

# JLX280-012-PN 使用说明书

## (不带字库 IC)

### 目 录

序号	内 容 标 题	页 码
1	概述	2
2	特点	2
3	外形及接口引脚功能	3~4
4	基本原理	4
5	技术参数	4~5
6	时序特性	5~7
7	指令功能及硬件接口与编程案例	8~末页

## 1. 概述

晶联讯电子专注于液晶屏及液晶模块的研发、制造。所生产 JLX280-012-PN 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。

JLX280-012-PN 可以显示 320 列\*240 行点阵彩色图片，或显示 20 个/行\*15 行 16\*16 点阵的汉字，或显示 40 个/行\*30 行 8\*8 点阵的英文、数字、符号。

本产品可选择带中文字库 IC 与不带中文字库 IC 两种。

## 2. JLX280-012-PN 彩色图像型点阵液晶模块的特性

2.1 结构轻、薄、带背光。

2.2 IC 采用 ILI9341, 功能强大，稳定性好

2.3 显示内容：

- 320\*240 点阵彩色图片；

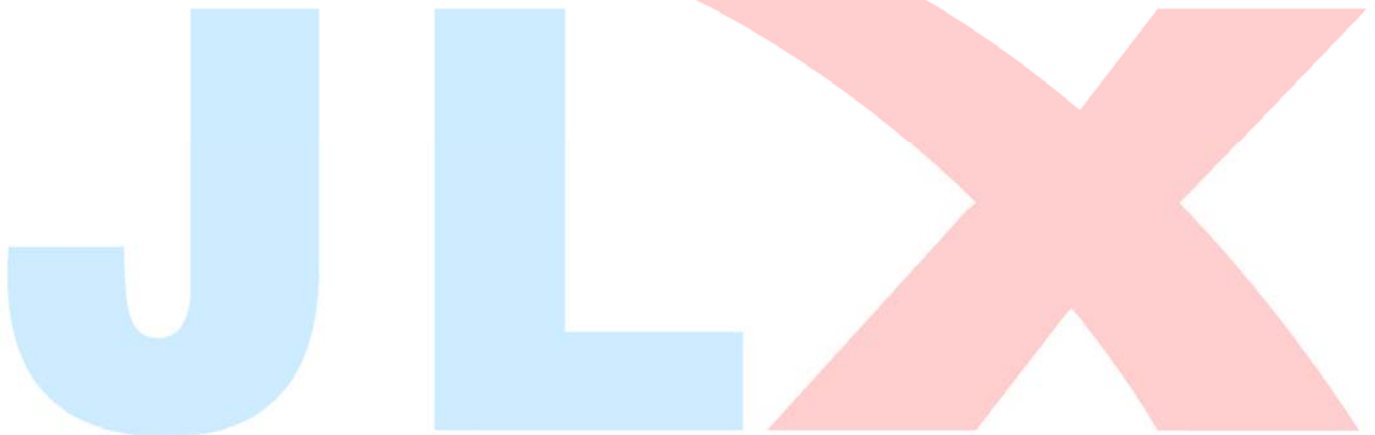
- 可選用 32\*32 点阵或其他点阵的图片来自编汉字，按照 32\*32 点阵汉字来计算可显示 10 个字/行\*7 行。

