

JLX16080G-370-BN 使用说明书

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	5
5	技术参数	6
6	时序特性	6~10
7	指令功能及硬件接口与编程案例	10~尾页

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 JLX16080G-370 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。

JLX16080G-370 可以显示 160 列*80 行点阵单色图片，或显示 16*16 点阵的汉字 10 个*5 行，或显示 8*16 点阵的英文、数字、符号 20 个*5 行。或显示 5*8 点阵的英文、数字、符号 26 个*10 行。

2. JLX16080G-370 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙，焊接式 FPC。

2.2 IC 采用矽创公司 ST75256, 功能强大，稳定性好

2.3 功耗低：不带背光 1mW (3.3V*0.3mA)，带背光不大于 150mW (3.3V*45mA)；

2.4 显示内容：

(1) 160*80 点阵单色图片，或其它小于 160*80 点阵的单色图片；

(2) 可选用 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 10 字*5 行；

(3) 按照 12*12 点阵汉字来计算可显示 13 字*6 行；

(4) 按照 8*16 点阵汉字来计算可显示 20 字*5 行；

(5) 按照 5*8 点阵汉字来计算可显示 26 字*10 行；

2.5 指令功能强：可软件调对比度、行列扫描方向可改（可旋转 180 度使用）；

并口时：可以“读-改-写”；

2.6 接口简单方便：采用 4 线 SPI 串行接口，或选择并口（6800 时序和 8080 时序可选），或 I2C（IIC）接口。

2.7 工作温度宽：-20℃ - 70℃；

2.7 储存温度宽：-30℃ - 80℃；

3.1 外形尺寸及接口引脚功能

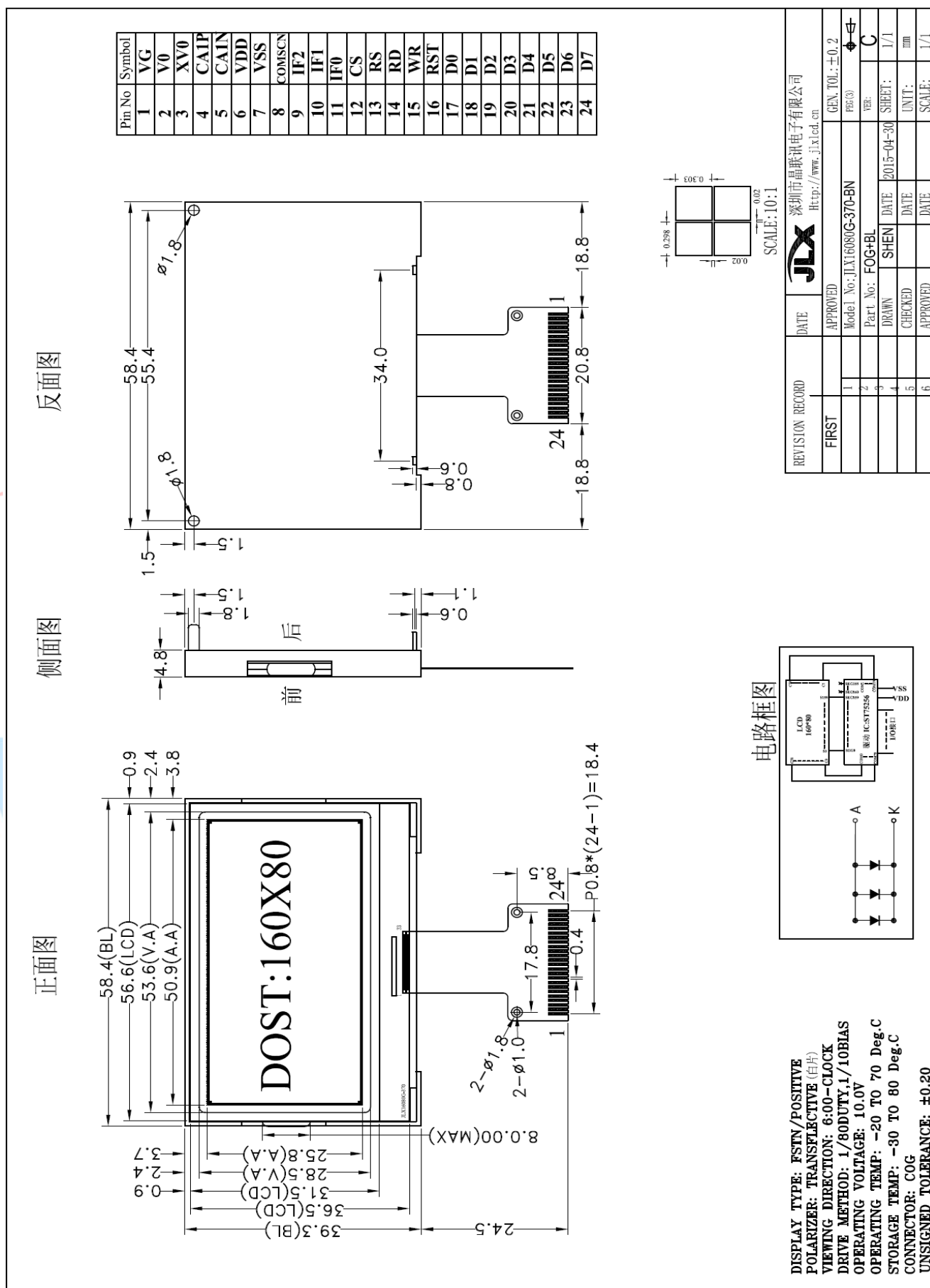


图 1. 外形尺寸

模块的接口引脚功能

3.2 模块的接口引脚功能

3.2.1 并行时接口引脚功能

引 线 号	符 号	名 称	功 能
1	VG	偏压电路	LCD 偏置驱动电压, VG 与 VSS 之间接一个电容
2	V0	倍压电路	V0 与 XV0 之间接一个电容
3	XV0	倍压电路	
4	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
5	CA1N	倍压电路	
6	VDD	电源电路	供电电源正极
7	VSS	接地	0V
8	COMSCN	镜像选择指令	VDD
9	IF2	I	L: 接低电平
10	IF1	I	H: 接高电平
11	IF0	I	L: 接低电平
12	CS	片选	低电平片选
13	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”)
14	E (RD)	使能信号	6800 时序: 使能信号
15	RW (WR)	读/写	6800 时序: H: 读数据 0: 写数据
16	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
17~24	D0~D7	I/O	并行接口时, 数据总线 DB0~DB7

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

引 线 号	符 号	名 称	功 能
1	VG	偏压电路	LCD 偏置驱动电压, VG 与 VSS 之间接一个电容
2	V0	倍压电路	V0 与 XV0 之间接一个电容
3	XV0	倍压电路	
4	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
5	CA1N	倍压电路	
6	VDD	电源电路	供电电源正极
7	VSS	接地	0V
8	COMSCN	镜像选择指令	VDD
9	IF2	I	L: 接低电平
10	IF1	I	L: 接低电平
11	IF0	I	L: 接低电平
12	CS	片选	低电平片选
13	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”)
14	E (RD)	使能信号	串行接口, RD 接高电平
15	RW (WR)	读、写	串行接口, RW 接高电平
16	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
17	D0 (SCK)	I/O	串行时钟
18~20	D1 ~ D3 (SDA)	I/O	串行数据
21~24	D4~D7	I/O	串行接口, D4~D7 引脚建议接 VDD

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	VG	偏压电路	LCD 偏置驱动电压，VG 与 VSS 之间接一个电容
2	V0	倍压电路	V0 与 XV0 之间接一个电容
3	XV0	倍压电路	
4	CA1P	倍压电路	CA1P 与 CA1N 之间接一个电容
5	CA1N	倍压电路	
6	VDD	电源电路	供电电源正极
7	VSS	接地	0V
8	COMSCN	镜像选择指令	VDD
9	IF2	I	L:接低电平
10	IF1	I	L:接低电平
11	IF0	I	H:接低电平
12	CS	片选	I2C 接口，此引脚接 VSS
13	A0(RS)	寄存器选择信号	I2C 接口，此引脚接高电平
14	E(RD)	使能信号	I2C 接口，不用，此引脚接高电平
15	RW(WR)	读、写	I2C 接口，不用，此引脚接高电平
16	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
17	D0(SCK)	I/O	串行时钟
18~20	D1 ~ D3 (SDA)	I/O	串行数据
21~22	D4-D5	I/O	I2C 接口，D4-D5 引脚接 VDD
23~24	D6-D7	I/O	I2C 接口，D6-D7 是从属地址接 VSS

表 3: I²C 总线接口引脚功能

4. 基本原理

4.1 液晶屏 (LCD)

在 LCD 上排列着 160×80 点阵,160 个列信号与驱动 IC 相连,80 个行信号也与驱动 IC 相连,IC 邦定在 LCD 玻璃上 (这种加工工艺叫 COG) .

