红荷电子双色 OLED(128*64)点阵显示模块产品说明书

一、概述：

红荷电子双色 OLED12864 显示屏所使用的驱动是台湾所罗门公司生产的 SSD1303。

SSD1303 芯片内部功能模块简介

SSD1303 内部主要包括：行列驱动模块、电源控制模块、GDDRAM（Graphic Display Data RAM）、MPU 接口、命令控制模块、振荡器和时序发生器、区域彩色译码模块。SSD1303 内部完整的模块结构和丰富的指令集，给外部数据接口设计和软件设计减少了负担，使得我们的硬件设计也变的十分简单、方便。SSD1303 是集驱动与控制于一体的单色无源矩阵 OLED 图形显示系统，采用 CMOS 工艺制造。它拥有 132 个列驱动和 64 个行驱动引线，最大可用于驱动 132×64 点阵，同时可选择用于四种颜色的驱动，OLED 屏采用了共阴极驱动方式的。SSD1303 内置了对比度控制、GDDRAM（Graphic Display Data RAM ）和振荡器等，在很大程度上减少整个显示系统的外围元件以及降低功耗。

SSD1303 特点如下：

⑴ 大容量矩阵显示

⑵支持四种颜色的区域彩色显示以及每种颜色 64 级亮度控制

⑶工作电压低：VDD=2.4V－3.5V

⑷ OLED 驱动电压：VCC=7.0V－16.0V

⑸最大列电极输出电流：320uA

⑹最大行电极吸入电流：45mA

⑺内置 132×64 位 GDDRAM 缓冲器

⑻单色 256 级对比度控制

⑼ 内置振荡器

⑽ 帧速率及驱动路数可调整

⑾兼容 8bit 6800 和 8bit 8080 系列并行接口，以及串行接口方式。

⑿ 支持行列重映射

⒀ 垂直滚动显示支持

⒁ 水平滚动显示支持

⒂ 低功耗

引脚号	功能	备注
1	3.3V 电源输入	OLED 模块供电
2	GND	电源地
3	D6	数字 IO
4	D7	数字 IO
5	D4	数字 IO
6	D5	数字 IO
7	D2	数字 IO
8	D3	数字 IO
9	D0	数字 IO
10	D1	数字 IO
11	WR	写控制，低有效
12	RD	读控制，低有效

