LAMPIRAN 2. PROGRAM SISTEM

```
/*****************************************************
This program was produced by the
CodeWizardAVR V2.05.0 Advanced
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
Project :
Version :
Date    : 9/28/2014
Author  : www.Eca.ir *** www.Webkade.ir
Company :
Comments:


Chip type           : ATmega32
Program type        : Application
AVR Core Clock frequency: 11.059200 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 512
*****************************************************/

#include <mega32.h>

#include <delay.h>

#include <stdio.h>

#define kalibarsi   0.0048828125

//// Konfigurasi Driver Motor DC --------> Mikro

#define enable_Maz PORTD.4  //enable motor 1
#define dir_Maz    PORTD.3
#define rem_Maz    PORTD.2

// Timer1 overflow interrupt service routine

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
   TCNT1 = 255;
   a++;
   if(a<pwmaz) enable_Maz=0;
   else       enable_Maz=1;

}
```

```
#define ADC_VREF_TYPE 0x40

int dataadc;
float tegangan;
int set_pwm;
char status,nilai_tegangan;
char cek_flak=0;

// 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;
}

// Konfigurasi PORT LCD ----> Mikro

#define LCD_RS_HI PORTC|=(1<<7)
#define LCD_RS_LO PORTC&=~(1<<7)

#define LCD_EN_HI PORTC|=(1<<5)
#define LCD_EN_LO PORTC&=~(1<<5)

#define LCD_D4_HI PORTC|=(1<<4)
#define LCD_D4_LO PORTC&=~(1<<4)
#define LCD_D5_HI PORTC|=(1<<2)
#define LCD_D5_LO PORTC&=~(1<<2)
#define LCD_D6_HI PORTC|=(1<<6)
#define LCD_D6_LO PORTC&=~(1<<6)
#define LCD_D7_HI PORTC|=(1<<0)
#define LCD_D7_LO PORTC&=~(1<<0)

#define line1 0x00
#define line2 0x40


// Standard Input/Output functions
#define _ALTERNATE_PUTCHAR_
//=====================================================================
====
//=============================================
```

```
void LCD_STROBE()
{
   LCD_EN_HI;
   LCD_EN_LO;
}

void lcd_write(unsigned char datalcd)
{
if(datalcd & 0x80) {LCD_D7_HI;} else {LCD_D7_LO;}
if(datalcd & 0x40) {LCD_D6_HI;} else {LCD_D6_LO;}
if(datalcd & 0x20) {LCD_D5_HI;} else {LCD_D5_LO;}
if(datalcd & 0x10) {LCD_D4_HI;} else {LCD_D4_LO;}
LCD_STROBE();
if(datalcd & 0x08) {LCD_D7_HI;} else {LCD_D7_LO;}
if(datalcd & 0x04) {LCD_D6_HI;} else {LCD_D6_LO;}
if(datalcd & 0x02) {LCD_D5_HI;} else {LCD_D5_LO;}
if(datalcd & 0x01) {LCD_D4_HI;} else {LCD_D4_LO;}
LCD_STROBE();
delay_ms(5);
}

void lcd_clrscr()
{
LCD_RS_LO;
lcd_write(0x1);
delay_ms(2);
}


void lcd_putc(unsigned char c)
{
   LCD_RS_HI;
   lcd_write(c);
}


void lcd_goto(unsigned char poslcd)
{
   LCD_RS_LO;
   lcd_write(0x80 + poslcd);
}


void init_lcd()
{
   LCD_RS_LO;
   delay_ms(15);
   LCD_D4_HI;
   LCD_D5_HI;
```

```
   LCD_STROBE();
   delay_ms(5);

   LCD_STROBE();
   delay_us(100);
   LCD_STROBE();
   delay_ms(5);
   LCD_D4_LO;
   LCD_STROBE();
   delay_us(40);
   lcd_write(0x28);
   lcd_write(0x0C);
   lcd_write(0x06);
   lcd_write(0x01);
   delay_ms(10);
}

void lcd_putsf(char flash *progmem_s)
{
      unsigned char c;
      for(;;)
      {
            c = *progmem_s++;
         if(c==0)break;
         lcd_putc(c);
      }
}

void putchar(char c)
{
   lcd_putc(c);

}

//====================================baca eeprom
internal=========================
void eepromwrite(unsigned char Address,int Data)
{
         while(EECR & (1<<EEWE));
         EEAR = Address;
         EEDR = Data;
         EECR |= (1<<EEMWE);
         EECR |= (1<<EEWE);
}
 unsigned int eepromread(unsigned char address)
{
   while(EECR & (1<<EEWE));
         EEAR = address;
         EECR |= (1<<EERE);
```

```
    return EEDR;
}
//============================================================
==============
void tulis_data()
{
     eepromwrite(10,set_pwm);


}


int data_rpm()
{
    set_pwm = eepromread(10);

    return set_pwm;
}


void lcd_int(unsigned int datana,int baris)
{
    unsigned int tempbyte;
    unsigned char buff[5];
    tempbyte=datana;
    buff[4]=(tempbyte/10000)+48;
    buff[3]=((tempbyte%=10000)/1000)+48;
    buff[2]=((tempbyte%=1000)/100)+48;
    buff[1]=((tempbyte%=100)/10)+48;
    buff[0]=(tempbyte%10)+48;

    if (baris == 0){

