Listing Program

```
/****************************************************
 *
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech
 s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 5/28/2017
Author  : User
Company :
Comments:


Chip type               : ATmega8
Program type            : Application
AVR Core Clock frequency: 12.000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
****************************************************/

#include <mega8.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay.h>
#include <alcd.h>
#define right PINB.3
unsigned char temp[33],buf[33],loop=0;
unsigned char detik,count,detikbpm=0,input,inputs;
int   lm35;
float data1;
eeprom char einput=-2;
eeprom float
 save1[20]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
 ,18,19,20} ;
 eeprom unsigned char
 save2[20]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
 ,18,19,20} ;
```

```
// Timer2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Reinitialize Timer2 value
TCNT2=0x8A;
// Place your code here
loop++;
if (loop>=100)
    {
    detik--;
    detikbpm++;
    loop=0;
}}

#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;
}


void simpan(){
lcd_clear();
input=0;
 while(1){

if(!PINB.4){delay_ms(200);input=input+1;lcd_clear();
if(input>20){lcd_clear();break;}}
 lcd_gotoxy(0,0);
sprintf(buf,"data pasien %d",input);
lcd_puts(buf);
```

```
lcd_gotoxy(10,1);
ftoa(save1[input],1,buf);
lcd_puts(buf);
lcd_putchar(0xdf);
lcd_putsf("C");


lcd_gotoxy(0,1);
sprintf(buf,"%dbpm",save2[input]);
lcd_puts(buf);

  }
      }
void suhu(){
data1=read_adc(0);
lm35=(float)data1*500/1023;
       lcd_gotoxy(0,1);
       lcd_putsf("TEMP:");
        ftoa(lm35,1,buf);
       lcd_puts(buf);
       lcd_putchar(0xdf);
      lcd_putsf("C");
}

void tampil(){
einput=einput+1;
suhu();
while(1){
      if(!PINB.3){lcd_clear();break;}


if(!PINB.4){delay_ms(100);save1[input=einput+1]=lm35
;save2[input=einput+1]=count;



lcd_clear();lcd_gotoxy(0,0);sprintf(buf,"simpan data
%d",input);lcd_puts(buf);
      lcd_gotoxy(10,1);
ftoa(save1[input],1,buf);
lcd_puts(buf);
lcd_putchar(0xdf);
lcd_putsf("C")
lcd_gotoxy(0,1);
sprintf(buf,"%d bpm",save2[input]);
lcd_puts(buf);
      delay_ms(1000);lcd_clear();break;}


      lcd_gotoxy(0,0);
      sprintf(buf,"BPM : %d ",count);
      lcd_puts(buf);

      if(lm35<36.5)
{lcd_gotoxy(11,1);lcd_putsf    ("Hipo");}
       if(lm35>=36.5 &&
lm35<=37.5){lcd_gotoxy(11,1);lcd_putsf    ("Norm");}
       if(lm35>37.5)
{lcd_gotoxy(11,1);lcd_putsf    ("Hipe");}

      if(count<60){
      lcd_gotoxy(11,0);
      lcd_putsf("bradi");
      PORTC.1=1;PORTC.2=0;PORTC.3=0;
      }

      if(count>=60 && count<=100){
      lcd_gotoxy(11,0);
      lcd_putsf("normal");
      PORTC.1=0;PORTC.2=1;PORTC.3=0;


      }

      if(count>100){
      lcd_gotoxy(11,0);
      lcd_putsf("taci");
      PORTC.1=0;PORTC.2=0;PORTC.3=1;
      }

      }}

void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=P State6=P State5=P State4=P State3=T
State2=T State1=T State0=T
PORTB=0xF8;
DDRB=0x07;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTC=0x00;
DDRC=0x3E;

// 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=0x10;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: T0 pin Falling Edge
TCCR0=0x06;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 31.250 kHz
// Mode: Ph. correct PWM top=0x03FF
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA3;
```

```
TCCR1B=0x04;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x02;
OCR1AL=0xFF;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 11.719 kHz
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x07;
TCNT2=0x8A;
OCR2=0x00;

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

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x40;

// USART initialization
// USART disabled
UCSRB=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
ADMUX=ADC_VREF_TYPE & 0xff;
```

```
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C
Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTD Bit 7
// RD - PORTD Bit 1
// EN - PORTD Bit 6
// D4 - PORTD Bit 5
// D5 - PORTD Bit 0
// D6 - PORTD Bit 3
// D7 - PORTD Bit 2
// Characters/line: 16
lcd_init(16);

// Global enable interrupts
#asm("sei")

        lcd_gotoxy(0,0);
lcd_putsf("  Andi Gofer A");
lcd_gotoxy(3,1);
lcd_putsf("20143010037");
delay_ms(2000);
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("   BPM & SUHU");
delay_ms(2000);
lcd_clear();
while (1)
      {

      if(einput==19){einput=255;}

lcd_gotoxy(0,0);
lcd_putsf("RELAX...");
lcd_gotoxy(0,1);
lcd_putsf("PUSH START");
      PORTC.1=0;PORTC.2=0;PORTC.3=0;

       if(!PINB.4){delay_ms(100);simpan();}

if(!PINB.3){TCNT0=0;lcd_clear();detikbpm=0;while(1){
       lcd_gotoxy(14,1);
       sprintf(temp,"%d ",detikbpm);
       lcd_puts(temp);

      count=TCNT0;

      lcd_gotoxy(0,0);
      sprintf(buf,"BPM : %d ",count);
      lcd_puts(buf);

       lcd_gotoxy(0,1);
       lcd_putsf("TEMP: Reading...");
if(detikbpm==30){
count=TCNT0*2;
PORTC.4=1;delay_ms(500);lcd_clear();detikbpm=0;TCNT0
=0;PORTC.4=0;tampil();break;}

                         }   }}
      }
```

FOTO-FOTO MODUL