AT56TV16液晶显示控制器使用手册 (1111)

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(CPLD) EPM240 编程实现，性能稳定可靠。为提高读写速度、简化程序，采用总线连接方式，并且显示屏中每个点影射显示缓存中的一个字，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。为便于字符操作，该控制板还提供了多点和 8 点写屏方式。对于八点写屏方式，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*16 点阵汉字写屏操作。

控制板中的多点和 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。

适配 CPU：51,96,X86,8088,Z80,DSP,ARM

AT56TV16 的参数如下：

分辨率	640*480	接口方式	8 位总线	背光类型	LED
写屏方式	单点、多点、8 点	显示颜色	64K 色	背光亮度	200
显示方式	1 页显存	工作电压	5V	工作温度	-20-70
屏幕尺寸	5.6 寸	消耗功率	5V/500mA	保存温度	-20-80

二、控制板接口定义

AT56TV16 的对外接口是 20 针脚的双压排线（连接控制板与适配 CPU），具体定义如下：

引脚	符号	功能	备注
1	VCC	液晶屏逻辑电源	5V
2	VCC	液晶屏逻辑电源	5V
3	NC		
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	A3	寄存器地址	3.3/5V 电平
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	

三、功能寄存器及其说明

AT56TV16 控制板的功能寄存器的地址总线是由 A3 A2 A1 A0 组合而成，具体如下：

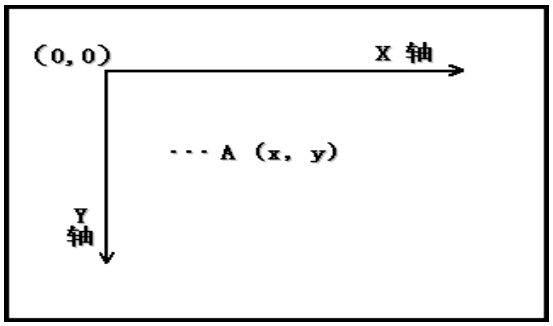
CS	A3A2A1A0	WR	RD	功能
0	0000	0-1	1	X 坐标低 8 位寄存器
0	0001	0-1	1	X 坐标高 8 位寄存器
0	0010	0-1	1	Y 坐标低 8 位寄存器
0	0011	0-1	1	Y 坐标高 8 位寄存器
0	0100	0-1	1	前景色低 8 位寄存器
0	0101	0-1	1	前景色高 8 位寄存器
0	0110	0-1	1	背景色低 8 位寄存器
0	0111	0-1	1	背景色高 8 位寄存器
0	1000	0-1	1	数据低 8 位寄存器
0	1001	0-1	1	数据高 8 位寄存器
0	1010	0-1	1	状态控制寄存器
0	1011	0-1	1	亮度调节
0	1100	0-1	1	（功能待定）
1	×	×	×	不选通

注： 所有寄存器只能写，不能读。

所以的功能寄存器只有在 CS 选通信号为低电平时有效，高电平时寄存器不选通，对寄存器写数据时，读（RD）信号置高电平。以下将对寄存器作详细说明。

1、 X 地址寄存器和 Y 地址寄存器

世龙液晶显示控制模块使用的坐标系与普通笛卡尔坐标系不同，而与通常的图像坐标系相同，即以可视区域的左上角作为坐标系原点，向右为 X 轴正向，向下为 Y 轴正向。如下图所示：



对于 AT56TV16 控制板，因为其控制的 LCD 的分辨率为 640*480，所以我们将液晶屏的横向（X 轴）分为 640 个坐标点，纵向（Y 轴）分为 480 个坐标点，因此 X、Y 坐标各需要两个字节即可完成对 X、Y 坐标数据的存储。具体如下表所示：

X 地址寄存器(X 取值范围 0～639)

A3A2A1A0 = 0001								A3A2A1A0 = 0000							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
×	×	×	×	×	×	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

Y 地址寄存器(Y 取值范围 0～479)

A3A2A1A0 = 0011								A3A2A1A0 = 0010							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
×	×	×	×	×	×	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

2、 前景色和背景色地址寄存器

AT56TV16 液晶显示控制模块控制显示的是 16 位 RGB 颜色模式，最多可表现出 65536 种颜色。16 位颜色的表示采用 5R 6G 5B 模式。即每个像素点用 16 位（两个字节）来表示，其中红色分量（R）占 5 位、绿色分量（G）占 6 位、蓝色分量（B）占 5 位。

前景色和背景色地址寄存器

A3A2A1A0=0101（前景色高），0111（背景色高）								A3A2A1A0=0100（前景色低），0110（背景色低）							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
红色表示位					绿色表示位						蓝色表示位				

例如： 纯红=0xF800 纯绿＝0x07E0 纯蓝＝0x001F

前景色是进行描点画图操作时描点区域内点、线、图形和文字所显示的颜色，背景色是指描点区域内前景色之外的颜色（即衬托前景色的颜色）。

3、 数据地址寄存器

单点写屏时数据寄存器写入的数据是显示的颜色值，多点和 8 点写屏时数据寄存器低 8 位写入的数据是对点位显示的是背景色还是前景色控制的控制字。

A3A2A1A0 = 1001								A3A2A1A0 = 1000							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

