

## WLT107V 液晶显示控制板技术手册

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 EPM240 编程实现，性能稳定可靠。WLT1-07V 可控制 640×480 真彩 TFT 液晶显示屏；为提高读写速度、简化程序，显示屏中每个点影射显示缓存中的一个字节，显示屏中的 XY 坐标与缓存器的行列号一一对应，因此，只需输入 XY 坐标，便可直接读写相应点数据，不用计算点在显示缓存中的位置。由于显示缓存采用高速 SRAM，对于 640×480 点阵彩色 LCD，可实现 2 页内容叠加显示，读写任意一页不会其他页内容。

**控制板中的多点 and 8 点写入方式在写字符和清屏操作上比一次写一个点的方式速度快十几倍。**

### 应用：

- 1、在做人机界面时，可用于弹出式菜单或页面，擦去弹出式菜单或页面也变得非常容易。
- 2、方便各种仪器多路波形、移动游标、菜单显示。

世龙显示产品型号说明：

WL S I -01-Q 43



适配 CPU: 51, 96, X86, 8088, Z80, DSP

### 一、接口定义：

#### 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 <sup>(1)</sup>	5V
2	VCC	液晶屏逻辑电源 <sup>(1)</sup>	5V
3	NC		
4	DATA0	数据总线	
5	DATA1	数据总线	
6	DATA2	数据总线	
7	DATA3	数据总线	
8	DATA4	数据总线	
9	DATA5	数据总线	
10	DATA6	数据总线	
11	DATA7	数据总线	
12	CS	片选信号，低电平对屏操作有效	
13	WR	写操作信号，低电平有效。	
14	RD	读操作信号，低电平有效。	

15	NC		
16	A0	寄存器地址	
17	A1	寄存器地址	
18	A2	寄存器地址	
19	GND	电源地	
20	GND	电源地	

(1) CS 与 A0、A1、A2 组合功能如下:

CS	A2A1A0	WR	RD	功能
0	000	0	1	X 低 8 位寄存器
0	001	0	1	X 高 8 位寄存器
0	010	0	1	Y 低 8 位寄存器
0	011	0	1	Y 高 8 位寄存器
0	100	0	1	前景色寄存器
0	101	0	1	背景色寄存器
0	110	0	1	写显示数据寄存器
0	111	0	1	状态控制寄存器
0	×	1	0	读显示数据
1	×	×	×	不选通

说明:

**所有寄存器只能写，不能读。**

a) X 坐标寄存器: XREG 地址: A2A1A0=000(低), 001(高), 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

×	×	×	×	×	×	×	×	X8	X7	X6	X5	X4	X3	X2	X1	X0
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

b) Y 坐标寄存器: YREG 地址: A=010, 011

D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0

×	×	×	×	×	×	×	×	L8	L7	L6	L5	L4	L3	L2	L1	L0
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

c) 状态字寄存器 PSW: 地址 A2A1A0=111

D7 D6 D5 D4 D3 D2 D1 D0

×	×	×	Prw	YINC	XINC	Wrcon1	Wrcon0
---	---	---	-----	------	------	--------	--------

×: 建议为 0

Prw 为读写页号: =0, 顶层; =1, 底层。

D3 和 D2 位根据用户需要可设置成以下两种功能:

**Xinc** 为 X 坐标自动增加控制位, =1 时允许 X 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) 写屏时自动加 8;

**YINC** 为行 (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** 为写入方式:

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=01 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 '01010101b' 则显示 '原色、前景色、原色、前景色、原色、前景色、原色、前景色'。

Wrcon1, Wrcon0=10 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’。

\*\*原色：显示屏原有颜色，前景色、背景色是事先存入前景、背景寄存器中的颜色值。

。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。



#### d) 双页叠加方式

控制板将显存分为两页，读写可分别操作，在显示时是叠加在一起显示的。对两页我们分别称之为顶层页、底层页，底层好比一块画布，顶层好比一块玻璃，当顶层玻璃擦干净之后，可看见底层图案。在此，我们定义一个新的概念--“透明色”，用透明色清屏，相当于将玻璃擦干净了。顶层玻璃可写入 255 种颜色+1 个透明色，底层可写入 256 种颜色。00H 值写入顶层为透明色，写入底层为黑色。如下图显示，底层为一幅照片，顶层用透明色‘擦’干净后，画上网格，两层叠加后顶层网格部分挡住照片，无遮挡部分露出照片。



- e) 数据寄存器 DATA: 地址 A2A1A0=110,  
 前景色寄存器 FRONT: A2A1A0=100,  
 背景色寄存器 BACK: A2A1A0=101