13	RES	OLED 复位，低有效
14	DC	高为数据，低为命令
15	CS	片选信号，低有效
16	NC	无连接

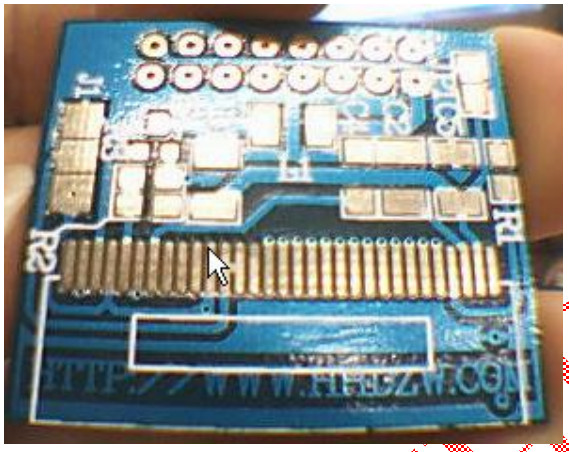
二、接口

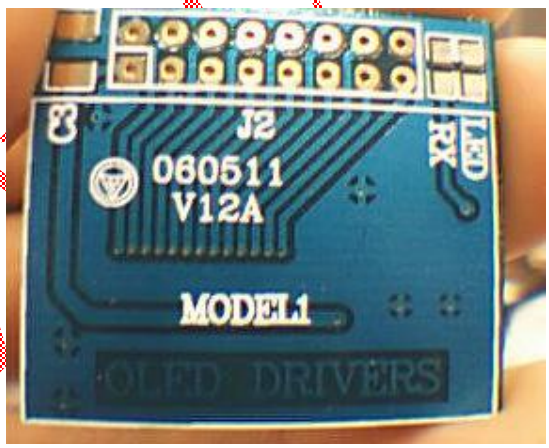
该模块的接口为双列 16 引脚。各引脚定义功能如下表所示：

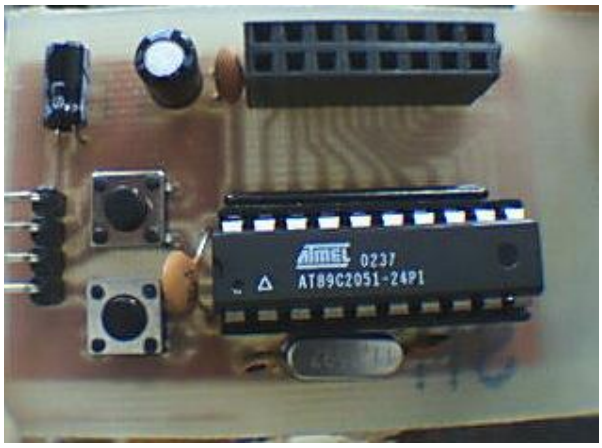
说明：

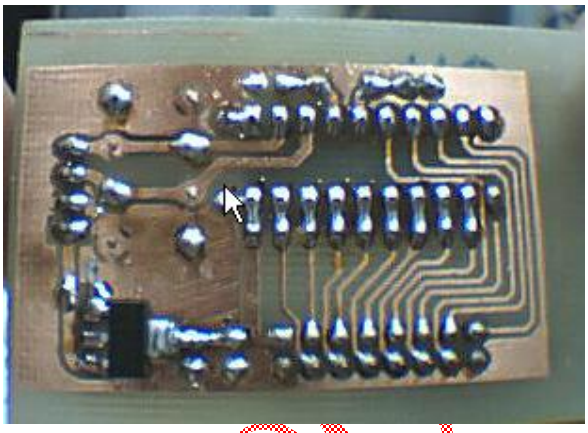
第 1 脚、第 2 脚为电源输入，IO 口供电和 DCDC 的高压都是由 3.3V 而来。此电压不能超过 3.3V。第 3 到 10 为 10 位并行数据接口。此屏支持串口接口方式。当跳线 J1 跳到 VCC 时，为 8 位并行模式，当 J1 跳到 GND 时为三线串行模式。1 到 15 脚为控制信号引脚。16 脚为外部 12V 高压输入。当不使用片的 12V DCDC 电压时，需要从外部接入 12V 电压。此时应该跳上 JP1。