- 可選用 16\*16 点阵或其他点阵的图片来自编汉字，按照 16\*16 点阵汉字来计算可显示 20 个字/行\*15 行。

2.4 指令功能强：例如可以用指令控制显示内容顺时针旋转 90°、逆时针旋转 90° 或倒立竖放。

2.5 接口简单方便。

2.6 工作温度宽：-20℃ - 70℃；



3. 外形尺寸及接口引脚功能

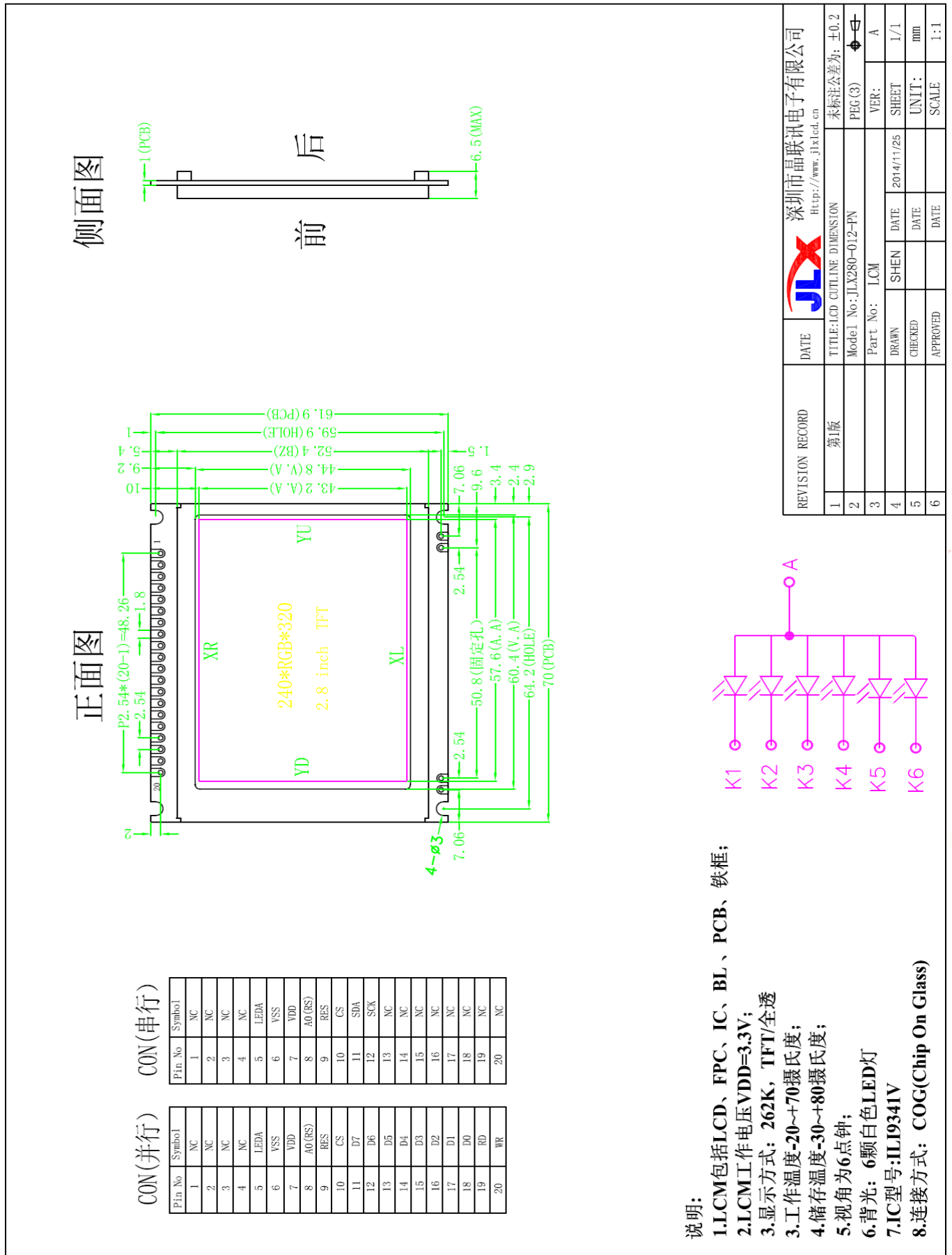


图 1. 外形尺寸

模块的接口引脚功能

表 1：模块的接口引脚功能

引线号	符号	名称	功能
1	NC	NC	空脚
2	NC	NC	空脚
3	NC	NC	空脚
4	NC	NC	空脚
5	LEDA	背光电源	背光电源正极，3.3V
6	VSS	接地	0V
7	VDD	电路电源	3.3V
8	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 "A0")
9	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
10	CS	片选	低电平片选
11-18	D7-D0	I/O	数据总线 DB7-DB0
19	RD (E)	使能信号	使能信号
20	WR	读/写	H: 读数据 0: 写数据

#### 4. 基本原理

##### 4.1 液晶屏 (LCD)

在 LCD 上排列着 320×240 点阵, 320 个列信号与驱动 IC 相连, 240 个行信号也与驱动 IC 相连, IC 绑定在 LCD 玻璃上 (这种加工工艺叫 COG)。

##### 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；

背光板是白色。

正常工作电流为：60~120mA (LED 灯数共 6 颗, 每颗灯是 10~20 mA)

工作电压：同 VDD 电压 (LED 灯本身的电压是 3.0V, 但是在 PCB 上已加了限流电阻, 所以可以同 VDD 电压)；

#### 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	-0.3	3.3	3.3	V
工作温度		-20		+70	°C
储存温度		-30		+80	°C

表 2：最大极限参数

5.2 直流 (DC) 参数

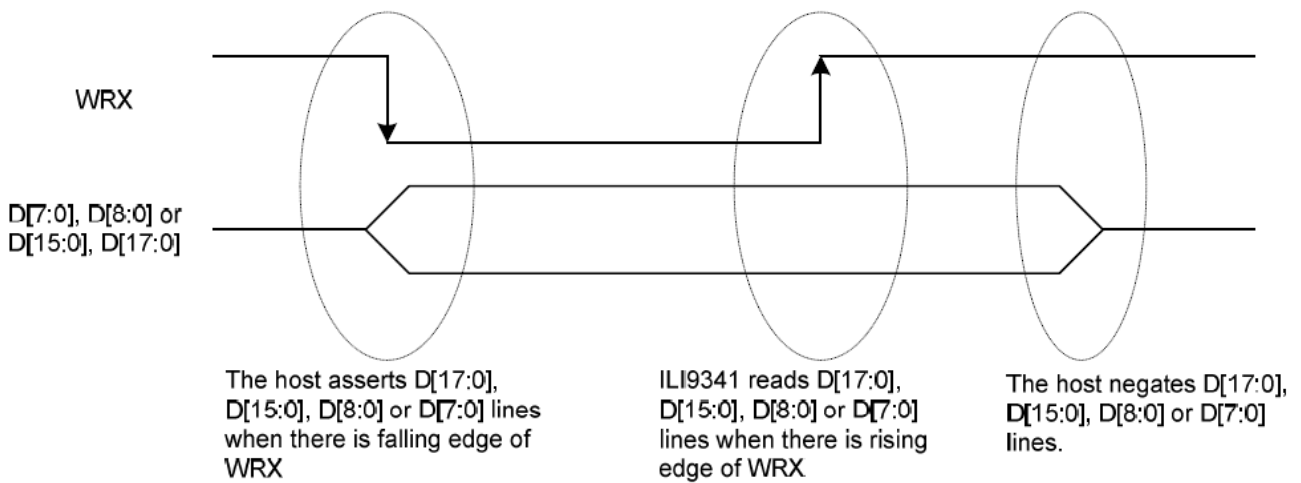
名称	符号	测试条件	标准值			单位
			最小	典型值	最大	
工作电压	VDD		2.8	3.0	3.3	V
背光工作电压	VLED		2.9	3.0	3.1	V
背光工作电流	ILED	VLED=3.0V, 共 5 颗 LED 灯并联	60	90	120	mA