    if (datana <10){lcd_goto(line1); lcd_putc(buff[0]);}
    else if(datana <100) {lcd_goto(line1); lcd_putc(buff[1]);lcd_putc(buff[0]);}
    else if(datana <1000) {lcd_goto(line1);
lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
    else if(datana <10000) {lcd_goto(line1);
lcd_putc(buff[3]);lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
    else {lcd_goto(line1);
lcd_putc(buff[4]);lcd_putc(buff[3]);lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
    }
    if(baris == 1)
    {

    if (datana <10){lcd_goto(line2); lcd_putc(buff[0]);}
    else if(datana <100) {lcd_goto(line2); lcd_putc(buff[1]);lcd_putc(buff[0]);}
    else if(datana <1000) {lcd_goto(line2);
lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
```

```
   else if(datana <10000) {lcd_goto(line2);
lcd_putc(buff[3]);lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
   else {lcd_goto(line2);
lcd_putc(buff[4]);lcd_putc(buff[3]);lcd_putc(buff[2]);lcd_putc(buff[1]);lcd_putc(buff[0]);}
   }
}

void set_dig( int dig)
{

  lcd_clrscr();
  if(dig == 0)
  {

    lcd_goto(line1);
    printf("***set_sensor****");

    lcd_int(set_pwm,1);

  }

}

void sevdata()
{
  if(button_next == 0)
  {
    while(button_next == 0);
    status++;
    if(status == 2)
    {
      tulis_data();
      status=0;
    }
  }
  if(button_vplus == 0)
  {
    while(button_vplus == 0);

    if(status == 1){
      set_pwm++;
      if(set_pwm > 255) set_pwm=0;
    }
  }
  if(button_vminus == 0)
  {
   while(button_vminus == 0);
    if(status == 1){
      set_pwm--;
```

```
    if(set_pwm < 0) set_pwm=255;
   }
  }
  delay_ms(100);
}
//==========================================Kontrol
Motor==============================
void run(int motor,int direction,int speed)
{
 if(motor==1)
 {
   if(direction==0)
   {
        rem_Maz=1; dir_Maz=0;

   }
   else if(direction==1)
   {

        rem_Maz=1;  dir_Maz=1;
   }
   pwmaz=speed;
 }
}

void cetak_nilai()
{
float tegangan,P,I,D,last_err,tmp_speed,;
  //int data = 3;
  int speed,speed1;
  float error;

  float data3 = 3.68;
  //float cetak;

   dataadc = read_adc(0);
   tegangan = (float)dataadc*5/1023;

  error = data3 - tegangan;

  P = 100*error;
  I  = (0.5*(error + last_err))*0.01;
  D = (0.2*(error - last_err))/0.01;

  error = last_err;
  tmp_speed speed = speed + (int) P + (int) I + (int) D;

  delay_ms(1);
```

```
    if(data3 == tegangan)
    {
        // berhenti
        rem_Maz=0; rem_Maz=0;
        delay_ms (100);

    }
   if(data3 < tegangan)
   {
        //kekiri
        run(1,0,speed1);

   }
   if(data3 > tegangan)
   {
        //kanan
        run(1,1,speed);
   }

    lcd_goto(line1);
    printf("vout= %0.2f volt",tegangan);
    lcd_goto(line2);
    printf("setpoin= %0.2f",data3);
    delay_ms(500);


}
void main(void)
{
// Declare your local variables here
  int i,j;

// 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=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x07;
DDRB=0x00;

// 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=255;
```

```
// 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=0xFC;
//DDRD=0xFC;
PORTD=0x1C;
DDRD=0x1C;

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// 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=0x00;
TCCR1B=0x01;
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: Timer2 Stopped
// Mode: Normal top=0xFF
// 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=0x04;

// 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: 691.200 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

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

// TWI initialization
// TWI disabled
TWCR=0x00;
init_lcd();

i=data_rpm();
j=data_rpm();

#asm("sei")

while (1)
    {

    while(status == 0)
    {
     cetak_nilai();

     if(status==0){if(status==0)sevdata();break;}
```

```
    }
    while(status!=0)
    {
     if(status == 1)
     {
       sevdata();
       set_dig(0);

     }
    }


     // run(1,1,data_rpm());   //kiri
     // run(1,1,data_rpm());   // kanan
     //rem_Maz=1; dir_Maz=1;  enable_Maz=0;
    }
}
```