

# 12864(0108)系列图形点阵液晶显示模块使用指引

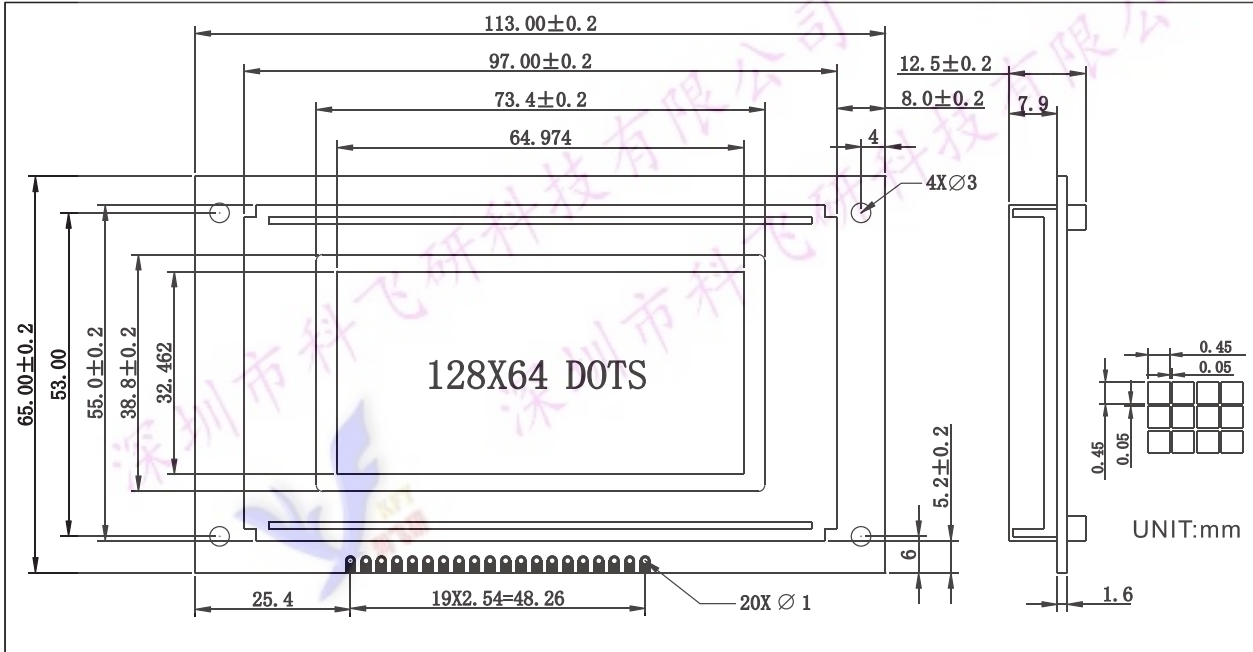
---

深圳市科飞研科技有限公司

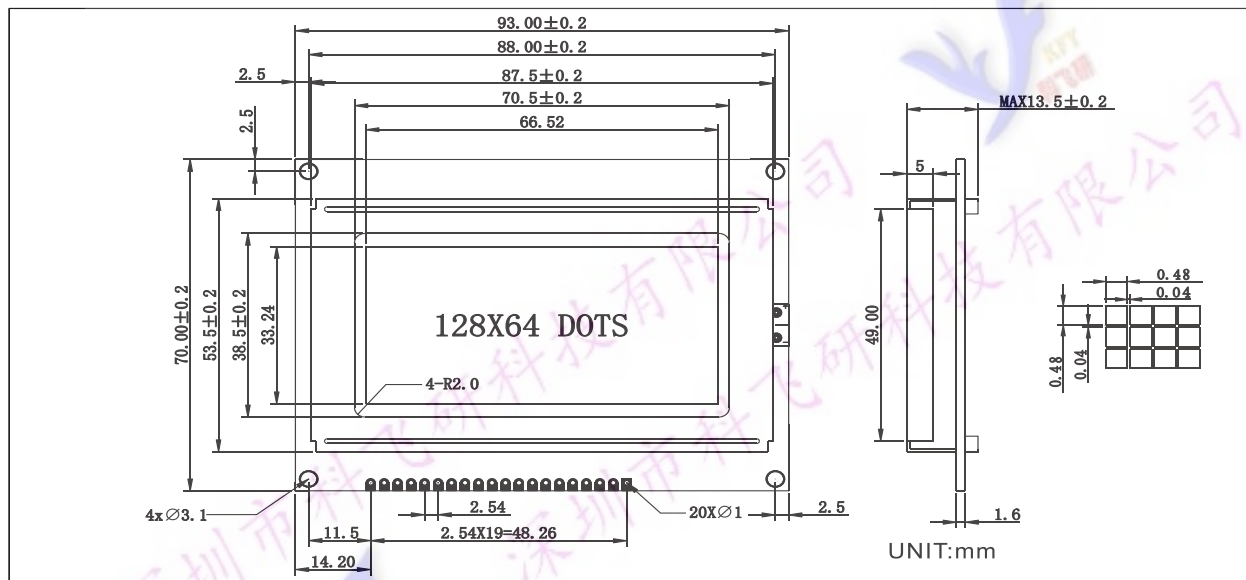
## 一、12864 液晶显示模块概述

12864 液晶显示模块是 128×64 点阵型液晶显示模块，可显示各种字符及图形，可与 CPU 直接接口，具有 8 位标准数据总线、6 条控制线及电源线。采用 KS0108 控制 IC。