表 3: 直流 (DC) 参数

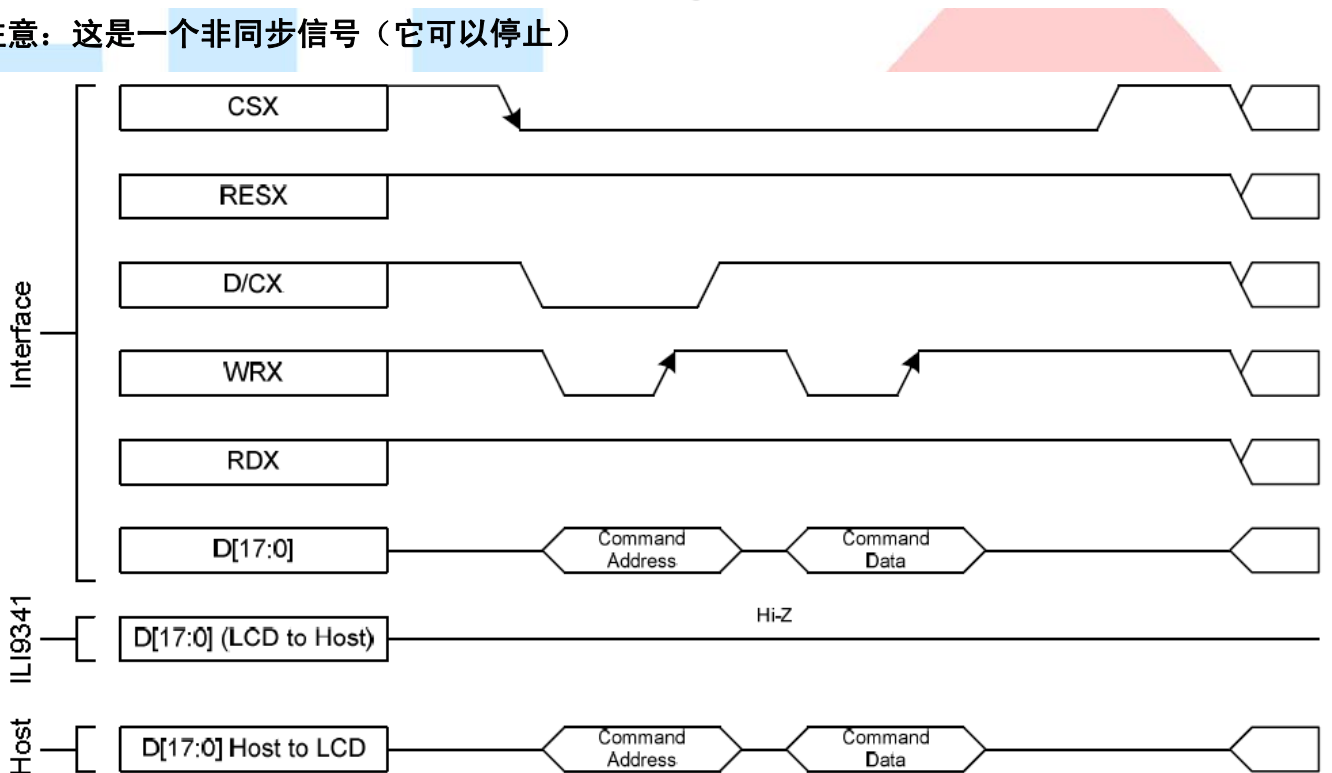
6. 读写时序特性

详见 IC 资料 “ILI9341”, 请找相关客服人员索要。

下图显示了一个写周期为 8080— I 单片机的接口。

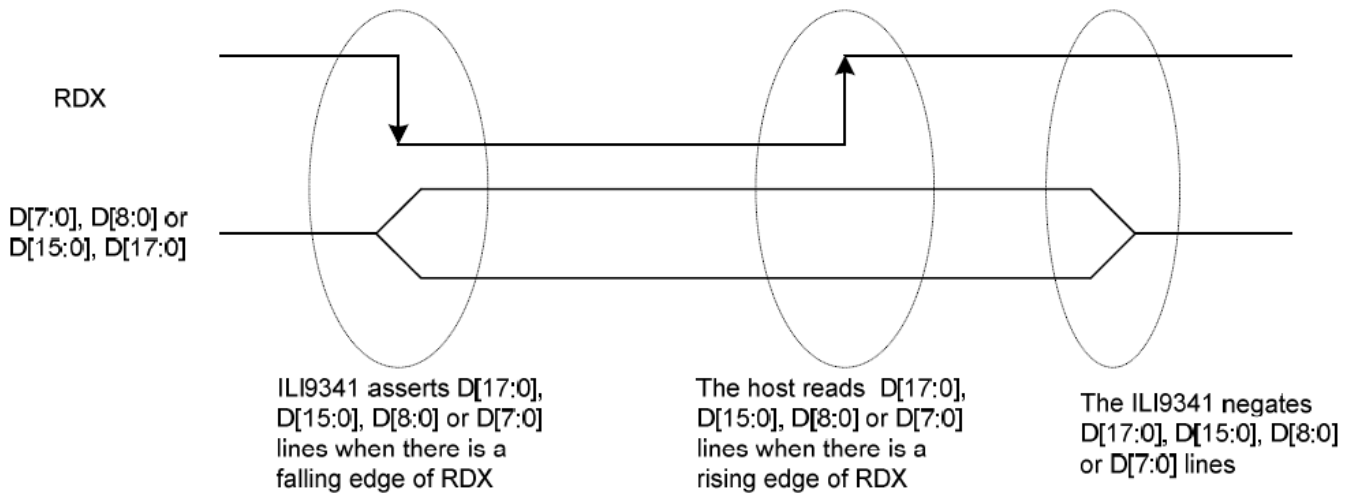


注意: 这是一个非同步信号 (它可以停止)

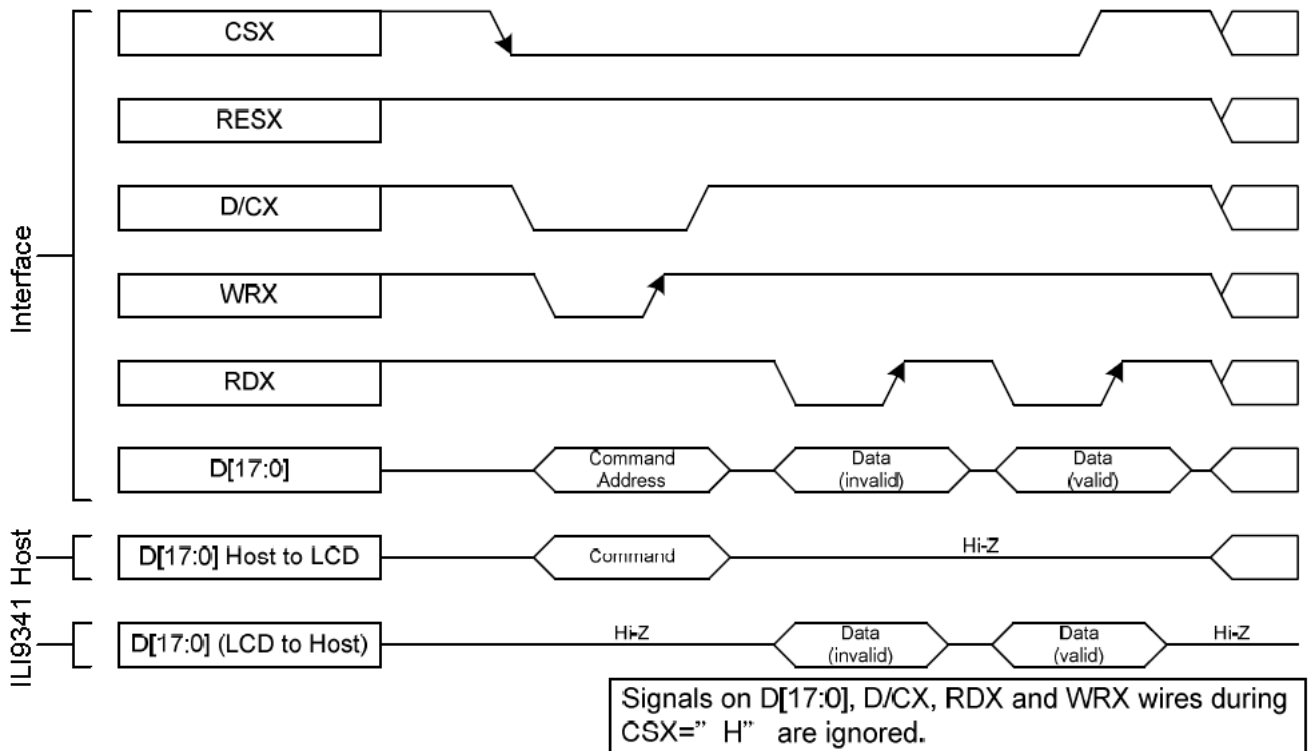


Signals on D[17:0], D/CX, RDX and WRX wires during CSX=" H" are ignored.