4.2 工作电路:

图 2 是 JLX16080G-370 图像点阵型模块的电路框图,它由驱动 IC ST75256 及几个电阻电容组成。
电路框图

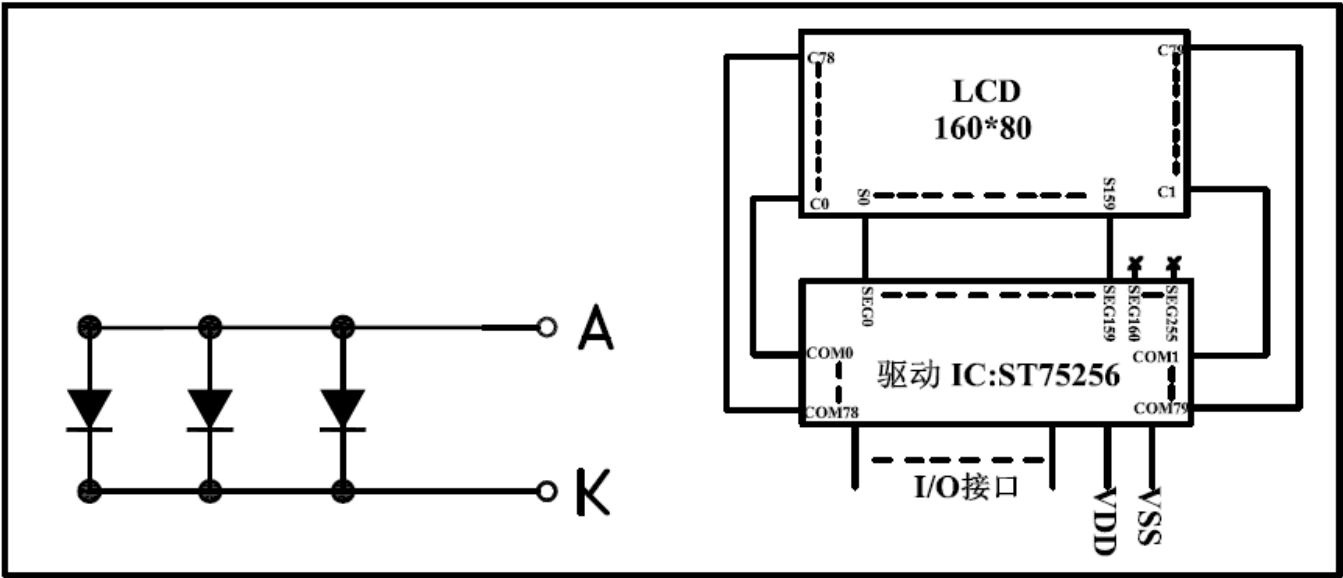


图 2: JLX16080G-370 图像点阵型液晶模块的电路框图

4.2 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: $-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$;

存储温度: $-30^{\circ}\text{C} \sim +80^{\circ}\text{C}$;

背光板可选择绿色、白色。

正常工作电流为: $24 \sim 60\text{mA}$ (LED 灯数共 3 颗);

工作电压: 3.0V ;

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	—	5.5	V
LCD 驱动电压	VDD - V0	-0.3	—	13.5	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	—	2.6	3.3	3.6	V
背光工作电压	VLED	—	2.9	3.0	3.1	V
输入高电平	VIH	—	0.8VDD	—	VDD	V
输入低电平	VIO	—	0	—	0.2VDD	V
输出高电平	VOH	$\text{IOH} = 0.2\text{mA}$	0.8VDD	—	VDD	V
输出低电平	V00	$\text{IOO} = 1.2\text{mA}$	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	24	45	60	mA

表 3: 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)

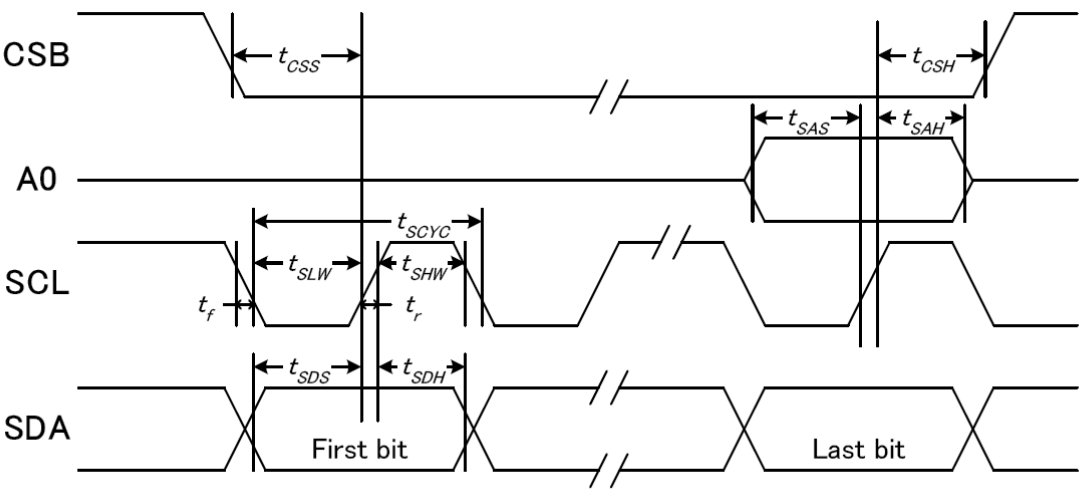


图 3. 从 CPU 写到 ST75256 (Writing Data from CPU to ST75256)

表 7. 写数据到 ST75256 的时序要求

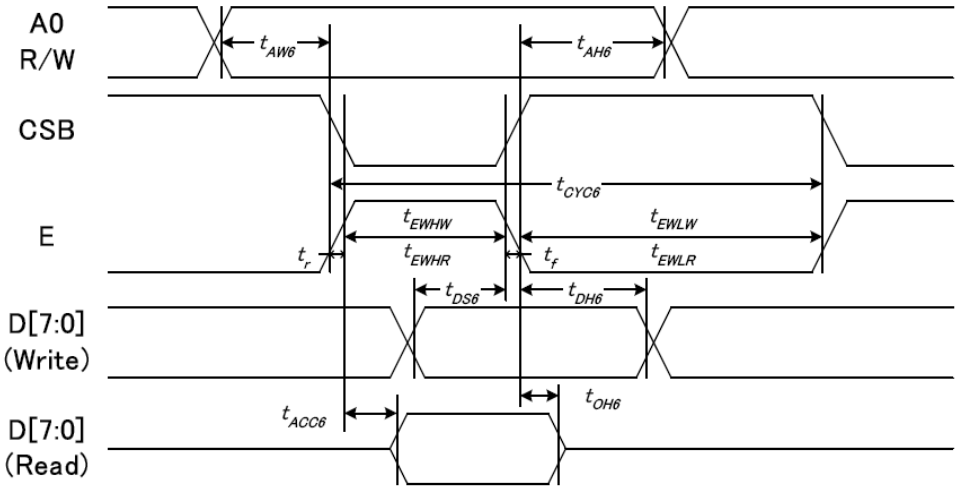
项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
4线 SPI串口时钟周期 (4-line SPI Clock Period)	tSCYC	引脚: SCL	80	---	---	ns
保持SCK高电平脉宽 (SCL "H" pulse width)	tSHW		30	---	---	ns
保持SCLK低电平脉宽 (SCL "L" pulse width)	tSLW		30	---	---	ns
地址建立时间 (Address setup time)	tSAS	引脚: A0	20	---	---	ns
地址保持时间 (Address hold time)	tSAH		20	---	---	ns
数据建立时间 (Data setup time)	tSDS	引脚: SID	20	---	---	ns
数据保持时间 (Data hold time)	tSDH		20	---	---	ns
片选信号建立时间 (CS-SCL time)	tCSS	引脚: CSB	20	---	---	ns
片选信号保持时间 (CS-SCL time)	tCSH		20	---	---	ns

VDD = 1.8~3.3V±5%, Ta = -30~85℃

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 ST75256 (Writing Data from CPU to ST75256)

图 4. 写数据到 ST75256 的时序要求 (6800 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	tAH6	20		---	ns
地址建立时间		tAW6	0		---	ns
系统循环时间	E	tCYC6	160		---	ns
使能“低”脉冲宽度		tEHLW	70		---	ns
使能“高”脉冲宽度		tEHWLW	70		---	ns
写数据建立时间	DB[7: 0]	tDS6	15		---	ns
写数据保持时间		tDH6	15		---	ns

VDD =1.8~3.3V±5%, Ta = -30~85℃

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,

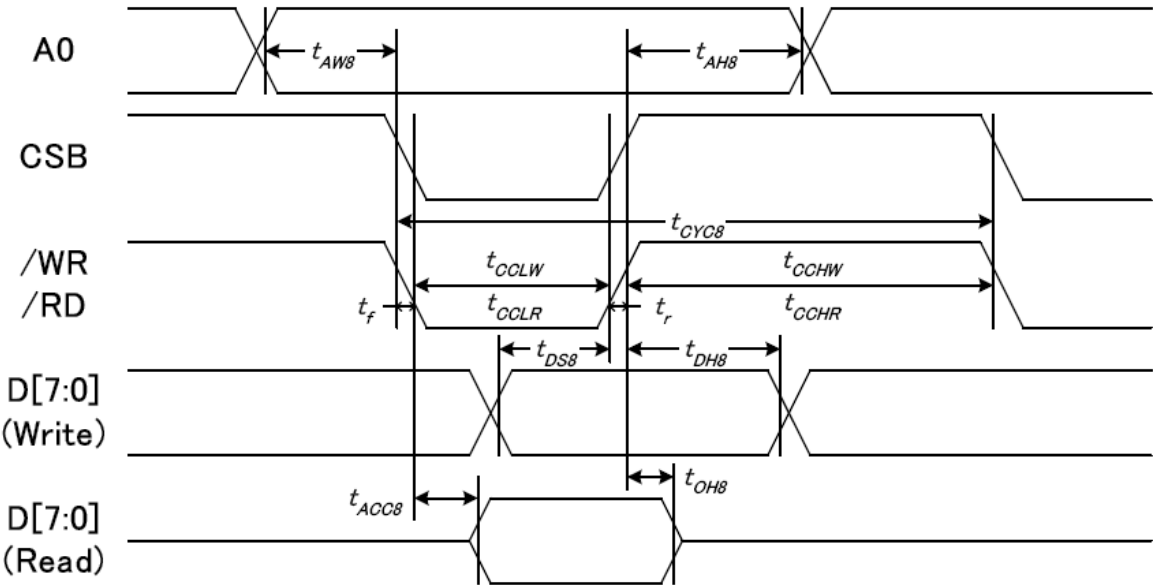
(TR + TF) ≤ (tcyc6 - tewlw - tewhw) 指定。

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 ST75256 (Writing Data from CPU to ST75256)

