

WLTl-06 液晶显示控制器使用手册（1009）

一、简介.....	1
二、控制板接口与设置	1
1、CPU 侧接口(双排压线)	1
2、WLTl06 控制板 LCD 侧接口（TTL 电平）	2
3、WLTl06 控制板 LCD 侧接口（LVDS 电平）	3
4、控制板外形尺寸:	3
三、指令操作说明	4
四、CPU 接口时序.....	7
写时序:	7
读时序:	8
自动清屏:	8
自动清屏步骤:	8
五、应用程序.....	9
1、汇编程序	错误！未定义书签。
2、C51 程序:	错误！未定义书签。
附录一、安徽世龙电子技术有限公司服务规范.....	16
附录二、液晶模块装配与使用注意事项、运输、产品责任等.....	17



地址:安徽省合肥市高新区天达路 71 号华亿科技苑 E 栋 3 层 邮编: 230088
技术支持电话: 0551-5329091、5329092、5329093、5359929 EXT 890、882
传 真:0551-5359928
E-Mail:support@worldlong.com
网 址:http://www.worldlong.com

一、简介

世龙显示控制产品核心电路采用 ALTERA 公司的大规模可编程集成电路编程实现，性能稳定可靠。

WLTl06V	TTL 电平，	控制 TFT640×480, 256 色，	2 页显存
WLTl06DH	TTL 电平，	控制 TFT800×480, 256 色，	2 页显存
WLTl06S	TTL 电平，	控制 TFT800×600, 256 色，	1 页显存
WLTl06X	TTL 电平，	控制 TFT1024×768, 256 色，	1 页显存
WLTl06VLV	LVDS 电平，	控制 TFT640×480, 256 色，	1 页显存
WLTl06LV	LVDS 电平，	控制 TFT800×600, 256 色，	1 页显存
WLTl06XLV	LVDS 电平，	控制 TFT1024×768, 256 色，	1 页显存

WLTl06 系列控制板是 WLTl04V 和 WLTl05 升级板，其速度、功能、显存容量均得到了提高：

- 1、自动清屏，清屏色取决于 FRONT 寄存器值，可由用户设定，800×600 屏清一屏时间为 19.2ms。
- 2、三种写屏方式。用户根据写屏内容自由切换单点、多点、八点写屏三种写屏方式。
- 3、显示屏上的 XY 坐标和存储器地址一一对应，不用计算显示屏上 XY 坐标与存储器地址的关系，只要将显示屏上的 XY 坐标写入地址寄存器，就可以直接写入数据。
- 4、用户通过改变状态控制寄存器相应位的状态，可选择 X、Y 坐标地址自动增加或不增加，单点写屏时 X 坐标自动加 1 适合图像显示；多点或 8 点写屏时，X 坐标自动加 8 适合字符显示和颜色填充；Y 坐标自动加 1 适合取样波形显示。X 坐标自动 1（8）时，写满一行自动换行。
- 5、存储空间为 1MSRAM。对任意一页读写，不影响当前页的显示。无“雪花”现象。

适配 CPU： 51,96,X86,8088,Z80,DSP,ARM

二、控制板接口与设置

1、CPU 侧接口(双排压线)

引脚	符号	功能	备注
1	VCC	液晶屏逻辑电源(+5V/3.3V)	5V
2	VCC	液晶屏逻辑电源(+5V/3.3V)	5V
3	NC		
4	DATA0	数据总线	2
5	DATA1	数据总线	2
6	DATA2	数据总线	2
7	DATA3	数据总线	2
8	DATA4	数据总线	2
9	DATA5	数据总线	2
10	DATA6	数据总线	2



11	DATA7	数据总线	2
12	/CS	片选信号，低电平对屏操作有效	2
13	/WR	写操作信号，低电平有效。	2
14	/RD	读操作信号，低电平有效。	2
15	NC		
16	A0	寄存器地址	2
17	A1	寄存器地址	2
18	A2	寄存器地址	2
19	GND	电源地	
20	GND	电源地	

注 1: WLT106LV, WLT106XLV 为 3.3V, 其余为 5V。

注 2: 逻辑电平 5V 和 3.3V 均可。

2、WLT106 控制板 LCD 侧接口 (TTL 电平) 控制 TFT640×480、TFT800×600、TFT1024×768

引脚	符 号	功 能	备注
1	GND	地	
2	XCK	数据采样信号	
3	Hsync	水平同步信号 (负)	
4	Vsync	垂直同步信号 (负)	
5	GND	地	
6	R0	红色数据信号 (LSB)	
7	R1	红色数据信号	
8	R2	红色数据信号	
9	R3	红色数据信号	
10	R4	红色数据信号	
11	R5	红色数据信号 (MSB)	
12	GND	地	
13	G0	绿色数据信号 (LSB)	
14	G1	绿色数据信号	
15	G2	绿色数据信号	
16	G3	绿色数据信号	
17	G4	绿色数据信号	
18	G5	绿色数据信号 (MSB)	
19	GND	地	
20	B0	蓝色数据信号 (LSB)	
21	B1	蓝色数据信号	
22	B2	蓝色数据信号	
23	B3	蓝色数据信号	
24	B4	蓝色数据信号	
25	B5	蓝色数据信号 (LMS)	
26	GND	地	