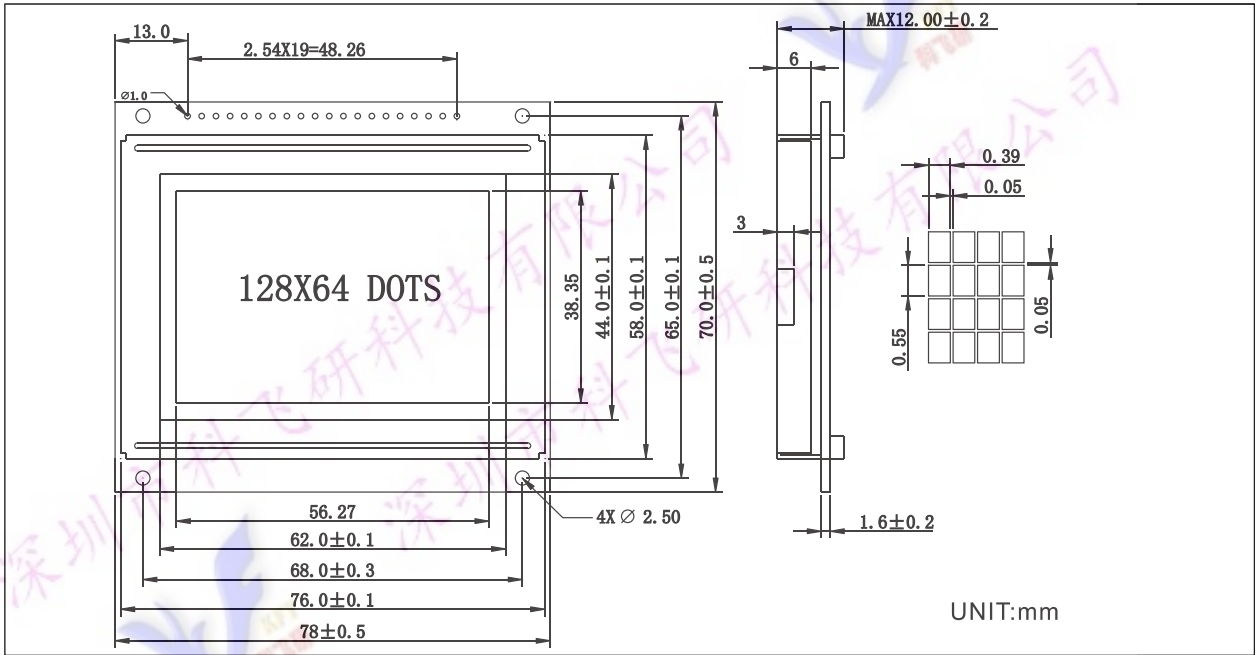
## 二、外形尺寸图



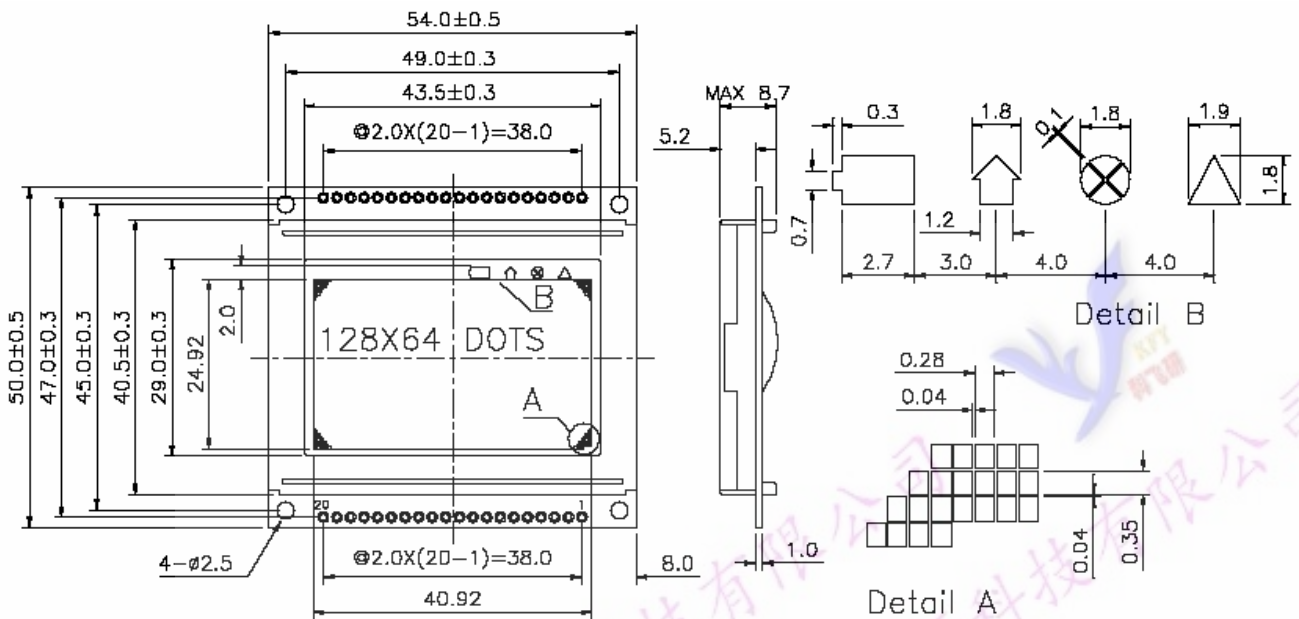
12864-3外形尺寸图



12864-1外形尺寸图



G12864A外形尺寸图



G12864I外形尺寸图

### 三、最大工作范围

- 1、逻辑工作电压(Vcc): 4.5~5.5V
- 2、电源地(GND): 0V
- 3、工作温度(Ta): -20~70℃ (宽温)
- 4、保存温度(Tstg): -30~80℃

### 四、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: 5.0mAmx (注: 不开背光的情况下)

## 五、接口说明

### 1. 接口说明表

管脚号	管脚	方向	说明
1	VSS	-	逻辑电源地。
2	VDD	-	逻辑电源+5V。
3	V0	I	LCD 调整电压, 应用时接 10K 电位器可调端
4	RS	I	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 将送入指令寄存器执行。
5	R/W	I	读\写选择: 高电平: 读数据; 低电平: 写数据。
6	E	I	读写使能, 高电平有效, 下降沿锁定数据。
7	DB0	I/O	数据输入输出引脚。
8	DB1	I/O	数据输入输出引脚。
9	DB2	I/O	数据输入输出引脚。
10	DB3	I/O	数据输入输出引脚。
11	DB4	I/O	数据输入输出引脚。
12	DB5	I/O	数据输入输出引脚。
13	DB6	I/O	数据输入输出引脚。
14	DB7	I/O	数据输入输出引脚。
15	CS1	I	片选择信号, 高电平时选择左半屏。
16	CS2	I	片选择信号, 高电平时选择右半屏。
17	/RET	I	复位信号, 低电平有效。
18	VEE	0	LCD 驱动, 负电压输出, 对地接 10K 电位器
19	LEDA	-	背光电源, LED+ (5V)。
20	LEDK	-	背光电源, LED- (0V)。

