

WLAT80 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 (CPLD) EPM240 编程实现，性能稳定可靠。WLAT80 专门控制台湾群创 8 寸 (at080tn42 点阵 800*600, led 背光) 真彩色 TFT 液晶显示屏，采用总线连接方式，可显示 256 色；为便于字符操作，该控制板提供了多点 and 8 点写屏方式，写一个字节，对应屏中 8 个点，只需写入 32 个字节就可完成一个 16 点阵汉字写屏操，比单点写屏方式速度快十几倍。控制板、显示屏及背光电源总功耗为 5V/300mA。

适配 CPU: 51, 96, X86, 8088, Z80, DSP, ARM

一、接口定义:

1、CPU 侧接口 (双排压线)

引脚	符号	功能	备注
1	VCC	控制板电源	5V
2	VCC	控制板电源	5V
3	NC	空脚	
4	DATA0	数据总线	3.3/5V 电平
5	DATA1	数据总线	3.3/5V 电平
6	DATA2	数据总线	3.3/5V 电平
7	DATA3	数据总线	3.3/5V 电平
8	DATA4	数据总线	3.3/5V 电平
9	DATA5	数据总线	3.3/5V 电平
10	DATA6	数据总线	3.3/5V 电平
11	DATA7	数据总线	3.3/5V 电平
12	/CS	片选信号，低电平对屏操作有效	3.3/5V 电平
13	/WR	写操作信号，低电平有效。	3.3/5V 电平
14	/RD	读操作信号，低电平有效。	3.3/5V 电平
15	NC	空脚	
16	A0	寄存器地址 ⁽³⁾	3.3/5V 电平
17	A1	寄存器地址 ⁽³⁾	3.3/5V 电平
18	A2	寄存器地址 ⁽³⁾	3.3/5V 电平
19	GND	电源地	
20	GND	电源地	

二、控制板操作指令系统

WLAT80 控制板中有 8 个寄存器，它们是行 (Y 坐标)、列 (X 坐标)、前景色、背景色、数据、状态控制寄存器，通过对相关寄存器操作，可实现单点读屏，单点、多点、8 点写屏。

所有寄存器只能写，不能读。

(1) CS 与 A0、A1、A2 组合功能如下:

CS	A2A1A0	WR	RD	功能
0	000	0	1	X 坐标低 8 位寄存器

0	001	0	1	X坐标高8位寄存器
0	010	0	1	Y坐标低8位寄存器
0	011	0	1	Y坐标高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
	110	0	1	写显示数据寄存器
	111	0	1	状态控制寄存器
0	×	1	0	读显示数据
1	×	×	×	不选通

说明:

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

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

×	×	×	×	×	×	V9	V8	V7	V6	V5	V4	V3	V2	V1	V0
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

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

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
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

- c) 状态字寄存器 PSW: 地址 A2A1A0=111

D7 D6 D5 D4 D3 D2 D1 D0

Bril	Bri0	X	X	YINC	XINC	Wrcon1	Wrcon0
------	------	---	---	------	------	--------	--------

Bril, bri0 为亮度控制: =11, 最亮; =10, 较亮; =01, 较暗; =00, 黑屏。

D3 和 D2 位根据用户需要可设置成以下两种功能:

Xinc 为列 (X 坐标) 号自动增加控制位, =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) 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

