

OCM12232-1 图形点阵液晶显示模块 使用说明书

感谢您关注和使用我们的字符点阵系列液晶显示器产品，欢迎您提出您的要求、意见和建议，我们将竭诚为您服务、让您满意。您可以浏览 <http://www.gptlcm.cn> 了解最新的产品与应用信息，或拨打热线电话 **0758—2317153** 以及向 syl@gptlcm.cn 邮箱发 **E-mail** 获取具体的技术咨询与服务。

金鹏实业有限公司

Golden Palm Industry Co.,Ltd.

一、液晶驱动 IC 基本特性

- 1、具有低功耗、供应电压范围宽等特点。
- 2、具有 16common 和 61segment 输出，并可外接驱动 IC 扩展驱动。
- 3、具有 2560 位显示 RAM (DD RAM)，即 $80 \times 8 \times 4$ 位
- 4、具有与 68 系列或 80 系列相适配的 MPU 接口功能，并有专用的指令集，可完成文本显示或图形显示的功能设置

二、模块基本特性

视域尺寸: 60.5×18.0mm (12232-1/-2), 54.8×18.3mm (12232-3)

显示类型: 黄底黑字

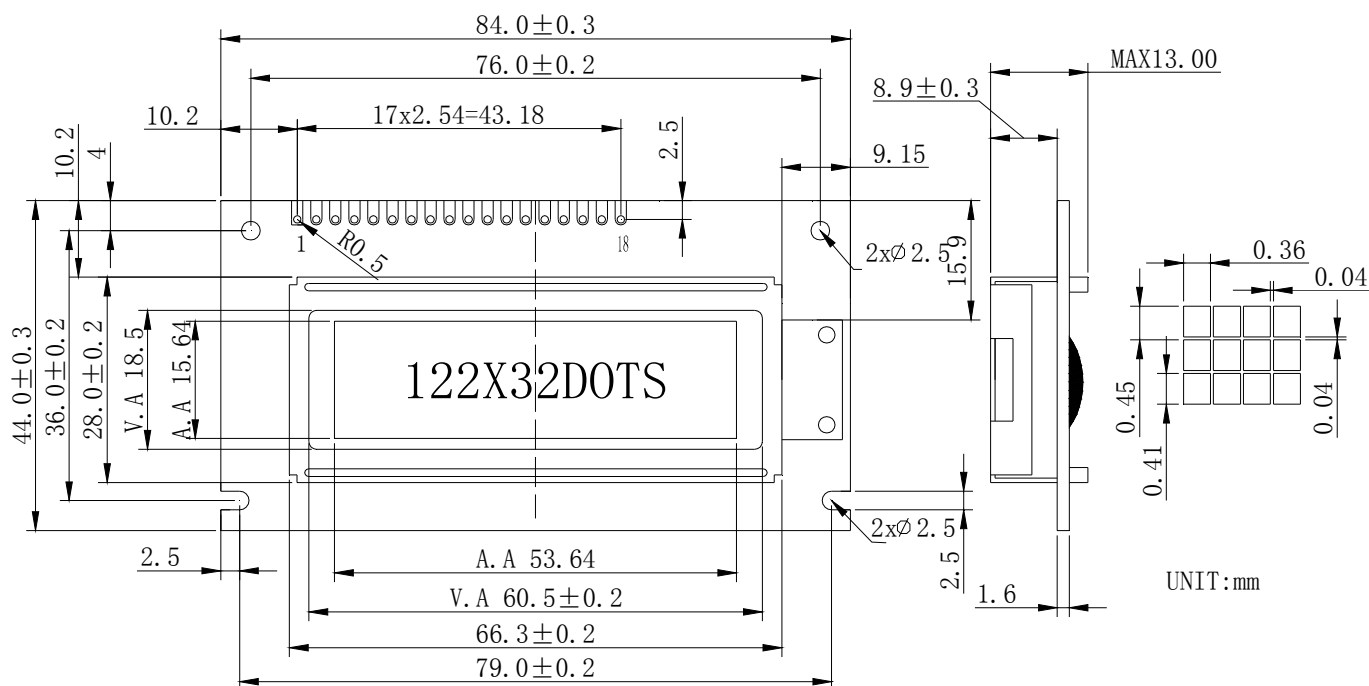
LCD 显示角度: 6 点钟直观

驱动方式: 1/32 duty, 1/6 bias

连接方式: 导电胶条, 铁框

- 补充说明: 模块外观尺寸可根据用户的要求进行适度调整。

三、外形尺寸图



四、工作参数

- 1、逻辑工作电压 (VDD-VSS): 5V
- 2、LCD 驱动电压 (Vdd-Vlcd): 3.0~13.5V
- 3、工作温度 (Ta): 0~55℃ (常温) / -20~70℃ (宽温)
- 4、保存温度 (Tstg): -10~65℃