下图显示的读周期为 8080— I 单片机的接口。



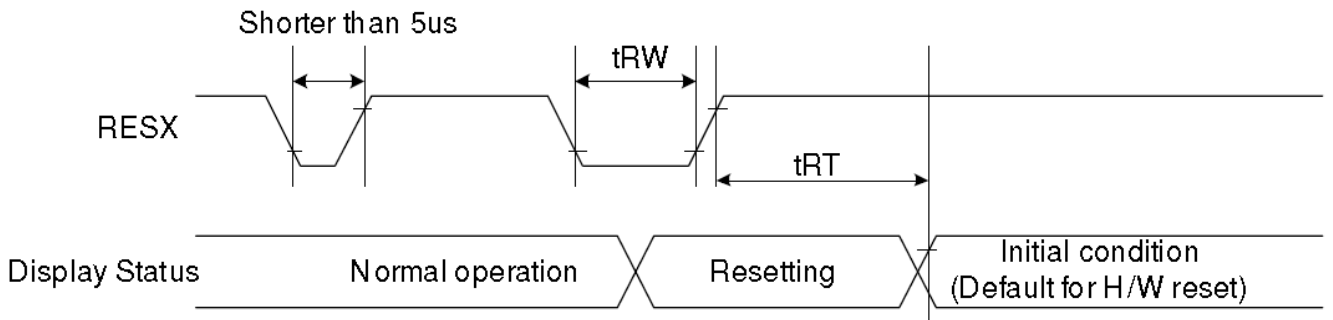
注：RDX 是一个非同步信号（它可以停止）。



注：读取数据只有在 D / CX 输入被拉高。如果 D / CX 驱动低在读然后显示信息的输出将高阻。

IM3	IM2	IM1	IM0	MCU-Interface Mode	CSX	WRX	RDX	D/CX	Function
0	0	0	0	8080 MCU 8-bit bus interface I	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.
0	0	0	1	8080 MCU 16-bit bus interface I	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.
0	0	1	0	8080 MCU 9-bit bus interface I	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.
0	0	1	1	8080 MCU 18-bit bus interface I	"L"		"H"	"L"	Write command code.
					"L"	"H"		"H"	Read internal status.
					"L"		"H"	"H"	Write parameter or display data.
					"L"	"H"		"H"	Reads parameter or display data.

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



图为电源启动后复位的时序  
表 6: 电源启动后复位的时序要求

Signal	Symbol	Parameter	Min	Max	Unit
RESX	tRW	Reset pulse duration	10		uS
	tRT	Reset cancel		5 (note 1,5)	mS
				120 (note 1,6,7)	mS

7. 指令功能:

7.1 指令表

指令表 8.

Regulative Command Set													
Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
No Operation	0	1	↑	XX	0	0	0	0	0	0	0	0	00h
Software Reset	0	1	↑	XX	0	0	0	0	0	0	0	1	01h
Read Display Identification Information	0	1	↑	XX	0	0	0	0	0	1	0	0	04h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	ID1 [7:0]							XX	
	1	↑	1	XX	ID2 [7:0]							XX	
	1	↑	1	XX	ID3 [7:0]							XX	
Read Display Status	0	1	↑	XX	0	0	0	0	1	0	0	1	09h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [31:25]							X	00
	1	↑	1	XX	X	D [22:20]			D [19:16]				61
	1	↑	1	XX	X	X	X	X	X	D [10:8]			00
	1	↑	1	XX	D [7:5]			X	X	X	X	X	00
Read Display Power Mode	0	1	↑	XX	0	0	0	0	1	0	1	0	0Ah
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	08
Read Display MADCTL	0	1	↑	XX	0	0	0	0	1	0	1	1	0Bh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	00
Read Display Pixel Format	0	1	↑	XX	0	0	0	0	1	1	0	0	0Ch
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	RIM	DPI [2:0]		X	DBI [2:0]				06
Read Display Image Format	0	1	↑	XX	0	0	0	0	1	1	0	1	0Dh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	D [2:0]			00
Read Display Signal Mode	0	1	↑	XX	0	0	0	0	1	1	1	0	0Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:2]						0	0	00
Read Display Self-Diagnostic Result	0	1	↑	XX	0	0	0	0	1	1	1	1	0Fh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	D [7:6]			X	X	X	X	X	00
Enter Sleep Mode	0	1	↑	XX	0	0	0	1	0	0	0	0	10h
Sleep OUT	0	1	↑	XX	0	0	0	1	0	0	0	1	11h
Partial Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	0	12h
Normal Display Mode ON	0	1	↑	XX	0	0	0	1	0	0	1	1	13h
Display Inversion OFF	0	1	↑	XX	0	0	1	0	0	0	0	0	20h
Display Inversion ON	0	1	↑	XX	0	0	1	0	0	0	0	1	21h
Gamma Set	0	1	↑	XX	0	0	1	0	0	1	1	0	26h
	1	1	↑	XX	GC [7:0]							01	
Display OFF	0	1	↑	XX	0	0	1	0	1	0	0	0	28h
Display ON	0	1	↑	XX	0	0	1	0	1	0	0	1	29h
Column Address Set	0	1	↑	XX	0	0	1	0	1	0	1	0	2Ah
	1	1	↑	XX	SC [15:8]							XX	
	1	1	↑	XX	SC [7:0]							XX	
	1	1	↑	XX	EC [15:8]							XX	
	1	1	↑	XX	EC [7:0]							XX	
Page Address Set	0	1	↑	XX	0	0	1	0	1	0	1	1	2Bh
	1	1	↑	XX	SP [15:8]							XX	
	1	1	↑	XX	SP [7:0]							XX	
	1	1	↑	XX	EP [15:8]							XX	
	1	1	↑	XX	EP [7:0]							XX	



