

G24064A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
深圳市科飞研科技有限公司

深圳市科飞研科技有限公司

目录

1、 产品简介.....	2
2、 引用文件.....	2
3、 机械特性.....	2
4、 产品框图.....	2
5、 光电特性.....	3
6、 极限参数.....	3
7、 接口时序.....	3
8、 直流特性.....	4
9、 引脚描述.....	5
10、 命令描述、.....	6
11、 附录.....	12

1、产品简介

主要工艺: SMT
 显示内容: 240X64 点阵
 显示模式: STN, POSITIVE
 驱动条件: 1/128Duty 1/12Bias
 视向: 6: 00
 背光: LED
 工作温度: -20℃~+70℃
 储存温度: -30℃~+80℃
 控制 IC: T6963 C

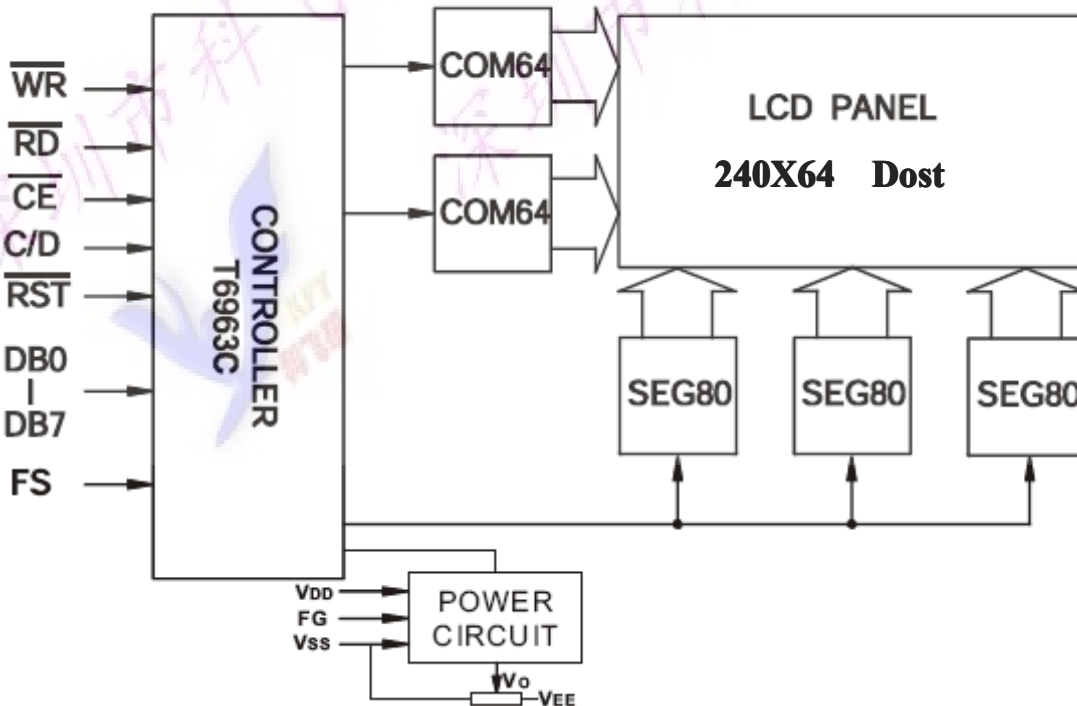
2、引用文件

T6963C 规格书

3、机械特性

类别	标准值	单位
模块	180.0(w)X65.0(h)X12.0 (t)	mm
有效显示区	132.0(w)X39.0(h)	mm
点大小	0.49(w)X0.49(h)	mm
点间隙	0.04(w)X0.04(h)	mm