27	DE	水平同步信号	
28	VCC	逻辑电源	3
29	VCC	逻辑电源	3
30	NC	无电气连接	
31	NC	无电气连接	
32	GND	地	
33	GND	地	
34	NC	无电气连接	

注 3: TTL 电平控制板背面有短路跳线, 控制板输出到显示屏的供电可接 3.3V 或 5V, 用户可根据不同显示屏的供电需求, 跳接到 3.3V 或 5V。

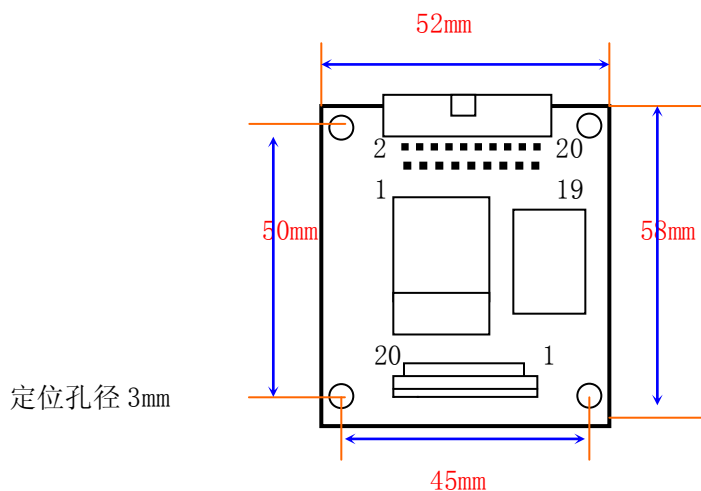
3、WLTl06 控制板 LCD 侧接口 (LVDS 电平) 控制 TFT800×600、TTL1024×768

引脚	符 号	功 能	备注
1	VCC	逻辑电源 (+3.3V)	
2	VCC	逻辑电源 (+3.3V)	
3	GND	地	
4	GND	地	
5	Rx0-	R0—R5, G0	
6	Rx0+		
7	GND	地	
8	Rx1-	G1—G5, B0, B1	
9	Rx1+		
10	GND	地	
11	Rx2-	B2—B5, Vsync, Hsync, De	
12	Rx2+		
13	GND	地	
14	CLK-	clock	
15	CLK+		
16	GND	地	
17	NC	无电气连接	
18	NC	无电气连接	
19	GND	地	
20	GND	地	

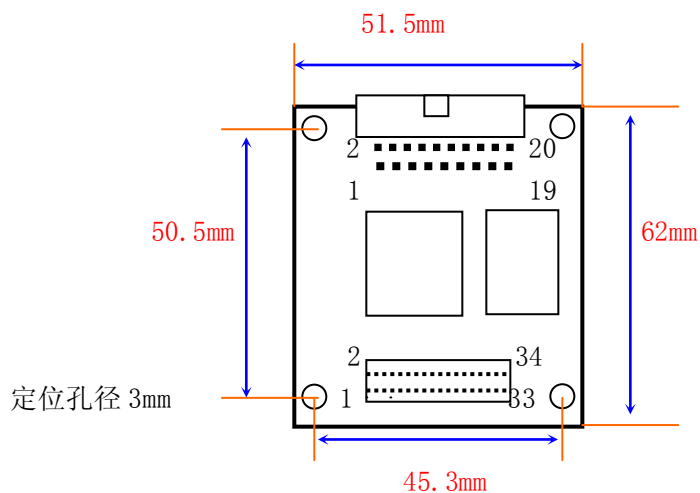
:

4、控制板外形尺寸

LVDS 电平



TTL 电平



三、指令操作说明

控制板中共有 8 个寄存器，分别为 XY 坐标、前景色、背景色、数据、状态寄存器，通过对这 8 个寄存器进行操作，可实现写屏功能。**所有寄存器只能写，不能读。**

XY 坐标、前景色、背景色、状态寄存器写入不分先后次序，如其内容没有变化，可不需要再写。但要注意，在自动加 1 (8) 控制位为 1 时，随着数据寄存器的写入，XY 寄存器数值在自动增加。

CS 与 A0、A1、A2 组合功能如下：

CS	A2A1A0	WR	RD	功能	寄存器符号
0	000	0	1	X 低 8 位寄存器	XREGL
0	001	0	1	X 高 8 位寄存器	XREGH
0	010	0	1	Y 低 8 位寄存器	YREGL

0	011	0	1	Y 高 8 位寄存器	YREGH
0	100	0	1	前景色寄存器	FRONT
0	101	0	1	背景色寄存器	BACK
0	110	0	1	写显示数据寄存器	DATA
0	111	0	1	状态控制寄存器	PSW
0	×	1	0	读当前坐标颜色值	
1	×	×	×	不选通	

a) X 地址寄存器 XREG: 地址: A2A1A0=000, 001, 低字节在前, 高字节在后

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
×	×	×	×	×	×	H9	H8	H7	H6	H5	H4	H3	H2	H1	H0