图 4. 写数据到 ST75256 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	tAH8	20		---	ns
地址建立时间		tAW8	0		---	ns
系统循环时间	/WR	tCYC8	160		---	ns
使能“低”脉冲宽度		tCCLW	70		---	ns
使能“高”脉冲宽度		tCCHW	70		---	ns
写数据建立时间	DB	tDS8	15		---	ns
写数据保持时间		tDH8	15		---	ns

VDD =1.8~3.3V±5%, Ta = -30~85℃

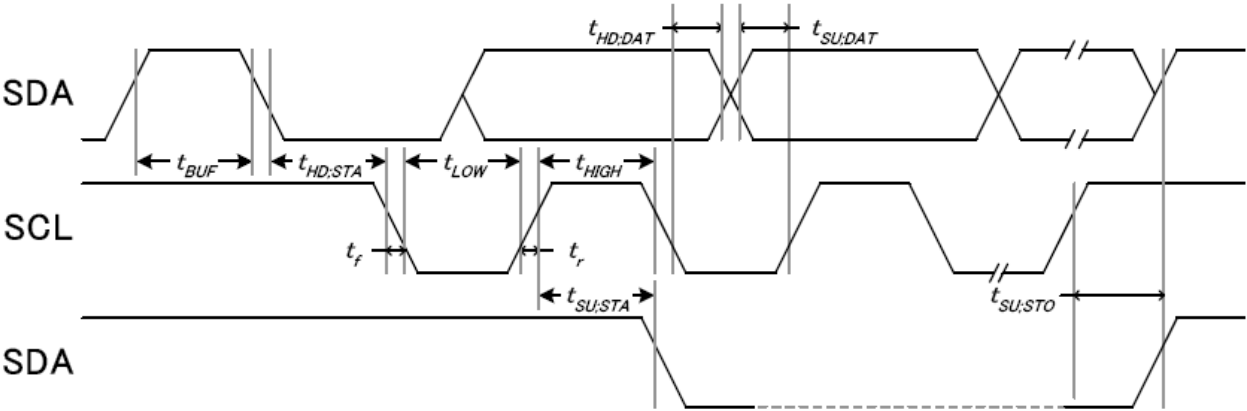
输入信号的上升时间和下降时间（TR，TF）是在 15 纳秒或更少的规定。当系统循环时间非常快，

（TR + TF）≤（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 ST75256 (Writing Data from CPU to ST75256)

图 4. 写数据到 ST75256 的时序要求 (I²C 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	
SCL时钟频率	CSL	FSCLK	---		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）:

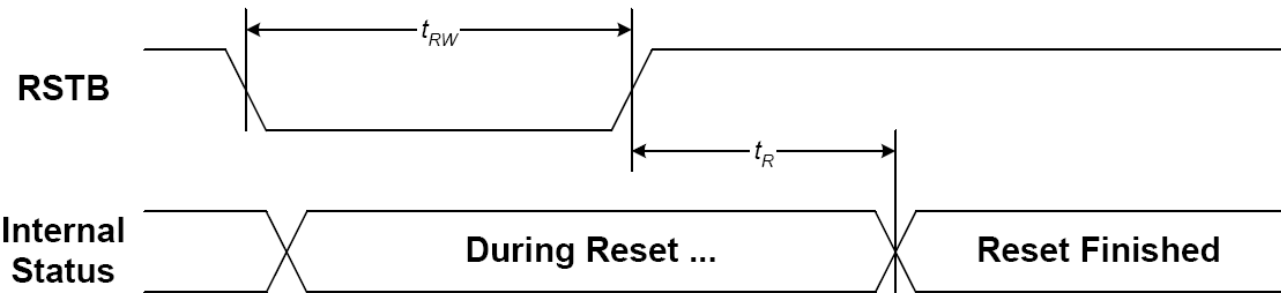


图 5： 电源启动后复位的时序

表 6： 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}		--	--	1	us
复位保持低电平的时间	T_{RD}	引脚： RESET, WR	1	--	--	ms

7. 指令功能:

7.1 指令表

INSTRUCTION	A0	R/W	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
1.Extension Command	0	0	0	0	1	1	EXT1	0	0	EXT0	Set extension instruction
Ext[1:0]=0,0 (Extension Command 1)											
2.Display ON/OFF	0	0	1	0	1	0	1	1	1	DSP	Set LCD display DSP=0: Display off DSP=1: Display on
3.Inverse Display	0	0	1	0	1	0	0	1	1	INV	Set inverse display INV=0: Normal display INV=1: Inverse display
4.All Pixel ON/OFF	0	0	0	0	1	0	0	0	1	AP	Set all pixel on mode AP=0: All pixel off mode AP=1: All pixel on mode
5.Display Control	0	0	1	1	0	0	1	0	1	0	Set display control CLD :Set CL dividing ratio DT[7:0] : Set the number of duty LF[4:0] : Set N-line inversion counter FI : Set the inversion type of frame at the end of common scan cycle
	1	0	0	0	0	0	0	CLD	0	0	
	1	0	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
	1	0	0	0	LF4	FI	LF3	LF2	LF1	LF0	
6.Power Save	0	0	1	0	0	1	0	1	0	SLP	Set power save mode SLP=0: Sleep out mode SLP=1: Sleep in mode
7.Set Page Address	0	0	0	1	1	1	0	1	0	1	Set page address Starting page address: 00h≤YS≤28h Ending page address: YS≤YE≤28h
	1	0	YS7	YS6	YS5	YS4	YS3	YS2	YS1	YS0	
	1	0	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0	
8.Set Column Address	0	0	0	0	0	1	0	1	0	1	Set column address Starting column address: 00h≤XS≤FFh Ending column address: XS≤XE≤FFh
	1	0	XS7	XS6	XS5	XS4	XS3	XS2	XS1	XS0	
	1	0	XE7	XE6	XE5	XE4	XE3	XE2	XE1	XE0	
9.Data Scan Direction	0	0	1	0	1	1	1	1	0	0	Set normal/ inverse display of address and address scan direction
	1	0	0	0	0	0	0	MV	MX	MY	
10.Write Data	0	0	0	1	0	1	1	1	0	0	Write data to DDRAM
	1	0	D7	D6	D5	D4	D3	D2	D1	D0	
11.Read Data	0	0	0	1	0	1	1	1	0	1	Read data from DDRAM (Only for parallel interface and I ² C)
	1	1	D7	D6	D5	D4	D3	D2	D1	D0	
12.Partial In	0	0	1	0	1	0	1	0	0	0	Set partial area Starting partial display address: 00h≤PTS≤A1h Ending partial display address: 00h≤PTE≤A1h
	1	0	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0	
	1	0	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0	

INSTRUCTION	A0	R/W	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
13. Partial Out	0	0	1	0	1	0	1	0	0	1	Exit the partial mode
14. Read/Modify/Write In	0	0	1	1	1	0	0	0	0	0	Enable read modify write
15. Read/Modify/Write Out	0	0	1	1	1	0	1	1	1	0	Disable read modify write
16. Scroll Area	0	0	1	0	1	0	1	0	1	0	Set scroll area TL[7:0] : Set top line address BL[7:0] : Set bottom line address NSL[7:0] : Number of specified line SCM[1:0] : Area scroll mode
	1	0	TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0	
	1	0	BL7	BL6	BL5	BL4	BL3	BL2	BL1	BL0	
	1	0	NSL7	NSL6	NSL5	NSL4	NSL3	NSL2	NSL1	NSL0	
	1	0	0	0	0	0	0	0	SCM1	SCM0	
17. Set Start Line	0	0	1	0	1	0	1	0	1	1	Set scroll start address $00h \leq SL \leq A1h$
	1	0	SL7	SL6	SL5	SL4	SL3	SL2	SL1	SL0	
18. OSC ON	0	0	1	1	0	1	0	0	0	1	Turn on the internal oscillator
19. OSC OFF	0	0	1	1	0	1	0	0	1	0	Turn off the internal oscillator
20. Power Control	0	0	0	0	1	0	0	0	0	0	Power circuit operation VB=0: OFF, VB=1: ON VF=0: OFF, VF=1: ON VR=0: OFF, VR=1: ON
	1	0	0	0	0	0	VB	0	VF	VR	
21. Set Vop	0	0	1	0	0	0	0	0	0	1	Set Vop
	1	0	0	0	Vop5	Vop4	Vop3	Vop2	Vop1	Vop0	
	1	0	0	0	0	0	0	Vop8	Vop7	Vop6	
22. Vop Control	0	0	1	1	0	1	0	1	1	VOL	Control Vop VOL=0: Vop increase one step VOL=1: Vop decrease one step
23. Read Register Mode	0	0	0	1	1	1	1	1	0	REG	Set read register mode REG=0: read the register value of Vop[5:0] REG=1: read the register value of Vop[8:6]
24. Nop	0	0	0	0	1	0	0	1	0	1	No operation
25. Read Status (Parallel and I ² C)	0	1	D7	D6	D5	D4	D3	D2	D1	D0	Read status byte (Parallel and I ² C)
26. Read Status (4-Line and 3-Line SPI)	0	0	1	1	1	1	1	1	1	0	Read status byte (4-Line and 3-Line SPI)
	0	1	D7	D6	D5	D4	D3	D2	D1	D0	
27. Data Format Select	0	0	0	0	0	0	1	DO	0	0	DO=0; LSB on bottom (Default) DO=1; LSB on top
28. Display Mode	0	0	1	1	1	1	0	0	0	0	Set display mode DM=0 : Mono (Default) DM=1 : 4Gray Scale Mode
	1	0	0	0	0	1	0	0	0	DM	