### 2. 接口说明表

管脚号	管脚	方向	说明
1	/CS1	I	片选择信号, 低电平时选择左半屏。
2	/CS2	I	片选择信号, 低电平时选择右半屏。
3	VSS	-	逻辑电源地。
4	VDD	-	逻辑电源。
5	V0	I	LCD 调整电压, 接 10K 电位器的中端
6	RS	I	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 将送入指令寄存器执行。
7	R/W	I	读\写选择, 高电平: 读数据; 低电平: 写数据。
8	E	I	读写使能, 高电平有效, 下降沿锁定数据。
9	DB0	I/O	数据输入输出引脚。
10	DB1	I/O	数据输入输出引脚。
11	DB2	I/O	数据输入输出引脚。
12	DB3	I/O	数据输入输出引脚。
13	DB4	I/O	数据输入输出引脚。
14	DB5	I/O	数据输入输出引脚。
15	DB6	I/O	数据输入输出引脚。
16	DB7	I/O	数据输入输出引脚。
17	/RET	I	复位信号, 低电平有效。
18	VEE	0	LCD 驱动负电压输出, 对地接一个 10K 电位器
19	LEDA	-	背光电源, LED+ (5V)。
20	LEDK	-	背光电源, LED- (0V)。

R/W	RS	DB7	DB6	DB2	DB1	DB0
L	L	L	L	H	H	H

功能：设置屏幕显示开/关。

DB0=H，开显示；DB0=L，关显示。不影响显示 RAM(DD RAM) 中的内容。

## 2、设置显示起始行

CODE:	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	H	H	行地址 (0~63)					

功能：执行该命令后，所设置的行将显示在屏幕的第一行。显示起始行是由 Z 地址计数器控制的，该命令自动将 A0-A5 位地址送入 Z 地址计数器，起始地址可以是 0-63 范围内任意一行。Z 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、设置页地址

CODE:	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	H	L	H	H	H	页地址 (0~7)		

功能：执行本指令后，下面的读写操作将在指定页内，直到重新设置。页地址就是 DD RAM 的行地址，页地址存储在 X 地址计数器中，A2-A0 可表示 8 页，读写数据对页地址没有影响，除本指令可改变页地址外，复位信号(RST)可把页地址计数器内容清零。

DDRAM 地址映像表

		Y 地址							
		0	1	2	.....	61	62	63	
DB0	∫	PAGE0						X=0	
DB7									
DB0	∫	PAGE1						X=1	
DB7									
		⋮							
		⋮							
		⋮							
		⋮							
DB0	∫	PAGE6						X=7	
DB7									
DB0	∫	PAGE7						X=8	
DB7									

## 4、设置列地址

CODE:	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	H	列地址 (0~63)					

功能：DD RAM 的列地址存储在 Y 地址计数器中，读写数据对列地址有影响，在对 DD RAM 进行读写操作后，Y 地址自动加一。

## 5、状态检测



CODE: R/W RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

H	L	BF	L	ON/OFF	RST	L	L	L	L
---	---	----	---	--------	-----	---	---	---	---

功能：读忙信号标志位 (BF)、复位标志位 (RST) 以及显示状态位 (ON/OFF)。

BF=H: 内部正在执行操作; BF=L: 空闲状态。

RST=H: 正处于复位初始化状态; RST=L: 正常状态。

ON/OFF=H: 表示显示关闭; ON/OFF=L: 表示显示开。

### 6、写显示数据

CODE: R/W RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

L	H	D7	D6	D5	D4	D3	D2	D1	D0
---	---	----	----	----	----	----	----	----	----

功能：写数据到 DD RAM, DD RAM 是存储图形显示数据的, 写指令执行后 Y 地址计数器自动加 1。D7-D0 位数据为 1 表示显示, 数据为 0 表示不显示。写数据到 DD RAM 前, 要先执行“设置页地址”及“设置列地址”命令。

