

JLX320240G-913-BN 使用说明书

(焊接式 FPC)

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	3-6
4	电路框图、背光参数	7
5	技术参数	7-8
6	时序特性	8-12
7	指令表及硬件接口、编程案例	13-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 JLX320240G-913-BN 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。

JLX320240G-913-BN 可以显示 320 列*240 行点阵单色的图片，或显示 10 个/行*7 行 32*32 点阵或显示 13 个/行*10 行 24*24 点阵的汉字，或显示 20 个/行*15 行 16*16 点阵的汉字。

2. JLX320240G-913-BN 图像型点阵液晶模块的特性

- 2.1 结构牢, 带档墙背光。
- 2.2 IC 采用矽创公司 ST75320, 功能强大, 稳定性好
- 2.3 功耗低。
- 2.4 接口简单方便: 可采用 4 线 SPI 串行接口、并行接口, I²C 接口。
- 2.5 工作温度宽: -20℃ - 70℃;
- 2.6 储存温度宽: -30℃ - 80℃;
- 2.7 显示内容:

- 320*240 点阵单色图片;
- 或显示 10 个×7 行 32*32 点阵的汉字;
- 或显示 13 个×10 行 24*24 点阵的汉字;
- 或显示 20 个×15 行 16*16 点阵的汉字;
- 或显示其他的 ASCII 码等;



3.2 模块的接口引脚功能

3.2.1 并行时接口引脚功能

表 1

引线号	符号	名称	功能
1	CS	片选	低电平片选
2	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶屏开始工作
3	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”)
4	RW(WR)	读/写	6800 时序: H: 读数据 0: 写数据
5	E (RD)	使能信号	6800 时序: 使能信号
6~13	DO~D7	I/O	并行接口时, 数据总线 DB0~DB7
14	VSS	接地	0V
15	IF2	IF2	L: 接低电平
16	IF1	IF1	H: 接高电平
17	IF0	IF1	L: 接低电平
18	VD1	VD1	VD1 接高电平, 使能内部电路
19	VDD	电源电路	供电电源正极
20	VDD	电源电路	供电电源正极
21	VSS	接地	0V
22	VC	倍压电路	VC 与 VSS 之间接一个电容
23	V2	倍压电路	V2 与 VSS 之间接一个电容
24	V1	倍压电路	V1 与 VSS 之间接一个电容
25	MV1	倍压电路	MV1 与 VSS 之间接一个电容
26	MV2	倍压电路	MV2 与 VSS 之间接一个电容
27	AVDD	倍压电路	AVDD 与 VSS 之间接一个电容
28	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
29	CA1N	倍压电路	
30	CA2P	倍压电路	CA2P 与 CA2N 之间接一个电容
31	CA2N	倍压电路	
32	CD1P	倍压电路	CD1P 与 CD1N 之间接一个电容
33	CD1N	倍压电路	
34	NVDD	倍压电路	NVDD 与 VSS 之间接一个电容
35	CD2N	倍压电路	CD2P 与 CD2N 之间接一个电容
36	CD2P	倍压电路	
37	MV3	倍压电路	MV3 与 VSS 之间接一个电容
38	CB1N	倍压电路	CB1P 与 CB1N 之间接一个电容
39	CB1P	倍压电路	
40	V3	倍压电路	V3 与 VSS 之间接一个电容
41	CE1P	倍压电路	CE1P 与 CE1N 之间接一个电容
42	CE1N	倍压电路	
43	CE3P	倍压电路	CE3P 与 CE1N 之间接一个电容
44	CE2N	倍压电路	
45	CE2P	倍压电路	CE2P 与 CE2N 之间接一个电容
46	VOUT	倍压电路	

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

表 2

引线号	符号	名称	功能
1	CS	片选	低电平片选
2	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶屏开始工作
3	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”)
4	RW (WR)	读/写	串行接口, 接高电平“H”
5	E (RD)	使能信号	串行接口, 接高电平“H”
6	D0	SCLK	串行接口, 串行时钟
7~9	D1、D2、D3	SDA	串行接口, 串行数据 (D1、D2、D3 短接一起做 SDA)
10~13	D4~D7	I/O	串行接口, D4、D5、D6、D7 都接高电平“H”
14	VSS	接地	0V
15	IF2	IF2	L: 接低电平
16	IF1	IF1	L: 接低电平
17	IF0	IF1	H: 接高电平
18	VD1	VD1	VD1 接高电平, 使能内部电路
19	VDD	电源电路	供电电源正极
20	VDD	电源电路	供电电源正极
21	VSS	接地	0V
22	VC	倍压电路	VC 与 VSS 之间接一个电容
23	V2	倍压电路	V2 与 VSS 之间接一个电容
24	V1	倍压电路	V1 与 VSS 之间接一个电容
25	MV1	倍压电路	MV1 与 VSS 之间接一个电容
26	MV2	倍压电路	MV2 与 VSS 之间接一个电容
27	AVDD	倍压电路	AVDD 与 VSS 之间接一个电容
28	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
29	CA1N	倍压电路	CA2P 与 CA2N 之间接一个电容
30	CA2P	倍压电路	
31	CA2N	倍压电路	CD1P 与 CD1N 之间接一个电容
32	CD1P	倍压电路	
33	CD1N	倍压电路	NVDD 与 VSS 之间接一个电容
34	NVDD	倍压电路	
35	CD2N	倍压电路	CD2P 与 CD2N 之间接一个电容
36	CD2P	倍压电路	
37	MV3	倍压电路	MV3 与 VSS 之间接一个电容
38	CB1N	倍压电路	CB1P 与 CB1N 之间接一个电容
39	CB1P	倍压电路	
40	V3	倍压电路	V3 与 VSS 之间接一个电容
41	CE1P	倍压电路	CE1P 与 CE1N 之间接一个电容
42	CE1N	倍压电路	
43	CE3P	倍压电路	CE3P 与 CE1N 之间接一个电容
44	CE2N	倍压电路	
45	CE2P	倍压电路	CE2P 与 CE2N 之间接一个电容
46	VOUT	倍压电路	
			VOUT 与 VSS 之间接一个电容

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	CS	片选	I ² C 接口, 接高电平“H”
2	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶屏开始工作
3	A0(RS)	寄存器选择信号	I ² C 接口, 接高电平“H”
4	RW(WR)	读/写	I ² C 接口, 接高电平“H”
5	E(RD)	使能信号	I ² C 接口, 接高电平“H”
6	D0	SCLK	I ² C 接口, 时钟信号
7~11	D1~D5	SDA	I ² C 接口, 数据信号 (D1、D2、D3、D4、D5 短接一起做 SDA)
12~13	D6、D7	I/O	I ² C 接口, 从属地址, 接低电平“L”
14	VSS	接地	0V
15	IF2	IF2	H:接高电平
16	IF1	IF1	H:接高电平
17	IF0	IF1	L:接低电平
18	VD1	VD1	VD1 接高电平, 使能内部电路
19	VDD	电源电路	供电电源正极
20	VDD	电源电路	供电电源正极
21	VSS	接地	0V
22	VC	倍压电路	VC 与 VSS 之间接一个电容
23	V2	倍压电路	V2 与 VSS 之间接一个电容
24	V1	倍压电路	V1 与 VSS 之间接一个电容
25	MV1	倍压电路	MV1 与 VSS 之间接一个电容
26	MV2	倍压电路	MV2 与 VSS 之间接一个电容
27	AVDD	倍压电路	AVDD 与 VSS 之间接一个电容
28	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
29	CA1N	倍压电路	CA2P 与 CA2N 之间接一个电容
30	CA2P	倍压电路	
31	CA2N	倍压电路	CD1P 与 CD1N 之间接一个电容
32	CD1P	倍压电路	
33	CD1N	倍压电路	NVDD 与 VSS 之间接一个电容
34	NVDD	倍压电路	
35	CD2N	倍压电路	CD2P 与 CD2N 之间接一个电容
36	CD2P	倍压电路	
37	MV3	倍压电路	MV3 与 VSS 之间接一个电容
38	CB1N	倍压电路	CB1P 与 CB1N 之间接一个电容
39	CB1P	倍压电路	
40	V3	倍压电路	V3 与 VSS 之间接一个电容
41	CE1P	倍压电路	CE1P 与 CE1N 之间接一个电容
42	CE1N	倍压电路	
43	CE3P	倍压电路	CE3P 与 CE1N 之间接一个电容
44	CE2N	倍压电路	
45	CE2P	倍压电路	CE2P 与 CE2N 之间接一个电容
46	VOUT	倍压电路	
			VOUT 与 VSS 之间接一个电容

 表 3: I²C 总线接口引脚功能

4. 电路框图

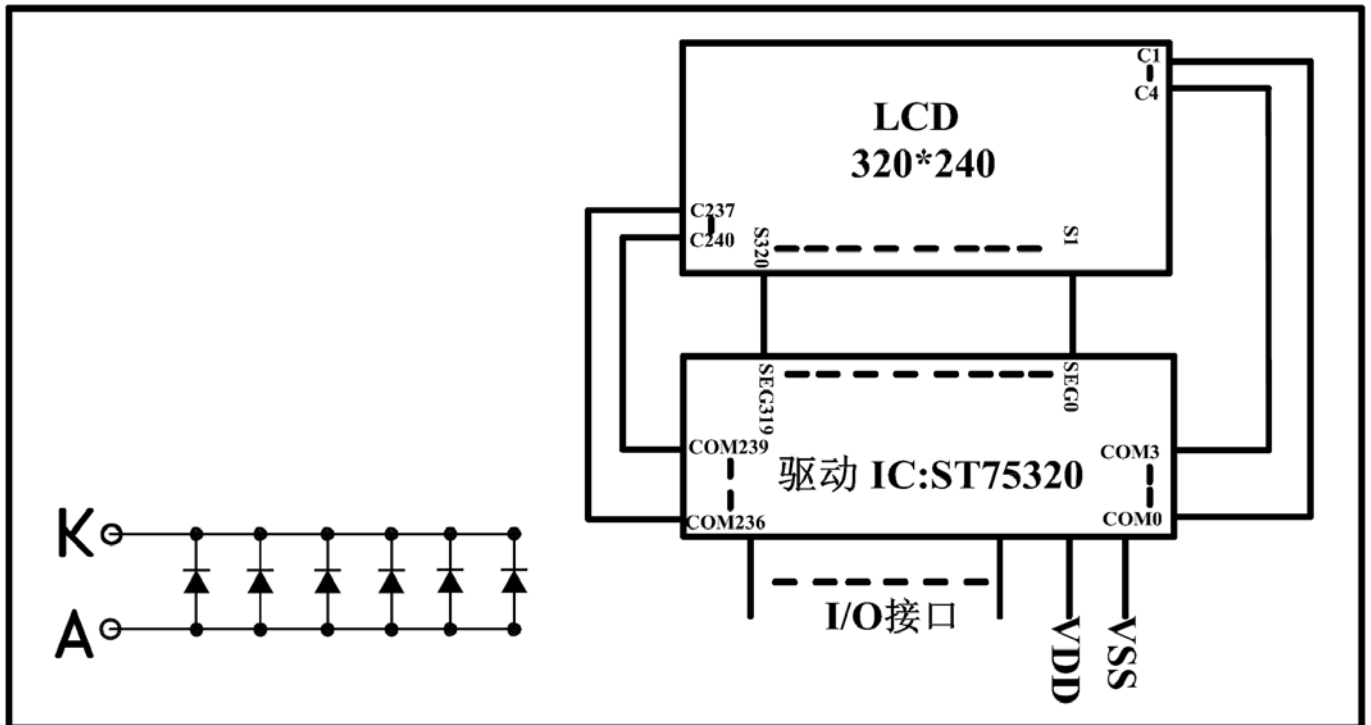


图 2: JLX320240G-913-BN 图像点阵型液晶模块的电路框图

4.1 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: $-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$;