4、 状态字地址寄存器 PSW

状态字寄存器 PSW 主要是对写点方式和 X、Y 坐标进行控制，多页显存时，还需要对读写页号和显示页号进行控制。

A3A2A1A0 = 1010							
D7	D6	D5	D4	D3	D2	D1	D0
Off/on	×	×	×	YINC	XINC	Write1	Write0

Off/on ： 背光亮度控制位，=0 打开背光电源，=1 关闭被光电源。

XINC ： 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）写屏时坐标自动加 8。

YINC ： 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 ： 单点写入方式，直接将颜色值写入数据寄存器，与前景色、背景色寄存器内容无关，每次写入一个点；单点写屏时，必须先写高字节(A=1001)，再写低字节(A=1000)。

WRCON1，WRCON0＝01 ： 多点写入方式，将点位信息写入数据寄存器，例如：写入数据寄存器的数据为‘01010101B’则表示一次显示的 8 个点的颜色依此是‘原色、前景色、原色、前景色、原色、前景色、原色、前景色’（即 0 表示原色，1 表示前景色）。

WRCON1，WRCON0＝10 ： 8 点写入方式，将点位信息写入数据寄存器，例如：写入数据寄存器的数据为‘01010101B’则表示一次显示的 8 个点的颜色依此是‘背景色、前景色、背景色、前景色、背景色、前景色、背景色、前景色’（即 0 表示背景色，1 表示前景色）。

多点和 8 点写入方式，一次可写入 8 个点，适用于写字符和清屏，将点位信息写入数据寄存器的低字节(A=1000)。原色是指显示屏原来显示颜色，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。

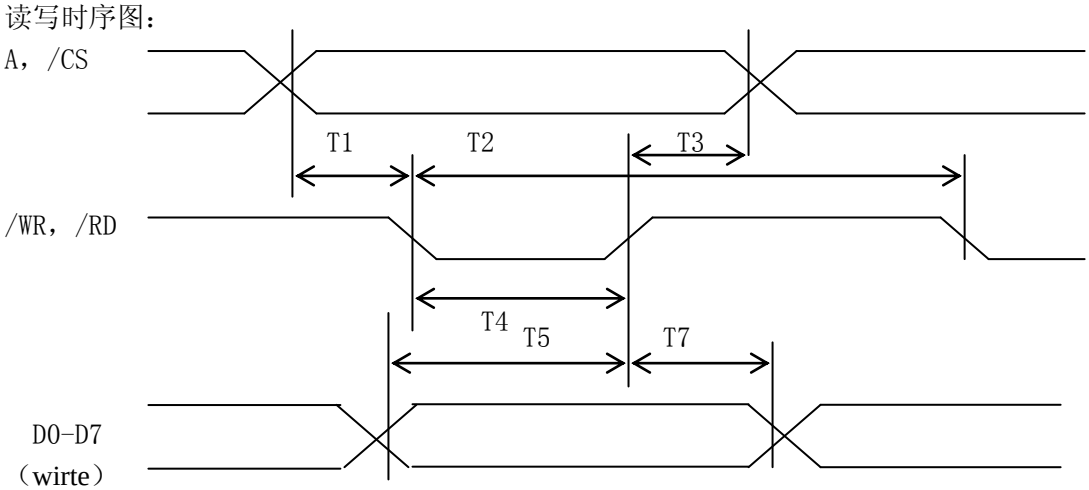
5、 亮度控制寄存器(A3 A2 A1 A0 = 1011)

