LAMPIRAN A
PROGRAM PADA PENGONTROL MIKRO AVR
ATMEGA 16

Program Pengontrolan Pendulum Terbalik Menggunakan *on/off*

```
/*****************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP
InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 1/5/2010
Author  : F4CG
Company : F4CG
Comments:


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

#include <mega16.h>
#include <stdio.h>
#include <delay.h>


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

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char
adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE
& 0xff);
// 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)
{char lcd_buffer[33];
float vd,vpot,x1,x2,g,h;
int temp,f,cek;

// Declare your local variables here

// 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=0x00;
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=0x00;

// Port D initialization
// Func7=Out Func6=Out Func5=Out
Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0
State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// 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;
// USART initialization
// Communication Parameters: 8 Data, 1
Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

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

// ADC initialization
// ADC Clock frequency: 86.400 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x87;

// LCD module initialization
lcd_init(16);

while (1)
     {temp=read_adc(0);
     vpot=((float)temp*0.0048);
     lcd_gotoxy(0,0);
     sprintf(lcd_buffer,"Vpot: %3.2f V",vpot);
     lcd_puts(lcd_buffer);
     delay_us(50);
cek++;
 if(cek==500){
 printf("%3.2f",vpot);cek=0;}

       vd=read_adc(1);
       g=((float)vd*0.0048);
       lcd_gotoxy(0,1);
       sprintf(lcd_buffer,"vd: %3.2f V",g);
       lcd_puts(lcd_buffer);
       delay_us(50);

if (g<=0.63)
       {
       PORTD.1=0;
       delay_us(10);
       PORTD.2=0;
       delay_us(10);
       }
if (g>=4.75)
       {
       PORTD.1=0;
       delay_us(10);
       PORTD.2=0;
       delay_us(10);
       }

if (g>=0.64 && g<=4.74)
{
if (vpot<=2.44)
       {
       PORTD.1=1;
       delay_us(50);
```

```
        PORTD.2=0;
        delay_us(50);
        }
if (vpot>=2.47)
        {
        PORTD.2=1;
        delay_us(50);
        PORTD.1=0;
        delay_us(50);
        }

if (vpot>=2.45 && vpot<=2.46)
       {

                  {PORTD.1=0;
                delay_us(50);
                 PORTD.2=0;
                 delay_us(50);}
         }}
       }}
```

Program Pengontrolan Pendulum Terbalik Menggunakan Pengontrol PID

```
/*****************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP
InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 1/8/2010
Author  : F4CG
Company : F4CG
Comments:


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

#include <mega16.h>
#include <delay.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


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

#define ADC_VREF_TYPE 0x40
unsigned char key,key1[6];
unsigned char keypad();
unsigned int k,a,c,d,j,cek;

eeprom float sp,kp,ki,kd;



// Read the AD conversion result
unsigned int read_adc(unsigned char
adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE
& 0xff);
// 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
unsigned int u,V;
unsigned char text[32],lcd_buffer[33];;

void main(void)

{ float pv,er,p,i,d,I,o,rate,PID,vd,vpot,g;
int temp;
// Declare your local variables here

// 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=Out Func1=Out
Func0=Out
// State7=P State6=P State5=P State4=P
State3=1 State2=1 State1=1 State0=1
PORTB=0xFF;
DDRB=0x0F;

// 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=Out Func2=Out Func1=In Func0=In
// State7=T State6=T State5=T State4=T
State3=0 State2=0 State1=T State0=T
PORTD=0x00;
DDRD=0x0C;

// 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: 10.800 kHz
// Mode: Fast PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x6F;
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;

// USART initialization
// Communication Parameters: 8 Data, 1
Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

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

// ADC initialization
// ADC Clock frequency: 86.400 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x87;

// LCD module initialization
lcd_init(16);
u=0;
while (1)
      {


// program tampilan awal:


awal:
lcd_clear();
key=keypad();
lcd_gotoxy(0,0);
lcd_putsf("A=INPUT  B=DATA");
lcd_gotoxy(0,1);
lcd_putsf("C=StArT MoT0r");
delay_ms(100);
if(key=='A'){
goto inputdata;}
else if(key=='B')
{goto lihat;}
 else if(key=='C')
{lcd_clear();goto adc;}
delay_ms(100);
 goto awal;

lihat:
       lcd_clear();
       lcd_gotoxy(0,0);
       sprintf(text,"SP: %3.3f",sp);
```