### 7、读显示数据

CODE: R/W RS DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

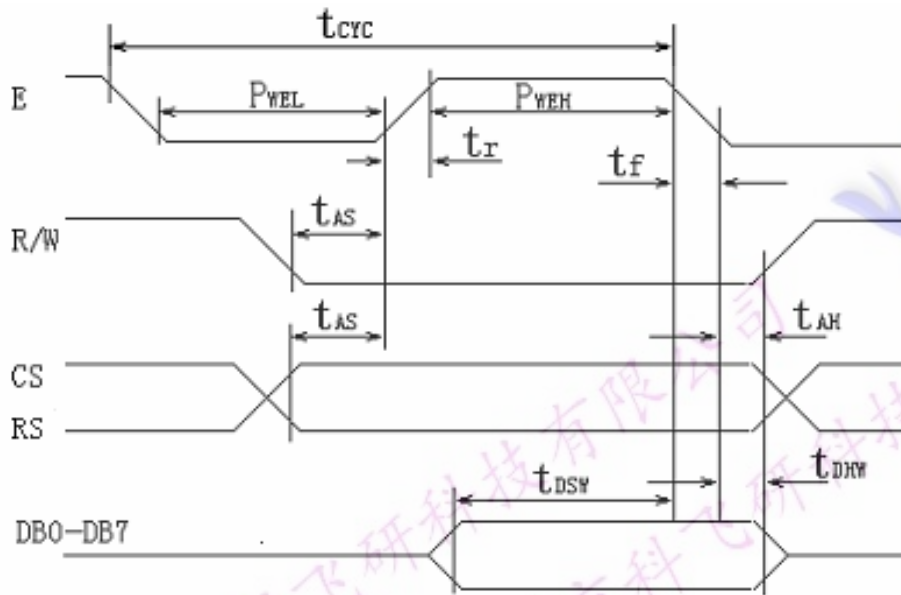
H	H	D7	D6	D5	D4	D3	D2	D1	D0
---	---	----	----	----	----	----	----	----	----