五、电气特性 (测试条件 Ta=25, Vdd=5.0±0.25V)

- 1、输入高电平 (Vih): 3.5Vmin
- 2、输入低电平 (Vil): 0.55Vmax
- 3、输出高电平 (Voh): 3.75Vmin
- 4、输出低电平 (Vol): 1.0Vmax
- 5、工作电流: 2.0mAmx

六、接口说明

序号	引脚名称	引脚说明
1	VDD	逻辑电源正
2	VSS	逻辑电源地
3	Vo	LCD 驱动电源
4	/RESET	复位端, 对于 68 系列 MPU: 上升沿 (L-H) 复位, 且复位后电平须保持为高电 (H) ; 对于 80 系列 MPU: 下降沿 (H-L) 复位, 且复位后电平须保持为低电平 (L) 。
5	E1	读写使能。对于 68 系列 MPU, 连接使能信号引脚, 高电平有效; 对于 80 系列 MPU, 连接 /RD 引脚, 低电平有效。
6	E2	读写使能。对于 68 系列 MPU, 连接使能信号引脚, 高电平有效; 对于 80 系列 MPU, 连接 /RD 引脚, 低电平有效。
7	R/W	读写选择, 对于 68 系列 MPU, 高电平时读数据, 低电平时写数据; 对于 80 系列 MPU, 低电平时允许数据传输, 上升沿时锁定数据。
8	Ao	数据/指令选择。高电平: 数据 D0-D7 将送入显示 RAM; 低电平: 数据 D0-D7 将送入指令执行器执行。
9	DB0	数据输入输出引脚 0
10	DB1	数据输入输出引脚 1
11	DB2	数据输入输出引脚 2
12	DB3	数据输入输出引脚 3
13	DB4	数据输入输出引脚 4
14	DB5	数据输入输出引脚 5
15	DB6	数据输入输出引脚 6
16	DB7	数据输入输出引脚 7
17	LED+	背光电源正
18	LED-	背光电源负

七、指令描述

1、显示模式设置（显示开关指令）

R/W

CODE:	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0
	L	H	L	H	L	H	L	H	H	H	D

功能：开/关屏幕显示，不改变显示 RAM(DD RAM) 中的内容，也不影响内部状态。D=1，开显示；D=0，关显示。如果在显示关闭的状态下选择静态驱动模式，那么内部电路将处于安全模式。
该指令不影响显示 RAM 的内容。

2、设置显示起始行

R/W

CODE:	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0
	L	H	L	H	H	L	显示起始行（1~31）				

功能：执行该命令后，所设置的行将显示在屏幕的第一行。起始地址可以是 0-31 范围内任意一行。行地址计数器具有循环计数功能，用于显示行扫描同步，当扫描完一行后自动加一。

3、页地址设置

R/W

CODE:	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0
	L	H	L	H	L	H	H	H	L	A1	A0

功能：设置页地址。当 MPU 要对 DD RAM 进行读写操作时，首先要设置页地址和列地址。本指令不影响显示。

A1	A0	页地址
0	0	0
0	1	1
1	0	2
1	1	3

4、列地址设置

R/W

CODE:	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0
	L	H	L	L	A6	A5	A4	A3	A2	A1	A0

功能：设置 DD RAM 中的列地址。当 MPU 要对 DD RAM 进行读写操作前，首先要设置页地址和列地址。执行读写命令后，列地址会自动加 1，直到达到 50H 才会停止，但页地址不变。

A6	A5	A4	A3	A2	A1	A0	列地址
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
1	0	0	1	1	1	0	4E
1	0	0	1	1	1	1	4F

5、读状态指令

R/W

CODE:	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0
	L	L	H	BUSY	ADC	ON/OFF	RESET	L	L	L	L

功能：检测内部状态。

BUSY 为忙信号位，BUSY =1：内部正在执行操作；BUSY =0：空闲状态。

ADC 为显示方向位, ADC=0: 反向显示; ADC=1: 正向显示。

ON/OFF 显示开关状态, ON/OFF=0: 显示打开, ON/OFF=1: 显示关闭。

RESET 复位状态, RESET=0: 正常, RESET=1: 内部正处于复位初始化状态。

6、写显示数据

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	H	H	L	Write Data							

功能: 将 8 位数据写入 DD RAM, 该指令执行后, 列地址自动加 1, 所以可以连续将数据写入 DD RAM 而不用重新设置列地址。

7、读显示数据

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	H	L	H	Read Data							

功能: 读出页地址和列地址限定的 DD RAM 地址内的数据。当“读-修改-写模式”关闭时, 每执行一次读指令, 列地址自动加 1, 所以可以连续从 DD RAM 读出数据而不用设置列地址。