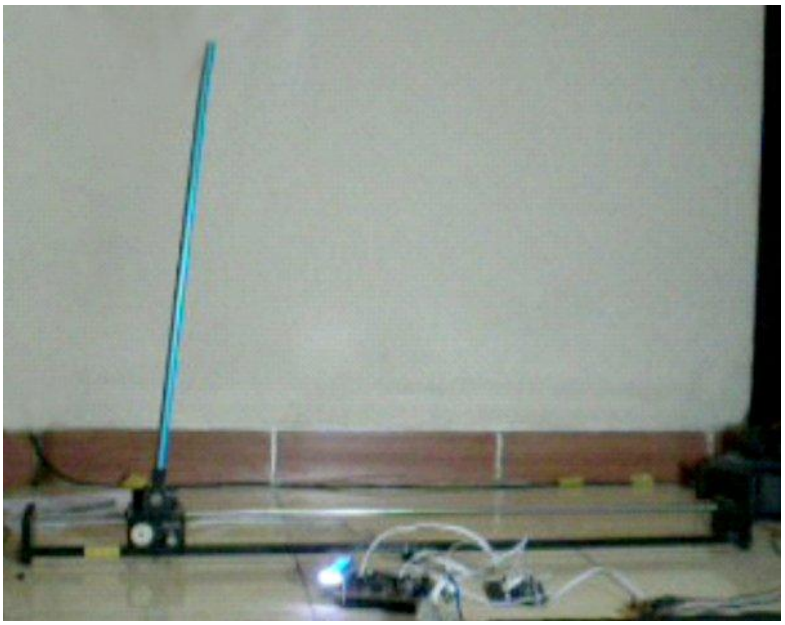
```
        lcd_puts(text);
        lcd_gotoxy(0,1);
          sprintf(text,"KP: %3.3f",kp);
         lcd_puts(text);
         delay_ms(2000);
          lcd_clear();

        lcd_gotoxy(0,0);
          sprintf(text,"Ki: %3.3f",ki);
           lcd_puts(text);
           lcd_gotoxy(0,1);
          sprintf(text,"Kd: %3.3f",kd);
         lcd_puts(text);
         delay_ms(2000);
          goto awal;
//    program
inputdata=========================
==============================
==============


     inputdata:
       key=keypad();
       lcd_clear();
       lcd_gotoxy(0,1);
       lcd_putsf("INPUT NILAI:");
       delay_ms(500);
        c=0;
       for(j=0;j<=5;j++)
       {
       key=keypad();
      key1[c]=key;
       lcd_gotoxy(0,1);
       lcd_clear();
             if(c==0)

{lcd_gotoxy(0,0);sprintf(text,"INPUT 1:");}
             else if(c==1)
                   {lcd_gotoxy(0,0);
lcd_putsf("INPUT 2:");lcd_gotoxy(0,1);
                  sprintf(text,"%c",key1[0]);}
             else if(c==2)
                   {lcd_gotoxy(0,0);
lcd_putsf("INPUT 3:");lcd_gotoxy(0,1);

sprintf(text,"%c%c",key1[0],key1[1]);}
             else if(c==3)
                   {lcd_gotoxy(0,0);
lcd_putsf("INPUT 4:");lcd_gotoxy(0,1);

sprintf(text,"%c%c%c",key1[0],key1[1],key
1[2]);}
             else if(c==4)
                   {lcd_gotoxy(0,0);
lcd_putsf("INPUT 5:");lcd_gotoxy(0,1);

sprintf(text,"%c%c%c%c",key1[0],key1[1],
key1[2],key1[3]);}
             else if(c==5)
                   {lcd_gotoxy(0,0);
lcd_putsf("INPUT 6:");lcd_gotoxy(0,1);

sprintf(text,"%c%c%c%c%c",key1[0],key1[
1],key1[2],key1[3],key1[4]);}
       lcd_puts(text);
       delay_ms(200);
       c++;
       if(key==247)
       {
             j=j-1;
             c=c-1; }  }
 lcd_clear();

sprintf(text,"%c%c%c%c%c%c",key1[0],ke
y1[1],key1[2],key1[3],key1[4],key1[5]);
      lcd_puts(text);
        delay_ms(2000);
ulang:
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("pilih A=SP");
lcd_gotoxy(6,1);
lcd_putsf("B=KP");
lcd_gotoxy(12,0);
lcd_putsf("C=Ki");
lcd_gotoxy(12,1);
lcd_putsf("D=Kd");
delay_ms(100);
 key=keypad();
if(key=='A')
       goto setpoint;
else if(key=='B')
       goto nilai_kp;
 else if(key=='C')
       goto nilai_ki;
else if(key=='D')
       goto nilai_kd;
else
delay_ms(100);
goto ulang;

setpoint:
       sp=atof(key1);
       lcd_clear();
       lcd_gotoxy(0,0);
       sprintf(text,"SP:%3.3f",sp);
       lcd_puts(text);
       delay_ms(2000);
       goto awal;
  nilai_kp:
```

```
        kp=atof(key1);
        lcd_clear();
        lcd_gotoxy(0,0);
          sprintf(text,"KP:%3.3f",kp);
       lcd_puts(text);
        delay_ms(2000);
        goto awal;
    nilai_ki:
        ki=atof(key1);
        lcd_clear();
        lcd_gotoxy(0,0);
          sprintf(text,"Ki:%3.3f",ki);
       lcd_puts(text);
        delay_ms(2000);
        goto awal;
     nilai_kd:
        kd=atof(key1);
        lcd_clear();
        lcd_gotoxy(0,0);
        sprintf(text,"Kd:%3.3f",kd);
       lcd_puts(text);
        delay_ms(2000);
        goto awal;


adc:
 temp=read_adc(0);
 vpot=((float)temp*0.0048);
 lcd_gotoxy(0,0);
 sprintf(lcd_buffer,"Vp:%3.2f
%3.2f",vpot,PID);
 lcd_puts(lcd_buffer);
 delay_us(50);
 cek++;
 if(cek==500){
 printf("%3.2f",vpot);cek=0;}

 vd=read_adc(1);
 g=((float)vd*0.0048);
 lcd_gotoxy(0,1);
 sprintf(lcd_buffer,"vd:%3.2f %3.2f",g,er);
 lcd_puts(lcd_buffer);
 delay_us(50);


//buat PID

pv=vpot;


        if(u==0)
              {o=sp-pv;
               i=0;
               goto hitungPID;}
        else goto hitungPID;

hitungPID:

              er=sp-pv;
           //P kontroller
              p=kp*er;
           // I konroller
               i=i+er;
              I=i*ki;
           // D kontroller
              rate=er-o;
              d=rate*kd;
           // PID
            PID = p+I+d;


              o=sp-pv;
               u++;

if (g<=0.63 || g>=4.75)
        {
        PORTD.2=0;
        delay_us(50);
        PORTD.3=0;
        delay_us(50);
        }

if (g>=0.64 && g<=4.74)
{

if (PID>0)
        {
        PORTD.2=1;
        delay_us(50);
        PORTD.3=0;
        delay_us(50);
             V=0,5*((PID^2)^1/2) + 6;
             OCR2= 28,33* V;
        }
if (PID<0)
        {
        PORTD.2=0;
        delay_us(50);
        PORTD.3=1;
        delay_us(50);
             V=0,5*((PID^2)^1/2) + 6;
             OCR2= 28,33* V;
        }

if (PID==0)
       {PORTD.2=0;
       delay_us(50);
       PORTD.3=0;
       delay_us(50);}

g<=2.10 && g>=1.9)
```