4、产品框图



5、光电特性

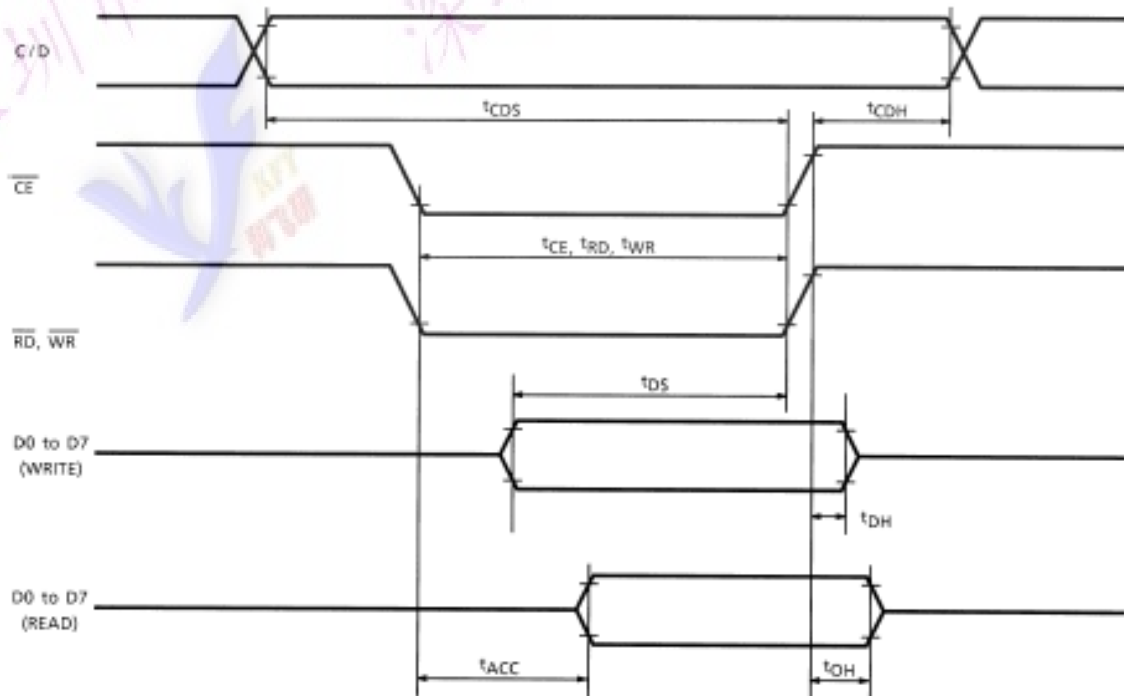
类别	符号	条件	最小值	TYP	最大值	单位
驱动电压	Vop.	25°C	8.8	12.0	15.0	V
响应时间	Ton	25°C	—	127	200	Ms
对比度	Toff	25°C	—	263	360	Ms
	CR	25°C	—	9	—	—
视角范围		25°C	—	60	—	DEG
交叉效应		25°C	—	1.2	—	—

6、极限参数

参数	符号	最小值	最大值	单位
逻辑电压	Vdd	-0.3	+5.5	V
驱动电压	Vout, VO	-0.3	-18	V
工作温度	Top	-20	+70	°C
存储温度	Tst	-30	+80	°C

7、接口时序

Bus Timing



Test Conditions (Unless Otherwise Noted, $V_{DD} = 5.0\text{ V} \pm 10\%$, $V_{SS} = 0\text{ V}$, $T_a = -20\text{ to }75^\circ\text{C}$)

Item	Symbol	Test Conditions	Min	Max	Unit
C / D Set-up Time	t_{CDS}	—	100	—	ns
C / D Hold Time	t_{CDH}	—	10	—	ns
\overline{CE} , \overline{RD} , \overline{WR} Pulse Width	t_{CE} , t_{RD} , t_{WR}	—	80	—	ns
Data Set-up Time	t_{DS}	—	80	—	ns
Data Hold Time	t_{DH}	—	40	—	ns
Access Time	t_{ACC}	—	—	150	ns
Output Hold Time	t_{OH}	—	10	50	ns

8、直流特性

Test Conditions (Unless Otherwise Noted, $V_{SS} = 0\text{ V}$, $V_{DD} = 5.0\text{ V} \pm 10\%$, $T_a = -20\text{ to }75^\circ\text{C}$)

Item	Symbol	Test Circuit	Test Condition	Min	Typ.	Max	Unit	Pin Name
Operating Voltage	V_{DD}	—	—	4.5	5.0	5.5	V	V_{DD}
Input	H Level	V_{IH}	—	$V_{DD} - 2.2$	—	V_{DD}	V	Input pins
	L Level	V_{IL}	—	0	—	0.8	V	Input pins
Output Voltage	H Level	V_{OH}	—	$V_{DD} - 0.3$	—	V_{DD}	V	Output pins
	L Level	V_{OL}	—	0	—	0.3	V	Output pins
Output Resistance	H Level	R_{OH}	$V_{OUT} = V_{DD} - 0.5\text{ V}$	—	—	400	Ω	Output pins
	L Level	R_{OL}	$V_{OUT} = 0.5\text{ V}$	—	—	400	Ω	Output pins
Input Pull-up Resistance	RPU	—	—	50	100	200	k Ω	(Note 1)
Operating Frequency	f_{OSC}	—	—	0.4	—	5.5	MHz	
Current Consumption (Operating)	$I_{DD(1)}$	—	$V_{DD} = 5.0\text{ V}$ $f_{OSC} = 3.0\text{ MHz}$ (Note 2)	—	3.3	6	mA	V_{DD}
Current Consumption (Halt)	$I_{DD(2)}$	—	$V_{DD} = 5.0\text{ V}$	—	—	3	μA	V_{DD}