Memory Write	0	1	↑	XX	0	0	1	0	1	1	0	0	2Ch
	1	1	↑		D [17:0]								XX
Color SET	0	1	↑	XX	0	0	1	0	1	1	0	1	2Dh
	1	↑	1	XX					R00 [5:0]				XX
	1	↑	1	XX					Rnn [5:0]				XX
	1	↑	1	XX					R31 [5:0]				XX
	1	↑	1	XX					G00 [5:0]				XX
	1	↑	1	XX					Gnn [5:0]				XX
	1	↑	1	XX					G64 [5:0]				XX
	1	↑	1	XX					B00 [5:0]				XX
	1	↑	1	XX					Bnn [5:0]				XX
	1	↑	1	XX					B31 [5:0]				XX
Memory Read	0	1	↑	XX	0	0	1	0	1	1	1	0	2Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1		D [17:0]								XX
Partial Area	0	1	↑	XX	0	0	1	1	0	0	0	0	30h
	1	1	↑	XX					SR [15:8]				00
	1	1	↑	XX					SR [7:0]				00
	1	1	↑	XX					ER [15:8]				01
	1	1	↑	XX					ER [7:0]				3F
Vertical Scrolling Definition	0	1	↑	XX	0	0	1	1	0	0	1	1	33h
	1	1	↑	XX					TFA [15:8]				00
	1	1	↑	XX					TFA [7:0]				00
	1	1	↑	XX					VSA [15:8]				01
	1	1	↑	XX					VSA [7:0]				40
	1	1	↑	XX					BFA [15:8]				00
	1	1	↑	XX					BFA [7:0]				00
Tearing Effect Line OFF	0	1	↑	XX	0	0	1	1	0	1	0	0	34h
Tearing Effect Line ON	0	1	↑	XX	0	0	1	1	0	1	0	1	35h
	1	1	↑	XX	X	X	X	X	X	X	X	M	00
Memory Access Control	0	1	↑	XX	0	0	1	1	0	1	1	0	36h
	1	1	↑	XX	MY	MX	MV	ML	BGR	MH	X	X	00
Vertical Scrolling Start Address	0	1	↑	XX	0	0	1	1	0	1	1	1	37h
	1	1	↑	XX					VSP [15:8]				00
	1	1	↑	XX					VSP [7:0]				00
Idle Mode OFF	0	1	↑	XX	0	0	1	1	1	0	0	0	38h
Idle Mode ON	0	1	↑	XX	0	0	1	1	1	0	0	1	39h
Pixel Format Set	0	1	↑	XX	0	0	1	1	1	0	1	0	3Ah
	1	1	↑	XX	X		DPI [2:0]		X		DBI [2:0]		66
Write Memory Continue	0	1	↑	XX	0	0	1	1	1	1	0	0	3Ch
	1	1	↑		D [17:0]								XX
Read Memory Continue	0	1	↑	XX	0	0	1	1	1	1	1	0	3Eh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1		D [17:0]								XX
Set Tear Scanline	0	1	↑	XX	0	1	0	0	0	1	0	0	44h
	1	1	↑	XX	X	X	X	X	X	X	X	STS [8]	00
	1	1	↑	XX					STS [7:0]				00
Get Scanline	0	1	↑	XX	0	1	0	0	0	1	0	1	45h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X		GTS [9:8]	00
	1	↑	1	XX					GTS [7:0]				00
Write Display Brightness	0	1	↑	XX	0	1	0	1	0	0	0	1	51h
	1	1	↑	XX					DBV [7:0]				00

Read Display Brightness	0	1	↑	XX	0	1	0	1	0	0	1	0	52h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	DBV [7:0]								00
Write CTRL Display	0	1	↑	XX	0	1	0	1	0	0	1	1	53h
	1	1	↑	XX	X	X	BCTRL	X	DD	BL	X	X	00
Read CTRL Display	0	1	↑	XX	0	1	0	1	0	1	0	0	54h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	BCTRL	X	DD	BL	X	X	00
Write Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	0	1	55h
	1	1	↑	XX	X	X	X	X	X	X	C [1:0]		00
Read Content Adaptive Brightness Control	0	1	↑	XX	0	1	0	1	0	1	1	0	56h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	X	X	X	X	X	X	C [1:0]		00
Write CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	1	1	1	0	5Eh
	1	1	↑	XX	CMB [7:0]								00
Read CABC Minimum Brightness	0	1	↑	XX	0	1	0	1	0	1	1	1	5Fh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	CMB [7:0]								00
Read ID1	0	1	↑	XX	1	1	0	1	1	0	1	0	DAh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	Module's Manufacture [7:0]								XX
Read ID2	0	1	↑	XX	1	1	0	1	1	0	1	1	DBh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver Version [7:0]								XX
Read ID3	0	1	↑	XX	1	1	0	1	1	1	0	0	DCh
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	LCD Module / Driver ID [7:0]								XX

Extended Command Set													
Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
RGB Interface Signal Control	0	1	↑	XX	1	0	1	1	0	0	0	0	B0h
	1	1	↑	XX	ByPass_MODE	RCM [1:0]		X	VSPL	HSPL	DPL	EPL	40
Frame Control (In Normal Mode)	0	1	↑	XX	1	0	1	1	0	0	0	1	B1h
	1	1	↑	XX	X	X	X	X	X	X	DIVA [1:0]		00
	1	1	↑	XX	X	X	X	RTNA [4:0]				1B	
Frame Control (In Idle Mode)	0	1	↑	XX	1	0	1	1	0	0	1	0	B2h
	1	1	↑	XX	X	X	X	X	X	X	DIVB [1:0]		00
	1	1	↑	XX	X	X	X	RTNB [4:0]				1B	
Frame Control (In Partial Mode)	0	1	↑	XX	1	0	1	1	0	0	1	1	B3h
	1	1	↑	XX	X	X	X	X	X	X	DIVC [1:0]		00
	1	1	↑	XX	X	X	X	RTNC [4:0]				1B	
Display Inversion Control	0	1	↑	XX	1	0	1	1	0	1	0	0	B4h
	1	1	↑	XX	X	X	X	X	X	NLA	NLB	NLC	02
Blanking Porch Control	0	1	↑	XX	1	0	1	1	0	1	0	1	B5h
	1	1	↑	XX	0	VFP [6:0]						02	
	1	1	↑	XX	0	VBP [6:0]						02	
	1	1	↑	XX	0	0	0	HFP [4:0]				0A	
	1	1	↑	XX	0	0	0	HBP [4:0]				14	