背光颜色: 白色。

正常工作电流为: $(8 \sim 20) \times 6 = 48 \sim 120\text{mA}$ (LED 灯数共 6 颗);

工作电压: 3.0;

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	3.5V	3.5	V
LCD 驱动电压	V3 - MV3	-0.3	—	33	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 4: 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	—	2.7	3.3	3.5	V

背光工作电压	VLED	—	2.9	3.0	3.1	V
输入高电平	VIH	—	0.7VDD	—	VDD	V
输入低电平	VIO	—	0	—	0.3VDD	V
输出高电平	VOH	IOH = 0.2mA	0.8VDD	—	VDD	V
输出低电平	VOO	IOO = 1.2mA	0	—	0.2VDD	V
模块工作电流	IDD	VDD = 3.0V	—	0.3	1.0	mA
背光工作电流	ILED	VLED=3.0V	48	90	120	mA

表 5: 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)

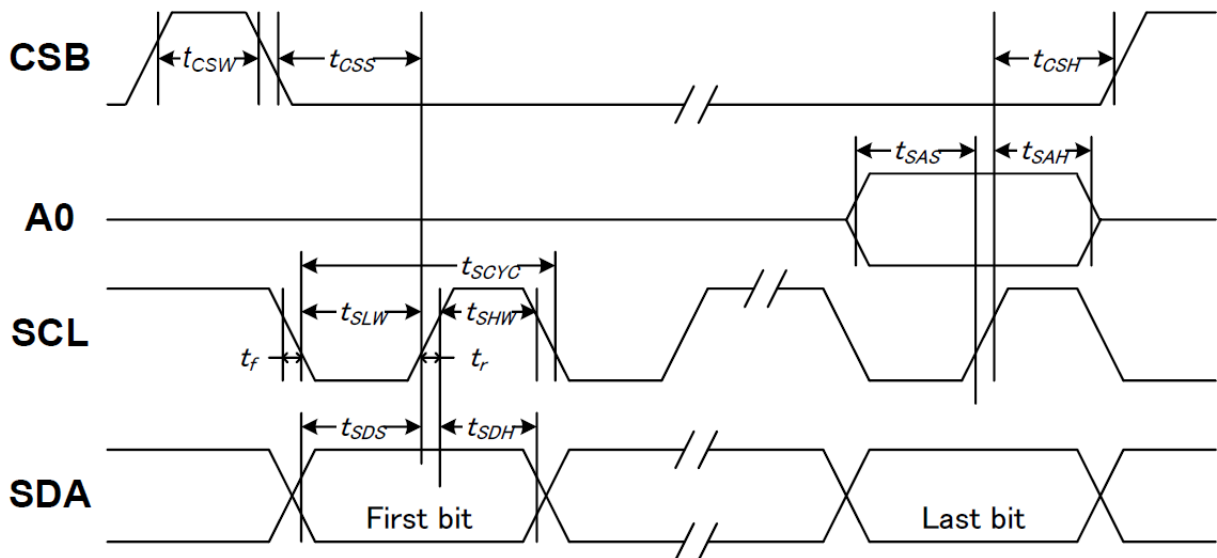


图 3. 从 CPU 写到 ST75320 (Writing Data from CPU to ST75320)

表 6. 写数据到 ST75320 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)	tSCYC	引脚: SCL	140	---	---	ns
保持SCK高电平脉宽 (SCL "H" pulse width)	tSHW		70	---	---	ns
保持SCLK低电平脉宽 (SCL "L" pulse width)	tSLW		60	---	---	ns
地址建立时间 (Address setup time)	tSAS	引脚: A0	20	---	---	ns
地址保持时间 (Address hold time)	tSAH		20	---	---	ns
数据建立时间 (Data setup time)	tSDS	引脚: SID	40	---	---	ns

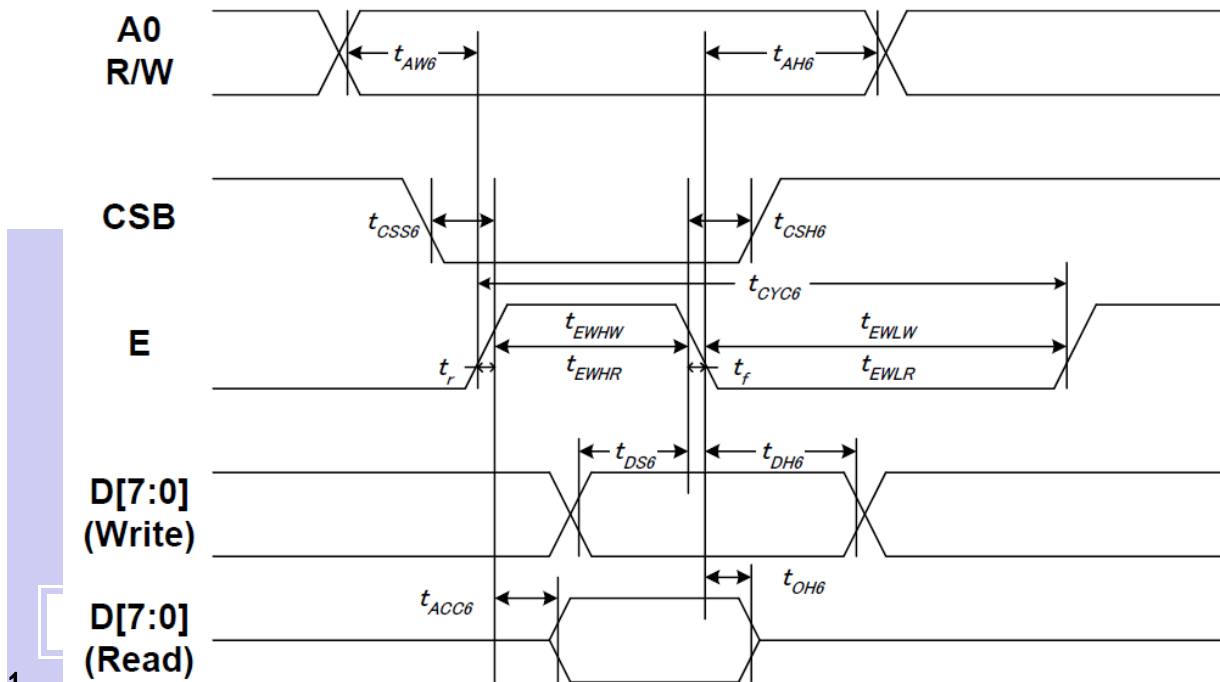
数据保持时间 (Data hold time)	tSDH		40	--	--	ns
片选信号建立时间 (CS-SCL time)	tCSS	引脚: CSB	60	--	--	ns
片选信号保持时间 (CS-SCL time)	tCSH		70	--	--	ns

VDD = 1.8~3.3V ± 5%, Ta = -30~85°C

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)



1. 从 CPU 写到 ST75320 (Writing Data from CPU to ST75320)

图 4. 写数据到 ST75320 的时序要求 (6800 系列 MPU)

表 7. 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	10		--	ns
地址建立时间		tAW6	0		--	ns
片选信号建立时间	CS	tCSS6	75			ns
片选信号保持时间		Tcsh6	50			ns
系统循环时间	E	tCYC6	320		--	ns
使能“低”脉冲宽度		tEHLW	150		--	ns
使能“高”脉冲宽度		tEHLW	150		--	ns
写数据建立时间	DB[7: 0]	tDS6	75		--	ns
写数据保持时间		tDH6	50		--	ns

VDD = 1.8~3.3V ± 5%, Ta = -30~85°C

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,

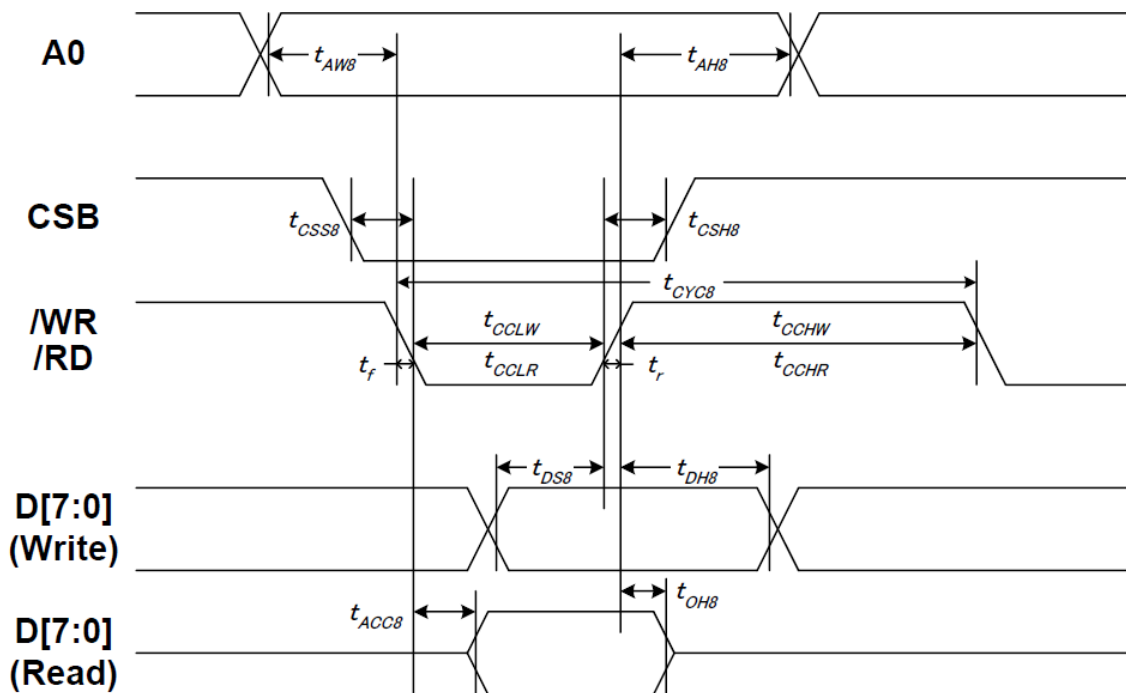
$(TR + TF) \leq (tcyc6 - tewlw - tewhw)$ 指定。