亮度控制寄存器主要是对液晶显示模块的液晶屏显示的亮度进行控制，AT56TV16 的亮度调节可分为十六个等级，具体如下 ：

亮度寄存器值	0～3	4～7 ⟶	60～63（每 4 个值分为一个亮度等级）
亮暗情况	最亮	最暗 ⟶	次亮 （逐渐变亮）

四、CPU 接口时序

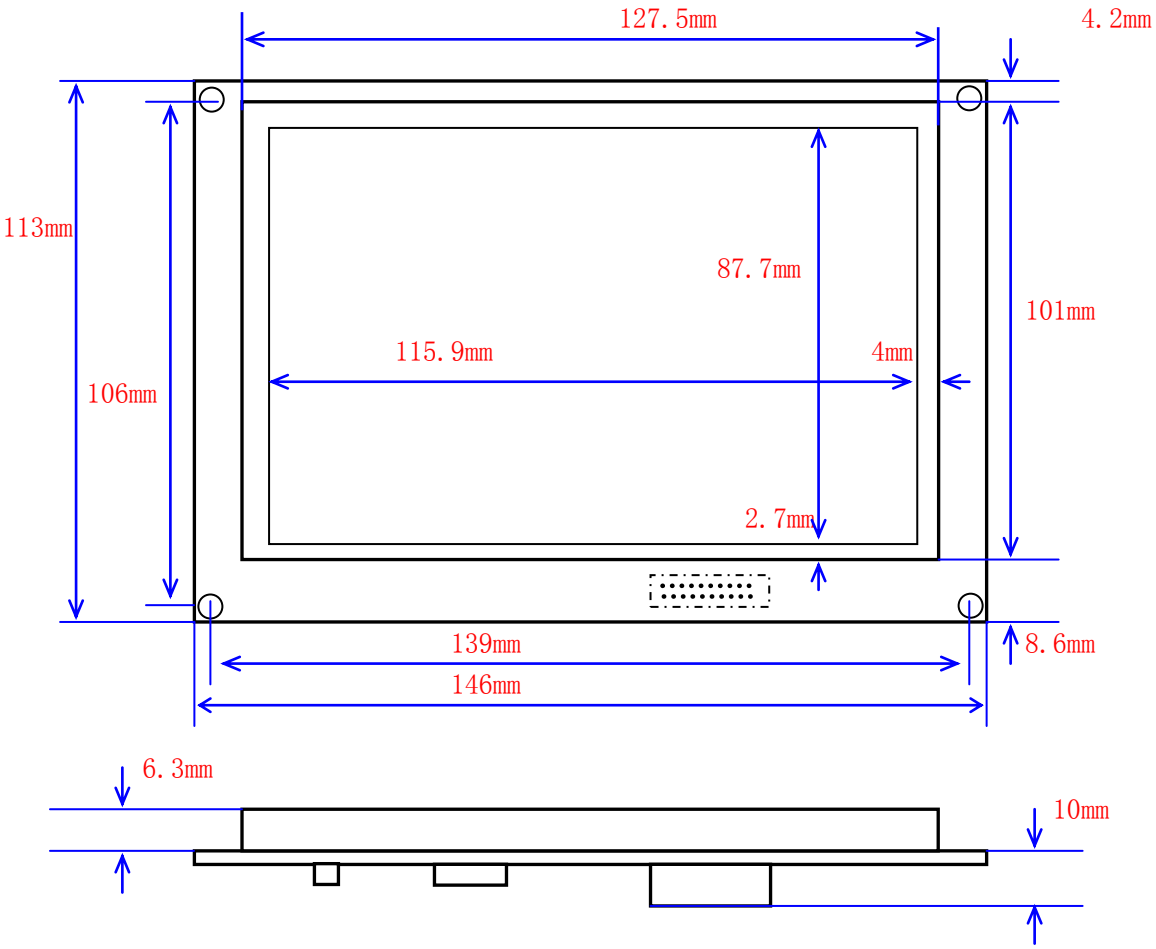
读写时序图：



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	20	-	ns
T5	写数据建立时间	25	-	ns
T7	写数据保持时间	10	-	ns

*****对数据寄存器(A=1000)写入操作后一段时间内(单点写入 100ns、多点或 8 点写入 500ns)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。**

五、控制板尺寸

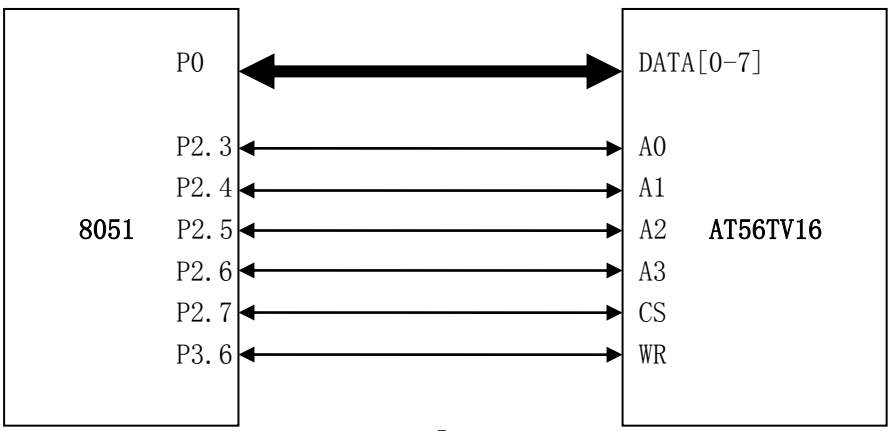


孔径 3.5mm,为便于用户安装，AT56TV16 已将控制板和屏用不锈钢外壳固定成一个整体。

六、应用程序

以下将提供 8051 和 ARM 两种适配 CPU 简单的控制显示的演示程序。

1、 C51 演示程序：8051 和 AT56TV16 之间数据的传输可以用总线（并口）的方式，也可以用串口的方式，总线方式比 I/O 方式速度要快很多，建议用户用总线方式，本示例程序是用总线的方式传输数据的，8051 和 AT56TV16 的接线图如下：