INSTRUCTION	A0	R/W	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
29.Set ICON	0	0	0	1	1	1	0	1	1	ICON	Enable/Disable ICON RAM ICON=1 ; Enable ICON RAM ICON=0 ; Disable ICON RAM
30.Set Master/Slave	0	0	0	1	1	0	1	1	1	MS	Select Master or Slave mode MS=0 ; CMD for Master (Default) MS=1 ; CMD for Slave
Ext[1:0]=0,1 (Extension Command 2)											
31.Set Gray Level	0	0	0	0	1	0	0	0	0	0	Set gray scale level GL[4:0]: Set Light Gray Level GD[4:0]: Set Dark Gray Level
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	GL4	GL3	GL2	GL1	GL0	
	1	0	0	0	0	GL4	GL3	GL2	GL1	GL0	
	1	0	0	0	0	GL4	GL3	GL2	GL1	GL0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	GD4	GD3	GD2	GD1	GD0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	GD4	GD3	GD2	GD1	GD0	
	1	0	0	0	0	GD4	GD3	GD2	GD1	GD0	
	1	0	0	0	0	GD4	GD3	GD2	GD1	GD0	
	1	0	0	0	0	0	0	0	0	0	
32.Analog Circuit Set	0	0	0	0	1	1	0	0	1	0	Set analog circuit BE[1:0]: Booster efficiency set BS[2:0]: Set bias ratio
	1	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	BE1	BE0	
	1	0	0	0	0	0	0	BS2	BS1	BS0	
33.Booster Level	0	0	0	1	0	1	0	0	0	1	Set booster level BST=0 : X8 BST=1 : X10
	1	0	1	1	1	1	1	0	1	BST	
34. Driving Select	0	0	0	1	0	0	0	0	0	DS	Power type DS=0: Internal (Default) DS=1 :External
35.Auto Read Control	0	0	1	1	0	1	0	1	1	1	Set auto-read instruction XARD=0: Enable auto read XARD=1: Disable auto read
	1	0	1	0	0	XARD	1	1	1	1	
36.OTP WR/RD Control	0	0	1	1	1	0	0	0	0	0	OTP WR/RD control WR/RD=0: Enable OTP read WR/RD=1: Enable OTP write
	1	0	0	0	WR/RD	0	0	0	0	0	
37.OTP Control Out	0	0	1	1	1	0	0	0	0	1	OTP control out
38.OTP Write	0	0	1	1	1	0	0	0	1	0	OTP write

INSTRUCTION	A0	R/W	COMMAND BYTE								DESCRIPTION	
			D7	D6	D5	D4	D3	D2	D1	D0		
39.OTP Read	0	0	1	1	1	0	0	0	1	1	OTP read	
40.OTP Selection Control	0	0	1	1	1	0	0	1	0	0	OTP selection control Ctrl=1: Disable OTP Selection Ctrl=0: Enable OTP Selection	
	1	0	1	Ctrl	0	1	1	0	0	1		
41.OTP Programming Setting	0	0	1	1	1	0	0	1	0	1	OTP programming setting	
	1	0	0	0	0	0	1	1	1	1		
42.Frame Rate	0	0	1	1	1	1	0	0	0	0	Frame rate setting in different temperature range	
	1	0	0	0	0	FRA4	FRA3	FRA2	FRA1	FRA0		
	1	0	0	0	0	FRB4	FRB3	FRB2	FRB1	FRB0		
	1	0	0	0	0	FRC4	FRC3	FRC2	FRC1	FRC0		
	1	0	0	0	0	FRD4	FRD3	FRD2	FRD1	FRD0		
43.Temperature Range	0	0	1	1	1	1	0	0	1	0	Temperature range setting	
	1	0	0	TA6	TA5	TA4	TA3	TA2	TA1	TA0		
	1	0	0	TB6	TB5	TB4	TB3	TB2	TB1	TB0		
	1	0	0	TC6	TC5	TC4	TC3	TC2	TC1	TC0		
44.Temperature Gradient Compensation	0	0	1	1	1	1	0	1	0	0	Set temperature gradient compensation coefficient	
	1	0	MT13	MT12	MT11	MT10	MT03	MT02	MT01	MT00		
	1	0	MT33	MT32	MT31	MT30	MT23	MT22	MT21	MT20		
	1	0	MT53	MT52	MT51	MT50	MT43	MT42	MT41	MT40		
	1	0	MT73	MT72	MT71	MT70	MT63	MT62	MT61	MT60		
	1	0	MT93	MT92	MT91	MT90	MT83	MT82	MT81	MT80		
	1	0	MTB3	MTB2	MTB1	MTB0	MTA3	MTA2	MTA1	MTA0		
	1	0	MTD3	MTD2	MTD1	MTD0	MTC3	MTC2	MTC1	MTC0		
	1	0	MTF3	MTF2	MTF1	MTF0	MTE3	MTE2	MTE1	MTE0		
Ext[1:0]=1,0(Extension Command 3)												
45.Set ID	0	0	1	1	0	1	0	1	0	1	Set ID	
	1	0	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0		
46 Read ID	0	0	0	1	1	1	1	1	1	RID	Read ID RID=1 ; Enable RID=0 ; Disable	
Ext[1:0]=1,1(Extension Command 4)												
47.Enable OTP	0	0	1	1	0	1	0	1	1	0	Enable OTP EOTP =0 ; Disable (Default)	
	1	0	0	0	0	EOTP	0	0	0	0	EOTP =1 ; Enable	

表 8. 指令表

请详细参考 IC 资料”ST75256.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 256*128 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:

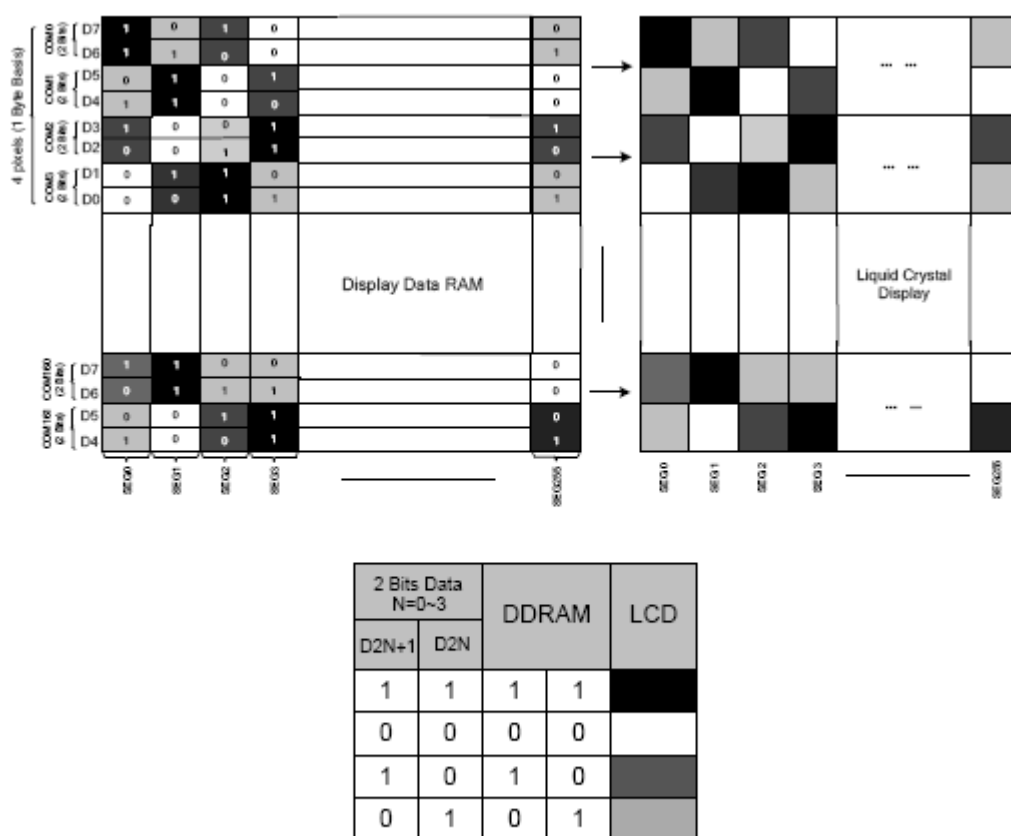
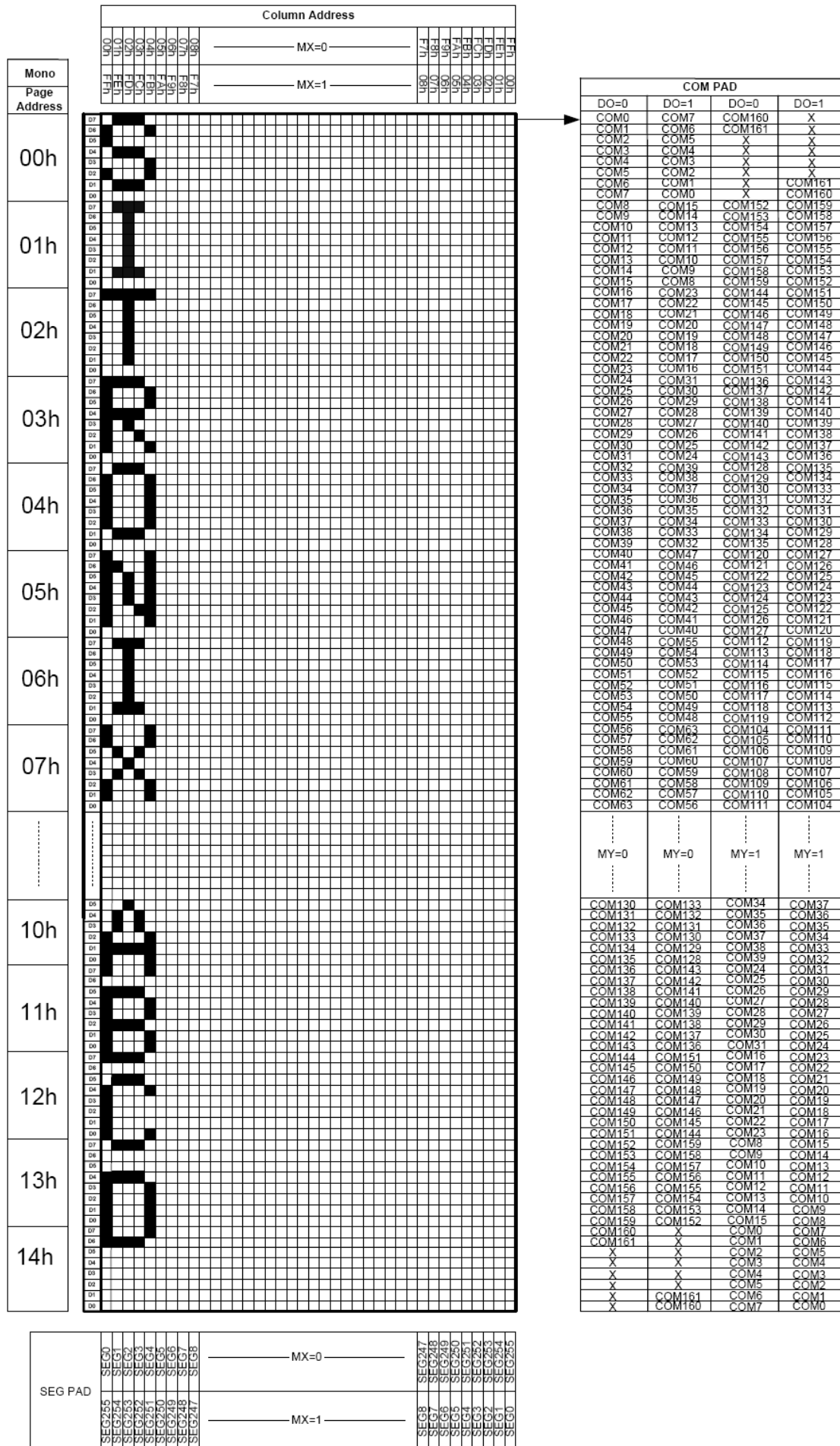