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 ST75320 (Writing Data from CPU to ST75320)

图 5. 写数据到 ST75320 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	tAW8	10		--	ns
地址建立时间		tAH8	0		--	ns
系统循环时间	/WR	tCYC8	320		--	ns
片选信号建立时间		tCSS6	75			ns
片选信号保持时间		Tcsh6	50			ns
使能“低”脉冲宽度		tCCLW	150		--	ns
使能“高”脉冲宽度		tCCHW	150		--	ns
写数据建立时间	DB	tDS8	50		--	ns
写数据保持时间		tDH8	50		--	ns

VDD = 1.8~3.3V ± 5%, Ta = -30~85°C

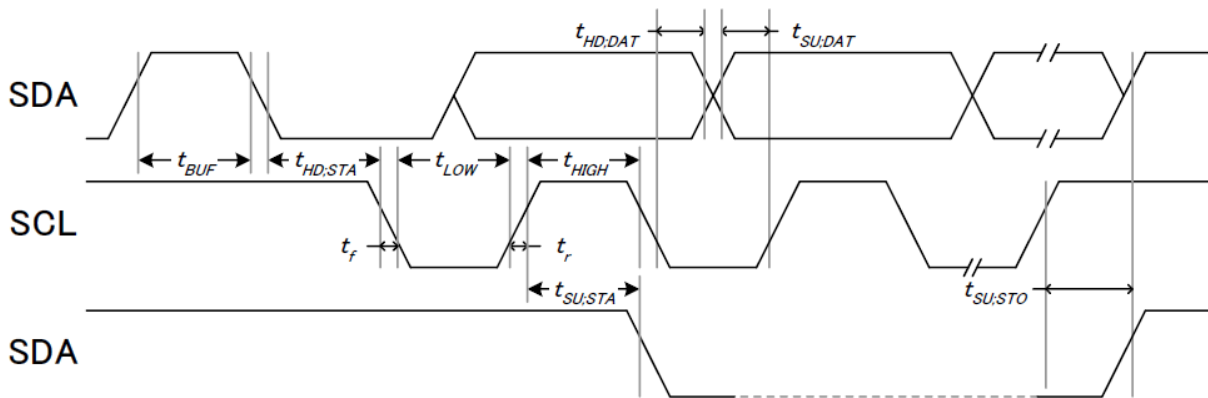
输入信号的上升时间和下降时间 (TR, TF) 是在 15 纳秒或更少的规定。当系统循环时间非常快,

$(TR + TF) \leq (tcyc8 - tcclw - tcchw)$ 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 I²C 接口的时序特性 (AC 参数)



从 CPU 写到 ST75320 (Writing Data from CPU to ST75320)

图 6. 写数据到 ST75320 的时序要求 (I²C 系列 MPU)

表 9. 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	FSCCLK	--		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	0.1		--	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	0.1			us
------------------	-----	------	-----	--	--	----

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):

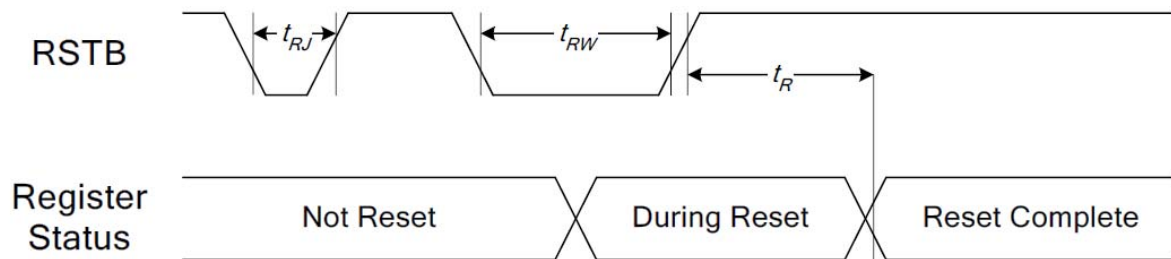


图 7: 电源启动后复位的时序

表 10: 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	T_{RW}		30	--	--	us
复位保持低电平的时间	T_R		--	--	10	us

7. 指令功能:

7.1 指令表

表 11

指令名称	指令码										
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(1) 显示开/关 (display on/off)	0	0	1	0	1	0	1	1	1	0	显示开/关: 0XAE:关, 0XAF: 开
(2) 正显/反显 (Inverse Display)	0	0	1	0	1	0	0	1	1	0	显示正显/反显 0XA6: 正显, 正常 0XA7: 反显
(3) 所有点阵开/关 (All Pixel ON/OFF)	0	0	1	0	1	0	0	0	1	0	0XA2: 所有点阵关 0XA3: 所有点阵开
(4) COM输出控制 (COM Output Status)	0	0	1	1	0	0	0	1	0	0	设置 COM 扫描模式, 0xc4
	1	0	-	-	-	-	-	0	SCAN	MY	SCAN =0; 正常扫描 SCAN =1; 隔行扫描 MY=0; COM0→COM239 MY=0; COM239→COM0
(5) 显示起始行 (Display Start Line)	0	0	1	0	0	0	1	0	1	0	设置显示开始行, 0x8a
	1	0	-	-	S5	S4	S3	S2	S1	S0	
(6) 页地址设置 (Set Page Address)	0	0	1	0	1	1	0	0	0	1	0Xb1: 页地址设置
	1	0	-	-	Y5	Y4	Y3	Y2	Y1	Y0	0X00: 起始页地址
(7) 列地址设置 (Set Column Address)	0	0	0	0	0	1	0	0	1	1	0X13: 列地址设置
	1	0	-	-	-	-	-	-	-	X8	0X00: 起始列地址
	1	0	X7	X6	X5	X4	X3	X2	X1	X0	0X00: 起始列地址
(8) 写数据到液晶屏 (Display Data Write)	0	0	0	0	0	1	1	1	0	1	0X1d: 写显示数据 DDRAM
	1	0	D7	D6	D5	D4	D3	D2	D1	D0	8 位显示数据
(9) 读液晶屏显示数据 (Display Data Read)	0	0	0	0	0	1	1	1	0	0	0X1C: 读数据
	1	1	D7	D6	D5	D4	D3	D2	D1	D0	8 位显示数据
(10) 液晶屏扫描方向	0	0	1	0	0	0	0	1	0	DIR	DIR=0; 列扫描 DIR=1; 行扫描
(11) 列扫描方向 (Column Address Direction)	0	0	1	0	1	0	0	0	0	MX	列扫描方向: 0xa0 或 0xa1
											MX=0: Column0→ Column 319 MX=0: Column 319→ Column0
(12) N行反显 (N-Line Inversion)	0	0	0	0	1	1	0	1	1	0	0X36: N 行反显
	1	0	-	-	NL5	NL4	NL3	NL2	NL1	NL0	
(13) N行反显开或关	0	0	1	1	1	0	0	1	0	NL	NL=0, 0xe4: N 行反显关 NL=1, 0xe5: N 行反显开
(14) 显示区域 (Display Area)	0	0	0	1	1	0	1	1	0	1	0X6C: 设置显示区域
	1	0	-	-	-	-	-	DTY2	DTY1	DTY0	DTY[2:0]=0x00~0x07
	1	0	-	-	SP5	SP4	SP3	SP2	SP1	SP0	SP[5:0]=0x00~0x4f
(15) 读改写	0	0	1	1	1	0	0	0	0	0	0XE0: 进入读改写模式
(16) 结束读改写	0	0	1	1	1	0	1	1	1	0	0XEE: 结束读改写模式
(17) 振荡电路开或关	0	0	1	0	1	0	1	0	1	OSC	内部振荡电路开或关
											0XAA: 振荡电路关 0XAB: 振荡电路开
(18) 工作时钟频率 (Operation Clock Frequency)	0	0	0	1	0	1	1	1	1	1	0X5F: 设置不同温度范围内的帧率
	1	0	FRB3	FRB2	FRB1	FRB0	FRA3	FRA3	FRA3	FRA3	
	1	0	FRD3	FRD3	FRD3	FRD3	FRD3	FRC3	FRC3	FRC3	
(19) 电源控制	0	0	0	0	1	0	0	1	0	1	0X26: 设置内建电路开或关

(Power Control)	1	0	-	VOUT	VAD	V3	VPF	VMV3	VMAD	VNF	
(20) 帧率水平	0	0	0	0	1	0	1	0	1	1	0X29: 设置帧速率的级别
	1	0	-	-	-	-	-	-	-	DBL	
(21) LCD偏压比设置 BIAS	0	0	1	0	1	0	0	0	1	0	0XA2: LCD 偏压比设置
	1	0	-	-	-	-	BS3	BS2	BS1	BS0	
(22)液晶内部电压设置 (Electronic Volume)	0	0	1	0	0	0	0	0	0	1	0X81: 设置对比度
	1	0	EV7	EV6	EV5	EV4	EV3	EV2	EV1	EVO	微调对比度, 范围 0X00-0XFF
	1	0	-	-	-	-	-	-	EV9	EV8	粗调对比度, 范围 0X00-0X03
(23) 电源放电 (Power Discharge)	0	0	1	1	1	0	1	0	1	0	0Xe9:
(24)睡眠模式 (Power Save)	0	0	1	0	1	0	1	0	0	PD	PD=0, 0XA8: 工作模式
											PD=1, 0XA9: 睡眠模工
(25) 温度梯度补偿 (Temperature Gradient Compensation)	0	0	0	1	0	0	1	1	1	0	0X4F: 设置温度梯度补偿
	1	0	TM1 [3:0]				TM0 [3:0]				
	1	0	TM3 [3:0]				TM2 [3:0]				
	1	0	TM5 [3:0]				TM4 [3:0]				
	1	0	TM7 [3:0]				TM6 [3:0]				
	1	0	TM9 [3:0]				TM8 [3:0]				
	1	0	TMB [3:0]				TMA [3:0]				
	1	0	TMD [3:0]				TMC [3:0]				
	1	0	TMF [3:0]				TME [3:0]				
(26)读状态 (Read Status)	0	0	1	0	0	0	1	1	1	0	0X80: 读 IC 状态
	0	1	D	OSC	AVD	V3	VFP	VMV3	VNAD	VFN	
	0	1	DISV	ITR	MY	PD	TD	NLFR	MLS	-	
(27)温度检测 (Temperature Detection)	0	0	0	1	1	0	1	0	0	TD	0X68: 不使能
											0X69: 使能
(28)LCD驱动模式 (LCD Driving method)	0	0	1	1	1	0	0	1	1	1	0XE7: LCD 驱动模式设置
			0	0	0	NLFR	1	0	0	1	
(29)空操作	0	0	1	1	1	0	0	0	1	1	0XE3: 空操作
(30) 频率补偿温度范围 (Frequency Compensation Temperature Range)	0	0	1	1	1	0	1	1	0	0	0XEC: 频率补偿温度范围设置
	1	0	-	TA6	TA5	TA4	TA3	TA2	TA1	TA0	
	1	0	-	TB6	TB5	TB4	TB3	TB2	TB1	TB0	
	1	0	-	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
(31) 温度滞后值 (Temperature Hysteresis value)	0	0	1	1	1	0	1	1	0	1	0XEC: 设置温度滞后值
	1	0	-	-	THV5	THV4	THV3	THV2	THV1	THV0	
	1	0	0	0	0	0	THF3	THF2	THF1	THF0	
(32) 当前温度数据 (Current Temperature Data)	0	0	1	1	1	0	1	1	1	1	0XEF: 监控当前温度
	0	1	T7	T6	T5	T4	T3	T2	T1	T0	
(33)读ID	0	0	1	0	0	0	1	1	1	1	0X8F: 读 ID 值
	0	1	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
(34)测试(Test)	0	0	1	1	1	1	1	1	TE	T	此功能不用