Note 1: Applied $\overline{T1}$, $\overline{T2}$, \overline{RESET}

Note 2: MDS = L, MD0 = L, MD1 = L, MD2 = H, MD3 = H, FS0 = L, FS1 = L, $\overline{SDSEL} = L$, $\overline{DUAL} = H$,
D7 to D0 = LHLHLHLH

9、引脚描述

引脚编号	引脚名称	引脚功能描述
1	FG	模块结构地
2	VSS	逻辑电源地
3	VDD	逻辑电源正
4	V0	LCD 驱动电压
5	/WR	当接口定义为 8080 接口时，/WR 为写入控制脚
6	/RD	当接口定义为 8080 接口时，/RD 为读控制脚，低有效
7	/CE	片选输入（辅助 IC）
8	C/D	命令/数据选择
9	NC	悬空
10	Reset	复位信号
11	DB0	数据输入输出引脚
12	DB1	数据输入输出引脚
13	DB2	数据输入输出引脚
14	DB3	数据输入输出引脚
15	DB4	数据输入输出引脚
16	DB5	数据输入输出引脚
17	DB6	数据输入输出引脚
18	DB7	数据输入输出引脚
19	FS	字体选择
20	VEE	LCD 驱动输出
21	LED+	背光电源正
22	LED-	背光电源负

10、命令描述、 指令表：

<i>COMMAND</i>	CODE	D1	D2	FUNCTION
<i>REGISTERS</i> <i>SETTING</i>	00100001	X address	Y address	Set Cursor Pointer
	00100010	Data	00H	Set Offset Register
	00100100	Low address	High address	Set Address Pointer
<i>SET CONONTROL</i> <i>WORD</i>	01000000	Low address	High address	Set Text Home Address
	01000001	Columns	00H	Set Text Area
	01000010	Low address	High address	Set Graphic Home Address
	01000011	Columns	00H	Set Graphic Area
<i>MODE SET</i>	1000x000	--	--	OR mode
	1000x001	--	--	EXOR mode
	1000x011	--	--	AND mode
	1000x100	--	--	Text Attribute mode
	10000xxx	--	--	Internal CG ROM mode
	10001xxx	--	--	External CG RAM mode
<i>DISPLAY MODE</i>	10010000	--	--	Display off
	1001xx10	--	--	Cursor on, blink off
	1001xx11	--	--	Cursor on, blink on
	100101xx	--	--	Text on, graphic off
	100110xx	--	--	Text off, graphic on
	100111xx	--	--	Text on , graphic on
<i>CURSOR</i> <i>PATTERN ELECT</i>	10100000	--	--	1-line cursor
	10100001	--	--	2-line cursor
	10100010	--	--	3-line cursor
	10100011	--	--	4-line cursor
	10100100	--	--	5-line cursor
	10100101	--	--	6-line cursor
	10100110	--	--	7-line cursor
	10100111	--	--	8-line cursor
<i>DATA AUTO</i> <i>READ/WRITE</i>	10110000	--	--	Set Data Auto Write
	10110001	--	--	Set Data Auto Read
	10110010	--	--	Auto Reset
<i>DATA</i> <i>READ/WRITE</i>	11000000	--	--	Data Write and Increment
	11000001	--	--	Data Read and Increment
	11000010	--	--	Data Write and Decrement
	11000011	--	--	Data Read and Decrement
	11000100	--	--	Data Write and Non variable
	11000101	--	--	Data Read and Non variable
<i>SCREEN PEEK</i>	11100000	--	--	Screen Peek
<i>SCREEN</i> <i>COPY</i>	11101000	--	--	Screen Copy
<i>BIT SET/RESET</i>	11110xxx	--	--	Bit reset
	11111xxx	--	--	Bit set
	1111x000	--	--	Bit0 (LSB)
	1111x001	--	--	Bit1
	1111x010	--	--	Bit2
	1111x011	--	--	Bit3
	1111x100	--	--	Bit4
	1111x101	--	--	Bit5
	1111x110	--	--	Bit6
	1111x111	--	--	Bit7 (MSB)

