

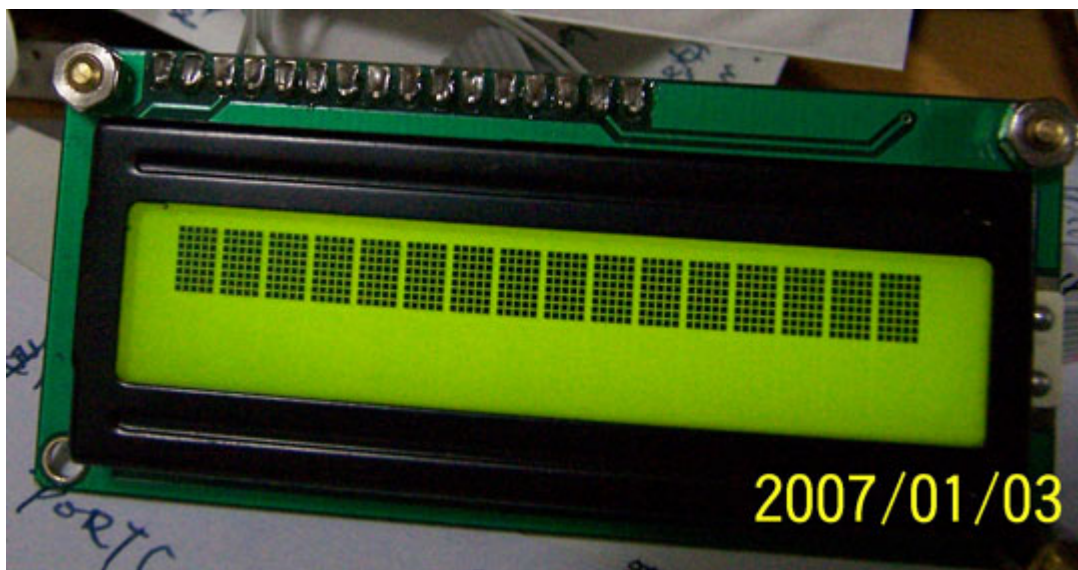
1602 液晶几种典型显示状态及解决方案

1602 应该说是很简单的一种器件，但是许多人初次使用的时候可能总是出问题，当然可能也有一次就试验成功的，但是像我这样的在 AVR ATmega16 上调 1602 居然花了近 3 天的时间，最后调到快崩溃的地步，对自己的智商和水平都产生了怀疑，最后解决问题却是充满了戏剧性。解决问题期间在 OURAVR 上查找了许多帖子，最后才在马超老师的一句话里发现了答案呵呵，所以我把我调 1602 液晶时遇到的情况描述一遍供大家参考，算是给 AVRFans 积攒一点资料。

硬件资料：AVR ATmega16，工作在外部晶振 11.0592MHz，MS1602C 字符液晶

状态 1

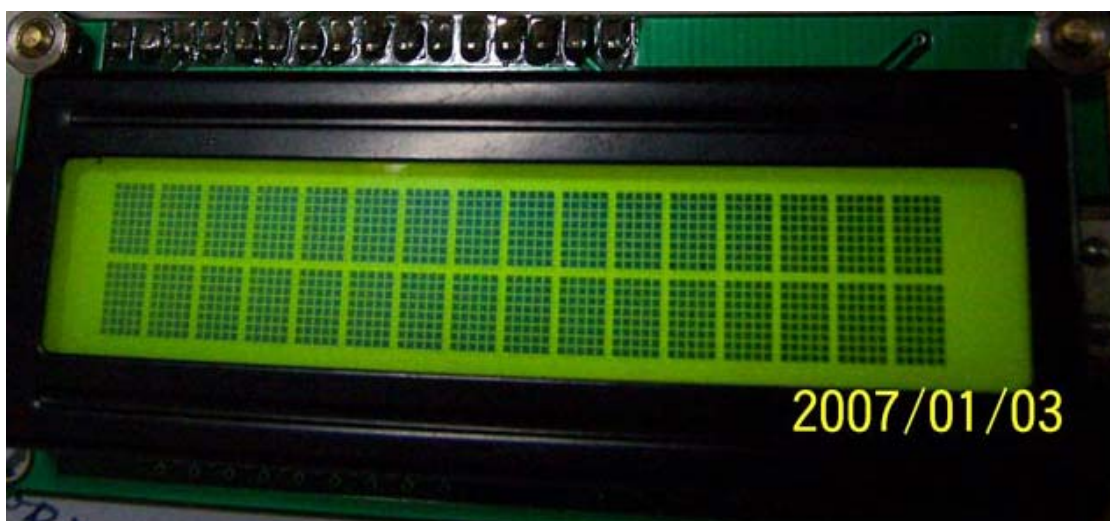
上电之后，1602 的第一行全黑，即 16 个 5x7 黑块



产生原因：液晶根本就没有进行初始化操作，需要检查连线或者程序

状态 2

屏上显示两排灰格



产生原因：程序中对液晶初始化不正常，应是部分初始化指令没有正常接收，建议按照标准初始化步骤调整程序，或者调整指令之间的延时（加大一些试试）

状态 3



此种不用解释了呵呵，调通了已经

1602 液晶大概就这三种状态了，我调的时候很费劲，8 位数据线方式接在 PC 口怎么都不出字符，怎么调延时都没用，总是显示两行灰格，最后看见马老师的提示，**PC 口有 4 个线默认是给 JTAG 用的，默认状态不做 IO 口**，于是重新将数据线接在 PB 口修改程序重新烧写后就好使~