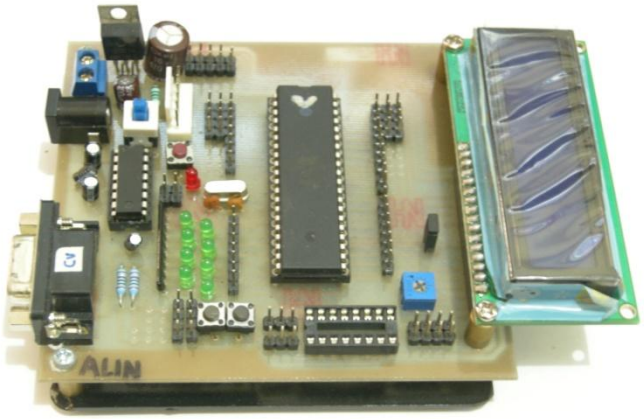
```
g<=4.55 && g>=4.3)
}


     }

 key=keypad();
if(key=='D')goto awal;
  else goto adc;
     };
}


unsigned char keypad()
{
PORTB=0b11111110;
if(PINB.4==0)return('*');
if(PINB.5==0)return('2');
if(PINB.6==0)return('3');
if(PINB.7==0)return('A');
//=======================
PORTB=0b11111101;
if(PINB.4==0)return('1');
if(PINB.5==0)return('5');
if(PINB.6==0)return('6');
if(PINB.7==0)return('B');
//=======================
PORTB=0b11111011;
if(PINB.4==0)return('4');
if(PINB.5==0)return('8');
if(PINB.6==0)return('9');
if(PINB.7==0)return('C');
//=======================
PORTB=0b11110111;
if(PINB.4==0)return('7');
if(PINB.5==0)return('0');
if(PINB.6==0)return('.');
if(PINB.7==0)return('D');
//=======================
 }
```

