LAMPIRAN

```
/*****************************************************
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    : 7/16/2017
Author  :
Company :
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 16.000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
```

```
#include <io.h>

// I2C Bus functions
#asm
   .equ __i2c_port=0x15 ;PORTC
   .equ __sda_bit=1
   .equ __scl_bit=0
#endasm
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>

unsigned char j,m,s;
unsigned char h,b,t;
unsigned char buffer[16];

char buf[33];
char text[16];
#define HX711_SCK PORTB.7
#define HX711_DT PINB.6
#define HIGH 1
#define LOW 0
```

```
char a=1;
eeprom char alarm_jam1,alarm_menit1,alarm_detik1;
char data;
char nilai[6];

int i,k;
char muncul_1=0, muncul_2=0;
char timer1,timer2;

#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;
}

long HX711_Buffer=0;
long Weight_Maopi=0, Weight_Shiwu=0;
float Weight=0,berat=0;
unsigned long HX711_Read(void)
```

```
{
unsigned long cout;
unsigned char i;
bool flag =0;
HX711_DT=HIGH;
delay_us(1);
HX711_SCK=LOW;
delay_us(1);

cout=0;
while(HX711_DT);
for(i=0;i<24;i++)
{
HX711_SCK=HIGH;
delay_us(1);
cout=cout<<1;
HX711_SCK=LOW;
delay_us(1);
if(HX711_DT) cout++;
}
HX711_SCK=HIGH;
cout ^=0x800000;
delay_us(1);
HX711_SCK=LOW;
delay_us(1);
return (cout);
}

void Get_Maopi()
```

```
{
HX711_Buffer=HX711_Read();
Weight_Maopi=HX711_Buffer/100;
}

unsigned int Get_Weight()
{
HX711_Buffer=HX711_Read();
HX711_Buffer=HX711_Buffer/100;

Weight_Shiwu=HX711_Buffer;
Weight_Shiwu=Weight_Shiwu-Weight_Maopi;
Weight_Shiwu=(unsigned int)((float)Weight_Shiwu/7.35+0.05);
return Weight_Shiwu;

}

void scand_keypad()
{
   char keluar=0;
   char lama=300;
   do
   {
      PORTB.4=0;PORTB.5=1;PORTC.7=1;
      delay_ms(30);
      if (PINB.0 == 0) {data=1;keluar=1;delay_ms(lama);}
      if (PINB.1 == 0) {data=4;keluar=1;delay_ms(lama);}
      if (PINB.2 == 0) {data=7;keluar=1;delay_ms(lama);}
      if (PINB.3 == 0) {delay_ms(300);}
```

```
      PORTB.4=1;PORTB.5=0;PORTC.7=1;
      delay_ms(30);
      if (PINB.0 == 0) {data=2;keluar=1;delay_ms(lama);}
      if (PINB.1 == 0) {data=5;keluar=1;delay_ms(lama);}
      if (PINB.2 == 0) {data=8;keluar=1;delay_ms(lama);}
      if (PINB.3 == 0) {data=0;keluar=1;delay_ms(lama);}

      PORTB.4=1;PORTB.5=1;PORTC.7=0;
      delay_ms(30);
      if (PINB.0 == 0) {data=3;keluar=1;delay_ms(lama);}
      if (PINB.1 == 0) {data=6;keluar=1;delay_ms(lama);}
      if (PINB.2 == 0) {data=9;keluar=1;delay_ms(lama);}
      if (PINB.3 == 0) {delay_ms(300);}
   }
   while (keluar ==0);
}

// Declare your global variables here

void main(void)
{
// Declare your local variables here
char kalibrasi=0;
unsigned int dataadc;
float SUHU;
char buff[33];
// 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=Out Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In
Func0=In

// State7=0 State6=T State5=0 State4=0 State3=P State2=P State1=P State0=P

PORTB=0x0F;

DDRB=0xB0;


// Port C initialization

// Func7=Out Func6=Out Func5=Out Func4=In Func3=In Func2=In Func1=In
Func0=In

// State7=0 State6=0 State5=0 State4=P State3=P State2=P State1=T State0=T

PORTC=0x1C;

DDRC=0xE0;


// Port D 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

PORTD=0x00;

DDRD=0x00;


// 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: 1000.000 kHz
// ADC Voltage Reference: AREF pin
```

```
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;

// I2C Bus initialization