Figure 21 DDRAM Mapping (4-Level Gray Scale Mode)

下图摘自 ST75256 IC 资料，可通过“ST75256.PDF”之第 37 页获取最佳效果。



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、背光源电源线、IO端口 (接口)
IO端口包括: 并口时: CS、RESET、RW、E、RS、DO--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 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:

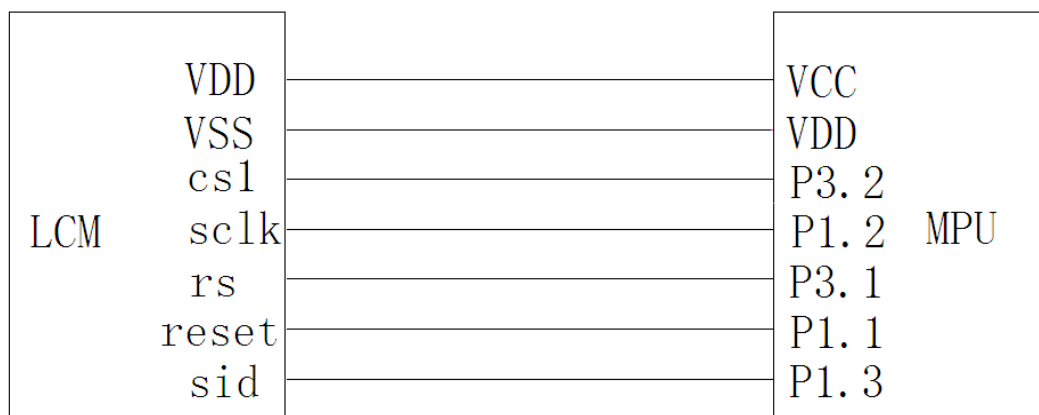
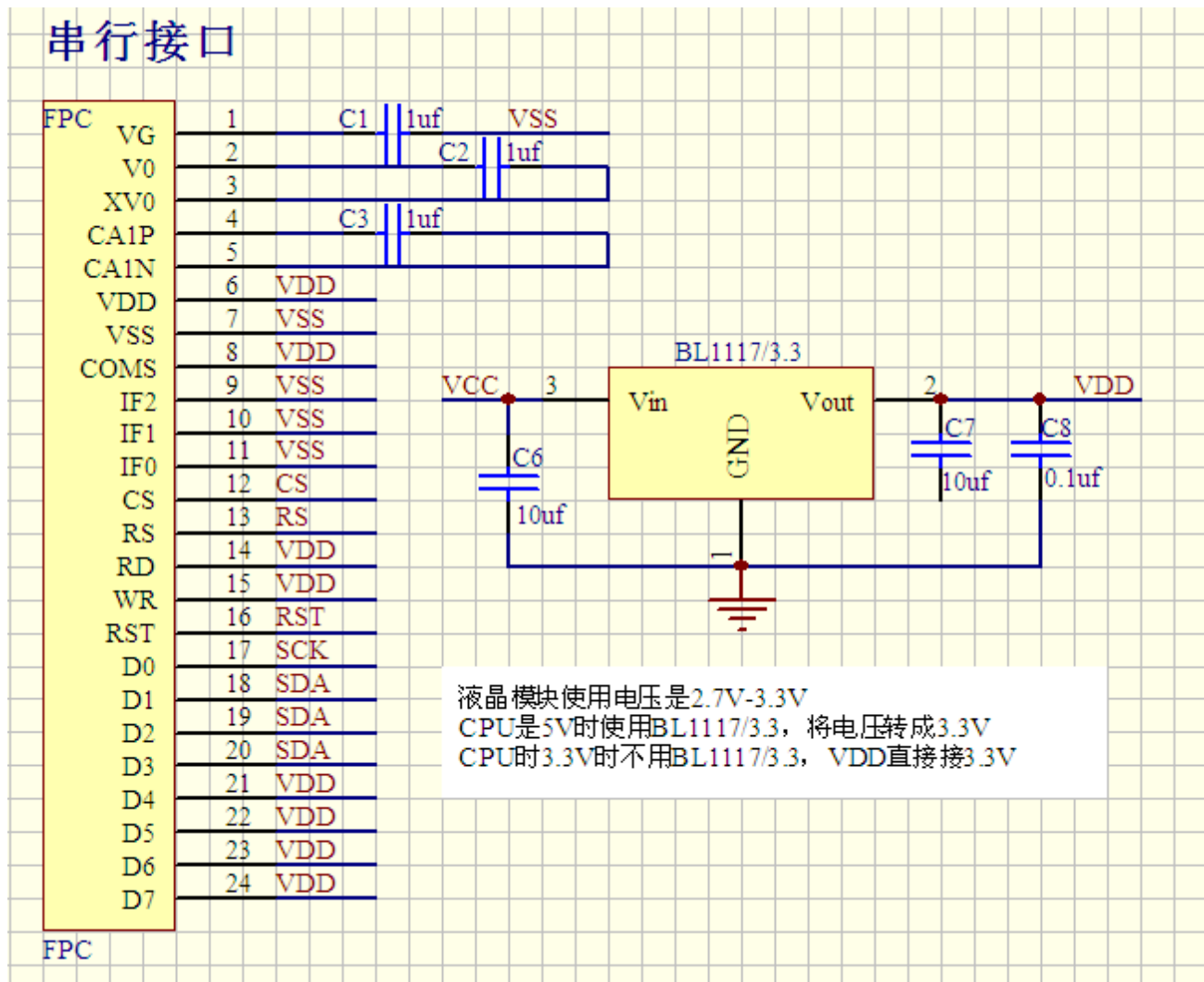


图 8. 串行接口



/* 液晶模块型号: JLX16080G-370-P-BN

并行接口

驱动 IC 是:ST75256

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

*/

#include <reg52.H>

#include <intrins.h>

```
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
```

#define uchar unsigned char

#define uint unsigned int

/*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++);
}
```

/*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(2000);
}
```

//写指令到 LCD 模块

```
void transfer_command_lcd(int data1)
{
    char i;
    lcd_cs1=0;
    lcd_rs=0;
    for(i=0;i<8;i++)
    {
        lcd_sclk=0;
        if(data1&0x80) lcd_sid=1;
        else lcd_sid=0;
        lcd_sclk=1;
        data1<<=1;
    }
    lcd_cs1=1;
}
```