表 8. 指令表

请详细参考 IC 资料”ST75320.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 320*240 点阵的屏分为 30 个“页”, 从第 0 “页” 到第 29 “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:

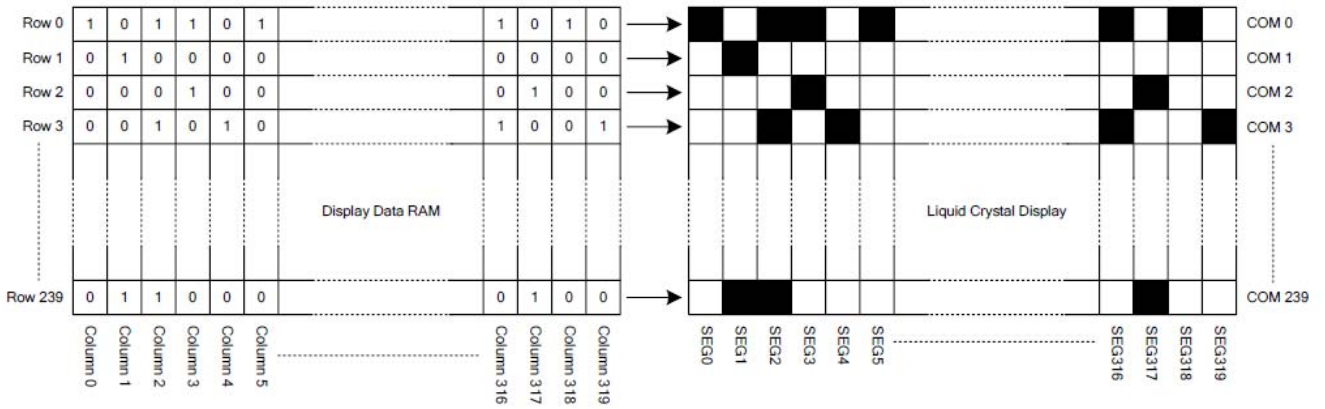


Figure 15 DDRAM Mapping

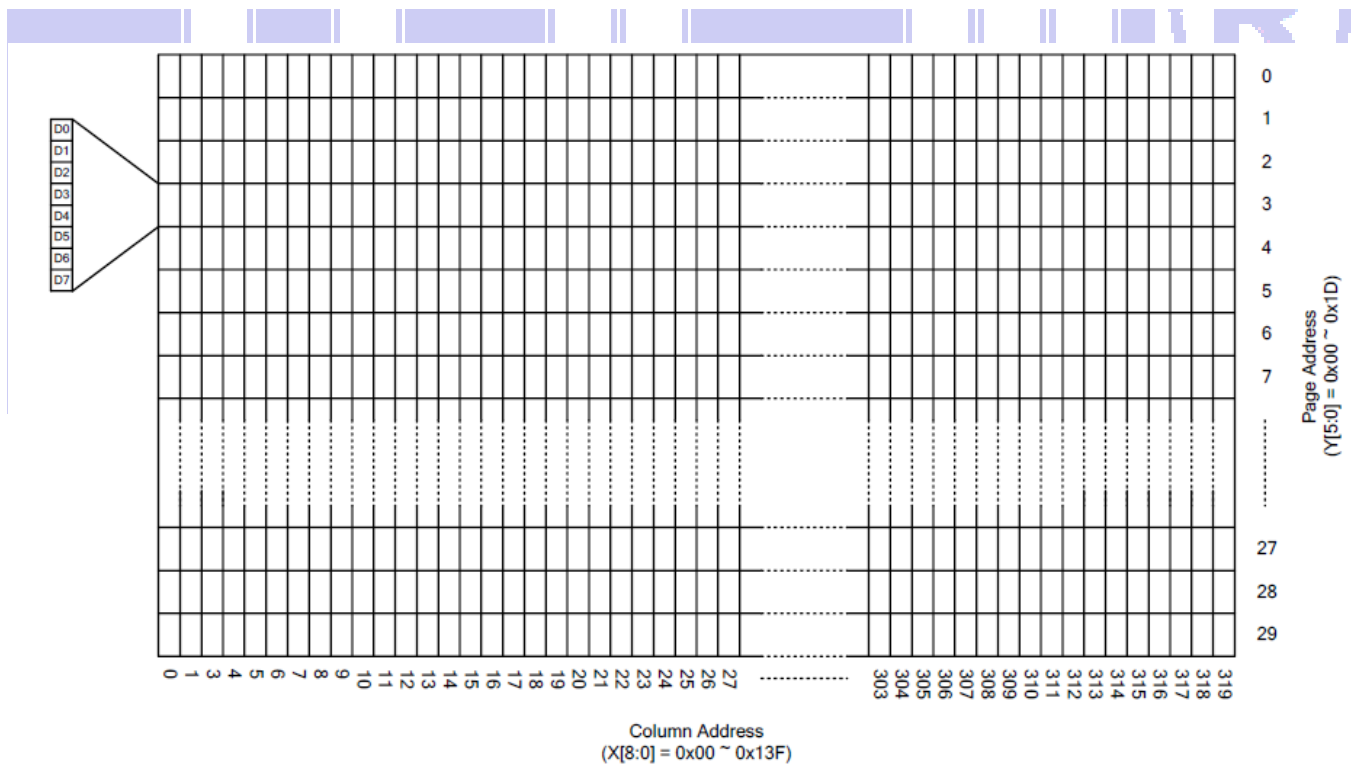


Figure 16 DDRAM Format

下图摘自 ST75320 IC 资料, 可通过 “ST75320.PDF” 之第 29 页获取最佳效果。

The relation between DDRAM and outputs with different MX or MY setting is shown below.

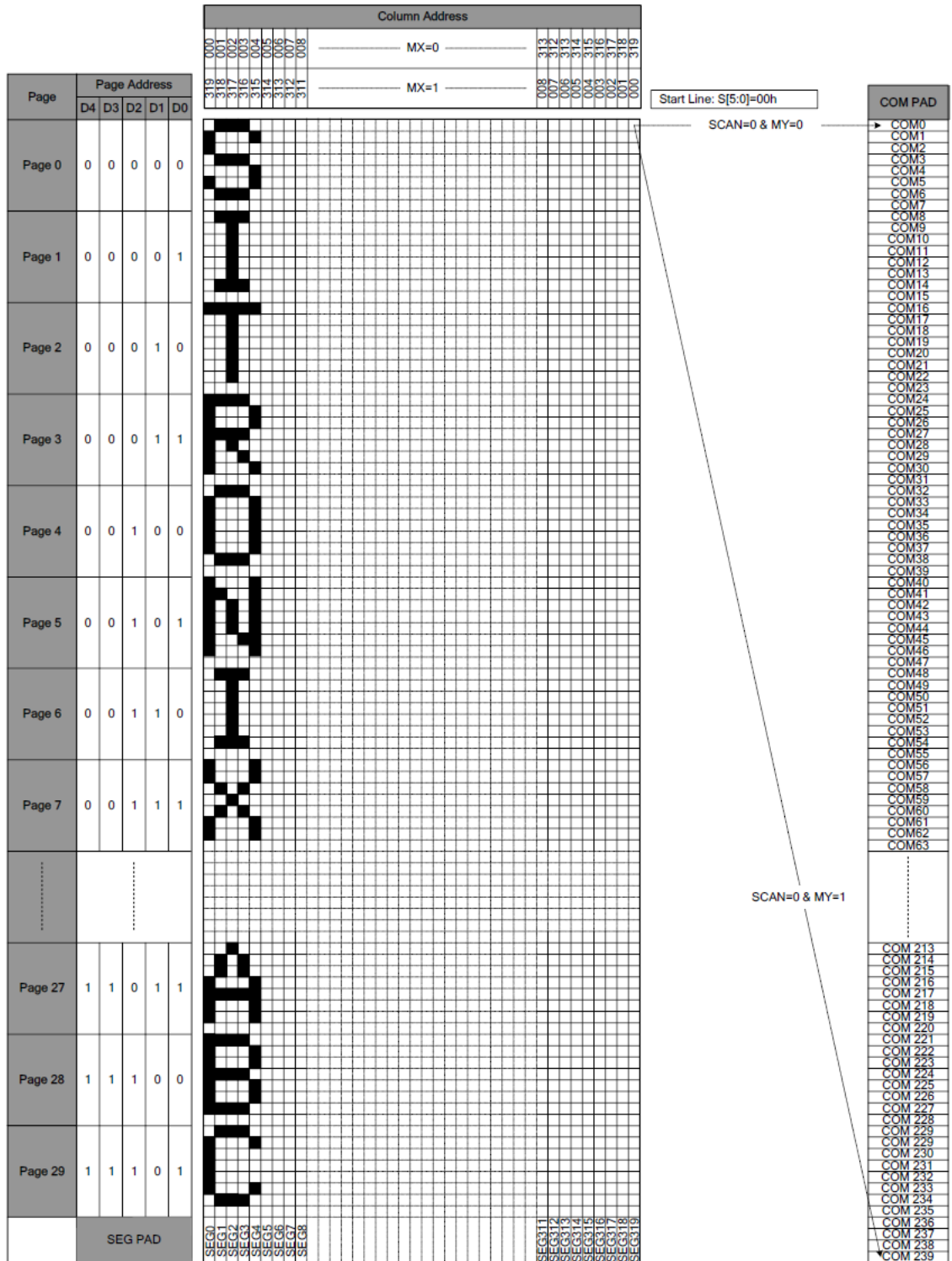


Figure 17 DDRAM Display Direction (Normal Scan)