注意: 再设置完列地址后, 首次读显示数据前必须执行一次空的“读显示数据”。这是因为设置完列地址后, 第一次读数据时, 出现在数据总线上的数据是列地址而不是所要读出的数据。

8、设置显示方向

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	L	H	L	H	L	H	L	L	L	L	D

功能: 该指令设置 DD RAM 中的列地址与段驱动输出的对应关系
显示当设置 D=0 时, 反向; D=1 时, 正向。

9、开/关静态驱动模式设置

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	L	H	L	H	L	H	L	L	H	L	D

功能: D=0 表正常驱动, D=1 表示打开静态显示。

如果在打开静态显示时, 执行关闭显示指令, 内部电路将被置为安全模式。

10、DUTY 选择

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	L	H	L	H	L	H	L	H	L	L	D

功能: 设置 D=0 表示 1/16DUTY, D=1 表示 1/32DUTY。

11、“读-修改-写”模式设置

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	L	H	L	H	H	H	L	L	L	L	L

功能: 执行该指令以后, 每执行一次写数据指令列地址自动加 1; 但执行读数据指令时列地址不会改变。这个状态一直持续到执行“END”指令。

注意: 在“读-修改-写”模式下, 除列地址设置指令之外, 其他指令照常执行。

12、END 指令

		R/W		D7	D6	D5	D4	D3	D2	D1	D0
CODE:	A0	/RD	/WR								
	L	H	L	H	H	H	L	H	H	H	L

功能：关闭“读-修改-写”模式，并把列地址指针恢复到打开“读-修改-写”模式前的位置。

13、复位指令

CODE:	R/W									
	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1 D0
	L	H	L	H	H	H	L	L	L	H L

功能：使模块内部初始化。
初始化内容：① 设置显示初始行为第一行；
②页地址设置为第三页。
复位指令对显示 RAM 没有影响。

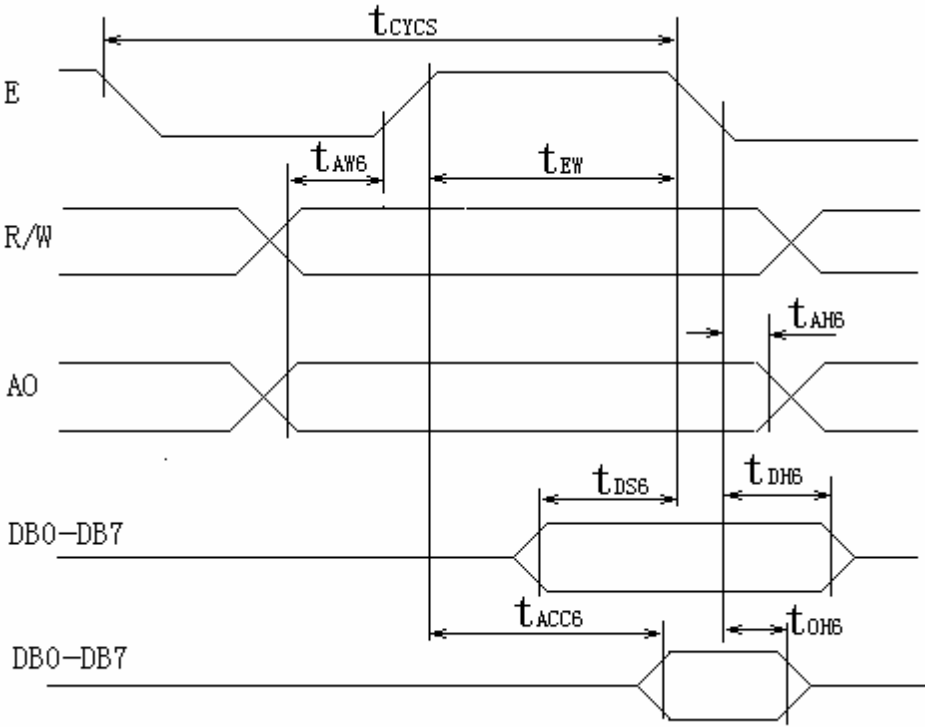
14、设置安全模式

通过关闭显示并打开静态显示的方法，可以设置安全模式，以减小功耗。
安全模式下的内部状态：

- 停止 LCD 驱动。Segment 和 Common 输出 VDD 电平。
- 停止晶体震荡并禁止外部时钟输入，晶振输入 OSC₂ 引脚处于不确定状态。
- 显示数据和内部模式不变。