另外，我还发现，**1602 数据线接在 PB 口时**，有时候上电之后显示工作正常，但有些时候根本就不工作，怀疑是 **ISP 下载线**有影响，拔去下载线，恢复正常。

下面附上我的初始化程序,参考了 OURAVR 上的一个帖子，具体是哪个我忘了不好意思，有所调整，按照太阳人那个数据手册里面推荐步骤进行的：

/*

本程序实现功能：在 1602 液晶屏上显示预期字符

硬件平台：AVR ATmega16,外部晶振 11.0592MHz

2007.01.04 avrnewbie@163.com

*/

```
#define LCDPORT PORTB           //数据总线在 PB 口
```

```
#define LCDDDR DDRB
```

```
#define LCDPIN PINB
```

```
#define En_H PORTA |= _BV(PA2)   //sbi(PORTA,2)           //En 接 Pa2
```

```
#define En_L PORTA &= ~_BV(PA2) //cbi(PORTA,2)
```

```
#define RW_R PORTA |= _BV(PA0)   //sbi(PORTA,0)           //RW 接 Pa0
```

```
#define RW_W PORTA &= ~_BV(PA0)  //cbi(PORTA,0)
```

```
#define RS_H PORTA |= _BV(PA3)   //sbi(PORTA,3)           //RS 接 Pa3
```

```
#define RS_L PORTA &= ~_BV(PA3) //cbi(PORTA,3)
```

```
#define DelayE _delay_us(2)
```

```
//#define Clear_Screen Write_Command(0x01)
```

```
//延时程序
```

```
void Delay(uint16_t time)
```

```
{
    while(time>0)
    {
        time--;
    }
}

//产生一个使能脉冲
void En_Toggle(void)
{
    En_H;      //拉高使能位
    DelaytE;    //保持高电平一定时间
    En_L;      //拉低使能位，产生一个下降沿
    DelaytE;    //保持低电平一定时间
}

//不断检测 LCD 忙标志 BF，直到其为 0，表示可以执行下一条指令
void Wait_Until_Ready(void)
{
    LCDDDR = 0x00;    //单片机设为输入，用以读取 LCD 忙标志

    RS_L;             //所读为状态位
    RW_R;             //设为读状态

    DelaytE;
    En_H;
    DelaytE;
    loop_until_bit_is_clear(LCDPIN,7); //不断循环，直至 BF=0
    En_L;
}

void Write_Command(uint8_t Command,uint8_t busyFlag)
{
    uint8_t busy;
    busy = busyFlag;

    if (busy == 1)      //busy=1 表示系统希望检测忙信号
        Wait_Until_Ready(); //等待指令执行完毕

    LCDDDR = 0xff;
    RS_L;             //写入的是命令字
    RW_W;             //置为写状态
    LCDPORT = Command;
    _delay_us(25);
}
```

```
    En_Toggle();           //产生使能脉冲，使之在下降沿开始执行指令

}

void Write_Data(uint8_t data)
{
    Wait_Until_Ready();    //等待指令执行完毕
    RS_H;                 //写入的是数据
    RW_W;                 //置为写状态
    LCDDDR = 0xff;
    LCDPORT = data;
    En_Toggle();           //产生使能脉冲，使之在下降沿开始执行指令
}

void Initialize_LCD(void)
{
    //主程序中在进入初始化程序前已经延时 400ms
    LCDDDR = 0xff;        //LCD 控制线在 PD 口，故将其设为输出
    //此处可以根据实际情况调整，或者在主程序中设置

    DDRA = 0xff;          //E\RS\RW 等接在 PA 口
    _delay_ms(15);
    Write_Command(0x38,0); //设为 8 位接口模式，显示 2 行字符,busy=0 不检测忙信号
    _delay_ms(5);
    Write_Command(0x38,0);
    _delay_ms(5);
    Write_Command(0x38,0);
    Write_Command(0x38,1);
    Write_Command(0x08,1); //显示关闭
    Write_Command(0x01,1); //显示清屏
    Write_Command(0x06,1); //写入新数据后光标右移
    Write_Command(0x0f,1); //显示功能开，显示光标
}
```

AVRnewbie@163.com

我的 BLOG，欢迎大家访问 <http://avrnewbie.21ic.org/>