控制板与 8051 连接示意图

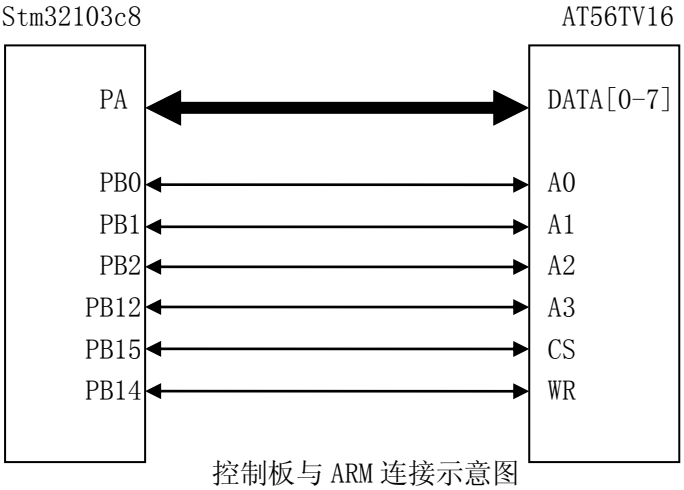
C51 演示程序（演示程序的显示结果见 11 页）

```
#include <reg52.h>
#include <zi48_50.h>
#define uchar unsigned char
#define uint unsigned int
uchar xdata *xlreg=(uchar *)0x0000;          //A3A2A1A0=0000
uchar xdata *xhreg=(uchar *)0x0800;          //A3A2A1A0=0001
uchar xdata *ylreg=(uchar *)0x1000;          //A3A2A1A0=0010
uchar xdata *yhreg=(uchar *)0x1800;          //A3A2A1A0=0011
uchar xdata *frontlreg=(uchar *)0x2000;      //A3A2A1A0=0100
uchar xdata *fronthreg=(uchar *)0x2800;      //A3A2A1A0=0101
uchar xdata *backlreg=(uchar *)0x3000;       //A3A2A1A0=0110
uchar xdata *backhreg=(uchar *)0x3800;       //A3A2A1A0=0111
uchar xdata *dlreg=(uchar *)0x4000;          //A3A2A1A0=1000
uchar xdata *dhreg=(uchar *)0x4800;          //A3A2A1A0=1001
uchar xdata *pswreg=(uchar *)0x5000;         //A3A2A1A0=1010
uchar xdata *brireg=(uchar *)0x5800;         //A3A2A1A0=1011
//画实心矩形 矩形左上角坐标（x，y） 矩形宽=l*8 矩形高=h
void drawrectangle(uint color,uint x,uint y,uint l,uint h,uchar psw)
{
    uint i,j;
    *frontlreg=color;
    *fronthreg=color>>8;
    *pswreg=psw;        //psw=0x05 ;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++)
        *dlreg=0xff;        //一次写入 8 点，只需写低字节 8 点颜色均为前景色 color
    }
}
//写汉字程序一次写 8 点  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
void ColorWriteMultiWord(uint x, uint y, uchar xNum, uchar yNum, uchar *pData,uint
frontcolor,uint backcolor,uchar psw)
{
    uchar i,j;
    uint n=0;
```

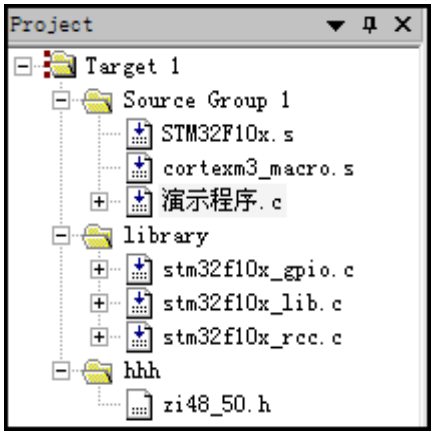
```
    *pswreg=psw;              //psw=0x05 ;多点写屏，=0x06;8 点写屏。
    *frontlreg = frontcolor;
    *fronthreg = frontcolor>>8;
    *backlreg = backcolor;
    *backhreg = backcolor>>8;
    for(i = 0; i < yNum; i ++)
    {
        *ylreg = y;
        *yhreg = y>>8;
        y++;
        *xlreg = x;
        *xhreg=x>>8;
        for(j = 0; j < xNum; j ++)
        *dlreg = pData[n++];          //多点和 8 点写屏,与高字节无关
    }
}
//画圆   半径 r
void round(uint color,uint x,uint y,uint r, uchar psw)
{
    int i,j,k ;
    *pswreg=psw;
    for(j=0;j<2*r;j++)
    {
        *ylreg = y;
        *yhreg = y>>8;
        y++;
        *xlreg = x;
        *xhreg =x>>8;
        for(i=0;i<2*r;i++)
        {
            k=(i-r)*(i-r)+(j-r)*(j-r);
            if(k<r*r) { *dhreg=color>>8;*dlreg=color;  }
            else  { *dhreg=0xff;*dlreg=0xff;  }
        }
    }
}void main()   //主程序
{
    drawrectangle (0xffff,0,0,80,480,0x05);   //全屏清白色
    ColorWriteMultiWord(220,100,6,50,shi,0x001f,0x07e0,0x06);        //8 点写入"世"
    ColorWriteMultiWord(220+1*50,100,6,50,lon,0x001f,0x07e0,0x06);  //8 点写入"龙"
    ColorWriteMultiWord(220+2*50,100,6,50,dian,0x001f,0x07e0,0x06); //8 点写入"电"
    ColorWriteMultiWord(220+3*50,100,6,50,zi,0x001f,0x07e0,0x06);   //8 点写入"子"
    ColorWriteMultiWord(220,200,6,50,shi,0xf800,0x00,0x05);       //多点写入"世"
```

```
    ColorWriteMultiWord(220+1*50,200,6,50,lon,0xf800,0x00,0x05); //多点写入"龙"
    ColorWriteMultiWord(220+2*50,200,6,50,dian,0xf800,0x00,0x05);//多点写入"电"
    ColorWriteMultiWord(220+3*50,200,6,50,zi,0xf800,0x00,0x05);  //多点写入"子"
    round(0xf800,250,320,30,0x04);          // 圆 R
    round(0x07e0,320,320,30,0x04);          // 圆 G
    round(0x001f,390,320,30,0x04);          // 圆 B
    while(1);  //等待
}
```

