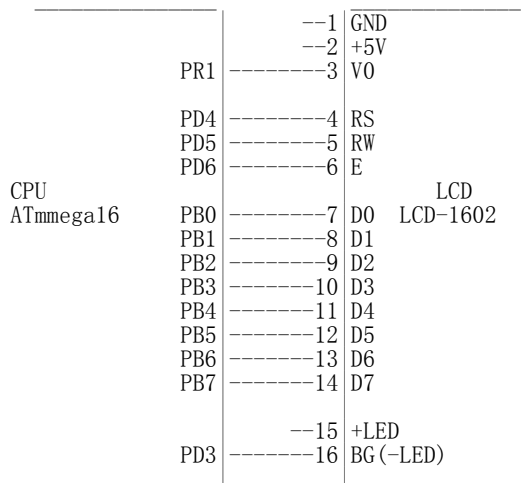


```
/*=====
编程： 失恋过度      QQ:175264198      EMIL: jcxysx@126.com
```

```
代码功能： LCD-1602显示ADC电压值
```

```
编译软件 (ICCAVR_6.31)      CPU: MEGAL16      内部晶振8M
```

```
=====
接线图如下：
```



```
=====*/
#include<ioM16V.h> //包含一些定义
#include<macros.h>
//LCD-1602的IO定义
#define BG_H      asm("sbi 0x12,3")    //BG设为高电平
#define BG_L      asm("cbi 0x12,3")    //BG设为低电平
#define RS_H      asm("sbi 0x12,4")    //RS设为高电平
#define RS_L      asm("cbi 0x12,4")    //RS设为低电平
#define RW_H      asm("sbi 0x12,5")    //RW设为高电平
#define RW_L      asm("cbi 0x12,5")    //RW设为低电平
#define E_H       asm("sbi 0x12,6")    //E 设为高电平
#define E_L       asm("cbi 0x12,6")    //E 设为低电平
//ADC的一些定义
#define Vref 2560                        //定义ADC检测基准电压值2.56V
unsigned char ADC_MUX=0,ADC_OK;
unsigned char lcd_d[]={'0','1','2','3','4','5','6','7','8','9'};
unsigned int  ADC_VA=0,ADC_VAtemp=0;
unsigned int  x,y,z,n;                  //用于显示的位换算变量
//=====
//微秒级延时程序
void delay_us(int time)
{
    do { time--; }
    while (time > 1);
}
//=====
//毫秒级延时程序
void delay_ms(unsigned int time)
{
    while(time != 0)
    { delay_us(1000);time--;}
}
//=====
//读取lcd是否内部操作(忙碌)状态
char Lcd_Busy()
{
    char r;
    DDRB = 0x00;                        //端口B设为输入方式
    E_L;RS_L;RW_H;                      //E=0(使能),RS=0(命令),RW=1(读)
    delay_us(5);                        //液晶延时子程序
    E_H;
    delay_us(5);                        //液晶延时子程序
    r = PINB & 0x80;                    //读取lcd_data第八位
    E_L;
    DDRB=0xff;                          //端口B设为输出方式
    return r;                          //读取结果返回
}
```

```

}
//=====
//向Lcd发送命令程序
void Lcd_Command(unsigned char Command)
{
    while(Lcd_Busy()); //判断lcd是否内部操作状态
    E_L;RS_L;RW_L; //E=0(使能),RS=0(命令),RW=0(写)
    delay_us(5); //液晶延时子程序
    E_H;
    PORTB = Command; //向Lcd发送命令
    delay_us(5); //液晶延时子程序
    E_L;
}
//=====
//向lcd写入一个字符程序
void Lcd_Write(unsigned char Data)
{
    while(Lcd_Busy()); //判断lcd是否内部操作状态
    E_L;RS_H;RW_L; //E=0(使能),RS=1(数据),RW=0(写)
    delay_us(5); //液晶延时子程序
    E_H;
    PORTB = Data; //向lcd写入一个字符
    delay_us(5); //液晶延时子程序
    E_L;
}
/*=====
LCD第1行显示地址1~16(0x80~0x )
LCD第2行显示地址1~16(0xc0~0x )
=====*/
//初始化LCD 8位接口,2行x16字符的工作方式
void Lcd_Init()
{
    DDRB = 0xff; //端口B设为输出方式
    DDRD = 0xff; //端口D设为输出方式