可通过打开显示或关闭静态显示的方法关闭安全模式。

八、接口时序



1. 与 68 系列 MPU 接口读写操作时序图

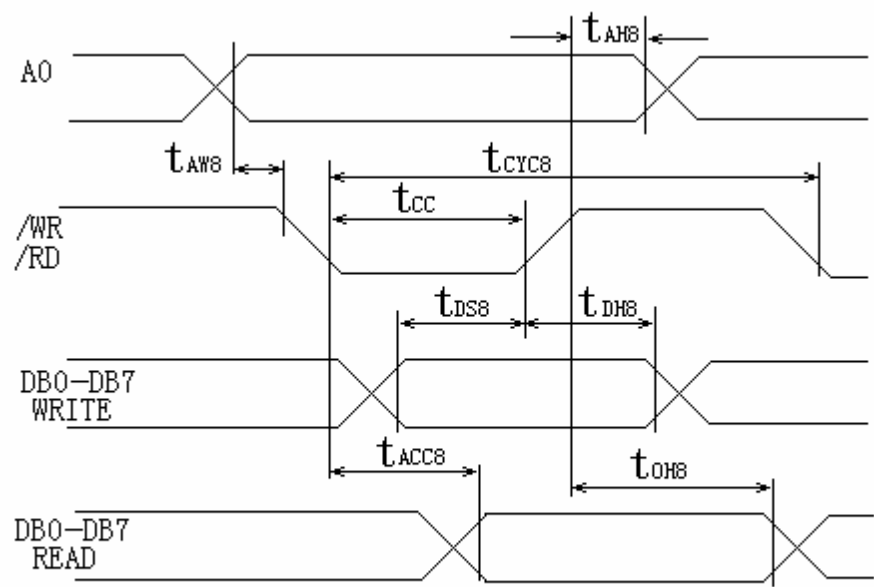
时序 1

时序参数表（VDD=5.0±10%, VSS=0V, Ta=-20~+75℃）

名称	符号	最小值	最大值	单位
地址建立时间	Taw6	20		ns

地址保持时间		Tah6	10		ns
系统时钟周期		Tcy6	1000		ns
E 脉冲宽度	读	Tew	100		ns
	写		80		ns
数据建立时间		Tds6	80		ns
写数据保持时间		Tdh6	10		ns
存取时间		Tacc6		90	ns
读数据保持时间		Tch6	10	60	ns

2. 与 80 系列 MPU 接口读写操作时序图



时序 2

时序参数表（VDD=5.0±10%, VSS=0V, Ta=-20~+70℃）

名称	符号	最小值	最大值	单位
地址建立时间	Taw8	20		ns
地址保持时间	Tah8	10		ns
/RW, /WR 周期	Tcyc8	1000		ns
/RW, /WR 脉冲宽度	Tcc	200		ns
数据建立时间	Tds8	80		ns
写数据保持时间	Tdh8	10		ns
读存取时间	Tacc8		90	ns
读数据保持时间	Tch8	10	60	ns

九、显示数据存储器 (DDRAM) 与地址的对应关系

(显示设定为 1/32 DUTY, 显示起始行为 10th)

Page Address D1, D2=	DATA	Display Pattern	Line Address	Common Output Example Common Output
0, 0	D ₀	0 Page	00	COM16 COM17 COM18 COM19 COM20 COM21 COM22 COM23 COM24 COM25 COM26 COM27 COM28 COM29 COM30 COM31
	D ₁		01	
	D ₂		02	
	D ₃		03	
	D ₄		04	
	D ₅		05	
	D ₆		06	
	D ₇		07	
0, 1	D ₀	1 Page	08	
	D ₁		09	
	D ₂		0A	
	D ₃		0B	
	D ₄		0C	
	D ₅		0D	
	D ₆		0E	
	D ₇		0F	
1, 0	D ₀	2 Page	10	COM 0 COM 1 COM 2 COM 3 COM 4 COM 5 COM 6 COM 7 COM 8 COM 9 COM10 COM11 COM12 COM13 COM14 COM15
	D ₁		11	
	D ₂		12	
	D ₃		13	
	D ₄		14	
	D ₅		15	
	D ₆		16	
	D ₇		17	
1, 1	D ₀	3 Page	18	
	D ₁		19	
	D ₂		1A	
	D ₃		1B	
	D ₄		1C	
	D ₅		1D	
	D ₆		1E	
	D ₇		1F	
Column	A D ₀ =0	00 01 02 03 04 05 06 07	4F	1/16
Address	C D ₀ =1	4F 4E 4D 4C 4B 4A 49 48	00	
Segment Term.		0 1 2 3 4 5 6 7	60 79	

