

/**

This program was produced by the
CodeWizardAVR V2.03.4 Standard
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 7/17/2017
Author :
Company :
Comments:

Chip type : ATmega16
Program type : Application
Clock frequency : 12.000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

*/

```
#include <mega16.h>
#include <stdio.h>
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

#include <delay.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
char buka=0,tutup=0;

int i;
```

```

char buff[33];

void buka_jendela(void)
{
if(buka==1)
{
for(i=0;i<200;i++)
{
PORTD.0=1;
delay_us(1200);
PORTD.0=0;
delay_us(18900);
}
buka=0;
tutup=1;
}
}

void tutup_jendela(void)
{
if(tutup==1)
{
for(i=0;i<200;i++)
{
PORTD.0=1;
delay_us(600);
PORTD.0=0;
delay_us(18500);
}
tutup=0;
buka=1;
}
}
// Declare your global variables here

void main(void)
{
// Declare your local variables here
unsigned int dataadc;
float SUHU;
char mhs=0;
char proses1=0;
char proses2=0;
char proses3=0;
int i;
char masuk=0,keluar=0;
char mhs_A=0,mhs_B=0;

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

```

```

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=In Func0=In
// State7=T State6=T State5=T State4=0 State3=0 State2=0 State1=P State0=P
PORTB=0x03;
DDRB=0x1C;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out Func0=Out
// State7=P State6=P State5=P State4=T State3=T State2=0 State1=0 State0=0
PORTD=0xE0;
DDRD=0x07;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;

```

```
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
```

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
// ADC initialization
// ADC Clock frequency: 750.000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
```

```
// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf(" RUANG ELEKTRO ");
```

```
for(i=0;i<200;i++)
{
PORTD.0=1;
delay_us(600);
PORTD.0=0;
delay_us(18500);
}
for(i=0;i<200;i++)
{
PORTD.0=1;
```

```

delay_us(1200);
PORTD.0=0;
delay_us(18500);
}
for(i=0;i<200;i++)
{
PORTD.0=1;
delay_us(600);
PORTD.0=0;
delay_us(18500);
}
PORTD.1=1;
PORTD.2=1;
delay_ms(3000);
PORTD.1=0;
PORTD.2=0;
buka=1;
tutup=1;
while (1)
{
// Place your code here
while(proses1==0)
{
lcd_gotoxy(0,0);
lcd_putsf(" RUANG ELEKTRO ");

if(PINB.0==1){masuk=1;}
while(masuk==1)
{
if(PINB.1==1){mhs_A=1;}

while(mhs_A==1){mhs++;delay_ms(3000);proses1=1;proses2=0;lcd_clear();masuk=0;mhs_A=0;}

}

sprintf(buff,"Mhs: %d",mhs);
lcd_gotoxy(0,1);
lcd_puts(buff);
//if(PINB.0==1){mhs++;delay_ms(1000);proses1=1;proses2=0;lcd_clear();}
PORTD.2=0; //kipas off
PORTD.1=0; //lampu off
}

while(proses2==0)
{
for(i=0;i<10;i++)
{
dataadc=read_adc(3);
}
SUHU=(((float)dataadc*500)/1024);
sprintf(buff,"SUHU: %0.2fC ",SUHU);

```

```

lcd_gotoxy(0,0);
lcd_puts(buff);

if(SUHU>=25){PORTD.2=1;tutup_jendela();}
if(SUHU<=20){PORTD.2=0;buka_jendela();}

if(PINB.1==1){keluar=1;}
while(keluar==1)
{
if(PINB.0==1){mhs_B=1;}
while(mhs_B==1){mhs++;delay_ms(3000);keluar=0;mhs_B=0;}
}

if(PINB.0==1){masuk=1;}
while(masuk==1)
{
if(PINB.1==1){mhs_A=1;}
while(mhs_A==1){mhs--;delay_ms(3000);masuk=0;mhs_A=0;}
}
sprintf(buff,"Mhs : %d ",mhs);
lcd_gotoxy(0,1);
lcd_puts(buff);

if(PIND.5==1){PORTD.1=1;} //lampu hidup
if(PIND.5==0){PORTD.1=0;} //lampu mati
if(mhs==0){proses1=0;lcd_clear();proses2=1;}
delay_ms(500);
}
};
}

```