功能：从 DD RAM 读数据, 读指令执行后 Y 地址计数器自动加 1。从 DD RAM 读数据前要先执行“设置页地址”及“设置列地址”命令。

注：设置列地址后, 首次读 DDRAM 中数据时, 须连续读操作两次, 第二次才为正确数据。读内部状态则不须要此操作。

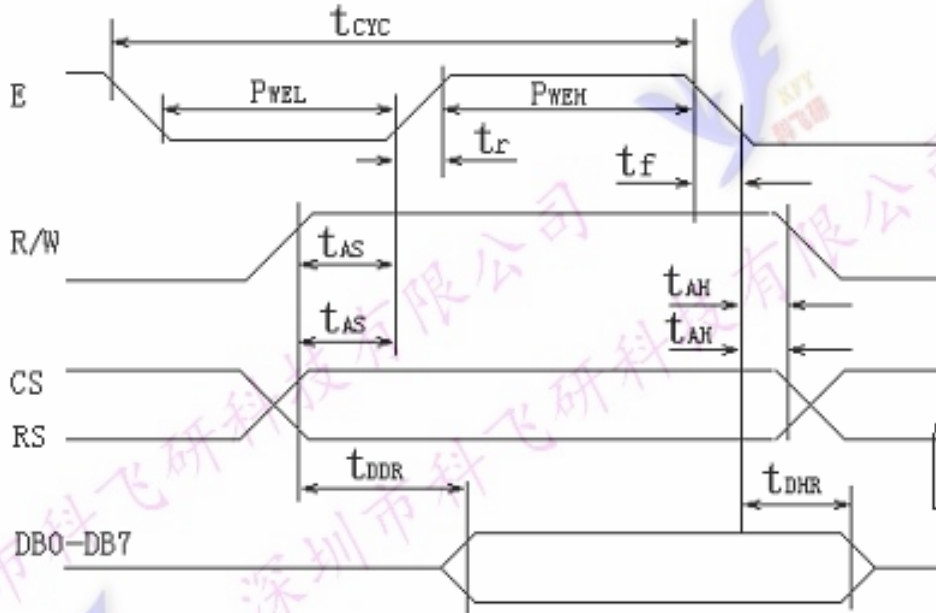
## 七、接口时序

### 1. 写操作时序



时序 1

### 2. 读操作时序



时序 2

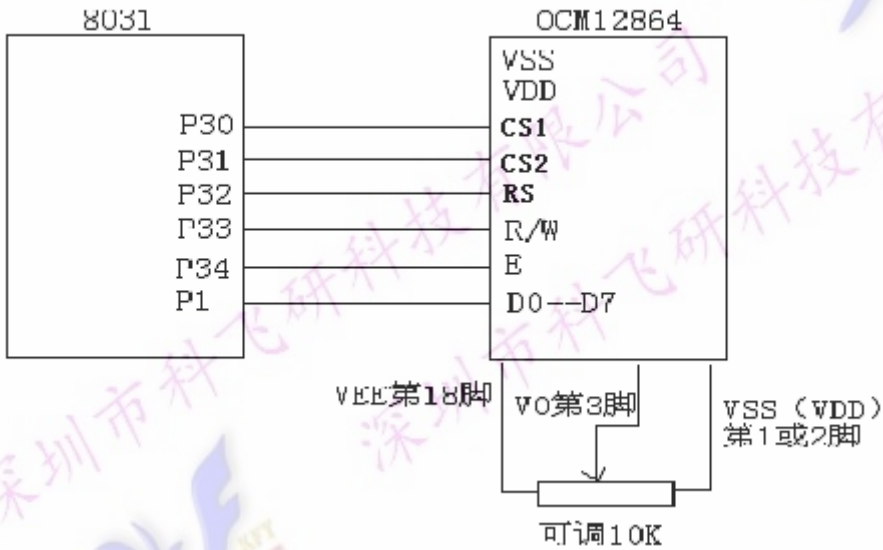
时序参数表:

名称	符号	最小值	典型值	最大值	单位
E 周期时间	Tcyc	1000			ns
E 高电平宽度	Pweh	450			ns
E 低电平宽度	Pwel	450			ns
E 上升时间	Tr			25	ns
E 下降时间	Tf			25	ns
地址建立时间	Tas	140			ns
地址保持时间	taw	10			ns
数据建立时间	Tdsw	200			ns
数据延迟时间	Tddr			320	ns
写数据保持时间	Tdhw	10			ns
读数据保持时间	Tdhr	20			ns

### 八、屏幕显示与 DDRAM 地址映射关系

	Y1	Y2	Y3	Y4	.....	Y62	Y63	Y64		
X=0	Line 0	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB0
	Line 1	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB1
	Line 2	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB2
	Line 3	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB3
	Line 4	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB4
	Line 5	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB5
	Line 6	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB6
	Line 7	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB7
					.....					
					.....					
X=7	Line60	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB4
	Line61	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB5
	Line62	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB6
	Line63	1/0	1/0	1/0	1/0	.....	1/0	1/0	1/0	DB7

## 九、测试硬件电路



## 十、读写模块程序举例

## ● 写指令子程序 (INST)

```

CLR    RS
CLR    R_W
MOV    P1,    A
SETB   E
NOP
CLR    E
RET

```

## ● 写数据子程序 (DATA)

```

SETB   RS
CLR    R_W
MOV    P1,    A
SETB   E
NOP
CLR    E
RET

```

## ● 写一页显示 RAM 数据 (假设指令子程序为 INST, 数据子程序为 DATA)

```

MOV    A,    #0B8H
LCALL INST                ; 置页地址为 0 页
MOV    A,    #40H
LCALL INST                ; 置列地址为 0 列
MOV    R2,   #40H
MOV    R1,   #00H
MOV    DPTR, #ADDR
LOOP: MOV    A,    R1
      MOVC   A,    @A+DPTR
      LCALL DATA
      INC    R1
      DJNZ  R2,   LOOP