TFT640×480 H9—H0 取值范围从 0 至 639
TFT800×600 H9—H0 取值范围从 0 至 799
TFT1024×768 H9—H0 取值范围从 0 至 1023

b) Y 地址寄存器 YREG: 地址: A2A1A0=010, 011, 低字节在前, 高字节在后

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
×	×	×	×	×	×	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0

TFT640×480 L8—L0 取值范围从 0 至 479。
TFT800×600 L9—L0 取值范围从 0 至 599。
TFT1024×768 L9—L0 取值范围从 0 至 767。

b) 状态寄存器 PSW: 地址 A2A1A0=111

WLTl06V 和 WLTl06DH 控制板

D7	D6	D5	D4	D3	D2	D1	D0
CLR	0	IP	P	YINC	XINC	Writel	Write0

WLTl06S, WLTl06LV 控制板

D7	D6	D5	D4	D3	D2	D1	D0
CLR	0	0	0	YINC	XINC	Writel	Write0

IP 显示页号:

800×480 和 640×480 为 2 页显存

P 读写页号:

800×480 和 640×480 为 2 页显存

CLR 自动清屏控制位: 参见自动清屏时序图

=1 时自动清屏, XINC=1, writelwrite0=10, 颜色值等于 FRONT 寄存器内容, 只清一页, 页号由 P 决定, =0 时正常读写。

Xinc 为 X 坐标值自动增加控制位:

=1 时允许 X 坐标自动增加, 写满一行后自动换行, =0 时则禁止增加。单点写屏时, X 自动加

1, 多点 (或 8 点) 写屏时 X 自动加 8;

YINC 为 Y 坐标自动加 1 控制位:

=1 时允许自动加 1, =0 时则禁止加 1;

WRCON1, WRCON0 为写屏方式: (老用户请注意 WLTl06 有别于和 WLTl04、WLTl05 板)

WRCON1, WRCON0=00 时为单点写入方式, 直接将颜色值写入数据寄存器, 而与前景色、背景色寄存器内容无关;

WRCON1, WRCON0=10 时为多点写入方式, 将点位信息写入数据寄存器, 如写入数据寄存器为 ‘01010101’ 则显示 ‘原色、前景色、原色、前景色、原色、前景色、原色、前景色’。

原色: 显示屏中原有图案颜色, 用这种写屏方式可将字符或图形写到底图上而不影响底图。

WRCON1, WRCON0=11 时为 8 点写入方式, 将点位信息写入数据寄存器, 如写入数据寄存器为 ‘01010101b’ 则显示 ‘背景色、前景色、背景色、前景色、背景色、前景色、背景色、前景色’。

注意: 多点和 8 点写屏时, 输入的 X 地址必须是偶数。

如下图, 显示屏原有颜色是一幅照片, “多点” 两个字是用多点写入方式写入的, 只写前景色蓝色, 不写背景色; 而 “8 点” 两字是用 8 点写入方式写入的, 前景色为绿色, 背景色为白色, 前景和背景同时写入。



c) 数据寄存器 DATA: 地址 A2A1A0=110,

前景色寄存器 FRONT: 地址 A2A1A0=100,

背景色寄存器 BACK: 地址 A2A1A0=101

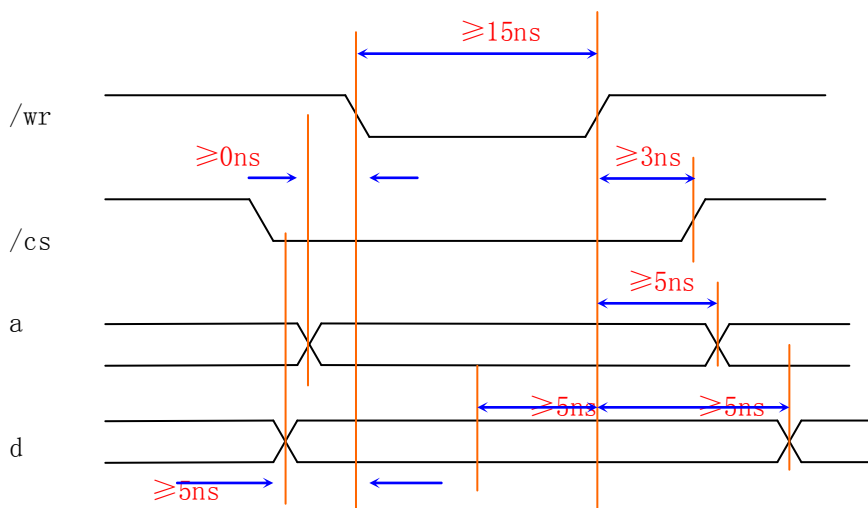
当单点写屏时, 数据寄存器的内容为该点的颜色值, 当多点或 8 点写屏时, 数据寄存器的内容为要写的图案点位信息。

	D7	D6	D5	D4	D3	D2	D1	D0
	R2	R1	R0	G2	G1	G0	B1	B0
256 色								
	颜色灰度	R2、R1、R0			G2、G1、G0			B1、B0
基本颜色	最黑	000			000			00
	亮蓝	000			000			11
	亮绿	000			111			00
	亮青	000			111			11
	亮红	111			000			00
	亮紫	111			000			11
	亮黄	111			111			00
	亮白	111			111			11