2、 ARM 演示程序：选用的 ARM 型号为 stm32103c8，采用 GPIO 端口的方式通信数据，其接线方式和工程文件菜单如下图所示：



控制板与 ARM 连接示意图



工程文件菜单

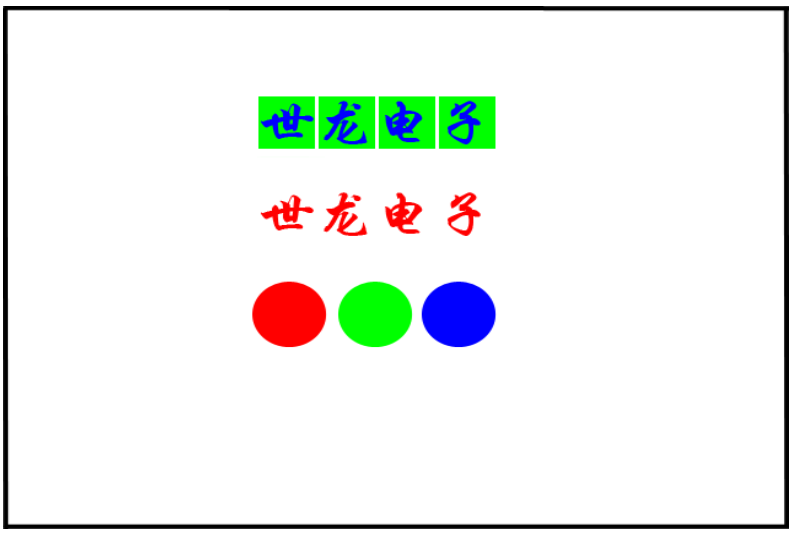
Stm32 演示程序（演示程序的显示效果见 11 页）

```
#include "stm32f10x_lib.h"
#include "zi48_50.h"
#define uchar unsigned char
#define uint unsigned int
#define xlreg  0x0000   //A3A2A1A0=0000
#define xhreg  0x0001   //A3A2A1A0=0001
#define ylreg  0x0002   //A3A2A1A0=0010
#define yhreg  0x0003   //A3A2A1A0=0011
#define flreg  0x0004   //A3A2A1A0=0100
#define fhreg  0x0005   //A3A2A1A0=0101
#define blreg  0x0006   //A3A2A1A0=0110
#define bhreg  0x0007   //A3A2A1A0=0111
#define dlreg  0x1000   //A3A2A1A0=1000
#define dhreg  0x1001   //A3A2A1A0=1001
#define pswreg 0x1002   //A3A2A1A0=1010
#define pswreg 0x1003   //A3A2A1A0=1011
void RCC_Configuration(void)     //时钟配置
{
    RCC_DeInit();  //复位 RCC
    RCC_HSEConfig(RCC_HSE_ON);  //Enable HSE
```

```
    while (RCC_GetFlagStatus(RCC_FLAG_HSERDY) == RESET);  // Wait till HSE is read
    RCC_SYSCLKConfig(RCC_SYSCLKSource_HSE);  //将 HSE 设置为系统时钟源
    RCC_HCLKConfig(RCC_SYSCLK_Div1);   //配置 HCLK 时钟，HCLK 时钟等于 SYSCLK
    RCC_PCLK2Config(RCC_HCLK_Div1);    //配置高速 APB 时钟，APB2 时钟等于 HCLK
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_GPIOB,ENABLE);
                                                          //使能 GPIOA GPIOB 时钟
}
void GPIOA_Configuration(void)  //GPIOA 配置
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin=GPIO_Pin_7|GPIO_Pin_6|GPIO_Pin_5|GPIO_Pin_4
                      |GPIO_Pin_3|GPIO_Pin_2|GPIO_Pin_1|GPIO_Pin_0;  //使能选择引脚
    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;   //引脚最大输出频率为 50HZ
    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;   //推拉输出备用
    GPIO_Init(GPIOA,&GPIO_InitStructure);    //GPIOA 初始化
}
void GPIOB_Configuration(void)  //GPIOB 配置
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_15|GPIO_Pin_14|GPIO_Pin_13
                                  |GPIO_Pin_12|GPIO_Pin_2|GPIO_Pin_1|GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_Init(GPIOB,&GPIO_InitStructure);
}
void writedata(uint a,uint dat) //对控制板写数据
{
    uint b;
    b=0xEFF0|a;
    GPIOB->ODR=b;
    GPIOA->ODR=dat;
    GPIOB->BRR=GPIO_Pin_15;     //CS=0
    GPIOB->BRR=GPIO_Pin_14;     //WR=0
    GPIOB->BSRR=GPIO_Pin_14;    //WR=1
    GPIOB->BSRR=GPIO_Pin_15;    //CS=1
}