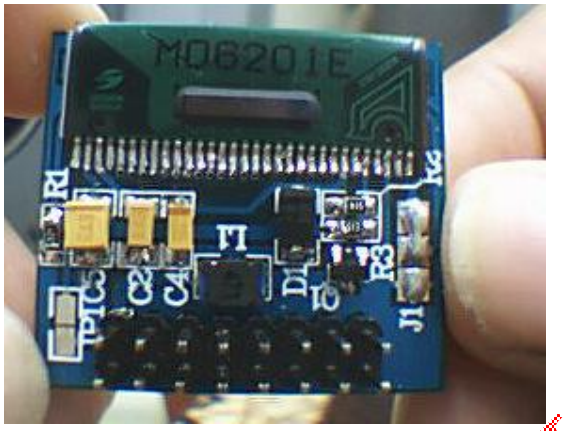
三、产品相关图片及简要说明：






ATMEL 0237
AT89C2051-24PI

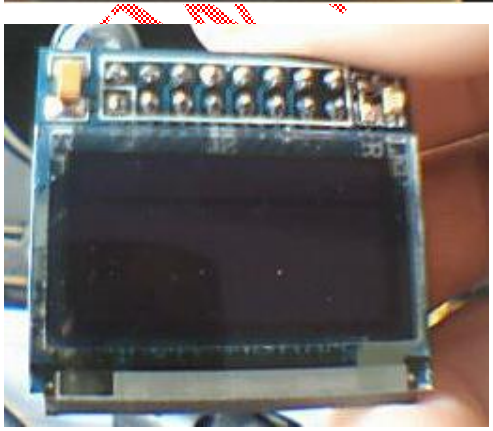



MD6201E


www.hhdzw.com
欢迎你使用红荷电
子产品，资料齐全，
请访问红荷电子网


欢迎你使用红荷电
子产品，资料齐全，
请访问红荷电子网

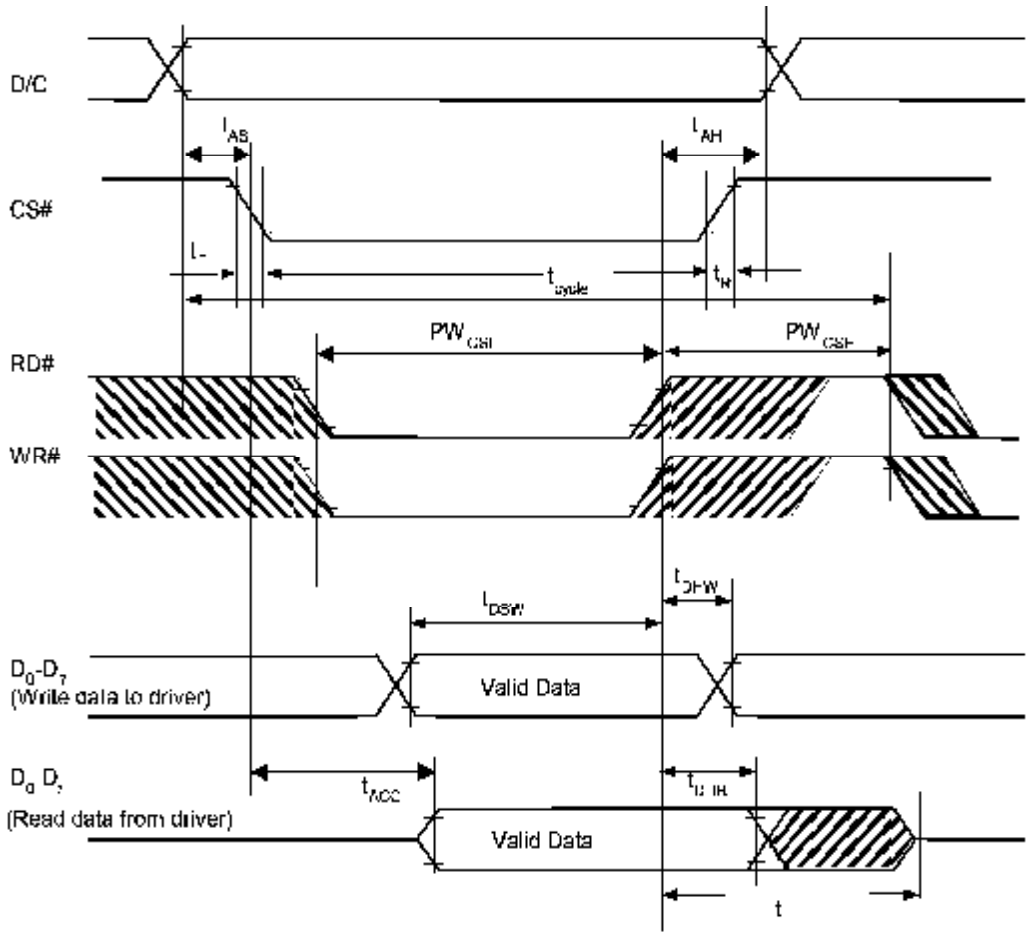

www.hhdzw.com
欢迎你使用红荷电
子产品，资料齐全，
请访问红荷电子网



J1 跳线功能说明：

J1 跳线为焊接跳线，当 J1 焊到 3.3V 时，模块使用 8 并行数据方式，当 J1 焊到 GND 时，使用的是 3 线串行数据传输模式。出厂设置都是 8 位并行数据传输模式。

项目	参数	最小值	典型值	最大值	单位
tcycle	时钟周期	300	-	-	ns
tAS	地址建立时间	0	-	-	ns
tAH	地址锁存时间	0	-	-	ns
tDSW	写数据时间	40	-	-	ns
tDHW	写数据保持时间	15	-	-	ns
tDHR	读数据保持时间	20	-	-	ns
tOH	输出无效时间	-	-	70	ns
tACC	有效时间	-	-	140	ns
tR	上升时间		-	15	ns
tF	下降时间	-	-	15	ns



四：指令说明：

D/C	Hex	D7	D6	D5	D4	D3	D2	D1	D0	指令功能	说明
0	00~0F	0	0	0	0	X3	X2	X1	X0	设置低位列地址	设置小于 16 位的地址，一行为 128 列，设置偏移地址时，低位地十加上高位地址就得到偏移地址。由 X3，X2，X1，X0 设定。上电复位时为 0000。
0	10~1F	0	0	0	1	X3	X2	X1	X0	设置高位列地址	设置高位列地址。真实的偏移地址值为：高位的偏移*16+低位偏移。由 X3，X2，X1，X0 确定。上电复位时为 0000

0	26	0	0	1	0	0	1	1	0	水平滚动设置	A[2:0] 设置每次滚动的列数。有效的设置为： 001b, 010b, 011b, 100b B[2:0] 水平移动的起始页地址设定 C[1:0]滚动帧频设定： 00b – 12 帧 01b – 64 帧 10b – 128 帧 11b – 256 帧 D[2:0]滚动结束页地址设定
0	A[2:0]	*	*	*	*	*	A2	A1	A0		
0	B[2:0]	*	*	*	*	*	B2	B1	B0		
0	C[1:0]	*	*	*	*	*	*	C1	C0		
0	D[2:0]	*	*	*	*	*	D2	D1	D0		
0	2F	0	0	1	0	1	1	1	1	水平滚动使能	水平滚动开始
0	2E	0	0	1	0	1	1	1	0	水平滚动关闭	停止水平滚动
0	81	1	0	0	0	0	0	0	1	对比度设置	双字节指令用来设置 256 级的对比度调节寄存器。上电时为一半， 0x80(127).(POR = 80h)
0	A[7:0]	A7	A6	A5	A4	A3	A2	A1	A0		
0	A0~ A1	1	0	1	0	0	0	0	X0	设置列映射	X0=0: 地址 0 映射到列 0，地址 127 映射到列 127。上电为此值。 X0=1: 地址 0 映射到列 127，地址 127 映射到列 0
0	A5	1	0	1	0	0	1	0	1	全部显示开	显示设置开，一般不用此指令，只是在上电初始化用。
0	A6~A7	1	0	1	0	0	1	1	X0	正常，反白显示设置	X0=0:正常显示 (POR) X0=1: 反白显示
0	A8	1	0	1	0	1	0	0	0	设置使用的行数	双字节指令 A[5:0] 决定使用的行数，上电时 64 使用行
0 0	AD	1 1	0 0	1 0	0 0	1 1	1 0	0 1	1 X0	-DC-DC 打开/关闭	X0=1 DC-DC 功能开启(POR) X0=0 DC-DC 关闭