指令介绍:

1、指针设置指令，格式如下：

D1, D2	0 0 1 0 0 N2 N1 N0
---------------	---------------------------

D1, D2为第一和第二个参数，后一个字节为指令代码，根据N0, N1, N2的取值，该指令有三种含义（N0, N1, N2不能有两个同时为1）

D1	D2	指令代码	功能
水平位置 (低七位有效)	垂直位置 (低五位有效)	21H (N0=1)	光标指针设置
地址 (低五位有效)	00H	22H (N1=1)	CGRAM偏置地址设置
低字节	高字节	24H (N2=1)	地址指针位置

- (1) 光标指针设置：D1表示光标在实际液晶屏上离左上角的横向距离（字符数），D2表示（纵向距离（字符行）
- (2) CGRAM偏置地填寄存器设置：设置了CGRAM在显示64KRAM内的高5位地址，CGRAM的实际地址为：

A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

偏置地址: C4 C3 C2 C1 C0

字符代码: D7 D6 D5 D4 D3 D2 D1 D0

行地址指针: +) R2 R1 R0

实际地址: V15 V14 V13 V12 V11 V10 V9 V8 V7 V6 V5 V4 V3 V2 V1 V0

- (3) 地址指针设置：设置将要进行操作的显示缓冲区（RAM）的一个单元地址，D1、D2为该单元地址的低位和高位地址。

2、显示区域设置，指令格式为：

D1, D2	0 0 1 0 0 0 0 N1 N0
---------------	----------------------------

根据N1, N0的不同取值，该指令有四种指令功能形式：

N1 N0	D1	D2	指令代码	功能
0 0	低字节	高字节	40H	文本区首址
0 1	字节数	00H	41H	文本区宽度（字节数/行）
1 0	低字节	高字节	42H	图形区首址

1	1	字节数	00H	43H	图形宽度（字节数/行）
---	---	-----	-----	-----	-------------

文本区和图形区首地址为对应显示屏上左上角字符位或字节位，修改该地址可以产生卷动效果。D1, D2分别为该地址的低位和高位字节。

文本区宽度（字节数/行）设置和图形区宽度（字节数/行）设置用于调整一行显示所占显示RAM的字节数，从而确定显示屏与显示RAM单元的对应关系。

T6963C硬件设置的显示窗口宽度是指T6963C扫描驱动的有效列数。需说明的是当硬件设置6×8字体时，图形显示区单元的低6位有效，对应显示屏上6×1显示位。

3、显示方式设置，指令格式为：

无参数	1 0 0 0 N3 N2 N1 N0
------------	----------------------------

N3：字符发生器选择位。

N3=1为外部字符发生器有效，此时内部字符发生器被屏蔽，字符代码全部提供给外部字符发生器使用，字符代码为00H~FFH。

N3=0为CGROM即内部字符发生器有效，限于CGROM字符代码为00H~7FH。因此选用80H~FFH字符代码时，将自动选择CGRAM。

N2~N0：合成显示方式控制位，其组合功能如下表：

N2 N1 N0	合成方式
0 0 0	逻辑“或”合成
0 0 1	逻辑“异或”合成
0 1 1	逻辑“与”合成
1 0 0	文本特征

当设置文本方式和图形方式无能打开时，上述合成显示方式设置才有效。其中的文本特征方式是指将图形区改为文本特征区。该区大小与文本区相同，每个字节作为对应文本区的每个字符显示的特征，包括字符显示与不显示、字符闪烁及字符的“负向”显示。通过这种方式，T6963C可以控制每个字符的文本特征。文本特征区内，字符的文本特征码由一个字节的低四位组成，即：

D7	D6	D5	D4	D3	D2	D1	D0
*	*	*	*	D3	D2	D1	D0

d3:字符闪烁控制位，d3=1为闪烁，d3=0为不闪烁；

d2~d0的组合如下：

d2 d1 d0	显示效果
0 0 0	正常显示
1 0 1	负向显示
0 1 1	禁止显示，空白

启用文本特征方式时可在原有图形区和文本区外用图形区域设置指令另开一区作为文本特征区，以保持原图形区的数据。显示缓冲区可划分如下：

单屏结构:

SAD1 SAD1' SAD2	图形显示区	显示 缓冲 区 R AM
	文本特征区	
	文本显示区	
	CGRAM (2K)	

4、显示开关，指令格式如下：

无参数	1 0 0 1 N3 N2 N1 N0
------------	----------------------------

N0: 1/0, 光标闪烁启用/禁止

N1: 1/0, 光标显示启用/禁止

N2: 1/0, 文本显示启用/禁止

N3: 1/0, 图形显示启用/禁止

5、光标形状选择，指令格式如下：

无参数	1 0 1 0 0 N2 N1 N0
------------	---------------------------

6、数据自动读、写方式设置：

无参数	1 0 1 1 0 0 N1 N0
------------	--------------------------

该指令执行后，MPU可以连续地读、写显示缓冲区RAM的数据，每读、写一次，地址指针自动增1。自动读、写结束时，必须写入自动结束命令以使T6963C退出自动读、写状态，开始接受其它指令。

N1, N0组合功能如下：

N1 N0	指令代码	功能
0 0	B0H	自动写设置
0 1	B1H	自动读设置
1 *	B2H/B3H	自动读、写结束

7、数据一次读、写方式，指令格式如下：

D1	1 1 0 0 0 N2 N1 N0
-----------	---------------------------

D1为需要写的数据，读时无此数据。

N2 N1 N0	指令代码	功能
0 0 0	C0H	数据写，地址加1
0 0 1	C1H	数据读，地址加1
0 1 0	C2H	数据写，地址减1
0 1 1	C3H	数据读，地址减1
1 0 0	C4H	数据写，地址不变
1 0 1	C5H	数据读，地址不变

8、屏读，指令格式为：

无参数	1 1 1 0 0 0 0 0
-----	-----------------

该指令将屏上地址指针处文本与图形合成后显示的一字节内容数据送到T6963C的数据栈内，等待MPU读出。地址指针应在图形区内设置。

9、屏拷贝，指令格式为：

无参数	1 1 1 0 1 0 0 0
-----	-----------------

该指令将屏上当前地址指针（图形区内）处开始的一行合成显示内容拷贝到相对应的图形显示区的一组单元内，该指令不能用于文本特征方式下或双屏结构液晶显示器的应用上。

10、位操作：

无参数	1 1 1 1 N3 N2 N1 N0
-----	---------------------

该指令可将显示缓冲区某单元的某一位清零或置1，该单元地址当前地址指针提供。

N3=1置1，N3=0清零。

N2~N0：操作位，对应该单元的D0~D7位。

11、T6963C的初始化设置一般都由管脚设置完成，因此其指令系统将集中于显示功能的设置上。T6963C的指令可带一个或两个参数，或无参数。每条指令的执行都是先送入参数（如果有的话），再送入指令代码。每次操作之前最好先进行状态字检测。T6963C的状态字如下所示：

STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0
------	------	------	------	------	------	------	------

STA0:指令读写状态 1:准备好 0:忙

STA1:数据读写状态 1:准备好 0:忙

STA2:数据自动读状态 1:准备好 0:忙

STA3:数据自动写状态 1:准备好 0:忙

STA4:未用

STA5:控制器运行检测可能性 1:可能 0:不能

STA6:屏读/拷贝出错状态 1:出错 0:正确

STA7:闪烁状态检测 1:正常显示 0:关显示

由于状态位作用不一样，因此执行不同指令必须检测不同状态位。在MPU一次读、写指令和数据时，STA0和STA1要同时有效——处于“准备好”状态。

当MPU读、写数组时，判断STA2或STA3状态。

屏读、屏拷贝指令使用STA6。

STA5和STA7反映T6963C内部运行状态。

11、附录

初始化程序参考:

```

/*-----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++)
            ;
}
/*-----写命令或数据到 LCD-----*/
void wr_od (uchar dat,uchar comm) //写一个数据和一个命令
{
    wr_data(dat);
    wr_comm(comm);
}
void wr_td (uchar dat1,uchar dath,uchar comm) //写两个数据和一个命令
{
    wr_data(dat1);
    wr_data(dath);
    wr_comm(comm);
}
void wr_xd (uint dat,uchar comm) //写一个 16 进制数据和一个命令
{
    uchar dat1,dath;
    dat1=dat;
    dath=dat>>8;
    wr_data(dat1);
    wr_data(dath);
    wr_comm(comm);
}
void wr_auto (uchar dat) //自动写数据
{
    chk_busy (1);
    cd=0;
    rd=1;
    data_or=dat;
    wr=0;
    wr=1;
}

void wr_comm (uchar comm) //写命令
{