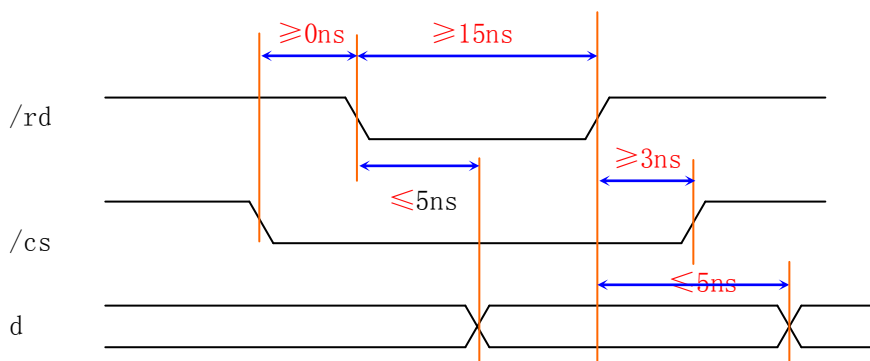
蓝色 灰度	最黑	000	000	00
	较暗	000	000	01
	较亮	000	000	10
	最亮	000	000	11
绿色 灰度	最黑	000	000	00
	较暗	000	001	00
	...	...	...	...
	较亮	000	110	00
	最亮	000	111	00
红色 灰度	最黑	000	000	00
	较暗	001	000	00
	...	...	...	...
	较亮	110	000	00
	最亮	111	000	00

四、CPU 接口时序

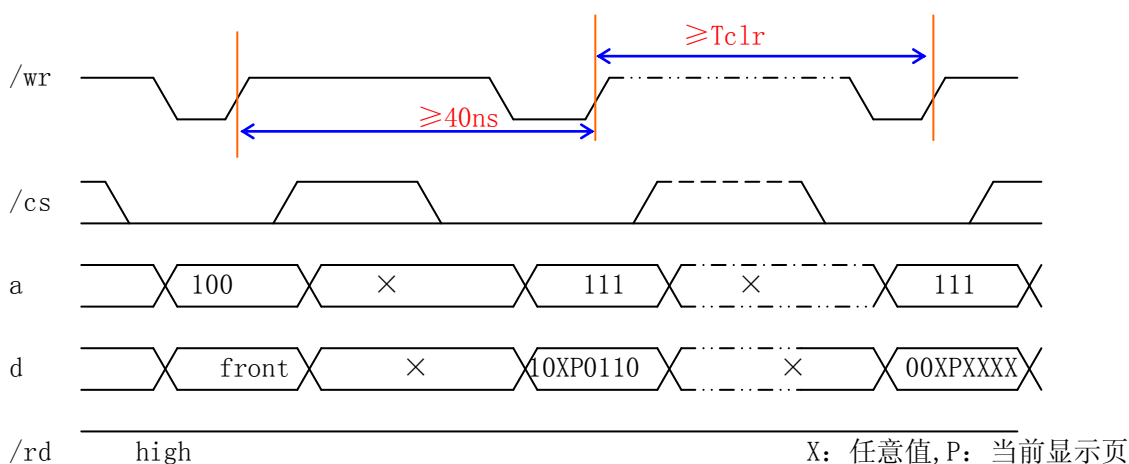
写时序:



读时序:



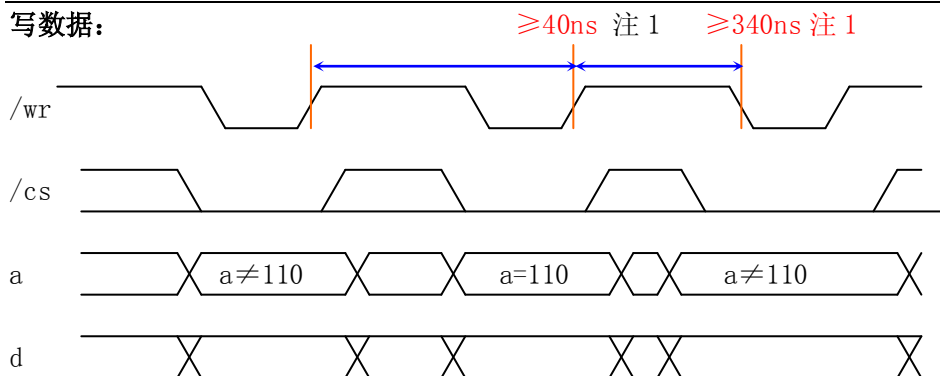
自动清屏:



自动清屏步骤:

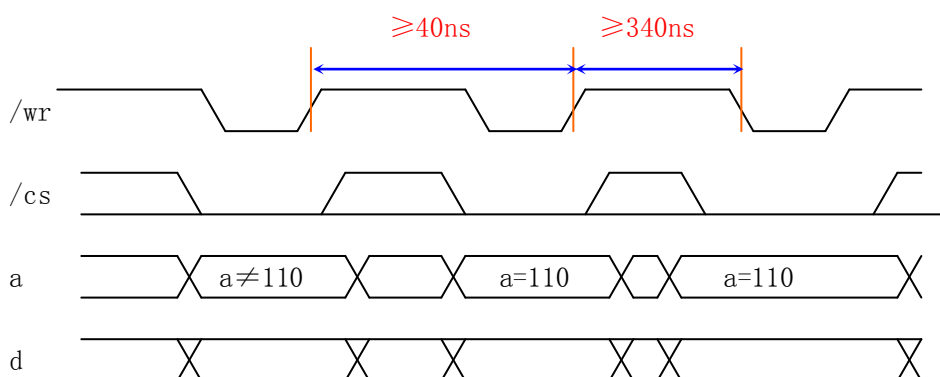
- 1、将颜色值写前景色 front 寄存器;
- 2、将 1XXP0101 (二进制) 写入状态 PSW 寄存器, 控制板开始自动清屏。D7=1 时自动清屏; D3=0, Y 坐标不自动加 1; D1D0=01, 多点写入方式, 一次写入 8 点; D2=1, X 坐标自动加 8。
- 3、等待 T_{clr} 后, 将 PSW 寄存器 D7 位复位, 清屏完毕, 全屏显示 front 色。
- 4、640×480 T_{clr} =12.3ms, 800×600 T_{clr} =19.2ms, 1024×768 T_{clr} =31.5ms

写数据:



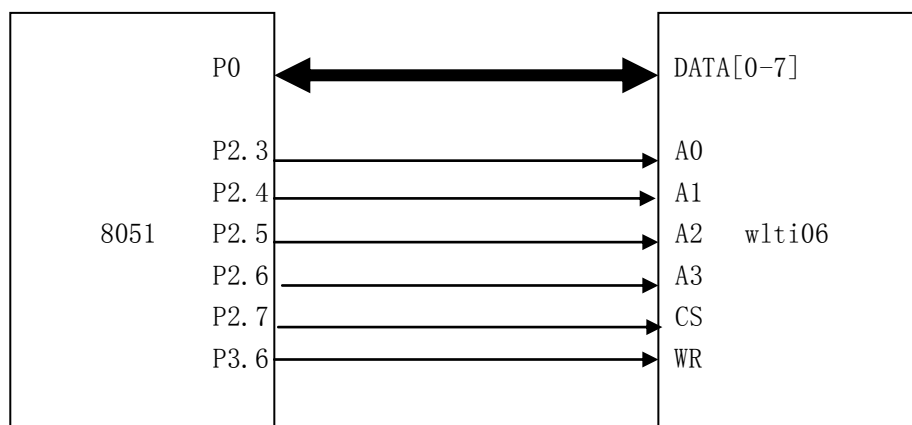
注 1: 对各寄存器写操作周期大于 40ns, 但对数据寄存器 (A=110) 写完数据后 340ns 之内不可对控制板进行写操作。

连续写数据 (自动加 1 加 8):



五、应用程序

下面提供 51 系列单片机总线和 I/O 口控制方式 C51 演示程序



控制板与 CPU 连接示意图

总线方式比 I/O 方式速度要快很多, 建议用户用总线方式

1、C51 程序 (总线)



```
#include <reg52.h>
#include <zi.h>
#define uchar unsigned char
#define uint unsigned int
uchar xdata *xlreg=(uchar *)0x0000;          //A=000
uchar xdata *xhreg=(uchar *)0x0800;          //A=001
uchar xdata *ylreg=(uchar *)0x1000;          //A=010
uchar xdata *yhreg=(uchar *)0x1800;          //A=011
uchar xdata *frontreg=(uchar *)0x2000;        //A=100
uchar xdata *backreg=(uchar *)0x2800;         //A=101
uchar xdata *dreg=(uchar *)0x3000;           //A=110
uchar xdata *pswreg=(uchar *)0x3800;         //A=111
//自动清屏
void autoclrscan(uchar color,uchar psw)
{
    //psw=0x06|page 页号
    *xlreg=0x00;
    *xhreg=0x00;
    *ylreg=0x00;
    *yhreg=0x00;
    *frontreg=color;
    *pswreg=psw|0x80;
    Delay();          //delay 13ms
    *pswreg=psw;
}
//画实心矩形（宽为 8 的倍数）
void drawrectangle(uchar color,uint x,uint y,uint l,uint h,uchar psw)
{
    //psw=0x06|page 页号
    //矩形宽=1*8
    uint i,j;
    *frontreg=color;
    *pswreg=psw;      //0x06|page;多点写屏方式,
    for(i=0;i<h;i++)
    {
        *ylreg = y;
        *yhreg = y>>8;
        y++;
        *xlreg = x;
        *xhreg=x>>8;
        for(j=0;j<l;j++)
        {
            *dreg=0xff;    //一次写入 8 点
            //delay 450ns
```



```
}  
}  
}  
//画一条直线____单点写入方式，一次写1点，  
void writeline(uchar color,uint x,uint y,uint l,uchar psw)  
    //color 线色  
    //x,y 起点坐标  
    //l 直线长度  
{  
    uint i;  
    *pswreg=psw;          // Xin=1:水平直线，或者 Yin=1:垂直直线，  
                          //Xin=1 且 Yin=1: 45 度斜线  
  
    *xlreg=x;  
    *xhreg=x>>8;  
    *ylreg=y;  
    *yhreg=y>>8;  
    for(i = 0; i <l; i++)    //  
    {  
        *dreg=color;  
    }  
}  
//写汉字程序____一次写8点  
void ColorWriteMultiWord(uint x, uint y, uchar xNum, uchar yNum, uchar *pData,uchar  
frontcolor,uchar backcolor,uchar psw)  
//write word at Screen  
//x,y:字符左上角位置  
//xNum=字符一行的点阵数/8，如16点阵字=2  
//yNum=字符行数，如16点阵字符=16  
//pData 字符首地址  
//frontcolor 字体色,backcolor 背景色  
{  
    uchar i,j;  
    uint n=0;  
    *pswreg=psw;          //psw=0x06|page;多点写屏，=0x07|page;8点写屏。Page 页号  
  
    *frontreg = frontcolor;    //front color  
    *backreg = backcolor;      //back color  
    for(i = 0; i < yNum; i++)  
    {  
  
        *ylreg = y;  
        *yhreg = y>>8;  
        y++;  