```



```

//12864参考程序
#include <reg52.h>

#define uint    unsigned int
#define uchar   unsigned char
#define disp_off 0x3e
#define disp_on  0x3f
#define disp_x   0xb8
#define disp_z   0xc0
#define disp_y   0x40
#define comm    0
#define dat     1
#define data_ora P1    //MCU P1<-----> LCM

sbit di =P3^0; //Data or Instrument Select,H:写数据,L:写指令
sbit rw =P3^1; //Write or Read,H:read,L:write
sbit e  =P3^2; //读写使能
sbit cs1=P3^3; //cs1=H,选择左半屏
sbit cs2=P3^4; //cs2=H,选择右半屏
sbit rst=P3^5; //Lcm reset,低有效
sbit bf =P1^7;
sbit res=P1^4;

void chk_busy (void);

uchar code hz11[]={
/*-- 宋体12; 此字体下对应的点阵为：宽x高=16x16  --*/
/*-- 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: 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: 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: 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: 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: 司 --*/
0x00,0x10,0x92,0x92,0x92,0x92,0x92,0x92,0x92,0x12,0x02,0x02,0xFE,0x00,0x00,
0x00,0x00,0x1F,0x04,0x04,0x04,0x04,0x04,0x04,0x0F,0x00,0x20,0x40,0x3F,0x00,0x00};

uchar code hz12[]={
/*-- 宋体12; 此字体下对应的点阵为：宽x高=16x16  --*/
/*-- 文字: 液 --*/
0xEF,0x9E,0xF9,0x1F,0xE7,0x7B,0x1B,0xE3,0x7B,0x9A,0x41,0xDB,0x5B,0x9B,0xFB,0xFF,
0xFB,0xFB,0x00,0xFF,0xFE,0xFF,0x00,0xBE,0xDE,0xED,0xF3,0xE4,0x9E,0x3F,0xBF,0xFF,
/*-- 文字: 晶 --*/
0xFF,0xFF,0xFF,0xFF,0x81,0xD5,0xD5,0xD5,0xD5,0xD5,0xD5,0x81,0xFF,0xFF,0xFF,0xFF,
0xFF,0x80,0xDA,0xDA,0xDA,0xDA,0x80,0xFF,0xFF,0x80,0xDA,0xDA,0xDA,0xDA,0x80,0xFF,
/*-- 文字: 显 --*/
0xFF,0xFF,0xFF,0xC1,0xD5,0x15,0xD5,0xD5,0xD5,0x15,0xD5,0xC1,0xFF,0xFF,0xFF,0xFF,
0xDF,0xDE,0xDD,0xD3,0xDF,0xC0,0xDF,0xDF,0xDF,0xC0,0xD7,0xDB,0xDC,0xDF,0xDF,0xFF,
/*-- 文字: 示 --*/
0xFF,0xDF,0xDF,0xDD,0xDD,0xDD,0xDD,0x1D,0xDD,0xDD,0xDD,0xDD,0xDF,0xDF,0xFF,
0xEF,0xF7,0xFB,0xFC,0xFF,0xBF,0x7F,0x80,0xFF,0xFF,0xFE,0xFD,0xF3,0xE7,0xFF,0xFF,
/*-- 文字: 器 --*/
0xBF,0xBF,0xB0,0xB6,0xB6,0x36,0x30,0x8F,0x3F,0x30,0xB6,0xA6,0x96,0xB0,0xFF,0xFF,
0xFD,0xFD,0x81,0xBA,0xBA,0xBB,0x83,0xFF,0x83,0xBB,0xBA,0xBA,0x81,0xF9,0xFD,0xFF};

```

```

uchar code tab31[]={
/*-- 调入图像：F:\?\?\? --*/
/*-- 宽度x高度=32x32 --*/

};

uchar code tab5[]={
/*-- 调入图像：F:\?\?\? --*/
/*-- 宽度x高度=128x64 --*/

};

/*-----延时子程序-----*/
void delay (uint us)
{
while(us--);
}
void delay1 (uint ms)
{
uint i,j;
for(i=0;i<ms;i++)
for(j=0;j<1000;j++)
;
}
/*-----写数据或命令到LCD-----*/
void wr_lcd (uchar dat_comm,uchar content)
{
chk_busy ();
di=dat_comm;
rw=0;
data_ora=content;
e=1;
;
e=0;
}
/*-----读LCD数据-----*/
uchar rd_lcd ()
{
uchar rddata;
chk_busy ();
di=1;
rw=1;
e=1;
;
rddata=data_ora;
e=0;
return rddata;
}
void chk_busy (void)
{
data_ora=0xff;
di=0;
rw=1;
;
e=1;
while(bf||res==1);
e=0;
}

/*-----写点阵-----*/
void lat_disp (uchar data1,uchar data2)
{
uchar i,j;
cs1=1;cs2=1;
wr_lcd (comm,disp_on);
for(j=0;j<8;j++)
{
wr_lcd (comm,disp_x+j);
wr_lcd (comm,disp_z);
wr_lcd (comm,disp_y);
for(i=0;i<32;i++)
{
wr_lcd (dat,data1);
wr_lcd (dat,data2);
}
}
}
}

```