0 0	AE-AF	1 1	0 0	1 1	0 0	1 1	1 1	1 1	0 1	显示开启，关闭	0xaf:显示打开 0xae:显示关闭
0 0	DA	1 0	1 0	0 0	1 X4	10	0 0	1 1	0 0	列地址映射配置	X4=0，标准映射(i.e. COM31，30，29…0；SEG0-132; COM31,32….62,63) X4=1(POR)，变换映射(i.e. COM62,60,58,…2,0; SEG0-132; COM1,3,5…61,63)

五、驱动头文件说明及格式说明：

51 和 AVR 使用相同的 C 语言指令，伪码的格式完全一致，只是底层驱动稍有不同。测试程序所使用的字库如下所示。

文件名：zimo.h

```
//说明：显示的结果是数组名，传送参数时，就传数组名可以得到显示结果。
unsigned char code h[]={0x10,0x04,0x1F,0xFC,0x00,0x84,0x01,0x00,0x01,0x00,0x01,0x04,0x00,0xFC,0x00,0x04};//'h'
unsigned char code w[]={0x01,0xF0,0x01,0x0C,0x00,0x30,0x01,0xC0,0x00,0x30,0x01,0x0C,0x01,0xF0,0x01,0x00};//'w'
unsigned char code d[]={0x00,0x00,0x00,0x70,0x00,0x88,0x01,0x04,0x01,0x04,0x11,0x08,0x1F,0xFC,0x00,0x04};//'d'
unsigned char code z[]={0x00,0x00,0x01,0x84,0x01,0x0C,0x01,0x34,0x01,0x44,0x01,0x84,0x01,0x0C,0x00,0x00};//'z'
unsigned char code dot[]={0x00,0x00,0x00,0x0C,0x00,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};//dot
unsigned char code c[]={0x00,0x00,0x00,0x70,0x00,0x88,0x01,0x04,0x01,0x04,0x01,0x04,0x00,0x88,0x00,0x00};//'c'
unsigned char code o[]={0x00,0x00,0x00,0xF8,0x01,0x04,0x01,0x04,0x01,0x04,0x01,0x04,0x00,0xF8,0x00,0x00};//'o'
unsigned char code m[]={0x01,0x04,0x01,0xFC,0x01,0x04,0x01,0x00,0x01,0xFC,0x01,0x04,0x01,0x00,0x00,0xFC};//'m'
unsigned char code mao[]={0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x0C,0x03,0x0C,0x00,0x00,0x00,0x00,0x00,0x00};//:
unsigned char code mp[]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};//
unsigned char code year[]={0x02,0x20,0x04,0x20,0x08,0x20,0x30,0x20,0xC7,0xE0,0x44,0x20,0x44,0x20,0x44,0x20,
0x7F,0xFF,0x44,0x20,0x44,0x20,0x44,0x20,0x44,0x20,0x40,0x20,0x00,0x20,0x00,0x00};//年
unsigned char code month[]={0x00,0x00,0x00,0x02,0x00,0x04,0x00,0x08,0x00,0x30,0xFF,0xC0,0x88,0x80,0x88,0x80,
0x88,0x80,0x88,0x84,0x88,0x82,0xFF,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};//月
unsigned char code day[]={0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFC,0x42,0x08,0x42,0x08,0x42,0x08,0x42,0x08,
0x42,0x08,0x42,0x08,0x42,0x08,0x7F,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};//日
unsigned char code xing[]={0x00,0x00,0x00,0x12,0x00,0x62,0x7D,0x82,0x54,0x92,0x54,0x92,0x54,0x92,0x57,0xFE,
```

0x54,0x92,0x54,0x92,0x54,0x92,0x54,0x92,0x7C,0x92,0x00,0x82,0x00,0x02,0x00,0x00};//星
unsigned char code qi[]={0x00,0x42,0x20,0x44,0xFF,0xD8,0x2A,0x40,0x2A,0x40,0x2A,0x50,0xFF,0xCC,0x20,0x46,
0x00,0x18,0x7F,0xE0,0x44,0x40,0x44,0x44,0x44,0x42,0x7F,0xFC,0x00,0x00,0x00,0x00};//期
unsigned char code yi[]={0x00,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,
0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x01,0x00,0x03,0x00,0x01,0x00,0x00,0x00};//一
unsigned char code er[]={0x00,0x00,0x00,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,0x20,0x08,
0x20,0x08,0x20,0x08,0x20,0x08,0x60,0x08,0x20,0x08,0x00,0x18,0x00,0x08,0x00,0x00};//二
unsigned char code san[]={0x00,0x00,0x20,0x04,0x21,0x04,0x21,0x04,0x21,0x04,0x21,0x04,0x21,0x04,0x21,0x04,
0x21,0x04,0x21,0x04,0x21,0x04,0x21,0x04,0x21,0x04,0x20,0x04,0x00,0x04,0x00,0x00};//三
unsigned char code si[]={0x00,0x00,0x7F,0xFE,0x40,0x14,0x40,0x24,0x40,0xC4,0x7F,0x04,0x40,0x04,0x40,0x04,
0x7F,0x84,0x40,0x44,0x40,0x44,0x40,0x44,0x40,0x44,0x7F,0xFE,0x00,0x00,0x00,0x00};//四
unsigned char code wu[]={0x00,0x04,0x40,0x04,0x41,0x04,0x41,0x04,0x41,0x04,0x41,0xFC,0x7F,0x04,0x41,0x04,
0x41,0x04,0x41,0x04,0x43,0xFC,0x41,0x04,0x40,0x04,0x00,0x0C,0x00,0x04,0x00,0x00};//五
unsigned char code liu[]={0x08,0x00,0x08,0x02,0x08,0x04,0x08,0x08,0x08,0x30,0x89,0xC0,0x48,0x80,0x78,0x00,
0x29,0x00,0x08,0x80,0x08,0x40,0x08,0x30,0x08,0x1E,0x08,0x0C,0x08,0x00,0x00,0x00};//六
unsigned char code ri[]={0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFC,0x42,0x08,0x42,0x08,0x42,0x08,0x42,0x08,
0x42,0x08,0x42,0x08,0x42,0x08,0x7F,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};//日