Display Function Control	0	1	↑	XX	1	0	1	1	0	1	1	0	B6h		
	1	1	↑	XX	X	X	X	X	PTG [1:0]		PT [1:0]		0A		
	1	1	↑	XX	REV	GS	SS	SM	ISC [3:0]				82		
	1	1	↑	XX	X	X	NL [5:0]						27		
	1	1	↑	XX	X	X	PCDIV [5:0]						XX		
Entry Mode Set	0	1	↑	XX	1	0	1	1	0	1	1	1	B7h		
	1	1	↑	XX	X	X	X	X	0	GON	DTE	GAS	07		
Backlight Control 1	0	1	↑	XX	1	0	1	1	1	0	0	0	B8h		
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX		
	1	1	↑	XX	X	X	X	X	TH_UI [3:0]				04		
Backlight Control 2	0	1	↑	XX	1	0	1	1	1	0	0	1	B9h		
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX		
	1	1	↑	XX	TH_MV [3:0]				TH_ST [3:0]				B8		
Backlight Control 3	0	1	↑	XX	1	0	1	1	1	0	1	0	BAh		
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX		
	1	1	↑	XX	X	X	X	X	DTH_UI [3:0]				04		
Backlight Control 4	0	1	↑	XX	1	0	1	1	1	0	1	1	BBh		
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX		
	1	1	↑	XX	DTH_MV [3:0]				DTH_ST [3:0]				C9		
Backlight Control 5	0	1	↑	XX	1	0	1	1	1	1	0	0	BCh		
	1	1	↑	XX	X	X	X	X	X	X	X	X	XX		
	1	1	↑	XX	DIM2 [3:0]				X	DIM1 [2:0]			44		
Backlight Control 7	0	1	↑	XX	1	0	1	1	1	1	1	0	BEh		
	1	1	↑	XX	PWM_DIV [7:0]										0F
	0	1	↑	XX	1	0	1	1	1	1	1	1	BFh		
Backlight Control 8	1	1	↑	XX	X	X	X	X	X	LEDONR	LEDONPOL	LEDPWMOPL	00		
	0	1	↑	XX	1	1	0	0	0	0	0	0	C0h		
Power Control 1	1	1	↑	XX	X	X	VRH [5:0]					26			
	0	1	↑	XX	1	1	0	0	0	0	0	1	C1h		
Power Control 2	1	1	↑	XX	X	X	X	X	X	BT [2:0]			00		
	0	1	↑	XX	1	1	0	0	0	1	0	1	C5h		
VCOM Control 1	1	1	↑	XX	X	VMH [6:0]						31			
	1	1	↑	XX	X	VML [6:0]						3C			
	0	1	↑	XX	1	1	0	0	0	1	1	1	C7h		
VCOM Control 2	1	1	↑	XX	nVM	VMF [6:0]						C0			
	0	1	↑	XX	1	1	0	1	0	0	0	0	D0h		
NV Memory Write	1	1	↑	XX	X	X	X	X	X	PGM_ADR [2:0]			00		
	1	1	↑	XX	PGM_DATA [7:0]							XX			
	0	1	↑	XX	1	1	0	1	0	0	0	1	D1h		
NV Memory Protection Key	1	1	↑	XX	KEY [23:16]								55		
	1	1	↑	XX	KEY [15:8]								AA		
	1	1	↑	XX	KEY [7:0]								66		
	0	1	↑	XX	1	1	0	1	0	0	1	0	D2h		
NV Memory Status Read	1	↑	1	XX	X	X	X	X	X	X	X	X	XX		
	1	↑	1	XX	X	ID2_CNT [2:0]			X	ID1_CNT [2:0]			XX		
	1	↑	1	XX	BUSY	VMF_CNT [2:0]			X	ID3_CNT [2:0]			XX		

Read ID4	0	↑	1	XX	1	1	0	1	0	0	1	1	D3h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	0	0	0	0	0	0	0	0	00
	1	↑	1	XX	1	0	0	1	0	0	1	1	93
	1	↑	1	XX	0	1	0	0	0	0	0	1	41
Positive Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	0	E0h
	1	1	↑	XX	X	X	X	X	VP0 [3:0]			08	
	1	1	↑	XX	X	X	VP1 [5:0]					0E	
	1	1	↑	XX	X	X	VP2 [5:0]					12	
	1	1	↑	XX	X	X	X	X	VP4 [3:0]			05	
	1	1	↑	XX	X	X	X	VP6 [4:0]				03	
	1	1	↑	XX	X	X	X	X	VP13 [3:0]			09	
	1	1	↑	XX	X	VP20 [6:0]						47	
	1	1	↑	XX	VP36 [3:0]			VP27 [3:0]				86	
	1	1	↑	XX	X	VP43 [6:0]						2B	
	1	1	↑	XX	X	X	X	X	VP50 [3:0]			0B	
	1	1	↑	XX	X	X	X	VP57 [4:0]				04	
	1	1	↑	XX	X	X	X	X	VP59 [3:0]			00	
	1	1	↑	XX	X	X	VP61 [5:0]					00	
	1	1	↑	XX	X	X	VP62 [5:0]					00	
1	1	↑	XX	X	X	X	X	VP63 [3:0]			00		
Negative Gamma Correction	0	1	↑	XX	1	1	1	0	0	0	0	1	E1h
	1	1	↑	XX	X	X	X	X	VN0 [3:0]			08	
	1	1	↑	XX	X	X	VN1 [5:0]					1A	
	1	1	↑	XX	X	X	VN2 [5:0]					20	
	1	1	↑	XX	X	X	X	X	VN4 [3:0]			07	
	1	1	↑	XX	X	X	X	VN6 [4:0]				0E	
	1	1	↑	XX	X	X	X	X	VN13 [3:0]			05	
	1	1	↑	XX	X	VN20 [6:0]						3A	
	1	1	↑	XX	VN36 [3:0]			VN27 [3:0]				8A	
	1	1	↑	XX	X	VN43 [6:0]						40	
	1	1	↑	XX	X	X	X	X	VN50 [3:0]			04	
	1	1	↑	XX	X	X	X	VN57 [4:0]				18	
	1	1	↑	XX	X	X	X	X	VN59 [3:0]			0F	
	1	1	↑	XX	X	X	VN61 [5:0]					3F	
	1	1	↑	XX	X	X	VN62 [5:0]					3F	
1	1	↑	XX	X	X	X	X	VN63 [3:0]			0F		
Digital Gamma Control 1	0	1	↑	XX	1	1	1	0	0	0	1	0	E2h
1 <sup>st</sup> Parameter	1	1	↑	XX	RCA0 [3:0]				BCA0 [3:0]				XX
:	1	1	↑	XX	RCAx [3:0]				BCAx [3:0]				XX
16 <sup>th</sup> Parameter	1	1	↑	XX	RCA15 [3:0]				BCA15 [3:0]				XX
Digital Gamma Control 2	0	1	↑	XX	1	1	1	0	0	0	1	1	E3h
1 <sup>st</sup> Parameter	1	1	↑	XX	RFA0 [3:0]				BFA0 [3:0]				XX
:	1	1	↑	XX	RFAx [3:0]				BFAX [3:0]				XX
64 <sup>th</sup> Parameter	1	1	↑	XX	RFA63 [3:0]				BFA63 [3:0]				XX
Interface Control	0	1	↑	XX	1	1	1	1	0	1	1	0	F6h
	1	1	↑	XX	MY_EOR	MX_EOR	MV_EOR	X	BGR_EOR	X	X	WEMODE	01
	1	1	↑	XX	X	X	EPF [1:0]		X	X	MDT [1:0]		00
	1	1	↑	XX	X	X	ENDIAN	X	DM [1:0]		RM	RIM	00