```

```

chk_busy (0);
cd=1;
rd=1;
data_ora=comm;
wr=0;
wr=1;
}
void wr_data (uchar dat)      //写数据
{
chk_busy (0);
cd=0;
rd=1;
data_ora=dat;
wr=0;
wr=1;
}
void chk_busy (uchar autowr)  //测状态
{
data_ora=0xff;
cd=1;
wr=1;
rd=0;
if(autowr)
{while(bf3==0)
;}
else
{while((bf0==0)||(bf1==0))
;}
rd=1;
}
/*-----初始化-----*/
void init_lcd (void)
{
rst=0;
;
rst=1;
fs=0;
ce=0;
wr=1;
rd=1;
wr_xd(addr_w,0x40);          //文本显示区首地址
wr_xd(addr_t,0x42);          //图形显示区首地址
wr_td(width,0x00,0x41);      //文本显示区宽度
wr_td(width,0x00,0x43);      //图形显示区宽度
wr_comm(0x81);                //逻辑"异或"
wr_td(0x02,0x00,0x22);       //CGRAM 偏置地址设置
wr_comm(0x9c);                //启用文本显示,启用图形显示
}
/*-----清 RAM-----*/
void clrRam (void)

```

```

{
    uchar i,j;
    wr_xd(addr_w,0x24);
    wr_comm(0xb0);
    for(j=0;j<72;j++)
    {
        for(i=0;i<width;i++)
            wr_auto(0x00);
    }
    wr_comm(0xb2);
}
/*-----显示点阵-----*/
void disp_dz (uchar data1,uchar data2)
{
    uchar i,j;
    wr_xd(addr_t,0x24);
    wr_comm(0xb0);
    for(j=0;j<16;j++)
    {
        for(i=0;i<width*2;i++)
            wr_auto(data1);
        for(i=0;i<width*2;i++)
            wr_auto(data2);
    }
    wr_comm(0xb2);
}
/*-----在 addr 处显示 8x1*y1 的图形-----*/
void disp_img (uint addr,uchar xl,uchar yl,uchar code *img)
{
    uchar i,j;
    for(j=0;j<yl;j++)
    {
        for(i=0;i<xl;i++)
        {
            wr_xd(addr+j*width+i,0x24);
            wr_od(img[j*xl+i],0xc0);
        }
    }
}
/*-----在 addr 处显示 row_y1 行(每行 row_xl 个)8x1*y1 的文字-----*/
void disp_chn (uint addr,uchar xl,uchar yl,uchar row_xl,uchar row_y1,uchar code *chn)
{
    uchar i,j,k,m;
    for(m=0;m<row_y1;m++)
    {
        for(k=0;k<row_xl;k++)
        {
            for(j=0;j<yl;j++)
            {
                for(i=0;i<xl;i++)
                {

```

```
        wr_xd(addr+m*yl*width+k*xl+j*width+i,0x24);
        wr_od(chn[(m*row_xl*xl*yl)+(k*xl*yl)+(j*xl)+i],0xc0);
    }
}
}
}
/*-----显示字符-----*/
void disp_eng (uchar code *eng)
{
    uchar i,j;
    wr_xd(addr_w,0x24);
    wr_comm(0xb0);
    for(j=0;j<8;j++)
    {
        for(i=0;i<width;i++)
            wr_auto(eng[j*width+i]);
    }
    wr_comm(0xb2);
}
/*-----主程序-----*/
void main ()
{
    SP=0x5f;
    init_lcd ();
    while (1)
    {
        clrRAM();
        disp_eng(tab1);
        delay1(800);
        clrRAM();
        disp_dz(0xcc,0xcc);
        delay1(800);
        clrRAM();
        disp_chn(0x01e0,3,24,10,2,tab3);
        delay1(800);
        clrRAM();
        disp_dz(0xcc,0x33);
        delay1(800);
        clrRAM();
        disp_img(0x00f0,30,64,tab5);
        delay1(800);
    }
}
```

内部字符代码表:

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	"	#	\$	%	&	'	()	*	+	,	=	>	/	
1	0	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
2	P	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	U	U	X	Y	Z	[\]	^	_
4	,	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	P	q	r	s	t	u	u	u	x	y	z	{		}	~	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

模块外型尺寸