//写数据到 LCD 模块

```
void transfer_data_lcd(int data1)
{
    char i;
    lcd_cs1=0;
    lcd_rs=1;
    for(i=0;i<8;i++)
    {
```

```

        lcd_sclk=0;
        if(data1&0x80) lcd_sid=1;
        else lcd_sid=0;
        lcd_sclk=1;
        data1<<=1;
    }
    lcd_cs1=1;
}

void initial_lcd()
{
    reset=0;
    delay(100);
    reset=1;
    delay(100);

    transfer_command_lcd(0x30); //EXT=0
    transfer_command_lcd(0x94); //Sleep out
    transfer_command_lcd(0x31); //EXT=1
    transfer_command_lcd(0xD7); //Autoread disable
    transfer_data_lcd(0X9F); //

    transfer_command_lcd(0x32); //Analog SET
    transfer_data_lcd(0x00); //OSC Frequency adjustment
    transfer_data_lcd(0x01); //Frequency on booster capacitors->6KHz
    transfer_data_lcd(0x03); //Bias=1/11

    transfer_command_lcd(0x20); // Gray Level
    transfer_data_lcd(0x01);
    transfer_data_lcd(0x03);
    transfer_data_lcd(0x05);
    transfer_data_lcd(0x07);
    transfer_data_lcd(0x09);
    transfer_data_lcd(0x0b);
    transfer_data_lcd(0x0d);
    transfer_data_lcd(0x10);
    transfer_data_lcd(0x11);
    transfer_data_lcd(0x13);
    transfer_data_lcd(0x15);
    transfer_data_lcd(0x17);
    transfer_data_lcd(0x19);
    transfer_data_lcd(0x1b);
    transfer_data_lcd(0x1d);
    transfer_data_lcd(0x1f);

    transfer_command_lcd(0x30); //EXT=0
    transfer_command_lcd(0x75); //Page Address setting
    transfer_data_lcd(0X00); // XS=0

```

```

transfer_data_lcd(0X14);    // XE=159 0x28
transfer_command_lcd(0x15); //Column Address setting
transfer_data_lcd(0X00);    // XS=0
transfer_data_lcd(0Xff);    // XE=256

transfer_command_lcd(0xBC); //Data scan direction
transfer_data_lcd(0x00);    //MX. MY=Normal
transfer_data_lcd(0xA6);

transfer_command_lcd(0xCA); //Display Control
transfer_data_lcd(0X00);    //
transfer_data_lcd(0X9F);    //Duty=160
transfer_data_lcd(0X20);    //Nline=off

transfer_command_lcd(0xF0); //Display Mode
transfer_data_lcd(0X10);    //10=Monochrome Mode, 11=4Gray

transfer_command_lcd(0x81); //EV control
transfer_data_lcd(0x27);    //VPR[5-0] //微调
transfer_data_lcd(0x03);    //VPR[8-6] //粗调
transfer_command_lcd(0x20); //Power control
transfer_data_lcd(0x0B);    //D0=regulator ; D1=follower ; D3=booste, on:1 off:0
delay_us(100);
transfer_command_lcd(0xAF); //Display on
}

/*写 LCD 行列地址: X 为起始的列地址, Y 为起始的行地址, x_total,y_total 分别为列地址及行地址的起点到终点的差值 */
void lcd_address(int x,int y,x_total,y_total)
{
    x=x-1;
    y=y+9;

    transfer_command_lcd(0x15); //Set Column Address
    transfer_data_lcd(x);
    transfer_data_lcd(x+x_total-1);

    transfer_command_lcd(0x75); //Set Page Address
    transfer_data_lcd(y);
    transfer_data_lcd(y+y_total-1);
    transfer_command_lcd(0x30);
    transfer_command_lcd(0x5c);
}

```

/*清屏*/

void clear_screen()

```
{
    int i, j;
    lcd_address(0, 0, 160, 11);
    for(i=0; i<17; i++)
    {
        for(j=0; j<160; j++)
        {
            transfer_data_lcd(0x00);
        }
    }
}
```

void test(int x, int y, uchar data1, uchar data2)

```
{
    int i, j;
    lcd_address(x, y, 160, 10);

    for(i=0; i<10; i++)
    {
        for(j=0; j<80; j++)
        {
            transfer_data_lcd(data1);
            transfer_data_lcd(data2);
        }
    }
}
```

/*显示 160*80 点阵的图像*/

void disp_160x80(int x, int y, char *dp)

```
{
    int i, j;
    lcd_address(x, y, 160, 10);
    for(i=0; i<10; i++)
    {
        for(j=0; j<160; j++)
        {
            transfer_data_lcd(*dp);
            dp++;
        }
    }
}
```

```
uchar code bmp2[]={
/*-- 调入了一幅图像: E:\work\图片收藏夹\黑白屏图片\JLX16080G-370. bmp  --*/
/*-- 宽度 x 高度=160x80  --*/
0x00, 0x08, 0x08, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7E, 0x54, 0x54, 0x54, 0x54, 0x54, 0x7E, 0x00, 0x00, 0x00, 0x00,
0x40, 0x7F, 0x49, 0x49, 0x49, 0x7F, 0x48, 0x88, 0x48, 0x38, 0x0F, 0x18, 0xE8, 0x48, 0x08, 0x00,
0x04, 0x84, 0x74, 0x27, 0x00, 0x42, 0x42, 0x7F, 0x42, 0x42, 0x42, 0x40, 0x7F, 0x00, 0x00, 0x00,
0x08, 0x86, 0x60, 0x07, 0x18, 0x21, 0x27, 0x38, 0x21, 0xA6, 0x7D, 0x24, 0x25, 0x26, 0x20, 0x00,
0x00, 0x00, 0x00, 0x00, 0x7E, 0x54, 0x54, 0x54, 0x54, 0x54, 0x7E, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x7C, 0x54, 0x57, 0x54, 0x54, 0x54, 0x57, 0x54, 0x7C, 0x00, 0x00, 0x00, 0x00,
0x00, 0x04, 0x04, 0x44, 0x44, 0x44, 0x44, 0x47, 0x44, 0x44, 0x44, 0x44, 0x04, 0x04, 0x00,
0x08, 0x0B, 0xFF, 0x0A, 0x09, 0x20, 0x2F, 0x2A, 0xFA, 0x2A, 0x2A, 0xFA, 0x2F, 0x20, 0x00, 0x00,
0x08, 0x08, 0xFF, 0x08, 0x08, 0x00, 0x10, 0x10, 0xFF, 0x10, 0x10, 0x10, 0x1F, 0x00, 0x00, 0x00,
0x00, 0x04, 0x04, 0xFC, 0x04, 0x04, 0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xFE, 0xA4, 0xA4, 0xA4, 0xA4, 0xFE, 0x00, 0x00, 0xFE, 0xA4, 0xA4, 0xA4, 0xA4, 0xFE, 0x00,
0x10, 0xF8, 0x10, 0x10, 0x20, 0xFF, 0xA0, 0x81, 0x82, 0x8C, 0xF0, 0x88, 0x84, 0x83, 0x82, 0x00,
0x00, 0x00, 0x00, 0xFE, 0x04, 0x08, 0x00, 0xFE, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x02, 0x1C, 0x00,
0x20, 0x20, 0xFF, 0x00, 0x80, 0x00, 0xFF, 0x82, 0x84, 0x48, 0x30, 0xD8, 0x86, 0x03, 0x02, 0x00,
0x00, 0xFE, 0xA4, 0xA4, 0xA4, 0xA4, 0xFE, 0x00, 0x00, 0xFE, 0xA4, 0xA4, 0xA4, 0xA4, 0xFE, 0x00,
0x04, 0x84, 0x44, 0x34, 0x04, 0xFC, 0x04, 0x04, 0x04, 0xFC, 0x14, 0x24, 0xC4, 0x04, 0x04, 0x00,
0x08, 0x10, 0x20, 0xC0, 0x00, 0x02, 0x01, 0xFE, 0x00, 0x00, 0x80, 0x40, 0x30, 0x18, 0x00, 0x00,
0xC0, 0x00, 0xFF, 0x00, 0x00, 0x21, 0xA1, 0xA2, 0xAC, 0xF0, 0xA8, 0xA4, 0xA6, 0x23, 0x22, 0x00,
0x10, 0x18, 0xF0, 0x20, 0xA1, 0x82, 0x8C, 0xB0, 0xC0, 0xA0, 0x90, 0x88, 0x8C, 0x86, 0x84, 0x00,
0x00, 0x0E, 0x10, 0x10, 0x10, 0x11, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x08, 0x48, 0x49, 0x7E, 0x48, 0x48, 0x7F, 0x48, 0x48, 0x08, 0x3F, 0x00, 0x00, 0xFF, 0x00, 0x00,
0x02, 0x02, 0x02, 0xFA, 0x8B, 0x8A, 0x8A, 0x8A, 0x8A, 0x8A, 0x8A, 0x8A, 0xFA, 0x02, 0x02, 0x02, 0x00,
0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0x1F, 0x10, 0x10, 0x00,
0x10, 0x1F, 0x10, 0x00, 0x00, 0x00, 0x00, 0x10, 0x18, 0x16, 0x01, 0x01, 0x16, 0x18, 0x10,
0x00, 0x08, 0x08, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x08, 0x11, 0x11, 0x18, 0x00, 0x00,
0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00, 0x00, 0x0E, 0x11, 0x10, 0x10, 0x11, 0x0E, 0x00,
0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00, 0x03, 0x0C, 0x10, 0x10, 0x10, 0x1C, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x10, 0x11, 0x11, 0x12, 0x0C, 0x00,
0x00, 0x1C, 0x10, 0x10, 0x13, 0x1C, 0x10, 0x00, 0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00,
0x00, 0x0C, 0x14, 0x24, 0x44, 0x84, 0x0C, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x02, 0x42, 0x92, 0x12, 0x12, 0x12, 0x92, 0x7E, 0x12, 0x12, 0x12, 0x52, 0x32, 0xD2, 0x02, 0x00,
0x00, 0x00, 0x00, 0x40, 0xE0, 0x40, 0x40, 0x44, 0x42, 0x41, 0x42, 0x7C, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x03, 0x01, 0x01, 0x01, 0xFE, 0x00, 0x00, 0x00,
0x04, 0xFC, 0x04, 0x04, 0x04, 0x04, 0x0C, 0x00, 0x04, 0x0C, 0x34, 0xC0, 0xC0, 0x34, 0x0C, 0x04,
0x00, 0x04, 0x04, 0xFC, 0x04, 0x04, 0x00, 0x00, 0x00, 0xF0, 0x88, 0x04, 0x04, 0x88, 0x70, 0x00,
0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0x00, 0x38, 0x44, 0x84, 0x84, 0x44, 0x38, 0x00,
0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0xE0, 0x18, 0x04, 0x04, 0x44, 0x78, 0x40, 0x00,
0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x18, 0x04, 0x04, 0x88, 0x70, 0x00,
0x00, 0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00,
0x00, 0x0C, 0x10, 0x11, 0x11, 0x12, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x07, 0x04, 0x04, 0x04, 0xFC, 0x24, 0x24, 0x24, 0x2F, 0x24, 0x00, 0x00, 0x00,
0x7F, 0x40, 0x48, 0x54, 0x63, 0x11, 0x13, 0x1D, 0xF1, 0x17, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00,
```