十、应用举例

1、模块接口与 8031 单片机测架连接表（以 OCM12232-1 模块为例）：

模块引脚	符号	8031 单片机引脚
3	V0	与 GND 间接一可调电阻
4	/RESET	P3.0
5	E1	P3.1
6	E2	P3.2
7	R/W	P3.3
8	A0	P3.4
9~16	D0~D7	P1 口

2、简单的测试程序(模拟 68MPU 接口时序)

```
#include <reg52.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

#define uchar unsigned char
#define data 1
#define comm 0
#define f_page 0xb8
#define f_row 0xc0
#define f_line 0x00

sbit rst=P3^0;
sbit e1 =P3^1;
sbit e2 =P3^2;
sbit rw =P3^3;
sbit a0 =P3^4;

sbit bf =P1^7;

void wr_lcd (uchar choel,uchar choe2,uchar data_comm,uchar content);
/*choel 为 1,控制左半屏,choe2 为 1,控制右半屏*/
void chk_busy (uchar choel,uchar choe2);
void delay (unsigned int us);
void delay1 (unsigned int ms);
//void init_lcd (void);
void disphz (uchar code *chn);
void dispzf (uchar code *chn);
void disptu (uchar code *img);
void wrlattice (uchar data1,uchar data2);
```

```

uchar code tab1[]={
/*-- 文字： 中 --*/
0x00, 0x00, 0xFC, 0x08, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0x08, 0xFC, 0x08, 0x00, 0x00,
0x00, 0x00, 0x07, 0x02, 0x02, 0x02, 0x02, 0xFF, 0x02, 0x02, 0x02, 0x02, 0x07, 0x00, 0x00, 0x00,
/*-- 文字： 科 --*/
0x10, 0x12, 0x92, 0x72, 0xFE, 0x51, 0x91, 0x00, 0x22, 0xCC, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
0x04, 0x02, 0x01, 0x00, 0xFF, 0x00, 0x04, 0x04, 0x04, 0x02, 0x02, 0x02, 0xFF, 0x01, 0x01, 0x00,
/*-- 文字： 国 --*/
0x00, 0xFE, 0x02, 0x0A, 0x8A, 0x8A, 0x8A, 0xFA, 0x8A, 0x8A, 0x8A, 0x0A, 0x02, 0xFE, 0x00, 0x00,
0x00, 0xFF, 0x40, 0x48, 0x48, 0x48, 0x48, 0x4F, 0x48, 0x49, 0x4E, 0x48, 0x40, 0xFF, 0x00, 0x00,
/*-- 文字： 技 --*/
0x08, 0x08, 0x88, 0xFF, 0x48, 0x28, 0x00, 0xC8, 0x48, 0x48, 0x7F, 0x48, 0xC8, 0x48, 0x08, 0x00,
0x01, 0x41, 0x80, 0x7F, 0x00, 0x40, 0x40, 0x20, 0x13, 0x0C, 0x0C, 0x12, 0x21, 0x60, 0x20, 0x00,
/*-- 文字： 肇 --*/
0x80, 0x60, 0x1C, 0x95, 0x96, 0x94, 0x9C, 0xC8, 0xA4, 0xAF, 0x94, 0x94, 0x2C, 0x44, 0x44, 0x00,
0x20, 0x22, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0xFF, 0x2A, 0x2A, 0x2A, 0x2F, 0x22, 0x22, 0x20, 0x00,
/*-- 文字： 有 --*/
0x00, 0x04, 0x84, 0x44, 0xE4, 0x34, 0x2C, 0x27, 0x24, 0x24, 0x24, 0xE4, 0x04, 0x04, 0x04, 0x00,
0x02, 0x01, 0x00, 0x00, 0xFF, 0x09, 0x09, 0x09, 0x29, 0x49, 0xC9, 0x7F, 0x00, 0x00, 0x00, 0x00,
/*-- 文字： 庆 --*/
0x00, 0x00, 0xFC, 0x44, 0x44, 0x44, 0x44, 0xC5, 0x7E, 0xC4, 0x44, 0x44, 0x44, 0x44, 0x44, 0x00,
0x40, 0x30, 0x0F, 0x40, 0x20, 0x10, 0x0C, 0x03, 0x00, 0x01, 0x06, 0x18, 0x30, 0x60, 0x20, 0x00,
/*-- 文字： 限 --*/
0xFE, 0x02, 0x32, 0x4E, 0x82, 0x00, 0xFE, 0x4A, 0xCA, 0x4A, 0x4A, 0x4A, 0x7E, 0x00, 0x00, 0x00,
0xFF, 0x00, 0x02, 0x04, 0x03, 0x00, 0xFF, 0x40, 0x20, 0x03, 0x0C, 0x12, 0x21, 0x60, 0x20, 0x00,
/*-- 文字： 金 --*/
0x40, 0x40, 0x20, 0x20, 0x50, 0x48, 0x44, 0xC3, 0x44, 0x48, 0x50, 0x50, 0x20, 0x60, 0x20, 0x00,
0x40, 0x40, 0x42, 0x42, 0x4A, 0x72, 0x42, 0x7F, 0x42, 0x62, 0x5A, 0x42, 0x42, 0x40, 0x40, 0x00,
/*-- 文字： 公 --*/
0x00, 0x00, 0x80, 0x40, 0x30, 0x0E, 0x84, 0x00, 0x00, 0x0E, 0x10, 0x60, 0xC0, 0x80, 0x80, 0x00,
0x00, 0x01, 0x20, 0x70, 0x28, 0x24, 0x23, 0x31, 0x10, 0x10, 0x14, 0x78, 0x30, 0x01, 0x00, 0x00,
/*-- 文字： 鹏 --*/
0x00, 0xFE, 0x12, 0xFE, 0x00, 0xFE, 0x12, 0xFE, 0x00, 0xFC, 0x0E, 0x35, 0x44, 0x7C, 0x00, 0x00,
0x20, 0x1F, 0x21, 0x7F, 0x20, 0x1F, 0x41, 0x7F, 0x08, 0x09, 0x09, 0x29, 0x49, 0x21, 0x1F, 0x00,
/*-- 文字： 司 --*/
0x00, 0x10, 0x92, 0x92, 0x92, 0x92, 0x92, 0x92, 0x92, 0x92, 0x12, 0x02, 0x02, 0xFE, 0x00, 0x00,
0x00, 0x00, 0x1F, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x0F, 0x00, 0x20, 0x40, 0x3F, 0x00, 0x00
};
uchar code tab2[]={
/*-- 文字： 8 --*/
0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00,
/*-- 文字： G --*/
0xC0, 0x30, 0x08, 0x08, 0x08, 0x38, 0x00, 0x00, 0x07, 0x18, 0x20, 0x20, 0x22, 0x1E, 0x02, 0x00,
/*-- 文字： 6 --*/
0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,
/*-- 文字： o --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x20, 0x1F, 0x00,
/*-- 文字： - --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
/*-- 文字： 1 --*/
0x00, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,
/*-- 文字： 7 --*/
0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00,
/*-- 文字： d --*/
0x00, 0x00, 0x00, 0x80, 0x80, 0x88, 0xF8, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x10, 0x3F, 0x20,
/*-- 文字： 5 --*/
0x00, 0xF8, 0x08, 0x88, 0x88, 0x08, 0x08, 0x00, 0x00, 0x19, 0x21, 0x20, 0x20, 0x11, 0x0E, 0x00,
/*-- 文字： e --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x22, 0x22, 0x22, 0x22, 0x13, 0x00,