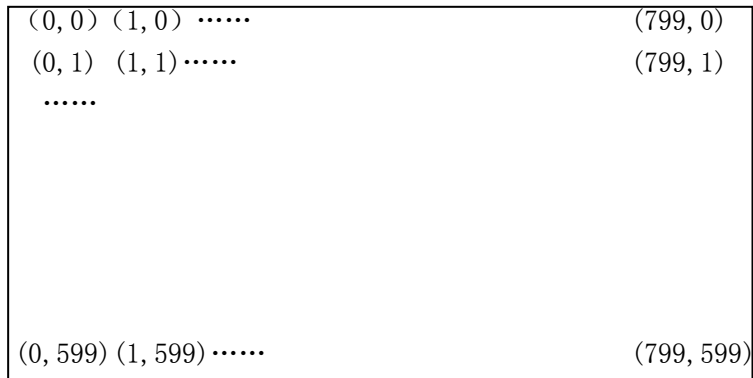
256 色

	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

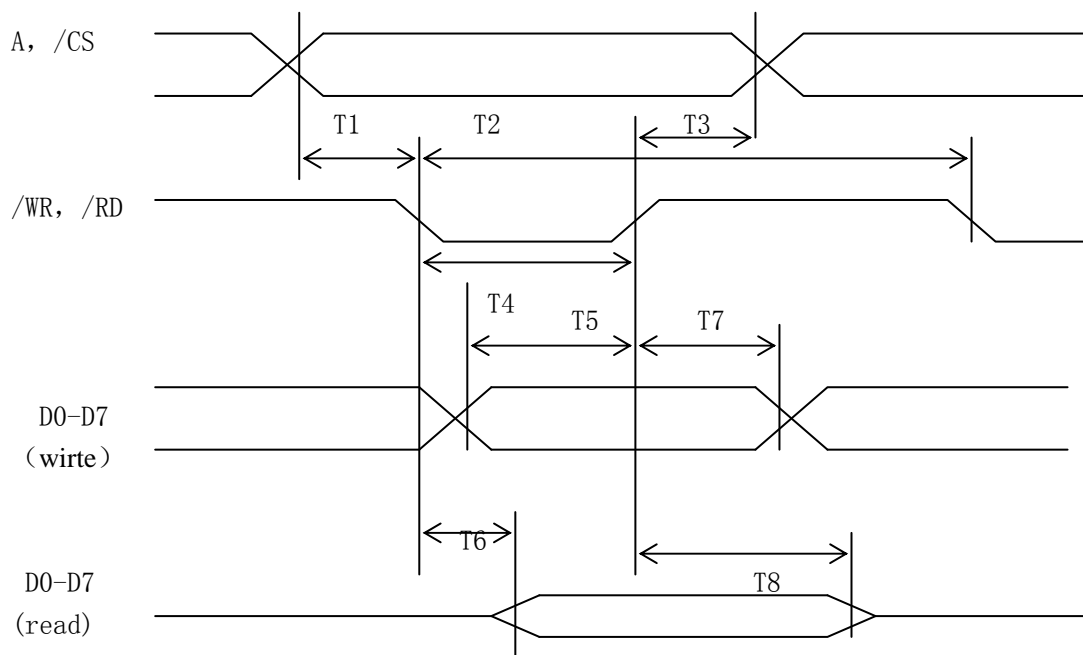
1、显示屏中坐标示意图(X, Y), X 为列号, Y 为行号



原点坐标在左上角。

二、电气信号参数:

读写时序图:

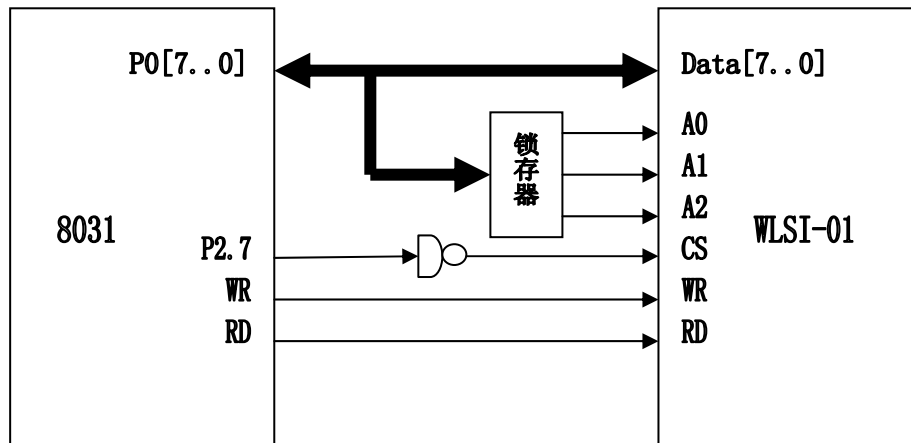


符号	参数说明	最小	最大	单位
T1	地址建立时间	5	-	ns
T3	地址保持时间	5	-	ns
T2	读写同期	20	-	ns
T4	读写脉冲宽度	10	-	ns
T5	写数据建立时间	5	-	ns

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 200ns、多点或 8 点写入 550ns) 不得对控制板进行任何操作, 以便控制板将 8 个点位颜色值写入显存, 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



8031 控制方式

1、C51 应用程序

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

WLAT80 控制板清屏程序—多点写入方式, 一次写 8 点(蓝色字为 WLAT80 控制 800*600)

```
void ClrScan(uchar color)
//color 清屏色
{
    uchar i, j;
    uint sPosX,sPosY; //uchar sPosY;
    XBYTE[0x8007]=0xc5; //多点写入, 且最亮
    XBYTE[0x8004] = color; //front color
    sPosY=0x0000
    sPosX=0x0000
    for(i = 0; i < 600; i ++) //一屏 600 行,for(i=0;i<240;i++)一屏 240 行,
    {
        sPosY++; //
        XWORD[0x8002] = sPosY; //XBYTE[0x0002]=sPosY;
        XWORD[0x8000] = sPosX.; //
```

```

for(j = 0; j < 100; j ++)          //一行 800 点,for(j=0;j<40;j++)
{
    XBYTE[0x8006] = 0xff;
    nop;                            //插入等待
}
}
}

```

WLAT80V 控制板写汉字程序—多点写入方式，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

```

{
    uchar i, j;
    uint  sPosX,sPosY;
    XBYTE[0x8007]=0xc5;          //最亮
    XBYTE[0x8004] = color;      //front color
    for(i = 0; i < yNum; i ++)
    {
        sPosY= y + i;
        XWORD[0x8002] = sPosY;
        sPosX= x;
        XWORD[0x8000] = sPosX.;
        for(j = 0; j < xNum; j ++)
        {
            XBYTE[0x8006] = pData[i*xNum + j];
            nop;                //插入等待
        }
    }
}
}

```