```



```
*xlreg = x;
*xhreg=x>>8;
for(j = 0; j < xNum; j ++)
{
    *dreg = pData[n++];    //多点 and 8 点写屏
    //_nop_();              //delay 500ns
}
}
}
//主程序（总线）
void main()
{
    uint i;
    uchar j;
    autoclrscan(0x00, 0x06);    //清黑色
    drawrectangle (0xff, 0, 0, 100, 600, 0x05);    //清白色，画矩形清屏
    for(i=0; i<256; i++)
    {writeline(i, i+30, 10, 100, 0x08);}    //用 256 种颜色画 256 条垂直线
    //用多点 and 8 点写屏方式显示“世龙电子”
    ColorWriteMultiWord(140, 100, 2, 16, shi, 0x03, 0x00, 0x06);    //多点写入“世龙电子”
    ColorWriteMultiWord(156, 100, 2, 16, lon, 0x03, 0x00, 0x06);
    ColorWriteMultiWord(172, 100, 2, 16, dian, 0x03, 0x00, 0x06);
    ColorWriteMultiWord(188, 100, 2, 16, zi, 0x03, 0x00, 0x06);

    ColorWriteMultiWord(140, 180, 2, 16, shi, 0x03, 0x00, 0x07);    //8 点写入“世龙电子”
    ColorWriteMultiWord(156, 180, 2, 16, lon, 0x03, 0x00, 0x07);
    ColorWriteMultiWord(172, 180, 2, 16, dian, 0x03, 0x00, 0x07);
    ColorWriteMultiWord(188, 180, 2, 16, zi, 0x03, 0x00, 0x07);
    while(1);
}
```

2、C51 程序（I/O 口）

```
#include <reg52.h>
#include <zi.h>
#define uchar unsigned char
#define uint unsigned int
#define xlreg 0x00
#define xhreg 0x08
#define ylreg 0x10
#define yhreg 0x18
#define frontreg 0x20
#define backreg 0x28
#define dreg 0x30
#define pswreg 0x38
```



```
sbit wr=P3^6;
sbit rd=P3^7;
sbit cs=P2^7;

void writedata(uchar a,uchar dat) //对控制板写数据
{
    P2=a;          //CS=0, WR=1, rd=1, a 寄存器地址,
    P0=dat;
    wr=0;
    wr=1;
    cs=1;
}

uchar readdata() //读显示屏当前页号、XY 坐标下的颜色值
{
    uchar dd;      //cs=0, a=任意值
    cs=0;
    rd=0;
    dd=P0;
    rd=1;
    cs=1;
    return dd;
}

//自动清屏
Void autoclrscan(uchar color,uchar psw)
{ //psw=0x06|page 页号
    writedata(xlreg, 0x00);
    writedata(xhreg, 0x00);
    writedata(ylreg, 0x00);
    writedata(yhreg, 0x00);
    writedata(frontreg, color);
    writedata(pswreg, psw|0x80);
    Delay();          //delay 13ms
    writedata(pswreg, psw);
}

//画实心矩形（宽为 8 的倍数）
void drawrectangle(uchar color,uint x,uint y,uint l,uint h,uchar psw)
{
    //psw=0x06|page 页号
    //矩形宽=1*8
    uint i,j;
    writedata(frontreg, color);
    writedata(pswreg, psw);
    for(i=0;i<h;i++)
```

```

{
    writedata(xlreg, x);
    writedata(xhreg, x>>8);
    writedata(ylreg, y);
    writedata(yhreg, y>>8);
    P2=dreg;
    P0=0xff;
    for(j=0; j<1; j++)
    {
        wr=0;wr=1;        //一次写入 8 点, 8 点颜色均为 color
        //delay 450ns 采用 writedata(dlreg, 0xff) 调用程度速度会很慢
    }
    y++;
}
cs=1;                //关闭控制板
}
//画一条直线____单点写入方式, 一次写 1 点,
void writeline(uchar color, uint x, uint y, uint l, uchar psw)
    //color 线色
    //x, y 起点坐标
    //l 直线长度
{
    uint i;
    writedata(pswreg, psw);        //Xin=1:水平直线, 或者 Yin=1:垂直直线,
                                    //Xin=1 且 Yin=1: 45 度斜线

    writedata(xlreg, x);
    writedata(xhreg, x>>8);
    writedata(ylreg, y);
    writedata(yhreg, y>>8);
    P2=dreg;
    P0=color;                //将色送到 P0 口
    for(i = 0; i < l; i ++) //采用 writedata(dlreg, 0xff) 调用程度速度慢

    {
        wr=0;wr=1;        //一次写入 1 个点, 只需写低字节即可,
    }
    cs=1;                //关闭控制板
}
//写汉字程序____一次写 8 点
void ColorWriteMultiWord(uint x, uint y, uchar xNum, uchar yNum, uchar *pData, uchar
frontcolor, uchar backcolor, uchar psw)
//write word at Screen
//x, y: 字符左上角位置