```

```

/*-- 文字: 8 --*/
0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00,
/*-- 文字: n --*/
0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x3F, 0x21, 0x00, 0x00, 0x20, 0x3F, 0x20,
/*-- 文字: - --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
/*-- 文字: P --*/
0x08, 0xF8, 0x08, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x20, 0x3F, 0x21, 0x01, 0x01, 0x01, 0x00, 0x00,
/*-- 文字: 2 --*/
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00,
/*-- 文字: a --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x19, 0x24, 0x22, 0x22, 0x22, 0x3F, 0x20,
/*-- 文字: 2 --*/
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00,
/*-- 文字: 1 --*/
0x00, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,
/*-- 文字: 2 --*/
0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00,
/*-- 文字: m --*/
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x3F, 0x20, 0x00, 0x3F,
/*-- 文字: 3 --*/
0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,
/*-- 文字: --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
/*-- 文字: 1 --*/
0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,
/*-- 文字: L --*/
0x08, 0xF8, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x20, 0x30, 0x00,
/*-- 文字: 9 --*/
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00,
/*-- 文字: C --*/
0xC0, 0x30, 0x08, 0x08, 0x08, 0x08, 0x38, 0x00, 0x07, 0x18, 0x20, 0x20, 0x20, 0x10, 0x08, 0x00,
/*-- 文字: 6 --*/
0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,
/*-- 文字: M --*/
0x08, 0xF8, 0xF8, 0x00, 0xF8, 0xF8, 0x08, 0x00, 0x20, 0x3F, 0x00, 0x3F, 0x00, 0x3F, 0x20, 0x00
};
uchar code tab3[]={
/*-- 调入了一幅图像: F:\梁\其它\H0C012232. bmp --*/
/*-- 宽度 x 高度=122x32 --*/
0x00, 0x00, 0x00, 0x70, 0xF8, 0xF8, 0xFC, 0xFC, 0xFE, 0xFE, 0xFE, 0xFE, 0xFC, 0xFC, 0xF0, 0xE0,
0xC0, 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, 0x80, 0x40, 0x20, 0x10, 0x48, 0xD4, 0xB4, 0x62, 0xC2, 0x82, 0x03, 0x02, 0x02,
0x02, 0x84, 0xC4, 0xE0, 0xA0, 0xE0, 0x60, 0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xE0, 0xB0, 0xF8, 0xF0, 0xF0, 0xF8, 0xFC, 0x60, 0x60, 0x30,
0x18, 0x00, 0x00, 0x00, 0x80, 0xC0, 0xE0, 0xE0, 0xF0, 0xF0, 0xF0, 0xD8, 0x58, 0x58, 0x58, 0x58,
0x58, 0xD8, 0xD8, 0x98, 0xB0, 0x30, 0x70, 0xE0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
0x07, 0x07, 0x02, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFC, 0xFC, 0xFC,
0xFC, 0xFC, 0xFC, 0xFC, 0xFC, 0xFC, 0xFC, 0xF8, 0xF0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x01, 0x00, 0x04,
0x0D, 0x1B, 0x36, 0x6D, 0xDB, 0xF6, 0xED, 0xFB, 0xFE, 0xFE, 0xFF, 0x7F, 0x1F, 0x07, 0x01, 0x00,
0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0xFC, 0xFE, 0xFE,
0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFC, 0xE0, 0xC0, 0xC0, 0xF8, 0xFE, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0x3E, 0xF8, 0x01, 0x37,
0x0C, 0x00, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0xFF, 0xC7, 0x03, 0x03, 0x03, 0x03, 0x03,
0xFF, 0x3F, 0x0F, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03, 0x0F, 0x3F, 0x7F, 0xFF, 0xEF,
0xBF, 0x7F, 0xFF, 0xC7, 0x07, 0x1F, 0xFE, 0xF8, 0xB0, 0x60, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x04, 0x08, 0x10, 0x20, 0x40, 0x40, 0x80, 0x83,