0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x1F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x07, 0x08, 0x11, 0x11, 0x18, 0x00, 0x00, 0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00,
0x00, 0x01, 0x01, 0x00, 0x01, 0x01, 0x01, 0x00, 0x00, 0x0E, 0x11, 0x10, 0x10, 0x11, 0x0E, 0x00,
0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00, 0x00, 0x00, 0x00, 0x07, 0x04, 0x04, 0x04, 0xFC,
0x24, 0x24, 0x24, 0x2F, 0x24, 0x00, 0x00, 0x00, 0x7F, 0x40, 0x48, 0x54, 0x63, 0x11, 0x13, 0x1D,
0xF1, 0x17, 0x11, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x18, 0x04, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x02, 0x0C, 0xE0, 0x48, 0x46, 0x40, 0x50, 0x48, 0x46, 0x40, 0xF0, 0x08, 0x06, 0x00, 0x00,
0xFF, 0x00, 0x40, 0x20, 0xC0, 0x20, 0x20, 0x20, 0x20, 0xFF, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00,
0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04, 0xFC, 0x04, 0x04, 0x00, 0x00,
0x00, 0xF0, 0x88, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00,
0x00, 0x04, 0x8C, 0x74, 0x70, 0x8C, 0x04, 0x00, 0x00, 0x38, 0x44, 0x84, 0x84, 0x44, 0x38, 0x00,
0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0x00, 0x02, 0x0C, 0xE0, 0x48, 0x46, 0x40, 0x50,
0x48, 0x46, 0x40, 0xF0, 0x08, 0x06, 0x00, 0x00, 0xFF, 0x00, 0x40, 0x20, 0xC0, 0x20, 0x20, 0x20,
0x20, 0xFF, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x03, 0x04, 0x08, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x10, 0x10, 0x91, 0x72, 0x55, 0x18, 0x00, 0x7F, 0x40, 0x40, 0x5F, 0x40, 0x40, 0x7F, 0x00, 0x00,
0x00, 0x7F, 0x40, 0x40, 0x48, 0x44, 0x44, 0x42, 0x41, 0x46, 0x78, 0x50, 0x40, 0x40, 0x00, 0x00,
0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x10, 0x11, 0x11, 0x10, 0x10, 0x00,
0x00, 0x0C, 0x10, 0x11, 0x11, 0x12, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x07, 0x08, 0x11, 0x11, 0x18, 0x00, 0x00, 0x00, 0x01, 0x01, 0x00, 0x01, 0x01, 0x01, 0x00,
0x00, 0x0E, 0x10, 0x10, 0x10, 0x11, 0x0E, 0x00, 0x00, 0x0E, 0x11, 0x10, 0x10, 0x11, 0x0E, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x10, 0x11, 0x11, 0x10, 0x10, 0x00,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xE0, 0x20, 0x24, 0x24, 0xFC, 0x24, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x40, 0x80, 0x00, 0xFF, 0x00, 0xE1, 0x42, 0x84, 0x18, 0x60, 0x80, 0x7C, 0x02, 0xC2, 0x0E, 0x00,
0x00, 0xFC, 0x04, 0x0C, 0x14, 0x24, 0x44, 0x84, 0x04, 0x84, 0x64, 0x3C, 0x14, 0x04, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x98, 0x84, 0x04, 0x04, 0x88, 0x70, 0x00,
0x00, 0x18, 0x04, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xF0, 0x88, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x04, 0x8C, 0x74, 0x70, 0x8C, 0x04, 0x00,
0x00, 0x0C, 0x14, 0x24, 0x44, 0x84, 0x0C, 0x00, 0x00, 0x38, 0x44, 0x84, 0x84, 0x44, 0x38, 0x00,
0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x98, 0x84, 0x04, 0x04, 0x88, 0x70, 0x00,
0x04, 0xFC, 0x04, 0x00, 0xFC, 0x04, 0x00, 0xFC, 0x04, 0x00, 0xFC, 0x04, 0x00, 0xFC, 0x04, 0x00, 0xFC,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1F, 0x10, 0x11, 0x11, 0x10, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x40, 0x5F, 0x41, 0x41, 0x7F, 0x01, 0x00, 0x7F, 0x48, 0x44, 0x42, 0x41, 0x46, 0x58, 0x40, 0x00,
0x04, 0x24, 0x24, 0x27, 0x24, 0x24, 0x24, 0x04, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x0F, 0x00, 0x00,
0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x11, 0x10, 0x10, 0x10, 0x1C, 0x00,
0x18, 0x10, 0x10, 0x1F, 0x10, 0x10, 0x18, 0x00, 0x00, 0x1C, 0x10, 0x10, 0x13, 0x1C, 0x10, 0x00,
0x00, 0x1F, 0x10, 0x11, 0x11, 0x10, 0x10, 0x00, 0x00, 0x0E, 0x10, 0x10, 0x10, 0x11, 0x0E, 0x00,
0x00, 0x1F, 0x10, 0x11, 0x11, 0x10, 0x10, 0x00, 0x00, 0x1C, 0x10, 0x10, 0x13, 0x1C, 0x10, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x98, 0x84, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00,
0x10, 0x10, 0x24, 0x22, 0x04, 0xF8, 0x00, 0xFE, 0x0A, 0x12, 0x62, 0x82, 0xC2, 0x3A, 0x02, 0x00,
0x10, 0x38, 0xD0, 0x10, 0x30, 0xA0, 0x72, 0x24, 0x08, 0x30, 0xC0, 0x04, 0x02, 0xFC, 0x00, 0x00,
0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x1C, 0x04, 0x84, 0x84, 0x44, 0x38, 0x00,
0x00, 0x00, 0x04, 0xFC, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00,
0x00, 0x98, 0x84, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x0C, 0x14, 0x24, 0x44, 0x84, 0x0C, 0x00,
0x00, 0x98, 0x84, 0x04, 0x04, 0x88, 0x70, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};
```