当本驱动用于 AVR 时，要将 code 改为 const。

以下为 51 的驱动头文件：

```
#ifndef  _SSD1303_H_
#define  _SSD1303_H_

//以下为接口定义，根据用户的实验板接口不同而自行定义

#define DAT P1
sbit RD=P3^2;
sbit DC=P3^3;
sbit RES=P3^4;
sbit CS=P3^5;
sbit WR=P3^7;
unsigned char code num[10][16]={
{0x00,0x00,0x07,0xF0,0x08,0x08,0x10,0x04,0x10,0x04,0x08,0x08,0x07,0xF0,0x00,0x00},/*"0"*/
{0x00,0x00,0x08,0x04,0x08,0x04,0x1F,0xFC,0x00,0x04,0x00,0x04,0x00,0x00,0x00,0x00},/*"1"*/
{0x00,0x00,0x0E,0x0C,0x10,0x14,0x10,0x24,0x10,0x44,0x11,0x84,0x0E,0x0C,0x00,0x00},/*"2"*/
{0x00,0x00,0x0C,0x18,0x10,0x04,0x11,0x04,0x11,0x04,0x12,0x88,0x0C,0x70,0x00,0x00},/*"3"*/
{0x00,0x00,0x00,0xE0,0x03,0x20,0x04,0x24,0x08,0x24,0x1F,0xFC,0x00,0x24,0x00,0x00},/*"4"*/
{0x00,0x00,0x1F,0x98,0x10,0x84,0x11,0x04,0x11,0x04,0x10,0x88,0x10,0x70,0x00,0x00},/*"5"*/
{0x00,0x00,0x07,0xF0,0x08,0x88,0x11,0x04,0x11,0x04,0x18,0x88,0x00,0x70,0x00,0x00},/*"6"*/
{0x00,0x00,0x1C,0x00,0x10,0x00,0x10,0xFC,0x13,0x00,0x1C,0x00,0x10,0x00,0x00,0x00},/*"7"*/
{0x00,0x00,0x0E,0x38,0x11,0x44,0x10,0x84,0x10,0x84,0x11,0x44,0x0E,0x38,0x00,0x00},/*"8"*/
{0x00,0x00,0x07,0x00,0x08,0x8C,0x10,0x44,0x10,0x44,0x08,0x88,0x07,0xF0,0x00,0x00} /*"9"*/
```

```
};
//*****************************主要操作函数*************************************
void WriteCommand(unsigned com);          //写命令程序
void WriteData(unsigned dat);             //写数据程序
void esbusini(void);                      //初始化总线程序
void ini_oled(void);                      //OLED 初始化
void ini_dis(void);                       //显示初始化(清除缓冲区)
/*****************************实现函数*****************************************
*函数原型:          unsigned char ReadCommand(void);
*功      能:          从 oled 上读当前命令到控制器。
*****************************************************************************/
void WriteCommand(unsigned com)
{
    CS=1;
    DC=0;
    WR=0;
    RD=1;
    CS=0;
    DAT=com;
    CS=1;
}
/*****************************实现函数*****************************************
*函数原型:          void WriteData(unsigned dat);
*功      能:          写数据到 oled 显示屏。
*****************************************************************************/
void WriteData(unsigned dat)
{
    CS=1;
    DC=1;
    WR=0;
    RD=1;
    CS=0;
    DAT=dat;
    CS=1;
}
/*****************************实现函数*****************************************
*函数原型:          void esbusini(void);
*功      能:          总线初始化。
*****************************************************************************/
void esbusini(void)
{
    RES=0;
    RES=1;
}
/*****************************实现函数*****************************************
```

```
*函数原型:        void ini_dis(void);
*功    能:        显示初始化。
*******************************************************************************/
void ini_dis(void)
{
    unsigned char i,j;
    for(i=0;i<8;i++)
    {
        WriteCommand (0xb0+i);      //设置显示位置—行
        WriteCommand (0x00);        //设置显示位置—列低地址
        WriteCommand (0x10);        //设置显示位置—列高地址
        for(j=0;j<128;j++)
            WriteData(0x00);          //屏幕显示，全亮
    }
}
/**************************实现函数********************************************
*函数原型:        void ini_oled(void);
*功    能:        oled 显示的准备工作。
*******************************************************************************/
void ini_oled(void)
{
    esbusini();
    /*********************************************
    // SSD1303 Initialization Command
    *********************************************/
    // Lower Column Address
    WriteCommand(0x00); /* Set Lower Column Address */
    // High Column Address
    WriteCommand(0x10); /* Set Higher Column Address*/
    // Display Start Line
    WriteCommand(0x40); /* Set Display Start Line */