```

```

0xFF, 0xFF, 0xFF, 0xFF, 0x83, 0x40, 0x40, 0x20, 0x10, 0x08, 0x04, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x03, 0x07, 0x0F, 0x1F, 0x3F, 0x3F, 0x7F, 0xFF,
0xFF, 0x7F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xDF, 0x3F, 0x3F, 0x0F, 0x61, 0x3F, 0x1F, 0x2F, 0x3F,
0x1F, 0x0F, 0x00, 0x0F, 0x07, 0x03, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x1E, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x60, 0x70, 0x3C, 0x1F, 0x07, 0x03, 0x01, 0x01, 0x00, 0x00,
0x01, 0x07, 0x07, 0x01, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7C, 0x44, 0xEC, 0x00,
0x7C, 0x44, 0x7C, 0x00, 0x7C, 0x40, 0x40, 0x00, 0x7C, 0x44, 0x38, 0x00, 0x7C, 0x14, 0x1C, 0x00,
0x7C, 0x40, 0x40, 0x00, 0x78, 0x24, 0x78, 0x00, 0x7C, 0x18, 0x18, 0x7C, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x20, 0x20, 0x20, 0x20, 0x10, 0x04, 0x07, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01,
0x05, 0x03, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

```

```

/*-----*/
void wr_lcd (uchar choel, uchar choe2, uchar data_comm, uchar content)
{
    chk_busy (choel, choe2);
    if (data_comm)
    {
        a0=1;    //data
        rw=0;    //write
    }
    else
    {
        a0=0;    //command
        rw=0;    //write
    }
    if (choel==1)
    {
        P1=content;    //output data or comm
        e1=1;
        ;
        e1=0;
    }
    if (choe2==1)
    {
        P1=content;    //output data or comm
        e2=1;
        ;
        e2=0;
    }
}

void chk_busy (uchar choel, uchar choe2)
{
    if (choel==1)
    {
        a0=0;
        rw=1;
        e1=1;
        while (bf==1)
            delay (2);
        e1=0;
    }
    if (choe2==1)
    {
        a0=0;
        rw=1;
    }
}