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

点亮液晶模块的步骤

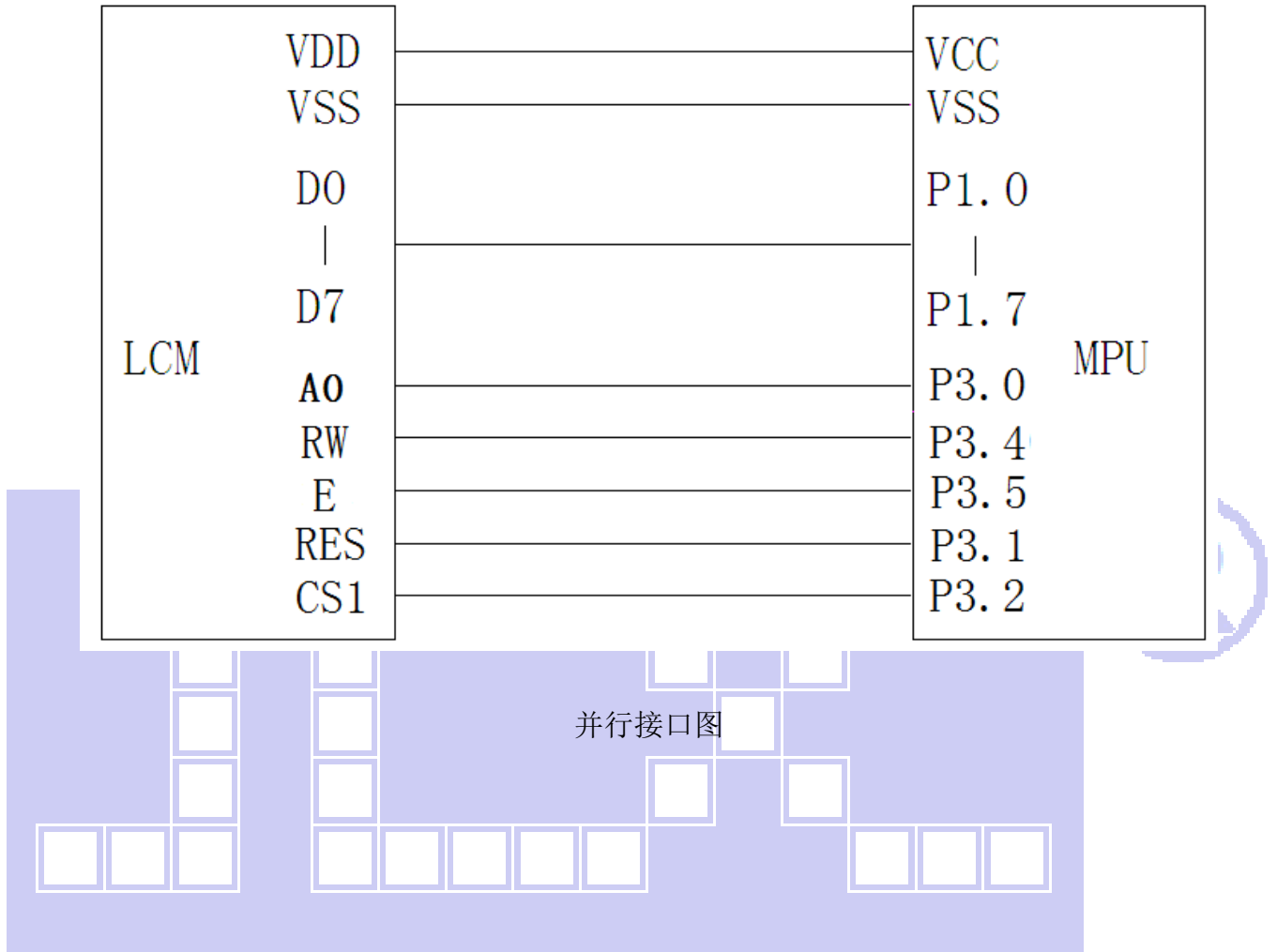
硬件准备:
开发板 (或专门设计的主板)、单片机、电源、连接线、仿真器或程序下载器 (又名烧录器)

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

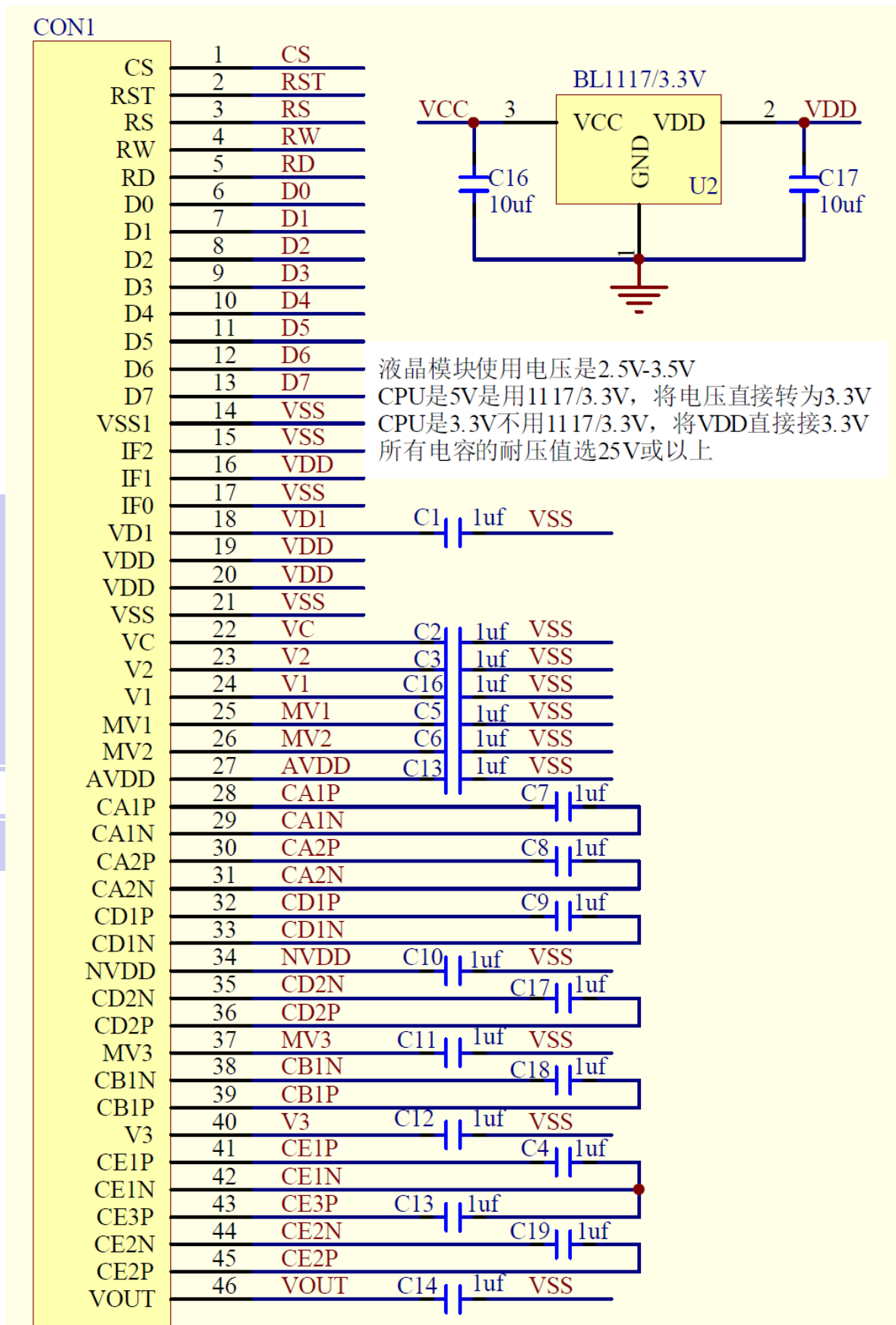
编写软件
背光给合适的直流电可以点亮, 但液晶屏里面没有程序, 只给电不能让液晶屏显示 (我们通常说“点亮”), 程序须另外编写, 并烧录 (下载) 到单片机里液晶模块才能工作。

7.4 接口方式及程序:

6.3.1 液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:



并行原理图



/* 液晶模块型号: JLX320240G-913, 并行接口

驱动 IC 是: ST75320

版权所有: 晶联讯电子: 网址 <http://www.jlxlcd.cn>;

```

*/
#include<reg51.h>
#include<intrins.h>
#include<chinese.h>
//=====
sbit RES=P3^1;
sbit CS1=P3^2;
sbit A0=P3^0;
sbit E=P3^5;
sbit RW=P3^4;
sbit key=P2^0;
//=====
void transfer_command_lcd(int data1)
{
    A0=0;
    E=0;
    RW=0;
    CS1=0;
    E=1;
    P1=data1;
    E=0;
    CS1=1;
}
//=====
void transfer_data_lcd(int data1)
{
    A0=1;
    E=0;
    RW=0;
    CS1=0;
    E=1;
    P1=data1;
    E=0;
    CS1=1;
}

/*等待一个按键，我的主板是用 P2.0 与 GND 之间接一个按键*/
void waitkey()
{
    repeat:
        if (key==1) goto repeat;
        else delay(3000);
}