## 7.4 初始化方法

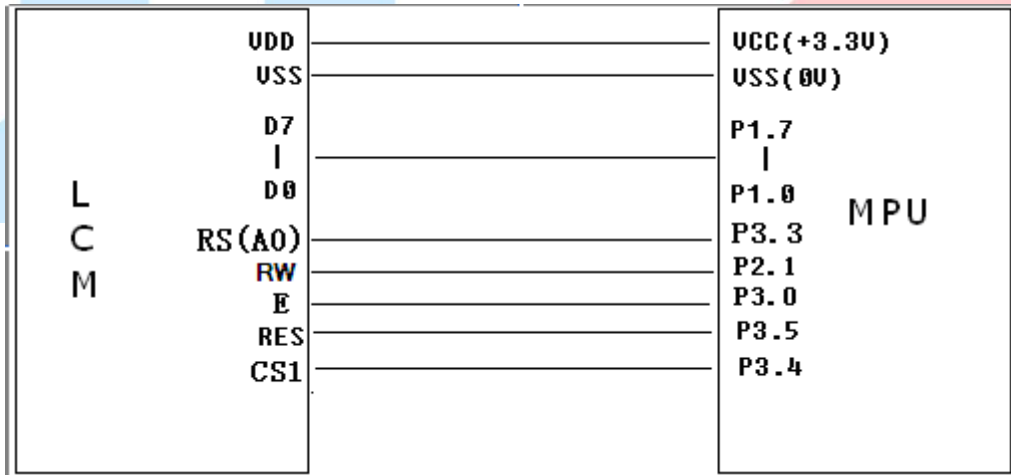
用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 过程请参考程序

**点亮液晶模块的步骤**

**硬件准备:**  
 开发板 (或专门设计的主板)、单片机、电源、连接线、仿真器或程序下载器 (又名烧录器)

**正确地接线**  
 根据说明书正确地与开发板连接, 连接的线包括: 液晶模块电源线、背光电源线、IO端口 (接口)  
 IO端口包括: 并口时: CS、RESET、RW、E、RS、D0—D7, 串口时: CS、SCLK、SDA、RESET、RS

**编写软件**  
 背光给合适的直流电可以点亮, 但液晶屏里面没有程序, 只给电不能让液晶屏显示 (我们通常说“点亮”), 程序须另外编写, 并烧录 (下载) 到单片机里液晶模块才能工作。



```
//=====
//液晶模块型号: JLX280-012-PN, |
//驱动 IC: IL19341 |
//接口: 8 位并行, 8080 时序 |
//版权所有: 深圳市晶联讯电子有限公司, 网站: http://www.jlxlcd.cn |
//=====
```

```
#include <reg51.h>
#include <intrins.h>
```

```
#include <160120_55.h> //存储 160*120 像素的 65k COLOR 的一幅彩图
//=====

sbit WR0=P2^1; //接口定义:WR0 就是 LCD 的 wr
sbit RD0=P3^0; //接口定义:RD0 就是 LCD 的 rd
sbit DC0=P3^3; //接口定义:DC0 就是 LCD 的 rs
sbit lcd_cs=P3^4; //接口定义:lcd_cs 就是 LCD 的 cs
sbit RST=P3^5; //接口定义:RST 就是 LCD 的 reset
sbit key=P2^0; //P2.0 口与 GND 之间接一个按键
//=====

//另外: 数据总线: P1 口

//定义彩屏旋转方向
#define normal 0x48
#define CW90 0xE8
#define CCW90 0x28
#define CW180 0x88

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

#define red 0xf800 //定义红色
#define blue 0x001f //定义蓝色
#define green 0x07e0 //定义绿色
#define white 0xffff //定义白色
#define black 0x0000 //定义黑色
#define orange 0xfc08 //定义橙色
#define yellow 0xffe0 //定义黄色
#define pink 0xf3f3 //定义粉红色
#define purple 0xa1d6 //定义紫色
#define brown 0x8200 //定义棕色
#define gray 0xc618 //定义灰色

//图片数据
uchar code pic2[];

//延时 1
//当用 STC12C5A60S2 单片机及晶振用 32MHz 时, K=293 就是 1 毫秒, 用 12MHz 晶振时, K=110 就是 1 毫秒
//当用 STC90C516RD+单片机及晶振用 32MHz 时, K=49 就是 1 毫秒
void delay_ms(long int i)
{
    long int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<50;k++);
}

//延时 2
//当用 STC12C5A60S2 单片机及晶振用 32MHz 时, K=2 就是 13 微秒
```

```
void delay_us(long int i)
{
    long int j,k;
    for (j=0;j<i;j++)
        for (k=0;k<2;k++);
}

void waitkey()
{
    repeat:
        if(key==1) goto repeat;
        else delay_ms(50);
}

//传送 8 位的数据
void data_out(uchar data1)
{
    //8080 8bit interface
    lcd_cs = 0;
    DC0 = 1;
    RDO = 1;
    P1=data1;
    WRO = 0;
    WRO = 1;
    lcd_cs = 1;
}

//传送 8 位的命令
void comm_out(uchar com)
{
    //8080 8bit interface
    DC0 = 0;
    lcd_cs = 0;
    RDO = 1;
    P1 = com;
    delay_us(2);
    WRO = 0;
    WRO = 1;
    lcd_cs = 1;
}

//传 16 位数据, 16 位数据一起赋值
void data_out_16(int data_16bit)
{
    data_out(data_16bit >>8);
    data_out(data_16bit);
}

//LCD 初始化
void LCD_initial()
{
```

```

delay_ms(50);
RST=0;           //低电平：复位
delay_ms(50);
RST=1;           //高电平：复位结束
delay_ms(10);
//开始初始化：
comm_out(0xCF);
data_out(0x00);
data_out(0xD9);
data_out(0X30);
comm_out(0xED);
data_out(0x64);
data_out(0x03);
data_out(0X12);
data_out(0X81);
comm_out(0xE8);
data_out(0x85);
data_out(0x00);
data_out(0x78);
comm_out(0xCB);
data_out(0x39);
data_out(0x2C);
data_out(0x00);
data_out(0x34);
data_out(0x02);
comm_out(0xF7);
data_out(0x20);
comm_out(0xEA);
data_out(0x00);
data_out(0x00);
comm_out(0xC0); //Power control
data_out(0x1B); //VRH[5:0]
comm_out(0xC1); //Power control
data_out(0x12); //SAP[2:0];BT[3:0]
comm_out(0xC5); //VCM control
data_out(0x32);
data_out(0x3C);
comm_out(0xC7); //VCM control2
data_out(0X9D);
comm_out(0x36); //行扫描顺序，列扫描顺序，横放/竖放
data_out(CW90); //定义：“normal”就是“0xc8”——正常竖放；
//定义：“CW180”就是“0x08”——在正常竖放基础上转180度竖放；
//定义：“CCW90”就是“0xa8”——在竖放基础上逆时针转90度横放；
//定义：“CW90”就是“0x68”——在竖放基础上顺时针转90度横放；

comm_out(0x3A);
data_out(0x55);
comm_out(0xB1);