```



```

//xNum=字符一行的点阵数/8，如 16 点阵字=2
//yNum=字符行数，如 16 点阵字符=16
//pData 字符首地址
//frontcolor 字体色,backcolor 背景色
{
    uchar i, j;
    uint n=0;
    writedata(pswreg,psw); //psw=0x06|page:多点写屏,=0x07|page:8点写屏。Page 页号
    writedata(frontreg,frontcolor); //front color
    writedata(backreg,backcolor); //back color
    for(i = 0; i < yNum; i ++)
    {
        writedata(ylreg,y);
        writedata(yhreg,y>>8);
        writedata(xlreg,x);
        writedata(xhreg,x>>8);
        y++;
        for(j = 0; j < xNum; j ++)
        {
            writedata(dreg, pData[n++]); //多点 and 8 点写屏,一次 8 个点
            //_nop_(); //500ns
        }
    }
}

//主程序 (I/O)
void main()
{
    uint i;
    uchar j;
    cs=1;wr=1;rd=1; //初始化 wr,cs,rd 全位高电平,以免端口争议
    autoclrscan(0x00,0x06); //清黑色
    drawrectangle (0xff,0,0,100,600,0x05); //清白色,画矩形清屏
    for(i=0;i<256;i++)
    {writeline(i,i+30,10,100,0x08);} //用 256 种颜色画 256 条垂直线
    //用多点 and 8 点写屏方式显示"世龙电子"
    ColorWriteMultiWord(140,100,2,16,shi,0x03,0x00,0x06); //多点写入"世龙电子"
    ColorWriteMultiWord(156,100,2,16,lon,0x03,0x00,0x06);
    ColorWriteMultiWord(172,100,2,16,dian,0x03,0x00,0x06);
    ColorWriteMultiWord(188,100,2,16,zi,0x03,0x00,0x06);
    ColorWriteMultiWord(140,180,2,16,shi,0x03,0x00,0x07); //8 点写入"世龙电子"
    ColorWriteMultiWord(156,180,2,16,lon,0x03,0x00,0x07);
    ColorWriteMultiWord(172,180,2,16,dian,0x03,0x00,0x07);
    ColorWriteMultiWord(188,180,2,16,zi,0x03,0x00,0x07);
}

```

```
while(1);
}
//字库”世龙电子”16x16点阵(zi.h)
unsigned char code shi[]={          // “世”16x16点阵
    0x00, 0x00, 0x00, 0xC0, 0x00, 0xE0, 0x03, 0x60,
    0x0B, 0x60, 0x0F, 0xFC, 0x3F, 0x44, 0x7B, 0x40,
    0x2B, 0xC0, 0x09, 0x40, 0x08, 0x00, 0x0F, 0xC0,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
unsigned char code lon[]={          // “龙”16x16点阵
    0x00, 0x00, 0x03, 0x80, 0x01, 0xF0, 0x01, 0x00,
    0x03, 0xE0, 0x07, 0x80, 0x1F, 0x00, 0x0F, 0xA0,
    0x05, 0x30, 0x09, 0x60, 0x1A, 0xC4, 0x13, 0x84,
    0x23, 0x0C, 0x01, 0xFC, 0x00, 0x00, 0x00, 0x00
};
unsigned char code dian[]={        // “电”16x16点阵
    0x00, 0x00, 0x06, 0x00, 0x07, 0x00, 0x03, 0x00,
    0x03, 0xE0, 0x07, 0x10, 0x1B, 0xB0, 0x1F, 0xA0,
    0x1B, 0x60, 0x0F, 0xC0, 0x0F, 0xC0, 0x02, 0x00,
    0x01, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
unsigned char code zi[]={          // “子”16x16点阵
    0x00, 0x00, 0x00, 0xE0, 0x03, 0xB0, 0x07, 0x60,
    0x07, 0xC0, 0x01, 0xC0, 0x00, 0xF0, 0x03, 0xF8,
    0x0F, 0x20, 0x0C, 0x20, 0x08, 0x20, 0x06, 0x20,
    0x03, 0xC0, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00
};
```