```



```
/*延时: 1 毫秒的 i 倍*/
```

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

```
/*延时: 1us 的 i 倍*/
```

```
void delay_us(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<1;k++);
}
```

```
//=====
```

```
void initial_lcd()
```

```
{
    RES=1;
    RES=0;
    delay(20);
    RES=1;
    delay(200);

    transfer_command_lcd(0xAE); //Display OFF
    transfer_command_lcd(0xEA); //Power Discharge Control
    transfer_data_lcd(0x00); //Discharge OFF
    transfer_command_lcd(0xA8); //sleep out

    transfer_command_lcd(0xAB); //OSC ON
    transfer_command_lcd(0x69); //Temperature Detection ON
    transfer_command_lcd(0x4E); //TC Setting
    transfer_data_lcd(0xff); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x44); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x12); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x11); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x11); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x11); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x22); //0mV/° C, should be adjusted by customer
    transfer_data_lcd(0x23); //0mV/° C, should be adjusted by customer

    transfer_command_lcd(0x39); //TC Flag
    transfer_data_lcd(0x00); //
    transfer_data_lcd(0x00); //
}
```



```

transfer_command_lcd(0x2B); //Frame Rate Level
transfer_data_lcd(0x00);    //

transfer_command_lcd(0x5F); //Set Frame Frequency
transfer_data_lcd(0x66);    //fFR=80Hz in all temperature range
transfer_data_lcd(0x66);    //should be adjusted by customer

transfer_command_lcd(0xEC); //FR Compensation Temp. Range
transfer_data_lcd(0x19);    //TA = -15 degree, should be adjusted by customer
transfer_data_lcd(0x64);    //TB = 60 degree, should be adjusted by customer
transfer_data_lcd(0x6e);    //TC = 70 degree, should be adjusted by customer

transfer_command_lcd(0xED); //Temp. Hysteresis Value (thermal sensitivity)
transfer_data_lcd(0x04);    //Vop threshold: +2° C
transfer_data_lcd(0x04);    //fFR threshold: +4° C
    
```

```

transfer_command_lcd(0xA6); //Display Inverse OFF
transfer_command_lcd(0xA4); //Disable Display All Pixel ON
    
```

```

transfer_command_lcd(0xC4); //COM Output Status
transfer_data_lcd(0x02);    //Interlace mode, MY=0

transfer_command_lcd(0xA1); //Column Address Direction: MX=0

transfer_command_lcd(0x6D); //Display Area
transfer_data_lcd(0x07);    //Duty = 1/240 duty
transfer_data_lcd(0x00);    //Start Group = 1

transfer_command_lcd(0x84); //Display Data Input Direction: Column
    
```

```

transfer_command_lcd(0x36); //Set N-Line
transfer_data_lcd(0x1e);    //N-Line
    
```

```

transfer_command_lcd(0xE4); //N-Line On
    
```

```

transfer_command_lcd(0xE7); //LCD Drive Method
transfer_data_lcd(0x19);    //NLFR=1//
    
```

```
//-----正常 VOP-----//
```

```

transfer_command_lcd(0x81); //0X81:设置对比度 Set EV=64h
transfer_data_lcd(0x4f);    //微调对比度, 范围 0X00-0XFF
transfer_data_lcd(0x01);    //粗调对比度, 范围 0X00-0X03
    
```

```

transfer_command_lcd(0xA2); //BIAS
transfer_data_lcd(0x0a);    //1/16 BIAS
    
```



```

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x20);    //AVDD ON
delay(10);

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x60);    //AVDD, MV3 & NAVDD ON
delay(10);

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x70);    //AVDD, MV3, NAVDD & V3 ON
delay(10);

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x78);    //AVDD, MV3, NAVDD, V3 & VPF ON
delay(10);

```

```

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x7c);    //AVDD, MV3, NAVDD, V3, VPF & VNF ON
delay(10);

```

```

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x7e);    //VOUT, AVDD, MV3, NAVDD, V3, VPF & VNF ON
delay(10);

```

```

transfer_command_lcd(0x25); //Power Control
transfer_data_lcd(0x7f);    //VOUT, AVDD, MV3, NAVDD, V3, VPF & VNF ON
delay(10);

```

```

clear_screen();
transfer_command_lcd(0xaf); //Display ON

```

```

}

```

```

void lcd_address(int y, int x)
{
    transfer_command_lcd(0xb1);
    transfer_data_lcd(y);
    transfer_command_lcd(0x13);
    transfer_data_lcd((x>>8)&0x0001);
    transfer_data_lcd(x);    //
    transfer_command_lcd(0x1d);
}

```

```

//=====
void clear_screen()

```



```

{
    int i, j;
    for(i=0;i<30;i++)
    {
        for(j=0;j<320;j++) //
        {
            lcd_address(i, j);
            transfer_data_lcd(0x00);
        }
    }
}

```

```

void display_string_8x16(uint page, uint 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_lcd(ascii_table_8x16[j][k+8*n]);/*显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x
                    为列地址, 最后为数据*/
                }
            }
            i++;
            column+=8;
        }
        else
            i++;
    }
}

```

//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>320)
    {
        column=0;
        page+=2;
    }
    if(address !=1)
    {
        for(k=0;k<2;k++)
        {
            for(i=0;i<16;i++)
            {
                lcd_address(page+k, column+i);
                transfer_data_lcd(Chinese_code_16x16[address]);
                address++;
            }
        }
        j +=2;
    }
    else
    {
        for(k=0;k<2;k++)
        {
            for(i=0;i<16;i++)
            {
                lcd_address(page+k, column+i);
                transfer_data_lcd(0x00);
            }
        }
        j++;
    }
}

```



```

        column+=16;
    }
}

```

//写入一组 32x32 点阵的汉字字符串（字符串表格中需含有此字）

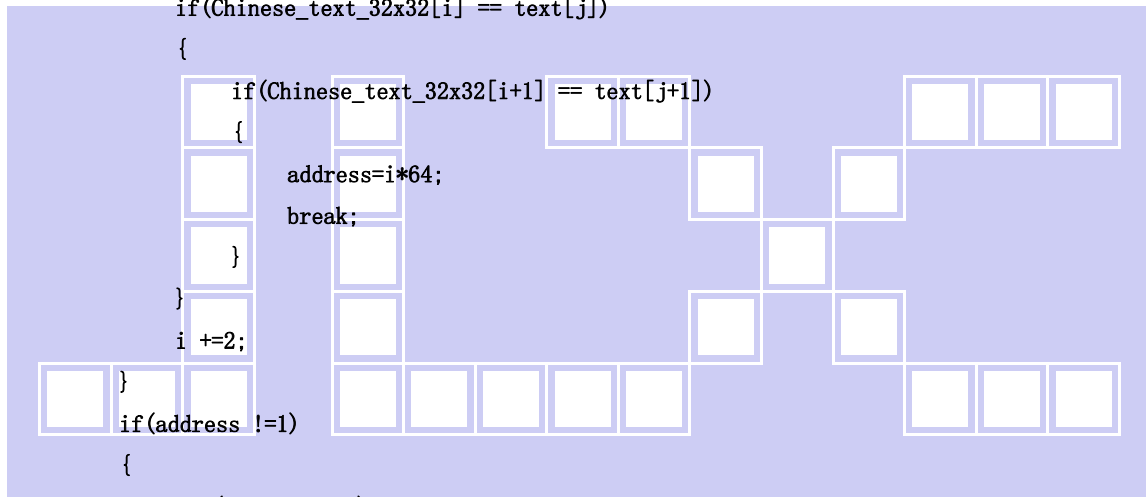
//括号里的参数：（页，列，汉字字符串）

```
void display_string_32x32(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_32x32[i]> 0x7e)
        {
            if(Chinese_text_32x32[i] == text[j])
            {
                if(Chinese_text_32x32[i+1] == text[j+1])
                {
                    address=i*64;
                    break;
                }
            }
            i +=2;
        }
        if(address !=1)
        {
            for(k=0;k<4;k++)
            {
                for(i=0;i<32;i++)
                {
                    lcd_address(page+k, column+i);
                    transfer_data_lcd(Chinese_code_32x32[address]);
                    address++;
                }
            }
            j +=2;
        }
        else
        {
            for(k=0;k<4;k++)
            {
                for(i=0;i<32;i++)

```



```

        {
            lcd_address(page+k, column+i);
            transfer_data_lcd(0x00);
        }
    }
    j++;
}
column+=32;
}
}

```

```
void test(uchar data1, uchar data2)
```

```

{
    int i, j;
    for(i=0; i<30; i++)
    {
        for(j=0; j<320; j++)
        {
            lcd_address(i, j);
            transfer_data_lcd(data1);
            transfer_data_lcd(data2);
        }
    }
}

```

```
//=====
```

```

void showpic(char *k)
{
    int i, j;
    for(i=0; i<30; i++)
    {
        for(j=0; j<320; j++) //
        {
            lcd_address(i, j);
            transfer_data_lcd(*k++);
        }
    }
}

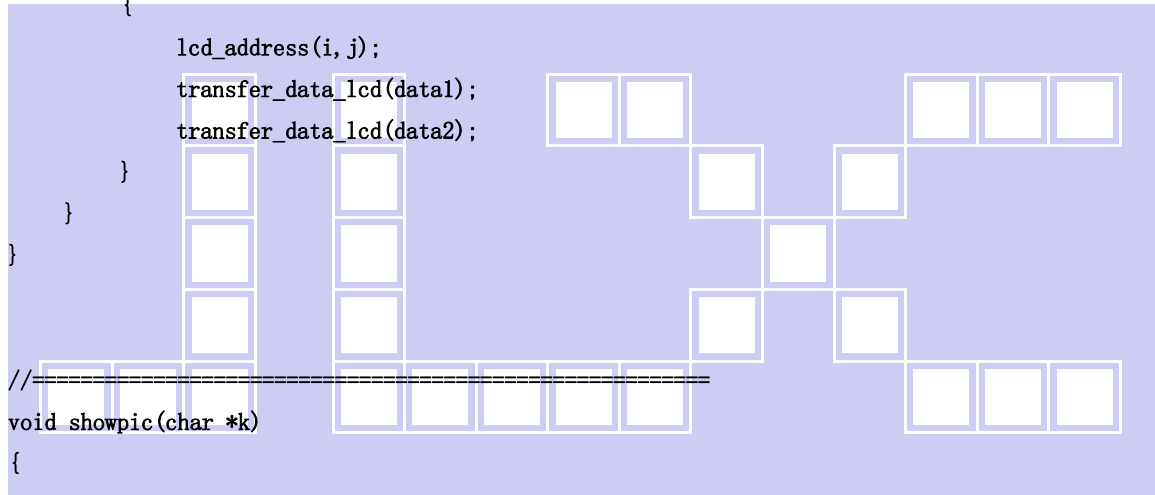
```

```
//=====
```

```

void main()
{
    initial_lcd();          //对液晶模块进行初始化设置;
    while(1)