    // Contrast Control Register
    WriteCommand(0x81); /* Set Contrast Control */
    WriteCommand(0x20); /* 0 ~ 255 */
    // Re-map
    WriteCommand(0xA0); /* [A0]:column address 0 is map
    to SEG0 ; [A1]: columnaddress 131 is map to SEG0*/
    // Entire Display ON/OFF
    WriteCommand(0xA4); /* A4=ON */
    // Normal or Inverse Display
    WriteCommand(0XA6); /* Normal Display*/
    // Multiplex Ratio
    WriteCommand(0xA8); /* Set Multiplex Ratio */
    WriteCommand(0x3f); /* Set to 36 Mux*/
    // Set DC-DC
```

```
    WriteCommand(0xAD); /* Set DC-DC */
    WriteCommand(0x8A); /* 8B=ON, 8A=Off */
    // Display ON/OFF
    WriteCommand(0xAE); /* AF=ON , AE=OFF*/
    // Display Offset
    WriteCommand(0xD3); /* Set Display Offset */
    WriteCommand(0x00); /* No offset */
    // Display Clock Divide
    WriteCommand(0xD5); /* Set Clock Divide */
    WriteCommand(0x20); /* Set to 80Hz */
    // Area Color Mode
    WriteCommand(0xD8); /* Set Area Color On or Off*/
    WriteCommand(0x00); /* Mono Mode */
    // COM Pins Hardware Configuration
    WriteCommand(0xDA); /* Set Pins HardwareConfiguration */
    WriteCommand(0x12);
    // VCOMH
    WriteCommand(0xDB); /* Set VCOMH */
    WriteCommand(0x00);
    // VP
    WriteCommand(0xD9); /* Set VP */
    WriteCommand(0x22); /* P1=2 , P2=2 */
    WriteCommand(0xc0);//配置成标准应用
    ini_dis();
}
/*********************************function*************************************************************/
//********************显示X，Y坐标处的一个字符。一行可以显示16个字符。X为0到15，Y为0到3**************
void disc(unsigned char X,unsigned char Y,unsigned char * c)
{
    unsigned char n;
    WriteCommand (0xb7-(Y<<1));
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
    for(n=0;n<=15;n+=2)
        WriteData(*(c+n));

    WriteCommand (0xb7-(Y<<1)-1);
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
```

```
    for(n=1;n<=15;n+=2)
        WriteData(*(c+n));
}
// X    is 0 to 7
void dish(unsigned char X,unsigned char Y,unsigned char * h)
{
    unsigned char n;
    WriteCommand (0xb7-(Y<<1));
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
    for(n=0;n<=31;n+=2)
        WriteData(*(h+n));

    WriteCommand (0xb7-(Y<<1)-1);
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
    for(n=1;n<=31;n+=2)
        WriteData(*(h+n));
}
void disn(unsigned char X,unsigned char Y,unsigned char n)
{
    unsigned char m;
    WriteCommand (0xb7-(Y<<1));
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
    //for(n=0;n<=15;n+=2)
    //    WriteData(*(c+n));
    for(m=0;m<=15;m+=2)
        WriteData(*(num[n]+m));

    WriteCommand (0xb7-(Y<<1)-1);
    if(X%2)
        WriteCommand (0x08);
    else
        WriteCommand (0x00);
    WriteCommand (0x10+(X>>1));
```

```
    for(m=1;m<=15;m+=2)
        WriteData(*(num[n]+m));
}
/*************************************************************************
*****************file is end here*********************************
*************************************************************/
#endif
```

以下为 AVR 的驱动头文件：

```
/*
*Copyright (c) 2005,桂林电子科技大学红荷电子科技
*All rights reserved .
*
*文件名称:ssd1303.h
*文件标识:见 C 程序描述
*当前版本:V1.2
*作者:陈超
*完成日期:2006.5.29
*
*取代版本:V1.1
*原作者:陈超
*完成日期:2006.5.28
*/
/*文件说明:
液晶的驱动程序头文件,适合的用 ssd1303\ssd1332 驱动的所有 oled 显示屏（包括彩色）
*/
/*
V1.2 新增加功能:1。可以直接写图片。两写入方式。
                2。一是写（上/下）半屏图。另一是写全屏图。
                3。新增加写单个汉字子函数。
                4。新增加写片上自带的 DCDC 程序段。
*/
//Start here//
//编译器:  GCC-AVR V20040502  日期: 2006-06-04 13:29:57
//目标芯片  : M16
//时钟: 8.0000Mhz
/*-------------------------------------------------------
OLED 引脚定义
1---VTEST
2---GND
3---V3.3
4---GND
5---GND
6 到 13--D7-D0
```