```
//-----
```

```
void main ()
```

```
{
```

```
    initial_lcd();
```

```
//对液晶模块进行初始化设置
```

```
    while(1)
```

```
    {
```

```
        clear_screen();
```

```
//清屏
```

```
        disp_160x80(1, 1, bmp2);
```

```
//显示一幅 240*160 点阵的黑白图。
```

```
        waitkey();
```

```
        clear_screen();
```

```
        test(1, 1, 0xff, 0xff);
```

```
        waitkey();
```

```
        clear_screen();
```

```
        test(1, 1, 0xaa, 0xaa);
```

```
        waitkey();
```

```
        clear_screen();
```

```
        test(1, 1, 0xff, 0x00);
```

```
        waitkey();
```

```
    }
```

```
}
```

7.5、并行接口

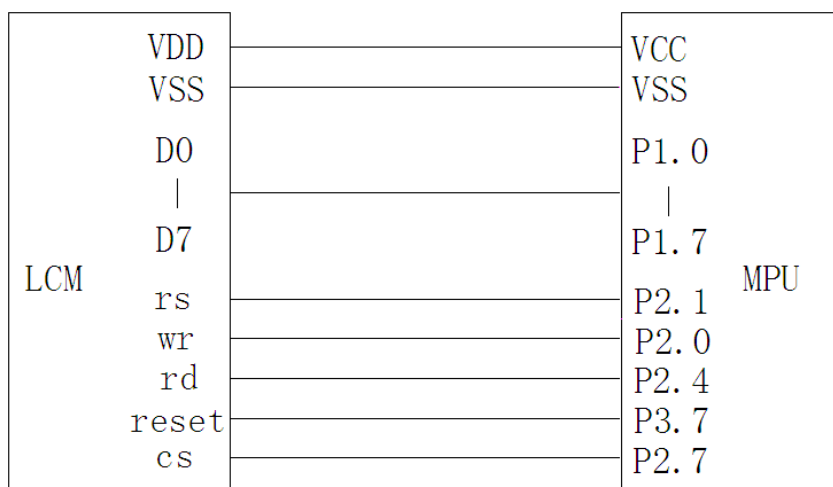
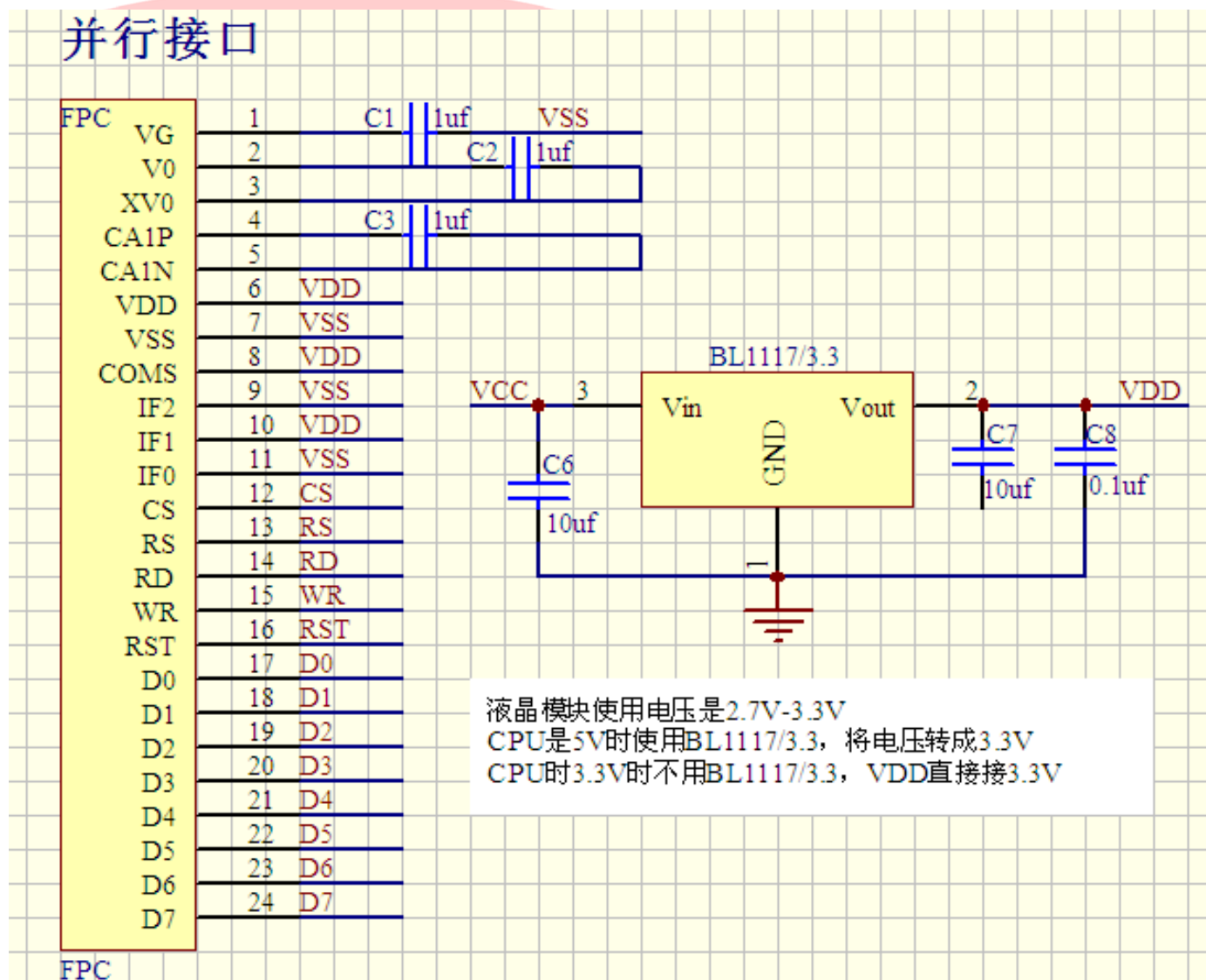


图 9. 并行接口



7.5.1、以下为并行接口方式范例程序

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：

并程序序：

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

sbit lcd_rs=P2^1;    /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit lcd_rd=P2^4;    /*接口定义:lcd_e 就是 LCD 的 rd*/
sbit lcd_wr=P2^0;    /*接口定义:lcd_rw 就是 LCD 的 wr*/
sbit lcd_reset=P3^7; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit lcd_cs1=P2^7;   /*接口定义:lcd_cs1 就是 LCD 的 cs1*/
sbit key  = P2^0;    //按键
```

//写指令到 LCD 模块

```
void transfer_command_lcd(int data1)
{
    lcd_cs1=0;
    lcd_rs=0;
    lcd_rd=0;
    lcd_wr=0;
    P1=data1;
    lcd_rd=1;
    lcd_cs1=1;
    lcd_rd=0;
}
```

//写数据到 LCD 模块

```
void transfer_data_lcd(int data1)
{
    lcd_cs1=0;
    lcd_rs=1;
    lcd_rd=0;
    lcd_wr=0;
    P1=data1;
    lcd_rd=1;
    lcd_cs1=1;
    lcd_rd=0;
}
```

7.6、IIC 接口

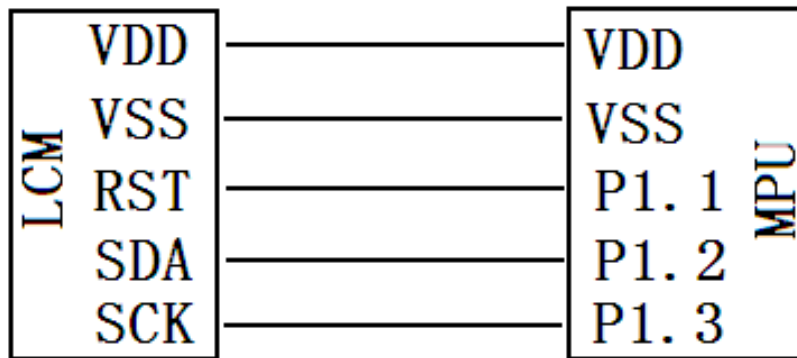
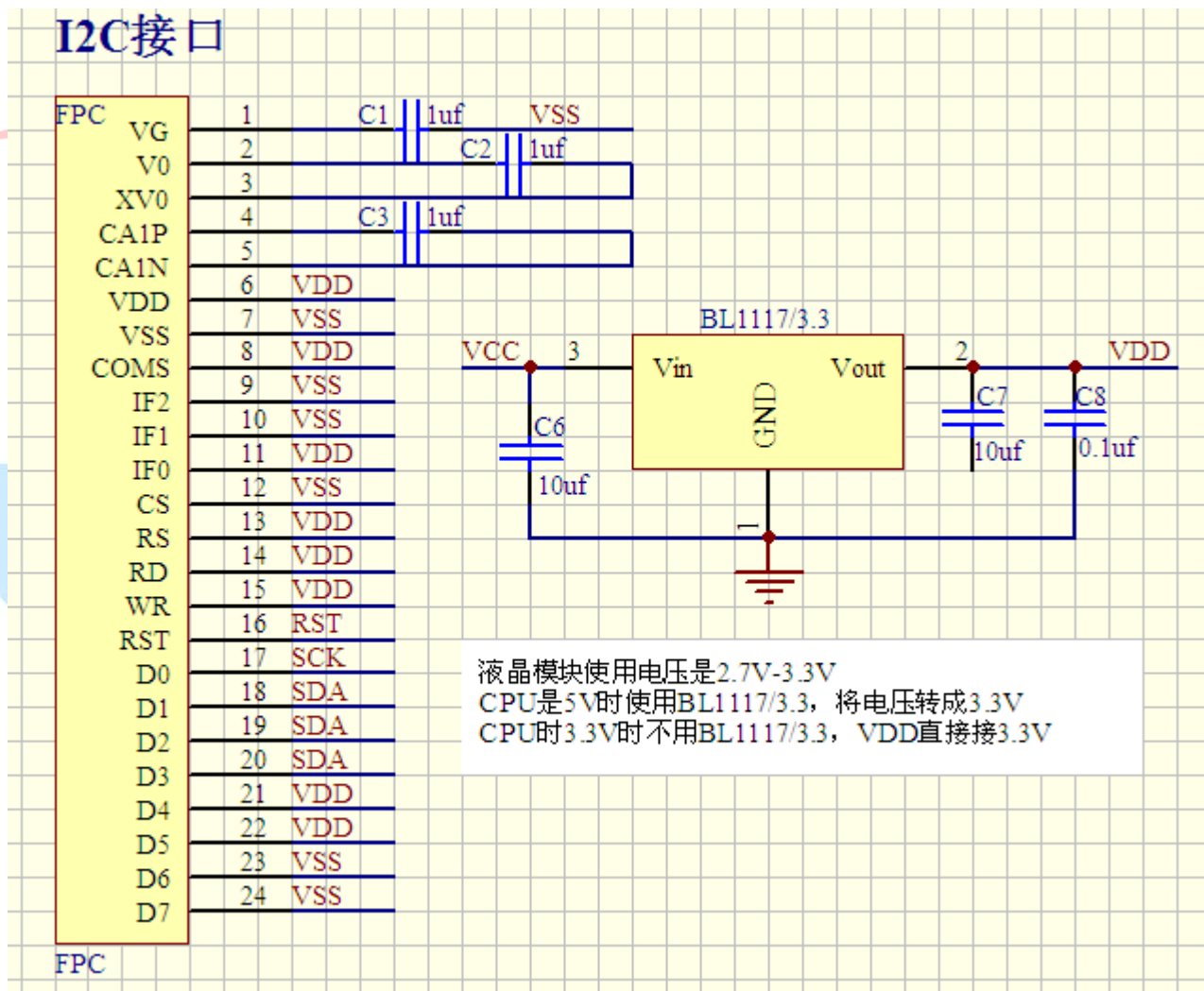


图 10. IIC



7.5.1、以下为并行接口方式范例程序

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: JLX19296G-370
   IIC 接口
   驱动 IC 是: ST75256
   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();  
}
```

