .MACRO
    r1=0xE2;               //Reset
    CALL F_LCD_Command_Send;
.ENDM
```

6、空命令设置

```
M_LCD_Command_Nop: .MACRO
    R1=0xE3;               //Nop
```



```
CALL F_LCD_Command_Send;  
.ENDM
```

7、COM 和 SEGMENT 设置

M_LCD_Solomon_Freq: .MACRO

```
//      R1=0x0AC0;                //COMs output : Normal  
R1=0xC8;                //COMs output : Inverse  
CALL F_LCD_Command_Send;
```

```
//      R1=0x0AA1;                //SEGs output : Normal  
R1=0xA0;                //SEGs output : Inverse  
CALL F_LCD_Command_Send;  
.ENDM
```

8、显示打开设置

M_LCD_Display_On: .MACRO

```
R1=0xAF;  
CALL F_LCD_Command_Send;  
.ENDM
```

9、全屏显示设置

M_LCD_ALL_On: .MACRO

```
R1=0xA5;                //All points on  
CALL F_LCD_Command_Send;  
.ENDM
```

10、正向设置

M_LCD_Normal_Display: .MACRO

```
R1=0xA4;                //Normal display  
CALL F_LCD_Command_Send;  
.ENDM
```

11、页地址设置

M_Set_LCD_Page_No:.MACRO

```
R1 |= 0xB0;             //Set Page No. (0~7)  
CALL F_LCD_Command_Send;  
.ENDM
```

12、显示起始行设置

M_Set_LCD_Start_Line: .MACRO

```
R1 |= 0x40;             //Set start line (00D ~ 63D)  
CALL F_LCD_Command_Send;
```



.ENDM

13、Com 反向设置

```
M_LCD_Common_Reverse: .MACRO
    R1 |= 0xC0;          //common reverse display
    CALL F_LCD_Command_Send;
.ENDM
```

14、ADC 反向设置

```
M_LCD_ADC_Reverse: .MACRO
    R1 |= 0xA1;          //Seg reverse display
    CALL F_LCD_Command_Send;
.ENDM
```

15、显示模式设置

```
M_LCD_DisplayMode_Reverse: .MACRO
    R1 |= 0xA7;          //
    CALL F_LCD_Command_Send;
.ENDM
```

16、写命令

```
//=====
// Function Name: F_LCD_Command_Send
// Description:      Send command data through simulate port.
// Input:           None
// Output:          None
// Destroy:         r1,r2;
// Used:            r1,r2;
// Stacks:          0;
//=====
```

```
.PUBLIC F_LCD_Command_Send;
.PUBLIC _F_LCD_Command_Send;
F_LCD_Command_Send:                                     //117cycle
_F_LCD_Command_Send: .PROC
.IFDEF Use68000Port
    //68000 interface
```

```
    r2=B_USBOperation+B_EPPin;
```

Read and USB operation

//6

```
    r1=[P_PortControl_Buf]; //9
```

```
    r1&=0xFFFF-B_AOP_Data-B_RWPPin-B_LCDOperation; //6
```

```
    r2|=r1; //3
```



```
[P_PortControl_Data]=r2;//9           //change control pin
r2^=B_EPPin;                          //6
[P_PortControl_Data]=r2;//9           //latch the data
retf;                                  //12

//8080 interface
//60
r2=B_USBOperation+B_RDPin;            //Disable Read and USB operation
//6

r1=[P_PortControl_Buf];               //9
r1&=0xFFFF-B_AOP_Data-B_WRPin-B_LCDOperation; //6
r2|=r1;                                //3
[P_PortControl_Data]=r2;               //9
r2^=B_WRPin;                           //6
[P_PortControl_Data]=r2;               //9
retf;                                  //12
.ENDIF
.ENDP
```

17、写数据

```
//=====
// Function Name: F_LCD_Data_Send
// Description:      Send display data through simulate port.
// Input:           None
// Output:          None
// Destroy:         r1,r2;
// Used:            r1,r2;
// Stacks:          0;
//=====
```

```
.PUBLIC F_LCD_Data_Send;
.PUBLIC _F_LCD_Data_Send;
F_LCD_Data_Send:                                //117cycle
_F_LCD_Data_Send: .PROC
//68000 interface
```

```
r2=B_USBOperation+B_EPPin+B_AOP_Data;          //Disable Read and USB operation