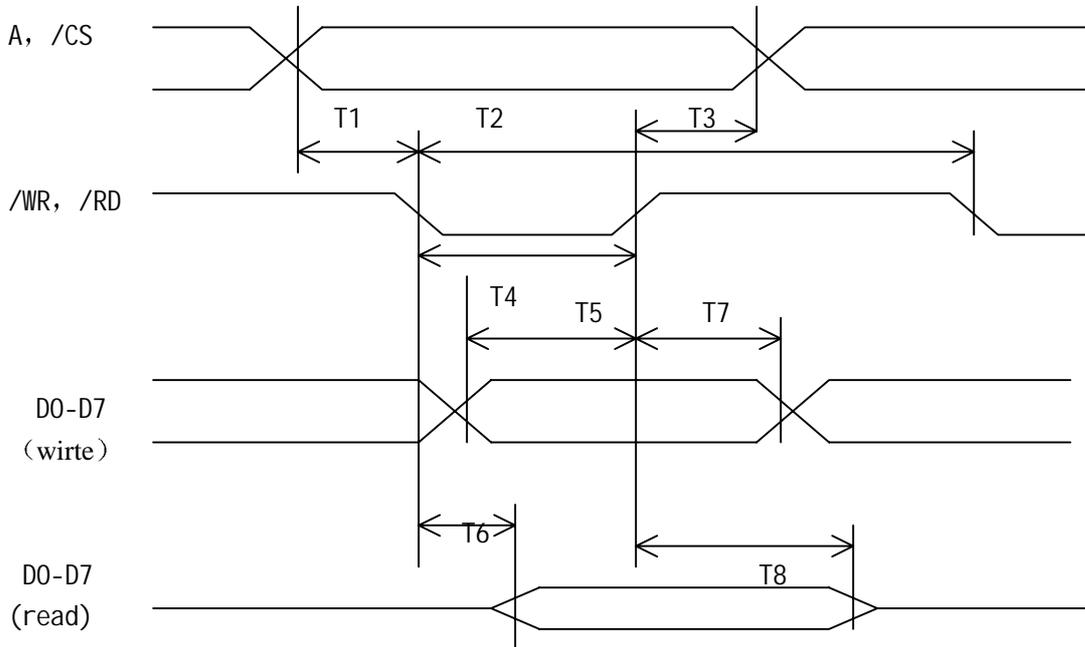
	D7	D6	D5	D4	D3	D2	D1	D0
256 色	R2	R1	R0	G2	G1	G0	B1	B0

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

## 二、电气信号参数:

读写时序图:

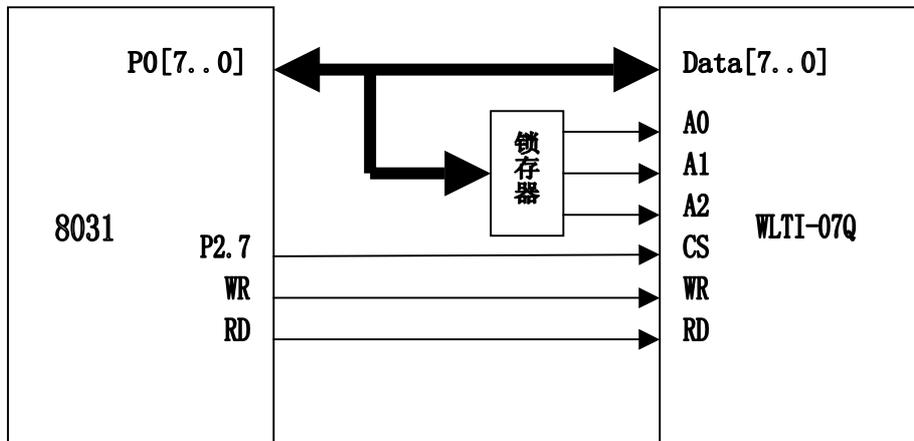


符号	参数说明	最小	最大	单位
T1	地址建立时间	0	-	ns
T3	地址保持时间	5	-	ns
T2	读写同期	20	-	ns
T4	读写脉冲宽度	10	-	ns
T5	写数据建立时间	5	-	ns
T7	写数据保持时间	5	-	ns
T6	读数据建立时间	5	-	ns
T8	读数据保持时间	10	-	ns

\*\*\*对数据寄存器 (A2A1A0=110) 写入操作后一段时间内 (单点写入 300ns、多点或 8 点写入 450ns) 不得对控制板进行任何操作, 以便控制板将 8 个点颜色值写入显存, 如 CPU 时钟很高, 可用插入空指令的办法实现等待。

#### 四、应用程序

下面提供 Intel 51 系列单片机 C51 演示程序



8031 控制方式

```
#define unsigned char uchar;
#define unsigned int uint;
```

TI07V 控制板清屏程序—多点写入方式，一次写 8 点

```
void ClrScan(uchar color)
//color 清屏色
{
    uchar i, j;
    uint sPosX;
    uint sPosY;
    XBYTE[0x0007]=0x05;           //写状态字，多点写顶层
    XBYTE[0x0004] = color;       //front color
    sPosY=0x0000                 //y=0
    sPosX=0x0000                 //x=0
    for(i = 0; i < 240; i ++ )   //一屏 240 行
    {
        sPosY++;
        XWORD[0x0002] = sPosY;
        XWORD[0x0000] = sPosX;   //
        for(j = 0; j < 40; j ++ ) //一行 320 点
        {
            XBYTE[0x0006] = 0xff; //8 个点全写前景色
            nop;                   //延时 450ns
        }
    }
}
```

TI07V 控制板写汉字程序—8点写入方式，一次写8点

```
void ColorWriteMultiWord(uint x, uint y, uchar xNum, uchar yNum, uchar *pData, uchar color)
```

```
//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
```

```
//color 字体色
```

```
//back 背景色
```

```
{  
    uchar i, j;  
    uint sPosX;  
    uint sPosY;  
    XBYTE[0x0007]=0x05;           //写状态字  
    XBYTE[0x0004] = color;        //front color  
    XBYTE[0x0005]=back;          //back color  
    for(i = 0; i < yNum; i ++)  
    {  
        sPosY= y + i;  
        XWORD[0x0002] = sPosY;  
        sPosX= x;  
        XWORD[0x0000] = sPosX.;  
        for(j = 0; j < xNum; j ++)  
        {  
            XBYTE[0x0006] = pData[i*xNum + j];  
            nop;                   //延时 450ns  
        }  
    }  
}
```