LAMPIRAN B

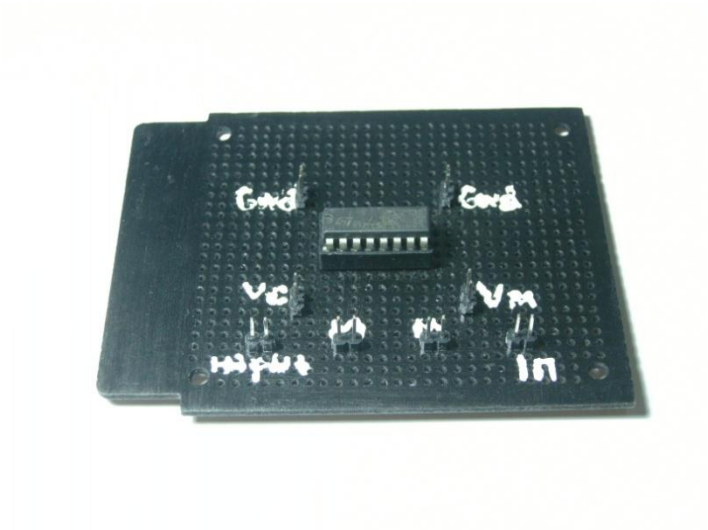
FOTO ALAT



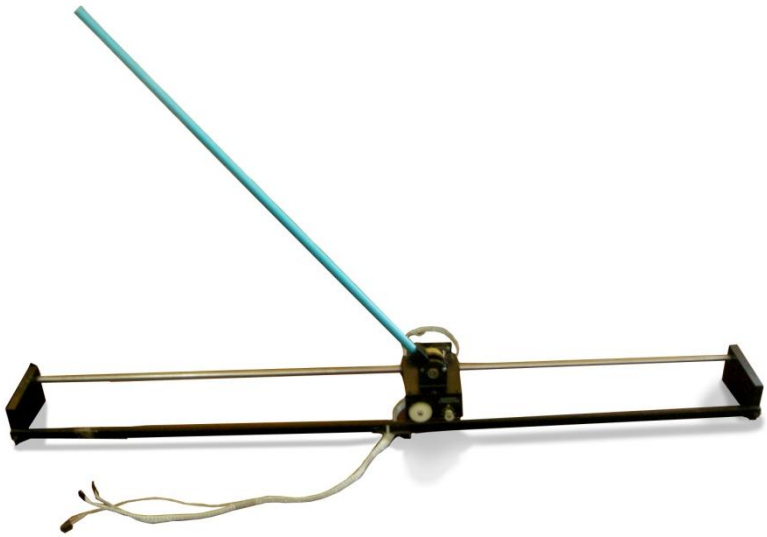
B-1 Foto Sistem Pendulum Terbalik



B-2 Foto Pengontrol Mikro AVR ATMEGA16



B-3 Foto *Motor Driver*



B-4 Foto Pendulum Terbalik

LAMPIRAN C

DATASHEET