r1=[P_PortControl_Buf];
```



```
r1&=0xFFFF-B_RWPPin-B_LCDOperation;  
r2|=r1;  
[P_PortControl_Data]=r2;           //change control pin  
r2^=B_EPPin;  
[P_PortControl_Data]=r2;           //latch data  
retf;  
.ENDP
```

九、与 61 板结合使用

范 例 已 经 包 在 IDE184 下 ， 路 径 为
unSPIDE184/SPCE061A/Example/model_Exa/LCD501_Characer。

【目的】

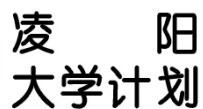
- 1、了解 SPLC501 的使用方法及相关函数
- 2、学习利用 SPLC501 显示字符

【设备】

- 1)装有 $\mu'nSPTM$ IDE 仿真环境的 PC 机一台。
- 2)61 板和液晶模组各一套。

【步骤】

- 1)根据硬件连接图连接好硬件。(实验箱默认的 LCD 连接方式)
- 2)将 $\mu'nSPTM$ IDE 打开后，建立一个新工程。
- 3)在该项目的源文件夹(SOURCE FILES)下建立一个新的 C 语言文件。
- 4)编写程序代码。
- 5)编译程序，软件调试。
- 6)注意观察 LCD 的现象

[illegible]

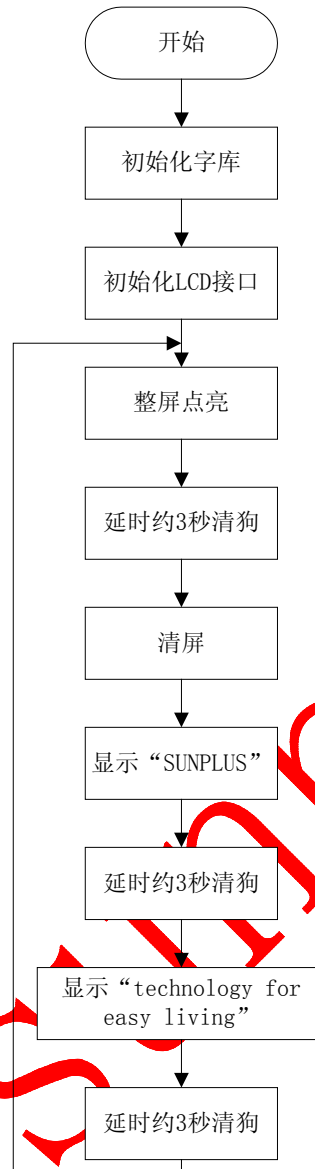
内藏 S PLC501 控制器图形液晶显示模组



内藏 SPLC501 控制器图形液晶显示模组



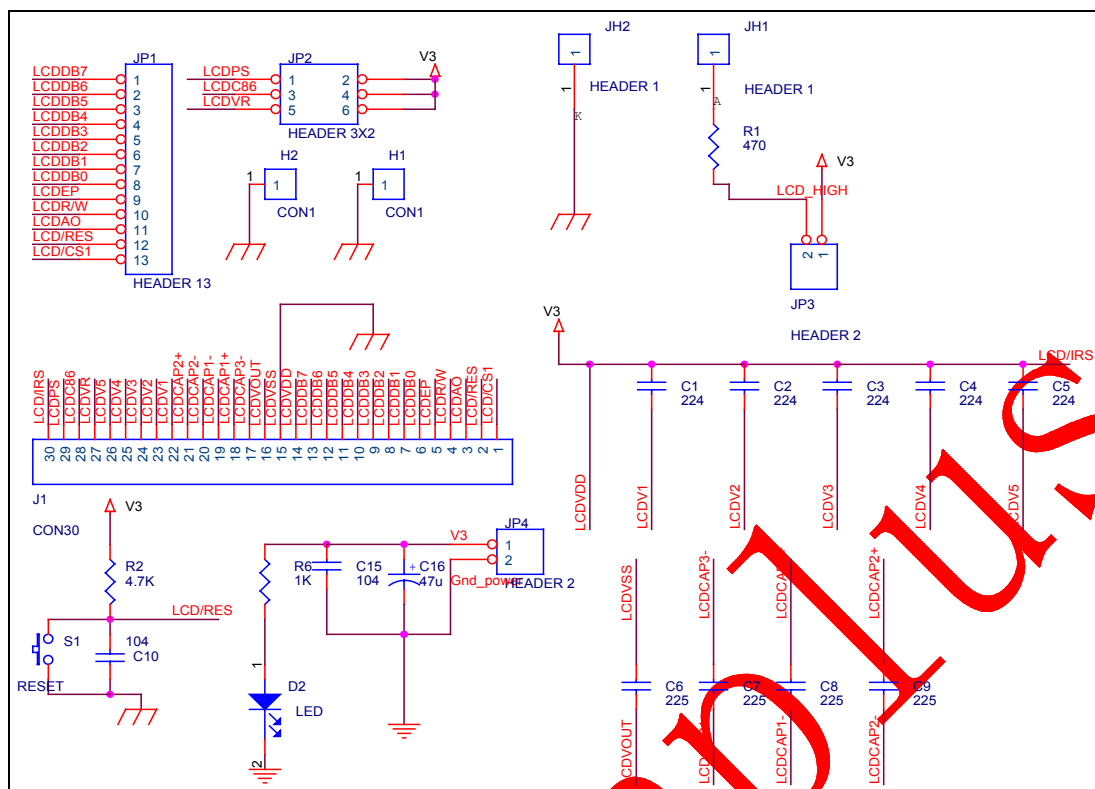
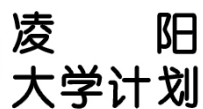
【程序流程图】



【液晶显示界面图】



内藏 SPLC501 控制器图形液晶显示模组



内藏 SPLC501 控制器图形液晶显示模组