//画实心矩形   左上角坐标（x0，y0） 矩形长=8＊1 矩形宽=w
void drawrectangle(uint color,uint x0,uint y0,uchar l,uint w,uchar psw)
{
    uint i,j;
    writedata(pswreg,psw);
    writedata(flreg,color);
    writedata(fhreg,color>>8);
```

```
    for(j=0;j<w;j++)
    {
        writedata(ylreg,y0);
        writedata(yhreg,y0>>8);
        writedata(xlreg,x0);
        writedata(xhreg,x0>>8);
        for(i=0;i<l;i++)
        writedata(dlreg,0xff);
        y0++;
    }
}
//画圆   半径 r
void round(uint color,uint x,uint y,uint r, uchar psw)
{
    int i,j,k ;
    writedata(pswreg,psw);
    for(j=0;j<2*r;j++)
    {
        writedata(ylreg,y);
        writedata(yhreg,y>>8);
        y++;
        writedata(xlreg,x);
        writedata(xhreg,x>>8);
        for(i=0;i<2*r;i++)
        {
            k=(i-r)*(i-r)+(j-r)*(j-r);
            if(k<r*r) { writedata(dhreg,color>>8);writedata(dlreg,color); }
            else  { writedata(dhreg,0xff);writedata(dlreg,0xff); }
        }
    }
}
//写字符 w=字符宽度/8
void writeword(uint fcolor,uint bcolor,uint x,uint y,uint l,uint h,uchar *pdata,uchar
psw)
{
    uint i,j,n=0;
    writedata(pswreg,psw);
    writedata(flreg,fcolor);
    writedata(fhreg,fcolor>>8);
    writedata(blreg,bcolor);
    writedata(bhreg,bcolor>>8);
    for(i=0;i<h;i++)
    {
```

```
        writedata(ylreg,y);
        writedata(yhreg,y>>8);
        writedata(xlreg,x);
        writedata(xhreg,x>>8);
        for(j=0;j<1;j++)
        writedata(dlreg,pdata[n++]);
        y++;
    }
}
int main() //主程序
{
    RCC_Configuration();
    GPIOA_Configuration();
    GPIOB_Configuration();
    GPIOB->BSRR=GPIO_Pin_13;   //WR=1
    drawrectangle(0xFFFF,0,0,80,480,0x05);  //全屏清白色
    writeword(0x001f,0x07e0,220,100,6,50,shi,0x06);   //8 点写入"世龙电子"
    writeword(0x001f,0x07e0,220+1*50,100,6,50,lon,0x06);
    writeword(0x001f,0x07e0,220+2*50,100,6,50,dian,0x06);
    writeword(0x001f,0x07e0,220+3*50,100,6,50,zi,0x06);
    writeword(0xf800,0x0000,220,200,6,50,shi,0x05);     //多点写入"世龙电子"
    writeword(0xf800,0x0000,220+1*50,200,6,50,lon,0x05);
    writeword(0xf800,0x0000,220+2*50,200,6,50,dian,0x05);
    writeword(0xf800,0x0000,220+3*50,200,6,50,zi,0x05);
    round(0xf800,250,320,30,0x04);          // 圆 R
    round(0x07e0,320,320,30,0x04);          // 圆 G
    round(0x001f,390,320,30,0x04);          // 圆 B
    while(1);
}
```