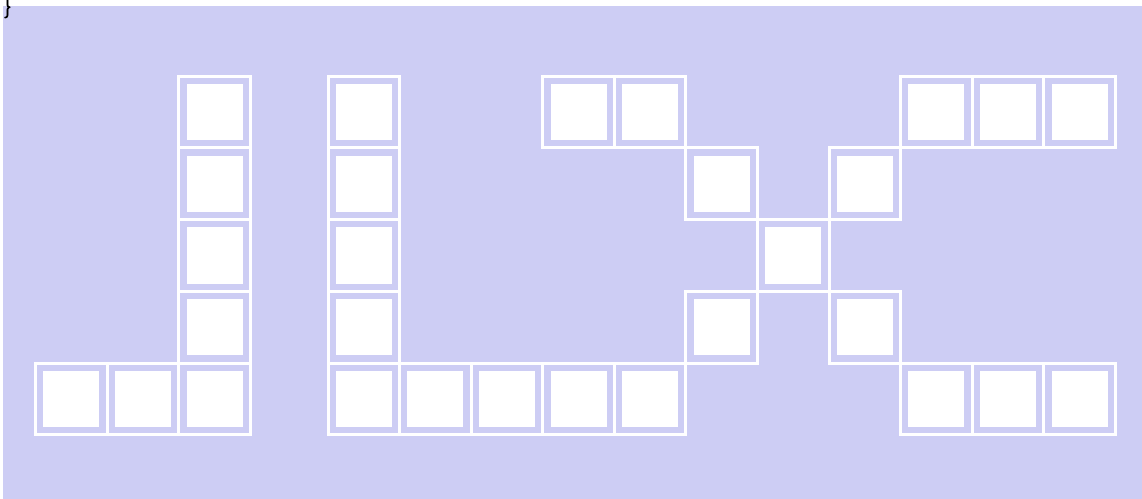
```



```

{
    clear_screen();
    display_string_8x16(4, 104, "JLX320240G-913");
    display_string_16x16(8, 63, "深圳市晶联讯电子有限公司");
    display_string_32x32(12, 31, "深圳市晶联讯电子");
    waitkey();
    showpic(bmp1);
    waitkey();
    showpic(bmp2);
    test(0x55, 0xaa);
    waitkey();
    test(0xaa, 0x55);
    waitkey();
    test(0xff, 0xff);
    waitkey();
}
}

```



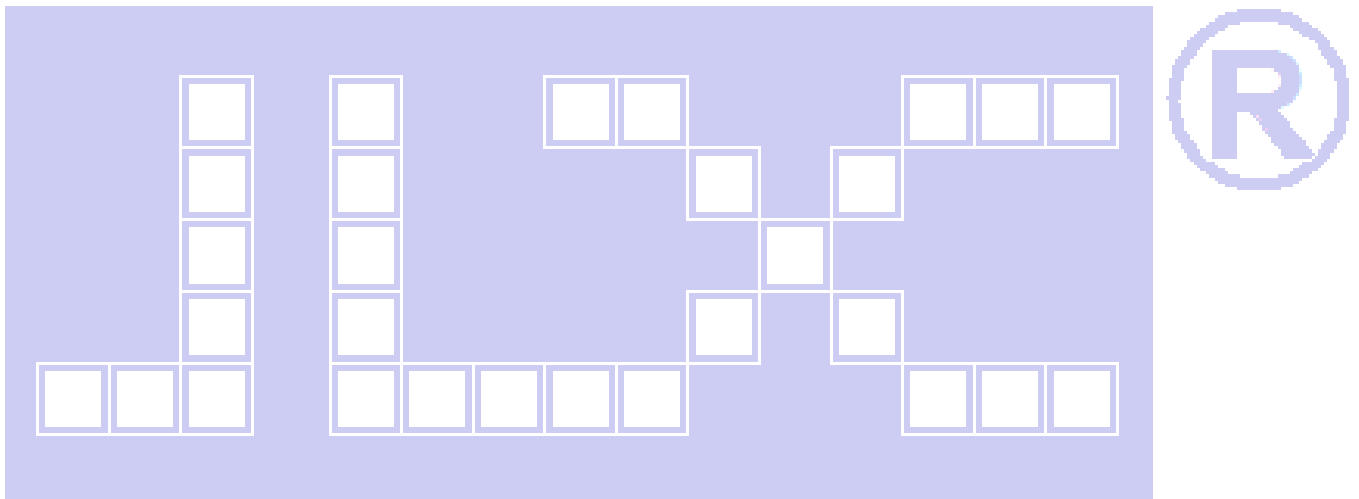
7.4 程序举例:

7.4.1 串行接口

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:



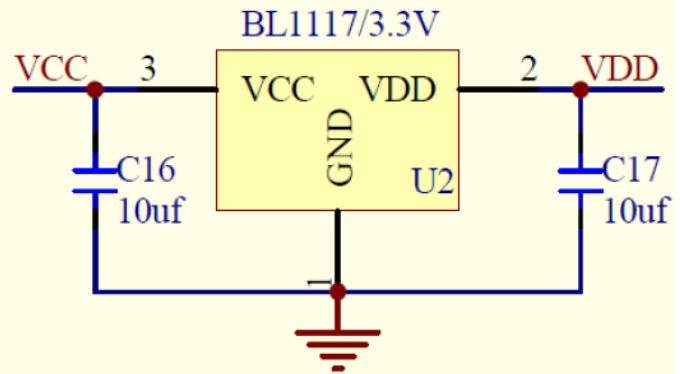
图 8. 串行接口



串行接口原理图

CON1

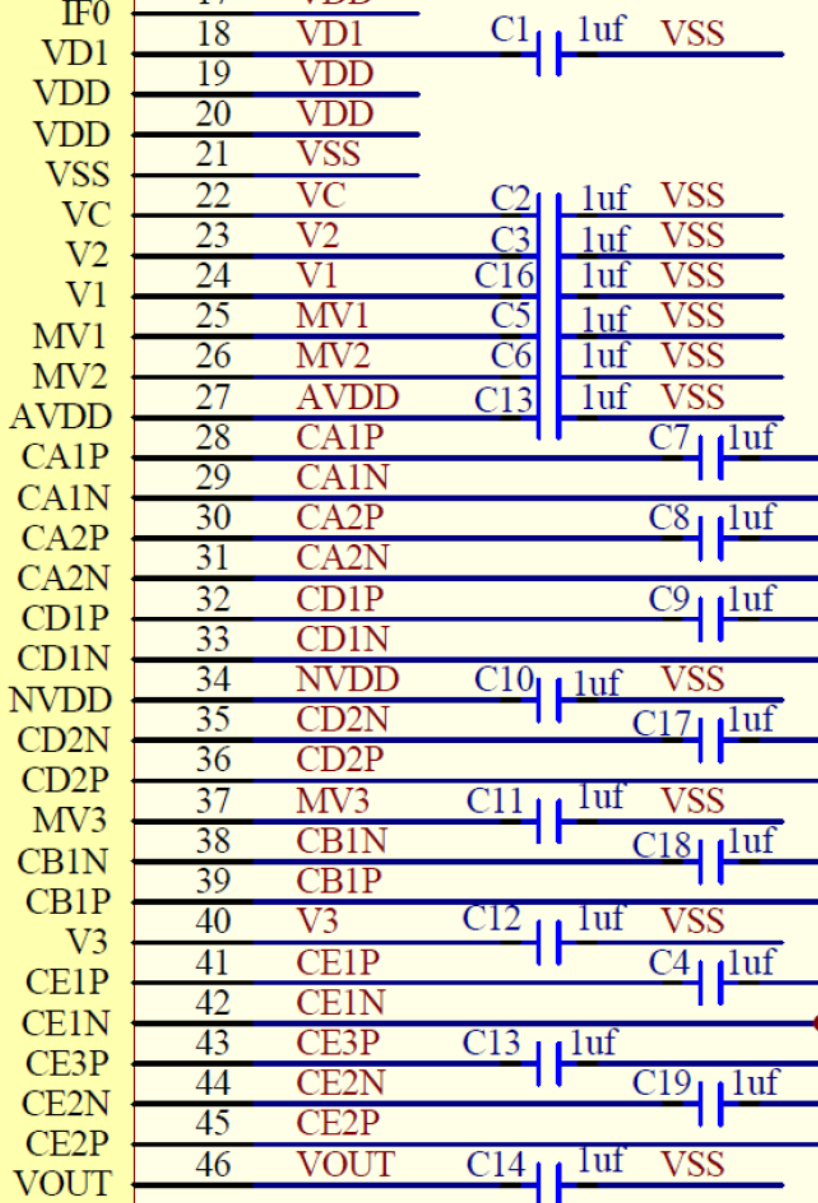
CS	1	CS
RST	2	RST
RS	3	RS
RW	4	VDD
RD	5	VDD
D0	6	SCLK
D1	7	SDA
D2	8	SDA
D3	9	SDA
D4	10	VDD
D5	11	VDD
D6	12	VDD
D7	13	VDD
VSS1	14	VSS
IF2	15	VSS
IF1	16	VSS
IF0	17	VDD
VD1	18	VD1
VDD	19	VDD
VDD	20	VDD
VSS	21	VSS
VC	22	VC
V2	23	V2
V1	24	V1
MV1	25	MV1
MV2	26	MV2
AVDD	27	AVDD
CA1P	28	CA1P
CA1N	29	CA1N
CA2P	30	CA2P
CA2N	31	CA2N
CD1P	32	CD1P
CD1N	33	CD1N
NVDD	34	NVDD
CD2N	35	CD2N
CD2P	36	CD2P
MV3	37	MV3
CB1N	38	CB1N
CB1P	39	CB1P
V3	40	V3
CE1P	41	CE1P
CE1N	42	CE1N
CE3P	43	CE3P
CE2N	44	CE2N
CE2P	45	CE2P
VOUT	46	VOUT



液晶模块使用电压是2.5V-3.5V

CPU是5V是用1117/3.3V，将电压直接转为3.3V

CPU是3.3V不用1117/3.3V，将VDD直接接3.3V



并行程序与串行只是接口定义、写数据和命令不一样，其它都一样

串行接口程序

```
sbit lcd_cs1 = P3^2;//CS
sbit lcd_reset= P1^1;//RST
sbit lcd_sclk = P1^2;//串行时钟
sbit lcd_rs = P3^1;//RS
sbit lcd_sid = P1^3;//串行数据
sbit key      = P2^0; //按键
```

//=====transfer command to LCM=====

```
void transfer_command_lcd(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;
        data1=data1<<=1;
    }
}
```

//-----transfer data to LCM-----