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);
//rtc_set_time(14,53,00);
//rtc_set_date(24,02,16);
// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("ALAT  PENGERING");
lcd_gotoxy(0,1);
lcd_putsf("    KERUPUK     ");
delay_ms(3000);
lcd_clear();
alarm_jam1=0xff;alarm_menit1=0xff;alarm_detik1=0xff;
Weight=Get_Weight();
berat=Weight;
kalibrasi=1;
while (1)
      {
      // Place your code here
      lcd_gotoxy(0,0);
```

```
lcd_putsf("SET JAM      [*]");
lcd_gotoxy(0,1);
lcd_putsf("SET TIMER    [#]");

while(kalibrasi==0)
{
dataadc=read_adc(1);
sprintf(buff,"ADC: %d  ",dataadc);
lcd_gotoxy(0,0);
lcd_puts(buff);
SUHU=(float)dataadc*500/1024;
sprintf(buff,"SUHU: %.2fC  ",SUHU);
lcd_gotoxy(0,1);
lcd_puts(buff);
delay_ms(500);

}

while((alarm_jam1!=0xff)&&(alarm_menit1!=0xff)&&(alarm_detik1!=0xff))
{
lcd_clear();

rtc_get_time(&j,&m,&s);
timer1=(s+5);
if(timer1>=60){timer1=(timer1-60);}
while(muncul_1==10)
{
rtc_get_time(&j,&m,&s);
sprintf(buffer,"JAM: %d:%d:%d  ",j,m,s); // LCD MENAMPILKAN JAM
```

```
    lcd_gotoxy(0,0);
    lcd_puts(buffer);

    sprintf(buffer,"TIM: %d:%d:%d  ",alarm_jam1,alarm_menit1,alarm_detik1);
//SKEDUL 1
    lcd_gotoxy(0,1);
    lcd_puts(buffer);

    muncul_2=10;
    if(timer1==s){muncul_1=0xff;lcd_clear();}
    if((alarm_jam1==j)&&(alarm_menit1==m)&&(alarm_detik1==s))
    {
    muncul_1=0xff;muncul_2=0xff;
    }
    }

    rtc_get_time(&j,&m,&s);
    timer2=(s+5);
    if(timer2>=60){timer2=(timer2-60);}
    while(muncul_2==10)
    {
    rtc_get_time(&j,&m,&s);
    dataadc=read_adc(1);
    SUHU=((float)dataadc*0.488);
    sprintf(buff,"Suhu : %.2fC ",SUHU);
    lcd_gotoxy(0,0);
    lcd_puts(buff);
    if(SUHU>=65){PORTC.6=0;}//PEMANAS OFF
    if(SUHU<=30){PORTC.6=1;}//PEMANAS ON
```

```
Weight=Get_Weight();
//sprintf(buff,"%0.0001f",Weight);
//lcd_gotoxy(0,0);
//lcd_puts(buff);
Weight=(Weight-berat)/0.598;
sprintf(buff,"Berat: %0.0001f g  ",Weight);
lcd_gotoxy(0,1);
lcd_puts(buff);

muncul_1=10;
if(timer2==s){muncul_2=0xff;lcd_clear();}
if((alarm_jam1==j)&&(alarm_menit1==m)&&(alarm_detik1==s))
{
muncul_2=0xff;muncul_1=0xff;
}
}

while((alarm_jam1==j)&&(alarm_menit1==m)&&(alarm_detik1==s))
{
PORTC.6=0;
alarm_jam1=0xff;alarm_menit1=0xff;alarm_detik1=0xff;
lcd_clear();
break;
}
}

PORTB.4=0;PORTB.5=1;PORTC.7=1;
delay_ms(30);
if (PINB.3 == 0) {a=0;lcd_clear();delay_ms(300);}
```

```
while(a==0)
{
 lcd_gotoxy(0,0);
 lcd_putsf("    SET  JAM    ");
 delay_ms(100);
 k=6;
 for (i=0;i<6;i++)
 {
    do {
        scand_keypad();delay_ms(500);
    } while (data>9);
    sprintf(buffer,"%x",data);
    lcd_gotoxy(i,1);
    lcd_puts(buffer);
    nilai[i+1]=data;
 }
 j = (nilai[1]*10)+nilai[2];
 m = (nilai[3]*10)+nilai[4];
 s = (nilai[5]*10)+nilai[6];

 rtc_init(0,0,0);
 rtc_set_time(j,m,s);
 a=1;
 lcd_clear();
}

PORTB.4=1;PORTB.5=1;PORTC.7=0;
delay_ms(30);
if (PINB.3 == 0) {a=0;lcd_clear();delay_ms(300);}
```

```
    while(a==0)
    {
     lcd_gotoxy(0,0);
     lcd_putsf("   SET  TIMER   ");
     delay_ms(100);
     k=6;
     for (i=0;i<6;i++)
     {
        do {
           scand_keypad();delay_ms(500);
        } while (data>9);
        sprintf(buffer,"%x",data);
        lcd_gotoxy(i,1);
        lcd_puts(buffer);
        nilai[i+1]=data;
     }
     j = (nilai[1]*10)+nilai[2];
     m = (nilai[3]*10)+nilai[4];
     s = (nilai[5]*10)+nilai[6];

     alarm_jam1=j;
     alarm_menit1=m;
     alarm_detik1=s;
     muncul_1=10;
     a=1;
     lcd_clear();
    }
    };
}
```