    Lcd_Command(0x38); //4次写命令进入控制模式
    Lcd_Command(0x38); //
    Lcd_Command(0x38); //
    Lcd_Command(0x38); //
    Lcd_Command(0x06); //令光标,0x0c=不显示,0x0d=显示闪动
    Lcd_Command(0x0c); //令显示器off
    Lcd_Command(0x01); //清除显示器
}
//=====
//写ASCII字符串程序
void asc_tran(unsigned char *asc)
{
    while((*asc) != 0) //判断字符串是否结束
    {
        Lcd_Write(*asc); //向lcd写入字符串
        asc++; //移下一个字符
    }
}
//=====
//IO初始化
void IO_init(void)
{
    DDRA=0X00;
    PORTA=0XFF;
    DDRB=0XFF;
    DDRD=0XFF;
    PORTD|=(1<<PD2); //关闭蜂鸣器IO
    BG_H; //背光开
}
//=====
//AD模块初始化
void adc_init(void)
{
    CLI();
    ADCSR = 0x00; //disable adc
    ADMUX = 0xc0; //select adc input 0 选择ADC1
    ACSR = (1<<ACD);
    ADCSRA = (1<<ADEN)|(1<<ADSC)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1);
    //AD使能、AD中断使能、开始转换、64分频
    MCUCR = 0x00;
    GICR = 0x00;

```

```

TIMSK = 0x00;
SEI();
}
//=====
//ADC转换中断函数
#pragma interrupt_handler adc_isr:15
void adc_isr(void)
{
    ADC_VAtemp=ADC&0X3FF;    //中断内读取IO的数据
    ADC_OK=1;                //禁止中断标志位
}
//=====
//返回转换后计算的值
unsigned int ADCtoBCD(unsigned int temp)
{
    return temp=(unsigned int)(((unsigned long)((unsigned long)temp*Vref))/1024);
}
//=====
//显示电压函数
void adc_disy(void)
{
    ADC_VA=1.94*ADCtoBCD(ADC_VAtemp); //传递计算结果到显示暂存变量
    if(ADC_MUX==0)
    {
        x=ADC_VA/1000;                //电压值分割为显示的4个位字符
        y=ADC_VA%1000/100;
        z=ADC_VA%1000%100/10;
        n=ADC_VA%1000%100%10;
        Lcd_Command(0x80);            //设置显示位址
        Lcd_Write(lcd_d[x]);          //向lcd写入字符串
        Lcd_Write(lcd_d[y]);
        Lcd_Write(lcd_d[z]);
        Lcd_Write(lcd_d[n]);
        asc_tran("mV");               //显示电压值单位mV
    }
}
//=====
//测试LCD主程序
void main()
{
    IO_init();        //初始化IO
    adc_init();        //初始化ADC
    Lcd_Init();        //初始化LCD
    while(1)
    {
        adc_disy();    //显示电压函数
        //if(ADC_MUX<4) ADC_MUX++; //循环读取PA0-PA3通道的电压值
        //else {ADC_MUX=0;}

        ADMUX = 0xc0|(ADC_MUX&0X0F); //读取PA0的电压值
        ADCSRA|=(1<<ADIE);          //读取高位
        ADCSRA|=(1<<ADSC);          //读取低位
        ADC_OK=0;                    //开中断标志位
        while(ADC_OK==0);            //等待中断标志位为1
        ADCSRA&=~(1<<ADIE);         //使能AD转换
        delay_us(20);               //延时可改变转换频率
    }
}

```