```



```
data_out(0x00);
data_out(0x1B);
comm_out(0xB6); //显示功能设置: 列/行 显示顺序
data_out(0x0A);
data_out(0x82); //改变 SOURCE 线的方向: 0xa2: 左到右, 0x82: 右到左
comm_out(0xF6);
data_out(0x01);
data_out(0x30);
comm_out(0xF2); // 3Gamma Function Disable
data_out(0x00);
comm_out(0x26); //Gamma curve selected
data_out(0x01);
comm_out(0xE0); //设置 GAMMA 值, 由玻璃及 IC 原厂设置
data_out(0x0F);
data_out(0x24);
data_out(0x1F);
data_out(0x0B);
data_out(0x0F);
data_out(0x05);
data_out(0x4A);
data_out(0X96);
data_out(0x39);
data_out(0x07);
data_out(0x11);
data_out(0x03);
data_out(0x11);
data_out(0x0D);
data_out(0x04);

comm_out(0XE1); //设置 GAMMA 值, 由玻璃及 IC 原厂设置
data_out(0x00);
data_out(0x1B);
data_out(0x20);
data_out(0x04);
data_out(0x10);
data_out(0x02);
data_out(0x35);
data_out(0x23);
data_out(0x46);
data_out(0x04);
data_out(0x0E);
data_out(0x0C);
data_out(0x2E);
data_out(0x32);
data_out(0x05);

comm_out(0x11); //退出睡眠
```

```

    delay_ms(50);    //这个很重要
    comm_out(0x29);  //打开显示: display ON.
}
//将单色的 8 位的数据 (代表 8 个像素点) 转换成彩色的数据传输给液晶屏
void mono_data_out(uint mono_data,uint font_color,uint back_color)
{
    uint i;
    for(i=0;i<8;i++)
    {
        if(mono_data&0x80)
        {
            data_out_16(font_color);    //当数据是 1 时, 显示字体颜色
        }
        else
        {
            data_out_16(back_color);    //当数据是 0 时, 显示底色
        }
        mono_data<<=1;
    }
}
//定义窗口坐标: 开始坐标 (XS,YS)以及窗口大小 (x_total,y_total)
void lcd_address(int XS,int YS,int x_total,int y_total)
{
    int XE,YE;
    XE=XS+x_total-1;
    YE=YS+y_total-1;
    comm_out(0x2a);    // 设置 X 开始及结束的地址
    data_out_16(XS); // X 开始地址(16 位)
    data_out_16(XE); // X 结束地址(16 位)

    comm_out(0x2b);    // 设置 Y 开始及结束的地址
    data_out_16(YS); // Y 开始地址(16 位)
    data_out_16(YE); // Y 结束地址(16 位)
    comm_out(0x2c);    // 写数据开始
}
//显示 160x120 点阵的彩色图像
void disp_160x120(uint y,uint x,char *dp)
{
    int i,j;
    for(j=0;j<120;j++)
    {
        lcd_address(x,y+j,160,120);
        for(i=0;i<160;i++)
        {
            data_out(*dp);
            dp++;
            data_out(*dp);    //这一句相当于写了一个像素点的数据, 彩色 1 个的 16 位的数据。
        }
    }
}

```

```

        dp++;
    }
}
}
//=====
//显示全屏单一色彩
void display_color(int color)
{
    int i, j;
    lcd_address(0, 0, 320, 240);
    for(i=0; i<240; i++)
    {
        for(j=0; j<320; j++)
        {
            data_out_16(color);
        }
    }
}
//显示 32x32 点阵的单色的图像
void disp_32x32(int x, int y, char *dp, int font_color, int back_color)
{
    int i, j;
    lcd_address(x, y, 32, 32);
    for(i=0; i<32; i++)
    {
        for(j=0; j<4; j++)
        {
            mono_data_out(*dp, font_color, back_color);
            dp++;
        }
    }
}

void main(void)
{
    LCD_initial();
    while(1)
    {
        display_color(blue);
        disp_160x120(0, 0, pic2);           //在 (y, x) 位置显示一幅 160x120 像素的图片
        disp_160x120(120, 0, pic2);       //在 (y, x) 位置显示一幅 160x120 像素的图片
        disp_160x120(0, 160, pic2);       //在 (y, x) 位置显示一幅 160x120 像素的图片
        disp_160x120(120, 160, pic2);     //在 (y, x) 位置显示一幅 160x120 像素的图片
        waitkey();
        display_color(white);             //全屏写 0x0000, 相当于清屏。
        delay_ms(10);
        //全屏显示单色:
        delay_ms(1000);
    }
}

```

```

        display_color(blue);
    waitkey();
        display_color(0xf800);
    waitkey();

        display_color(0x07e0);
    waitkey();
    display_color(0x001f);
    waitkey();
    display_color(0x0000);
    waitkey();
}
}

```