14--RD
15--WR
16--DC
17--RES
18--CS
19--BS2
20--BS1
--*/
#include <avr/io.h>
//#include <avr/delay.h>
#include <avr/interrupt.h>
/*--
下面是 AVR 与 OLED 连接信息
PA->D0--D7
PC3->RD
PC2->WR
PC1->DC
PC0->CS
PD7->RES
要使用本驱动，改变下面配置信息即可
--*/
#define OLED_RD_PORT PORTC //以下 4 个要设为相一个口
#define OLED_RD_DDR DDRC
#define OLED_WR_PORT PORTC
#define OLED_WR_DDR DDRC
#define OLED_DC_PORT PORTC
#define OLED_DC_DDR DDRC
#define OLED_CS_PORT PORTC
#define OLED_CS_DDR DDRC
//以上为四个重要操作的相关定义
#define OLED_RES_PORT PORTD //RES 个要设为同一个口
#define OLED_RES_DDR DDRD
//以上为复位操作
#define OLED_DATA_PORT PORTA
#define OLED_DATA_DDR DDRA
#define OLED_DATA_PIN PINA
//以上为数据端口定义
#define OLED_RD (1<<PC3)
#define OLED_WR (1<<PC2)
#define OLED_DC (1<<PC1)
#define OLED_CS (1<<PC0)
#define OLED_RES (1<<PD7)
//以上为接口定义