```
void transfer_data_lcd(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=data1<<=1;
    }
}
```



7.5、IIC 接口

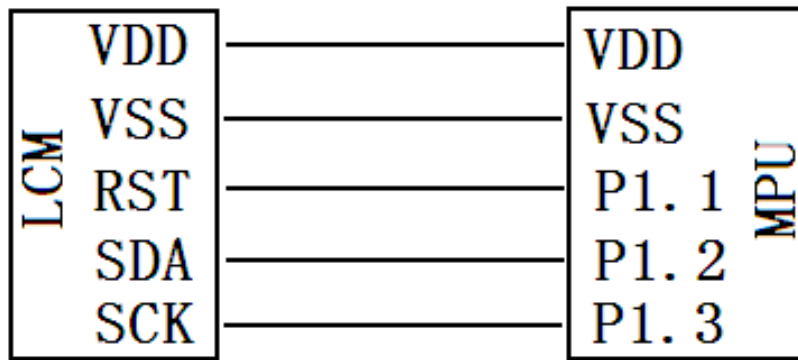
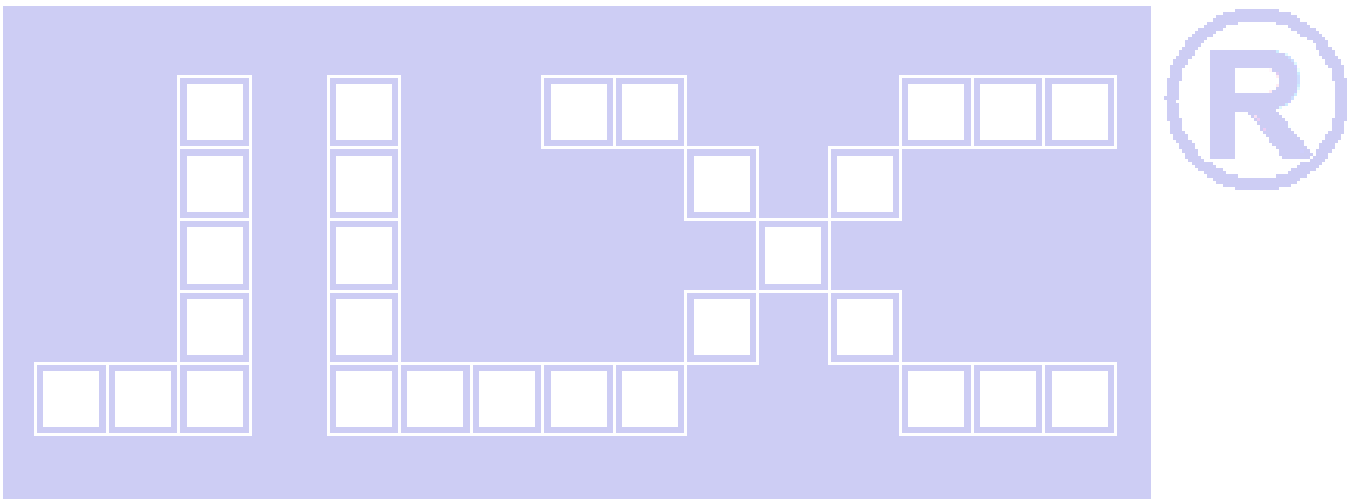


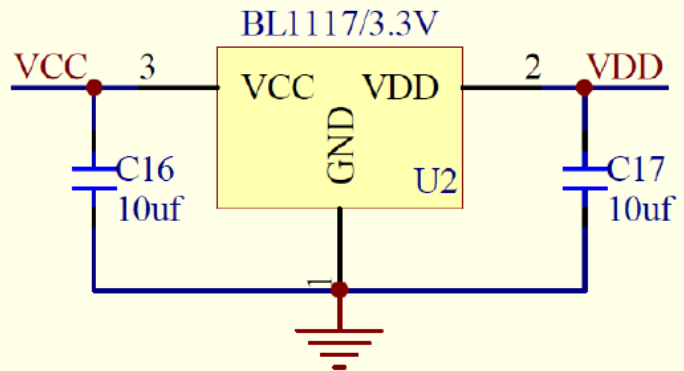
图 10. IIC

IIC 接口原理图



CON1

CS	1	VDD
RST	2	RST
RS	3	VDD
RW	4	VDD
RD	5	VDD
D0	6	SCLK
D1	7	SDA
D2	8	SDA
D3	9	SDA
D4	10	SDA
D5	11	SDA
D6	12	VDD
D7	13	VDD
VSS1	14	VSS
IF2	15	VDD
IF1	16	VDD
IF0	17	VSS
VD1	18	VD1
VDD	19	VDD
VDD	20	VDD
VSS	21	VSS
VC	22	VC
V2	23	V2
V1	24	V1
MV1	25	MV1
MV2	26	MV2
AVDD	27	AVDD
CA1P	28	CA1P
CA1N	29	CA1N
CA2P	30	CA2P
CA2N	31	CA2N
CD1P	32	CD1P
CD1N	33	CD1N
NVDD	34	NVDD
CD2N	35	CD2N
CD2P	36	CD2P
MV3	37	MV3
CB1N	38	CB1N
CB1P	39	CB1P
V3	40	V3
CE1P	41	CE1P
CE1N	42	CE1N
CE3P	43	CE3P
CE2N	44	CE2N
CE2P	45	CE2P
VOUT	46	VOUT

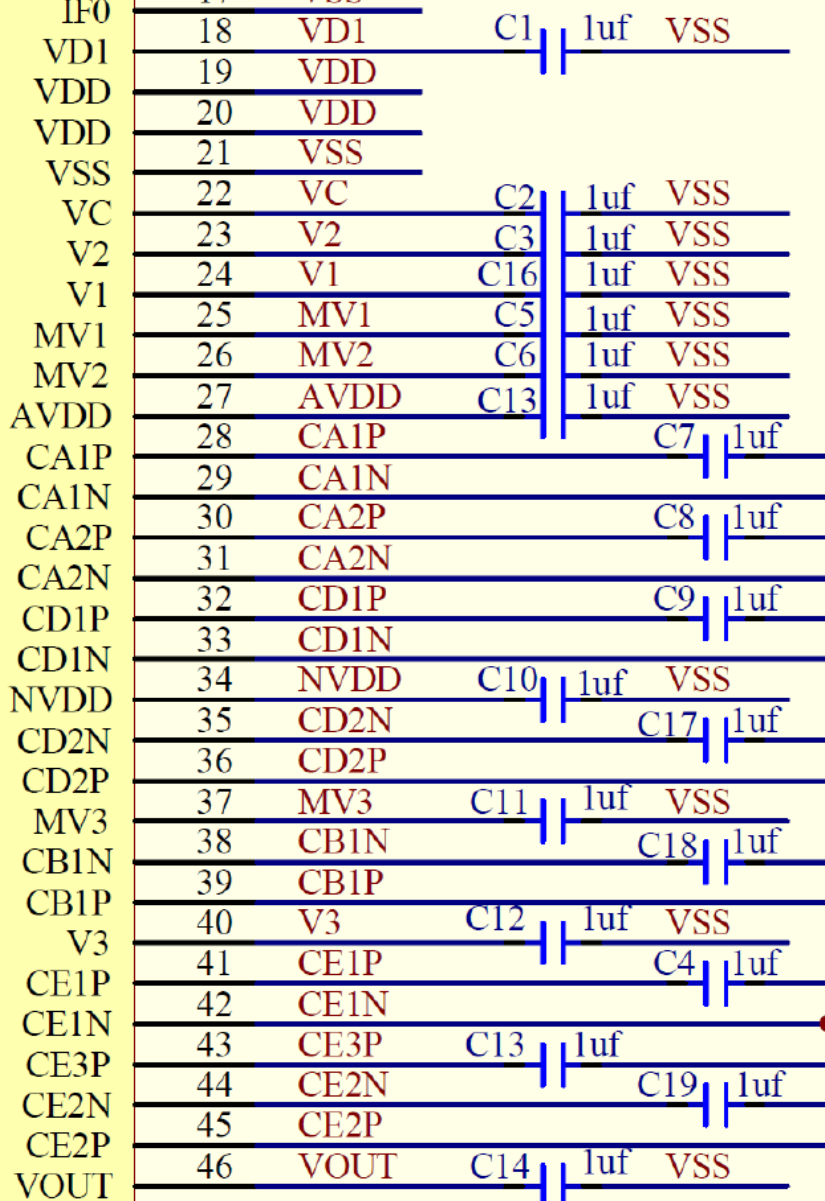


液晶模块使用电压是2.5V-3.5V

CPU是5V是用1117/3.3V, 将电压直接转为3.3V

CPU是3.3V不用1117/3.3V, 将VDD直接接3.3V

所有电容的耐压值选25V或以上



7.5.1、以下为 I2C 接口方式范例程序

与串行方式相比较，只需改变接口顺序以及传送数据、传送命令这两个函数即可：

```

/* 液晶模块型号：JLX320240G-913
   IIC 接口
   驱动 IC 是：ST75320
   版权所有：晶联讯电子：网址 http://www.jlxlcd.cn;
*/

```

```

#include <reg52.H>
#include <intrins.h>
#include <chinese_code.h>

```

```

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

```

```

void transfer(int data1)

```

```

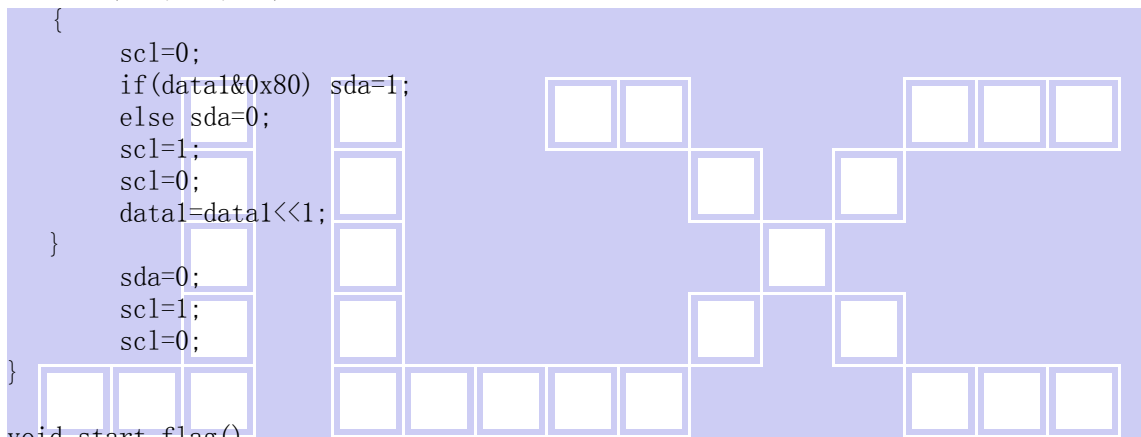
{
    int i;
    for(i=0;i<8;i++)

```

```

    {
        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*/
}

```

//写命令到液晶显示模块

```

void transfer_command(uchar com)

```

```

{
    start_flag();
    transfer(0x78);
    transfer(0x80);
    transfer(com);
    stop_flag();
}

```



```
//写数据到液晶显示模块  
void transfer_data(uchar dat)  
{  
    start_flag();  
    transfer(0x78);  
    transfer(0xC0);  
    transfer(dat);  
    stop_flag();  
}
```

-END-