附录一、安徽世龙电子技术有限公司服务规范

- 1、您购买安徽世龙电子技术有限公司各类控制板和液晶显示模块时，我公司将事先进行检测，确保您所购买的控制板和模块为完好的产品，液晶模块符合生产厂家提供的检测标准。
- 2、液晶模块属于元器件类产品，不属于设备，不能享受保修服务。
- 3、如果在使用过程中，您不小心损坏了液晶模块，我们将为您提供维修服务；
 - (1) 由于产品质量问题造成液晶模块显示不正常的，我公司将提供免费维修，必要时可以更换模块；
 - (2) 由于客户原因使模块受损的，我公司将尽力维修，如果我公司不能维修的，将返回生产厂家进行维修，这类情况将收取相应的维修成本费用。
- 4、如果由于液晶片的物理损伤造成液晶模块不正常工作的，一般该模块只能报废。

- 5、在我公司购买的液晶产品出现需要翻修的情况时，请认真填写《返修单》，如果没有《返修单》的请尽量使用情况和故障现象详细描述，和故障产品一并返回到我公司。

附录二、液晶模块装配与使用注意事项、运输、产品责任等

一、 处理保护膜

在装好的模块品表面有一保护膜，以防在装配时玷污显示表面，所有在整机装配结束前不得揭去，以免弄脏或损坏显示面。

二、 加装垫

在模块与前面板之间最好加一块 0.1 毫米左右的垫，面板还应保持平整，以免在装配后产生扭曲。

三、 严防静电

模块中的控制、驱动电路是低压、微功耗的 CMOS 电路，极易被静电击穿是一种不可修复的损坏，务必注意，不可大意。所以，在操作、装配以及使用中都应极其小心，要严防静电。为此：

- 1、不要用手随意去摸外引线、电路和 IC 等；
- 2、焊接使用的烙铁必须良好接地，没有漏电。

四、 装配操作时的注意事项

- 1、液晶模块是经心设计组装而成的，请勿随意自行加工、维修；
- 2、金属框爪不得随意扭动、拆卸；不要随意修改、加工 PCB 板外形、装配孔、线路及部件；不要随意改导电胶条，尤其注意有些模块侧面的柔性电缆需要保护，不能损伤；不要修改任何内部支架；不要碰、摔、折曲、扭动模块和背景部分等；

五、 焊接、在焊接模块外引线、接口电路时，应按如下规定进行：

- 1、如果液晶模块与其他外围电路的连接需要焊接或改变原有的连接头的话，请确认通过质检；
- 2、烙铁温度： $280 \pm 10^{\circ}\text{C}$ ；焊接时间： $< 3 \sim 4\text{S}$ ；焊接材料：共晶型、低熔点；重复焊不得超过 3 次。

六、模块的作用

1、液晶模块的外引线决不允许接错，不允许与 PCB 上不相关的焊盘、过孔等短路，否则可能造成过流、过压等液晶模块元器件有损的现象；

2、模块使用接入电源及断开电源时，必须在正电源（ $5 \pm 0.25\text{V}$ ）稳定接入以后，才能输入信号电平。如在电源稳定前或断开后输入信号电平，有可能损坏模块中的 IC 等电路。

3、点阵液晶模块显示时的对比度、视角与温度、驱动电压关系很大，所以，如果 V_{EE} 调整过高，不仅影响显示，还回缩短模块的使用寿命。

4、模块在规定工作温度范围内以下使用时，显示响应很慢，而在规定工作温度范围以上使用时，整个显示面又回呈现全显示状态，这种现象不是模块被损坏，只需恢复到规定温度范围内，一切又将恢复正常。（不应在超过存储极限温度的范围外使用或存储，如果温度低于结晶温度，液晶就会结晶，破坏定向层，使器件报废；如果温度过高，液晶将会变成各向同性的液晶，失去

液晶态，也就失去了液晶器件的功能。)

5、用力按压显示部位，会产生异常显示。这时切断电源，稍待片刻，重新上电，即恢复正常。

七、模块的存储

若长期（如几年）存储，我们推荐以下方式：

装入聚乙烯口袋（最好有防静电涂层）并将口密封；纺织暗处，避强光；决不能在表面压放任何物品；严格避免在极限温、湿度条件以外存放（液晶用的偏振片怕高温、怕潮湿）。

[运输损坏]

如果用户收到的货物在运输过程中已经损坏，要是包装受损的话，用户首先应该在得到送货人允许的前提下打开包装，如果货物受损，用户应该向运输公司索赔；否则一定要原封不动地保留货箱、包装材料及货物，并与安徽世龙电子技术有限公司联系。

[产品责任]

公司保证所有售出的产品符合生产厂家的质量要求，并对承担质量保证的责任，若用户在购买产品的 30 天内发现产品的质量确有问题，经安徽世龙电子技术有限公司或液晶生产厂家检测，系产品本身的质量问题，安徽世龙电子技术有限公司将负责维修或换货或退货，安徽世龙电子技术有限公司承担的产品责任不超过可户购买货品价值，并不对可户使用产品所造成的间接损失负责。由于可户对产品使用不当而导致产品的损坏（例如静电，焊接、连线不当，过流、过压使用等）、安徽世龙电子技术有限公司将不承担任何责任，但可尽力为可户提供维修服务，并将根据具体情况收取适当费用。