//OLDE INITIAL FUNCTION AND OPERATERS

```
unsigned char ReadData(void);
unsigned char ReadCommand(void);
void WriteCommand(unsigned com);
void WriteData(unsigned dat);
void esbusini(void);
void ini_oled(void);
void ini_dis(void);

/**************************实现函数**********************************************
*函数原型:          unsigned char ReadData(void);
*功      能:          从 oled 上读数据（显示 RAM 的数据）到控制器。
********************************************************************************/
unsigned char ReadData(void)
{
    unsigned char temp;
    OLED_DC_DDR|=OLED_DC;
    OLED_DC_PORT|=OLED_DC;
    OLED_RD_DDR|=OLED_RD;
    OLED_RD_PORT&=~OLED_RD;
    OLED_WR_DDR|=OLED_WR;
    OLED_WR_PORT|=OLED_WR;
    OLED_DATA_DDR=0X00;
    temp=OLED_DATA_PORT;
    return temp;
}
/**************************实现函数**********************************************
*函数原型:          unsigned char ReadCommand(void);
*功      能:          从 oled 上读当前命令到控制器。
********************************************************************************/
unsigned char ReadCommand(void)
{
    unsigned char temp;
    OLED_DC_DDR|=OLED_DC;
    OLED_DC_PORT&=~OLED_DC;
    OLED_RD_DDR|=OLED_RD;
    OLED_RD_PORT&=~OLED_RD;
    OLED_WR_DDR|=OLED_WR;
    OLED_WR_PORT|=OLED_WR;
    OLED_DATA_DDR=0X00;
    temp=OLED_DATA_PORT;
    return temp;
}
/**************************实现函数**********************************************
*函数原型:          unsigned char ReadCommand(void);
```

```
*功      能:         从 oled 上读当前命令到控制器。
*************************************************************************/
void WriteCommand(unsigned com)
{
    OLED_CS_DDR|=OLED_CS;
    OLED_CS_PORT|=OLED_CS;

    OLED_DC_DDR|=OLED_DC;          // USE AS OUTPUT
    OLED_DC_PORT&=~OLED_DC;            //DC=0;    WRITE COMMAND

    OLED_WR_DDR|=OLED_WR;          //AS OUT
    OLED_WR_PORT&=~OLED_WR;            //ENABLE EQUIL 0

    OLED_RD_DDR|=OLED_RD;          //DISABLE EQUIL 0
    OLED_RD_PORT|=OLED_RD;         //0

    OLED_CS_PORT&=~OLED_CS;

    OLED_DATA_DDR=0XFF;            //ALL AS OUTPUT
    OLED_DATA_PORT=com;            //OUTPUT THE INFORMATION

    OLED_CS_PORT|=OLED_CS;

}
/**************************实现函数**************************************
*函数原型:         void WriteData(unsigned dat);
*功      能:         写数据到 oled 显示屏。
*************************************************************************/
void WriteData(unsigned dat)
{
    OLED_CS_DDR|=OLED_CS;
    OLED_CS_PORT|=OLED_CS;

    OLED_DC_DDR|=OLED_DC;          //USE AS OUT
    OLED_DC_PORT|=OLED_DC;         //DC=1;

    OLED_WR_DDR|=OLED_WR;          //READ SET TO
    OLED_WR_PORT&=~OLED_WR;
    OLED_RD_DDR|=OLED_RD;
    OLED_RD_PORT|=OLED_RD;
    OLED_CS_PORT&=~OLED_CS;
    OLED_DATA_DDR=0XFF;            //USE AS OUT
    OLED_DATA_PORT=dat;            //SEND USEFUL INFORMATION
    OLED_CS_PORT|=OLED_CS;
```

```
}
/**************************实现函数**********************************************
*函数原型:          void esbusini(void);
*功    能:          总线初始化。
*********************************************************************************/
void esbusini(void)
{
    OLED_RES_DDR|=OLED_RES;
    OLED_RES_PORT|=OLED_RES;
}
/**************************实现函数**********************************************
*函数原型:          void ini_dis(void);
*功    能:          显示初始化。
*********************************************************************************/
void ini_dis(void)
{
    unsigned char i,j;
    for(i=0;i<8;i++)
    {
        WriteCommand (0xb0+i);      //设置显示位置—行
        WriteCommand (0x00);        //设置显示位置—列低地址
        WriteCommand (0x10);        //设置显示位置—列高地址
        for(j=0;j<128;j++)
        WriteData(0x00);            //屏幕显示，全亮
    }
}
/**************************实现函数**********************************************
*函数原型:          void ini_oled(void);
*功    能:          oled 显示的准备工作。
*********************************************************************************/
void ini_oled(void)
{
    esbusini();
    /*********************************************
    // SSD1303 Initialization Command
    *********************************************/
    // Lower Column Address
    WriteCommand(0x00); /* Set Lower Column Address */
    // High Column Address
    WriteCommand(0x10); /* Set Higher Column Address*/
    // Display Start Line
    WriteCommand(0x40); /* Set Display Start Line */
    // Contrast Control Register
    WriteCommand(0x81); /* Set Contrast Control */
    WriteCommand(0x20); /* 0 ~ 255 */
```

```
    // Re-map
    WriteCommand(0xA0); /* [A0]:column address 0 is map
    to SEG0 , [A1]: columnaddress 131 is map to SEG0*/
    // Entire Display ON/OFF
    WriteCommand(0xA4); /* A4=ON */
    // Normal or Inverse Display
    WriteCommand(0XA6); /* Normal Display*/
    // Multiplex Ratio
    WriteCommand(0xA8); /* Set Multiplex Ratio */
    WriteCommand(0x3f); /* Set to 36 Mux*/
    // Set DC-DC
    WriteCommand(0xAD); /* Set DC-DC */
    WriteCommand(0x8A); /* 8B=ON, 8A=Off */
    // Display ON/OFF
    WriteCommand(0xAE); /* AF=ON , AE=OFF*/
    // Display Offset
    WriteCommand(0xD3); /* Set Display Offset */
    WriteCommand(0x00); /* No offset */
    // Display Clock Divide
    WriteCommand(0xD5); /* Set Clock Divide */
    WriteCommand(0x20); /* Set to 80Hz */
    // Area Color Mode
    WriteCommand(0xD8); /* Set Area Color On or Off*/
    WriteCommand(0x00); /* Mono Mode */
    // COM Pins Hardware Configuration
    WriteCommand(0xDA); /* Set Pins HardwareConfiguration */
    WriteCommand(0x12);
    // VCOMH
    WriteCommand(0xDB); /* Set VCOMH */
    WriteCommand(0x00);
    // VP
    WriteCommand(0xD9); /* Set VP*/
    WriteCommand(0x22); /* P1=2 , P2=2 */

    WriteCommand(0xc0);//配置成标准应用

    ini_dis();
}
/*************************实现函数*********************************************
file end here
*******************************************************************************/
```

再说明一下小小的应用：

比如要在屏幕的第一行第一个位置显示字库中的“年”字，则用一条语句：
dish(0,0,year);//说明，第一个参数为 X 地址：第一个位置，第二为 Y 地址，第一行，nian 为字模中的一个数组名。

一个参考例程如下：

```
void main()
{
    ini_oled();
    WriteCommand(0xAD); /*  设置  DC-DC */
    WriteCommand(0x8B); /* 8B=ON, 8A=Off */

    disc(0,0,w);
    disc(1,1,h);
    dish(2,2,month);
    dish(3,4,day);
    disn(6,0,0);
    disn(7,0,1);
    disn(8,0,2);
    disn(9,0,3);
    WriteCommand(0xaf);
    while(1);

}
```

在我们的评估板(2051 系统板)上运行后看下结果，就会知道这些文件的含义了。

谢谢你对红荷电子的支持!

附注：当你使用 89S51 来接口此模块时，要注意电压的转换问题，S51 不能很稳定的工作在 3.3V，而与此屏相连接时，要用一片两片 573 做隔离(8 位数据口和 5 个控制接口)。只要给 573 的电源为 3.3V，就可以安全的使用此屏。