8051 和 ARM 的演示程序编译运行后的显示结果

字体的大小为 48*50 点阵汉子的数据文件“zi48_50.h”具体数据如下，以下为 ARM 演示程序字模数据的定义方式，8051 演示程序的数组需定义为 unsigned char code shi[]={…};

```
/*--  文字:  世  --*/
/*--  华文行楷 36;  此字体下对应的点阵为：宽 x 高=48x50   --*/
unsigned char shi[]={
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x80,0x00,0x00,0x00,0x00,0x1F,0xC0,0x00,
0x00,0x00,0x00,0x3F,0xE0,0x00,0x00,0x00,0x00,0x37,0xF0,0x00,0x00,0x00,0x00,0x77,
0xF0,0x00,0x00,0x00,0x3E,0x67,0xF0,0x00,0x00,0x00,0x1F,0xE7,0xF0,0x00,0x00,0x00,
0x0F,0xE7,0xF0,0x00,0x00,0x00,0x0F,0xE7,0xE0,0x00,0x00,0x07,0x8F,0xF7,0xE0,0x00,
0x00,0x0F,0xCF,0xF7,0xE0,0xE0,0x00,0x0F,0xCF,0xFF,0xFF,0xF8,0x00,0x1F,0xFF,0xFF,
0xFF,0xFC,0x00,0x1F,0xFF,0xFF,0xFF,0xFE,0x00,0xFF,0xFF,0xFF,0xE7,0xFE,0x07,0xFF,
0xFF,0xEF,0xE0,0xFC,0x3F,0xFF,0xFF,0xEF,0xC0,0x00,0x1F,0xFF,0xCF,0xEF,0xC0,0x00,
0x1F,0xFF,0xCF,0xFF,0xC0,0x00,0x0F,0xFF,0xCF,0xFF,0xC0,0x00,0x07,0xEF,0xCF,0xFF,
0xC0,0x00,0x01,0x8F,0xC7,0xFF,0xC0,0x00,0x00,0x0F,0xC7,0xFF,0x80,0x00,0x00,0x0F,
0xC3,0xFF,0x80,0x00,0x00,0x07,0xC0,0x07,0x00,0x00,0x00,0x07,0xC0,0x00,0x00,0x00,
0x00,0x07,0xC0,0x0F,0xF0,0x00,0x00,0x07,0xFF,0xFF,0xFC,0x00,0x00,0x03,0xFF,0xFF,
0xFC,0x00,0x00,0x03,0xFF,0xFF,0xFC,0x00,0x00,0x01,0xFF,0x00,0x00,0x00,0x00,0x00,
0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 };

/*--  文字:  龙  --*/
/*--  华文行楷 36;  此字体下对应的点阵为：宽 x 高=48x50   --*/
unsigned char lon[]={
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x0F,0x80,0x00,0x00,0x00,0x00,0x1F,0xC0,0x00,0x00,0x00,0x00,
0x3F,0xC0,0x00,0x00,0x00,0x00,0x0F,0xC7,0xE0,0x00,0x00,0x00,0x0F,0xCF,0xF8,0x00,
0x00,0x00,0x0F,0xC3,0xF8,0x00,0x00,0x00,0x1F,0x83,0xF8,0x00,0x00,0x00,0x1F,0x81,
0xF0,0x00,0x00,0x00,0x1F,0x00,0x00,0x00,0x00,0x00,0x3F,0xFE,0x00,0x00,0x00,0x00,
0x3F,0xFF,0x00,0x00,0x00,0x00,0x3F,0xFF,0x80,0x00,0x00,0x00,0x7F,0xFF,0x80,0x00,
0x00,0x01,0xFF,0xFF,0x00,0x00,0x00,0x03,0xFF,0xE0,0x00,0x00,0x00,0xFF,0xFF,0x80,
0x00,0x00,0x00,0xFF,0xFE,0x00,0x00,0x00,0x00,0x7F,0xFF,0x00,0x00,0x00,0x00,0x7F,
0xFF,0xE0,0x00,0x00,0x00,0x3F,0xFF,0xEE,0x00,0x00,0x00,0x0F,0xFF,0xC7,0x80,0x00,
0x00,0x03,0xEF,0xC7,0xC0,0x00,0x00,0x07,0xEF,0x8F,0xE0,0x00,0x00,0x07,0xCF,0x9F,
0xE0,0x00,0x00,0x0F,0xDF,0x1F,0xC0,0x00,0x00,0x0F,0x9F,0x3F,0x80,0x00,0x00,0x1F,
0x9F,0x7F,0x00,0x00,0x00,0x1F,0x3E,0xFE,0x02,0x00,0x00,0x3F,0x3F,0xFC,0x07,0x00,
0x00,0x7E,0x3F,0xF8,0x07,0x00,0x00,0x7E,0x3F,0xF0,0x07,0x00,0x01,0xFC,0x3F,0xE0,
0x07,0x80,0x01,0xF8,0x3F,0xC0,0x07,0x80,0x03,0xF0,0x7F,0x80,0x0F,0x80,0x03,0xF0,
```

0xFE, 0x00, 0x0F, 0x80, 0x03, 0xE0, 0xFE, 0x00, 0x1F, 0xC0, 0x00, 0x00, 0x3F, 0xC0, 0xFF, 0xC0,
0x00, 0x00, 0x1F, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0x07, 0xFF, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0xFF,
0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