```

```

    e2=1;
    while(bf==1)
        delay(2);
    e2=0;
}
}
void delay (unsigned int us)    //delay time
{
    while(us--);
}
void delay1 (unsigned int ms)
{
    unsigned int i, j;
    for(i=0; i<ms; i++)
        for(j=0; j<1000; j++)
            ;
}
/*-----初始化-----*/
void init_lcd (void)
{
    rst=1;        //68 系列模式, 复位触发(L->H, 保持电平:H), 80 系列与其相反
    ;
    wr_lcd (1, 1, comm, 0xe2);
    wr_lcd (1, 1, comm, 0xa4);
    wr_lcd (1, 1, comm, 0xa9);
    wr_lcd (1, 1, comm, 0xa0);
    wr_lcd (1, 1, comm, 0xee);
    wr_lcd (1, 1, comm, 0xaf);
}
/*-----显示汉字(16x16)-----*/
void disphz (uchar code *chn)
{
    uchar i, j, k;
    for(k=0; k<3; k++)
    {
        for(j=0; j<4; j++)
        {
            wr_lcd (1, 0, comm, f_page+j);
            wr_lcd (1, 0, comm, f_row);
            wr_lcd (1, 0, comm, f_line+12+k*16);
            for(i=0; i<16; i++)
                wr_lcd (1, 0, data, chn[k*64+j*16+i]);
        }
    }
    for(k=3; k<6; k++)
    {
        for(j=0; j<4; j++)
        {
            wr_lcd (0, 1, comm, f_page+j);
            wr_lcd (0, 1, comm, f_row);
            wr_lcd (0, 1, comm, f_line+k*16-48);
            for(i=0; i<16; i++)
                wr_lcd (0, 1, data, chn[k*64+j*16+i]);
        }
    }
}
/*-----显示字符(8x16)-----*/
void dispzf (uchar code *eng)
{

```

```

uchar i, j, k;
for(k=0;k<7;k++)
{
    for(j=0;j<4;j++)
    {
        wr_lcd (1, 0, comm, f_page+j);
        wr_lcd (1, 0, comm, f_row);
        wr_lcd (1, 0, comm, f_line+5+k*8);
        for(i=0;i<8;i++)
            wr_lcd (1, 0, data, eng[k*32+j*8+i]);
    }
}
for(k=7;k<14;k++)
{
    for(j=0;j<4;j++)
    {
        wr_lcd (0, 1, comm, f_page+j);
        wr_lcd (0, 1, comm, f_row);
        wr_lcd (0, 1, comm, f_line+k*8-56);
        for(i=0;i<8;i++)
            wr_lcd (0, 1, data, eng[k*32+j*8+i]);
    }
}
}
/*-----画图形-----*/
void disptu (uchar code *img)
{
    uchar i, j;
    for(j=0;j<4;j++)
    {
        wr_lcd (1, 1, comm, f_page+j);
        wr_lcd (1, 1, comm, f_row);
        wr_lcd (1, 1, comm, f_line);
        for(i=0;i<122;i++)
        {
            if(i<61)
            {wr_lcd (1, 0, comm, f_line+i);
              wr_lcd (1, 0, data, img[j*122+i]);}
            else
            {wr_lcd (0, 1, comm, f_line+i-61);
              wr_lcd (0, 1, data, img[j*122+i]);}
        }
    }
}
/*-----写点阵-----*/
void wrlattice (uchar data1, uchar data2)
{
    uchar i, j;
    for(j=0;j<4;j++)
    {
        wr_lcd (1, 1, comm, f_page+j);
        wr_lcd (1, 1, comm, f_row);
        wr_lcd (1, 1, comm, f_line);
        for(i=0;i<61;i=i+2)
        {
            wr_lcd (1, 1, data, data1);
            wr_lcd (1, 1, data, data2);
        }
    }
}

```

```
}
/*-----主程序-----*/
void main ()
{
    SP=0x5f;
    init_lcd ();
    while (1)
    {
        wrlattice (0x00,0x00);
        disphz (tab1);
        delay1 (300);
        wrlattice (0xff,0x00);
        delay1 (300);
        wrlattice (0x00,0x00);
        dispzf (tab2);
        delay1 (300);
        wrlattice (0x33,0x33);
        delay1 (300);
        disptu (tab3);
        delay1 (300);
    }
}
```