LAMPIRAN

```
/*****************************************************
/NUNUNG KRISNA APRIANTO
/*****************************************************
This program was produced by the
CodeWizardAVR V2.03.4 Standard
Automatic Program Generator
© Copyright 1998-2008 PavelHaiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 2/12/2016
Author  :
Company :
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****************************************************/

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

#include <delay.h>
unsigned char temp[6];
int data;
charsuhu,i=0;
charbuf[33];
charbef[33];
floatsuhu_celcius;

#define ADC_VREF_TYPE 0x40

// Read the 8 most significant bits
// of the AD conversion result
unsigned char 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;
}

// Declare your global variables here

void main(void)
{
// Declare your local variables here
unsigned char dataadc;
float SUHU;
unsignedintsuhu;
// 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=In Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=T State1=P State0=P
PORTB=0x03;
DDRB=0x08;

// 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=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=T State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
```

```
DDRD=0x7F;

PORTB=(1<<PORTB.3);
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 11059.200 kHz
// Mode: Fast PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x69;
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: 691.200 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);
PORTB.3=1;
lcd_gotoxy(0,0);
lcd_putsf(" SELAMAT DATANG ");
delay_ms(100);
lcd_clear();
while (1)
      {
      // Place your code here
suhu=read_adc(0);
suhu=read_adc(0);
suhu=read_adc(0);
      SUHU=(float)suhu*500/1024;

lcd_gotoxy(0,0);
lcd_putsf("SUHU:");
sprintf(buf,"%.2f C",SUHU);
lcd_gotoxy(6,0);
lcd_puts(buf);

if((SUHU>=25)&&(OCR0>=10)){PORTD.0=1;}
if((SUHU>=25)&&(OCR0<=10)){PORTD.0=1;}
if((SUHU<=25)&&(OCR0<=10)){PORTD.0=0;}
```

```
if((SUHU<=25)&&(OCR0>=10)){PORTD.0=0;}

if((SUHU>=26)&&(OCR0>=50)){PORTD.1=1;}
if((SUHU>=26)&&(OCR0<=50)){PORTD.1=1;}
if((SUHU<=26)&&(OCR0>=50)){PORTD.1=0;}
if((SUHU<=26)&&(OCR0<=50)){PORTD.1=0;}

if((SUHU>=27)&&(OCR0>=100)){PORTD.2=1;}
if((SUHU>=27)&&(OCR0<=100)){PORTD.2=1;}
if((SUHU<=27)&&(OCR0<=100)){PORTD.2=0;}
if((SUHU<=27)&&(OCR0>=100)){PORTD.2=0;}


lcd_gotoxy(0,1);
lcd_putsf("jml orang:");

if(PINB.1==0)
    {
if(OCR0<101)
    OCR0++;
lcd_gotoxy(11,1);lcd_putsf("    ");
    }
if(PINB.0==0)
    {
if(OCR0>0)
    OCR0--;
lcd_gotoxy(11,1);lcd_putsf("    ");
    }

itoa(OCR0,temp);
lcd_gotoxy(11,1);
lcd_puts(temp);
delay_ms(30);
    };
}
```