/*-- 文字: 电;华文行楷 36; 此字体下对应的点阵为：宽 x 高=48x50 --*/
unsigned char dian[]={
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00,
0x7F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x80, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x80, 0x00, 0x00,
0x00, 0x00, 0x3F, 0x80, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xC0,
0x00, 0x00, 0x00, 0x00, 0x1F, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xFE, 0x00, 0x00, 0x00, 0x00,
0x1F, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xF0, 0x00, 0x00, 0x01, 0xFF, 0xC1, 0xF8, 0x00,
0x01, 0xE3, 0xFF, 0x81, 0xFC, 0x00, 0x00, 0xF7, 0xFF, 0x81, 0xFE, 0x00, 0x00, 0xFF, 0xBF, 0xE1,
0xFE, 0x00, 0x00, 0xFF, 0x3F, 0xF3, 0xFE, 0x00, 0x00, 0xFE, 0x7F, 0xFB, 0xFE, 0x00, 0x00, 0xFE,
0xFF, 0xFF, 0xF8, 0x00, 0x00, 0xFF, 0xFF, 0xF7, 0xF0, 0x00, 0x00, 0xFF, 0xFF, 0xEF, 0xE0, 0x00,
0x00, 0xFF, 0xFF, 0xDF, 0xE0, 0x00, 0x00, 0x7E, 0x7F, 0xDF, 0xC0, 0x00, 0x00, 0x7E, 0xFF, 0xFF,
0x80, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x3F, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x3F,
0xFF, 0xFE, 0x00, 0x00, 0x00, 0x3F, 0xFF, 0xFE, 0x00, 0x00, 0x00, 0x1F, 0xFF, 0xFF, 0x00, 0x00,
0x00, 0x1F, 0xFF, 0x8F, 0x00, 0x00, 0x00, 0x0F, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFE, 0x00,
0x00, 0x00, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00,
0x1F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xF8, 0x00,
0x00, 0x00, 0x01, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

/*-- 文字: 子; 华文行楷 36; 此字体下对应的点阵为：宽 x 高=48x50 --*/
unsigned char zi[]={
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x80, 0x00, 0x00, 0x00,
0xC1, 0xFF, 0xC0, 0x00, 0x00, 0x00, 0xF7, 0xF7, 0xE0, 0x00, 0x00, 0x00, 0x7F, 0xE7, 0xE0, 0x00,
0x00, 0x00, 0x7F, 0xCF, 0xE0, 0x00, 0x00, 0x00, 0xFF, 0x9F, 0xE0, 0x00, 0x00, 0x01, 0xFF, 0x3F,
0xC0, 0x00, 0x00, 0x03, 0xFE, 0x3F, 0x80, 0x00, 0x00, 0x07, 0xFC, 0x7F, 0x00, 0x00, 0x00, 0x07,
0xF9, 0xFC, 0x00, 0x00, 0x00, 0x07, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x07, 0xEF, 0xF0, 0x00, 0x00,
0x00, 0x07, 0xCF, 0xFC, 0x00, 0x00, 0x00, 0x03, 0x0F, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF,
0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xE0, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xF0, 0x00, 0x00, 0x00,
0x07, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x1F, 0xFF, 0xFC, 0x00, 0x00, 0x00, 0x7F, 0xFF, 0xFE, 0x00,
0x00, 0x01, 0xFF, 0x9F, 0xFC, 0x00, 0x00, 0x07, 0xFC, 0x1F, 0xB8, 0x00, 0x00, 0x1F, 0xF8, 0x1F,
0x80, 0x00, 0x00, 0x3F, 0xF0, 0x0F, 0x80, 0x00, 0x00, 0x3F, 0xC0, 0x0F, 0x80, 0x00, 0x00, 0x1F,
0x80, 0x0F, 0x80, 0x00, 0x00, 0x1E, 0x00, 0x0F, 0x80, 0x00, 0x00, 0x1E, 0x00, 0x0F, 0x80, 0x00,

0x00, 0x1E, 0x00, 0x1F, 0x80, 0x00, 0x00, 0x0E, 0x00, 0x1F, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x1F,
0x00, 0x00, 0x00, 0x07, 0xC0, 0x1F, 0x00, 0x00, 0x00, 0x03, 0xF0, 0x3F, 0x00, 0x00, 0x00, 0x01,
0xFE, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xFE, 0x00, 0x00,
0x00, 0x00, 0x0F, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x07, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x01, 0xF0,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 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、 如果由于液晶屏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》的请尽量就使用情况和故障现象详细描述，并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

在装好的模块屏表面有一保护膜，以防在装配时玷污显示表面，所以在整机装配结束前不得揭去，以免弄赃或损坏显示面。

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、 液晶模块是精心设计组装而成的，请勿随意自行加工、维修；
2、 金属框架不得随意扭动、拆卸；不要随意修改、加工 PCB 板外形、装配孔、线路及部件；不要随意更改导电胶条，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连接需要焊接或改变原有的连接头的话，确认通过质检；

2、烙铁温度：280±10℃；焊接时间：＜3～4S；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±0.25V）稳定以后接入，才能输入电平信号。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 电路等。

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，还会缩短模块的使用寿命。

4、模块在规定工作温度范围以下使用时，显示响应很慢，而在规定工作温度范围以上使用时，整个显示面又会呈现全显示状态，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。可切断电源，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；放置暗处，避强光；决不能在表面压放任何物品；严格避免在极限温度、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，若是包装受损的话，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，并对其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用户购买货品价值，并不对用户使用产品所造成的间接损失负责。由于用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，但可尽力为用户提供维修服务，并将根据具体情况收取适当费用。