

LAMPIRAN - A

Instruksi Mikrokontroler

```
/******
```

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

Project :
Version :
Date : 8/5/2008
Author : Yakub Hartanto
Company : lab
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>
```

```
char var[5],i,konter,konter2,sudut[8],robah_kiri,robah_kanan;  
int temp,rotL,rotR;  
bit a,b;  
#include <hao.h>
```

```
#define RXB8 1  
#define TXB8 0  
#define UPE 2  
#define OVR 3  
#define FE 4  
#define UDRE 5  
#define RXC 7
```

```
#define FRAMING_ERROR (1<<FE)  
#define PARITY_ERROR (1<<UPE)  
#define DATA_OVERRUN (1<<OVR)  
#define DATA_REGISTER_EMPTY (1<<UDRE)  
#define RX_COMPLETE (1<<RXC)
```

```
// USART Receiver buffer  
#define RX_BUFFER_SIZE 8
```

```

char rx_buffer[RX_BUFFER_SIZE];
#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index]=data;
if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
};
};
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif
#endif

```

```

// USART Transmitter buffer
#define TX_BUFFER_SIZE 8
char tx_buffer[TX_BUFFER_SIZE];

#if TX_BUFFER_SIZE<256
unsigned char tx_wr_index,tx_rd_index,tx_counter;
#else
unsigned int tx_wr_index,tx_rd_index,tx_counter;
#endif

// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
if (tx_counter)
{
--tx_counter;
UDR=tx_buffer[tx_rd_index];
if (++tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
};
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
{
tx_buffer[tx_wr_index]=c;
if (++tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
++tx_counter;
}
else
UDR=c;
#asm("sei")
}

#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

```

```

#define ADC_VREF_TYPE 0x60
// 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);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

// Timer 0 output compare interrupt service routine
interrupt [TIM0_COMP] void timer0_comp_isr(void)
{
    // Place your code here
    PORTA.0=1;
    delay(sudut[1]);
    PORTA.0=0;

    PORTA.1=1;
    delay(sudut[2]);
    PORTA.1=0;

    PORTA.2=1;
    delay(sudut[3]);
    PORTA.2=0;

    PORTA.3=1;
    delay(sudut[4]);
    PORTA.3=0;

    PORTA.4=1;
    delay(sudut[5]);
    PORTA.4=0;

    PORTA.5=1;
    delay(sudut[5]);
    PORTA.5=0;

    PORTA.6=1;
    delay(sudut[7]);
    PORTA.6=0;
}

```

```

konter++;
if(konter==50)
{
var[2]=rotL+20;
var[3]=rotR+20;
rotL=0;
rotR=0;
konter=0;
}
var[4]=read_adc(7)+20;
}

```

```

// Timer 1 input capture interrupt service routine
interrupt [TIM1_CAPT] void timer1_capt_isr(void)
{

```

```

// Place your code here

```

```

//=====
=====

```

```

    if(a==0)
    {
        if(PIND.6==1)
        {
            a=1;
            rotL=rotL+1;
        }
    }
    else if(a==1)
    {
        if(PIND.6==0)
        {
            a=0;
            rotL=rotL+1;
        }
    }
}

```

```

    if(b==0)
    {
        if(PIND.7==1)
        {
            b=1;
            rotR=rotR+1;
        }
    }
    else if(b==1)

```

```

    {
        if(PIND.7==0)
        {
            b=0;
            rotR=rotR+1;
        }
    }
}

```

```

// Timer 2 output compare interrupt service routine
interrupt [TIM2_COMP] void timer2_comp_isr(void)

```

```

{
// Place your code here
if(konter2==25)
{
putchar(var[i]);
i++;
konter2=0;
}

```

```

konter2++;
if(i==5)
i=0;
}

```

```

// Declare your global variables here

```

```

void main(void)

```

```

{
// 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=0x7f;
// 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=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=Out Func4=Out Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x30;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
// Mode: CTC top=OCR0
// OC0 output: Disconnected
TCCR0=0x0D;
TCNT0=0x00;
OCR0=0xD7;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 11059.200 kHz
// Mode: Ph. & fr. cor. PWM top=ICR1
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: On
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA0;
TCCR1B=0x11;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x15;
ICR1L=0x9A;
OCR1AH=0x02;
OCR1AL=0x29;
OCR1BH=0x02;
OCR1BL=0x29;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
// Mode: CTC top=OCR2
// OC2 output: Disconnected
ASSR=0x00;
```



```
TCCR2=0x0F;
TCNT2=0x00;
OCR2=0xD7;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0xA2;

// 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=0xD8;
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: 691.200 kHz
// ADC Voltage Reference: AVCC 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;

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

```
a=PIND.6;
b=PIND.7;
rotL=0;
rotR=0;
var[0]='x';
var[1]='x';
```

```
sudut[1]=60;
sudut[2]=90;
sudut[3]=35;
sudut[4]=165;
sudut[5]=165;
sudut[6]=165;
sudut[7]=90;
```

```
while (1)
{
    // Place your code here
```

```
//=====
=====
```

```
    lengan:
    temp=getchar();
    switch(temp)
    {
        case 'a':temp=getchar();sudut[1]=temp;break;
        case 'b':temp=getchar();sudut[2]=temp;break;
        case 'c':temp=getchar();sudut[3]=temp;break;
        case 'd':temp=getchar();sudut[4]=temp;break;
        case 'e':temp=getchar();sudut[5]=temp;sudut[6]=temp;break;
        case 'f':temp=getchar();sudut[7]=temp;break;
        case 'r':goto roda;break;
    }
    goto lengan;
```

```
//=====
=====
```

```
    roda:
    temp=getchar();
    switch(temp)
    {
        case 'w':
            OCR1A=553;OCR1B=553;
            PORTB=0B00000110;
```

```

break;

case 's':
PORTB=0;
break;

case 'a':
OCR1A=553;OCR1B=553;
PORTB=0B00000101;
break;

case 'd':
OCR1A=553;OCR1B=553;
PORTB=0B00001010;
break;

case 'x':
OCR1A=553;OCR1B=553;
PORTB=0B00001001;
break;

case '+':
if(OCR1A==5530)
goto tambah;
OCR1A=OCR1A+553;OCR1B=OCR1B+553;
tambah:
break;

case '-':
if(OCR1A==553)
goto kurang;
OCR1A=OCR1A-553;OCR1B=OCR1B-553;
kurang:
break;

case 'l':
OCR1A=1;OCR1B=1;
PORTB=0;
goto lengan;
break;
}

goto roda;
};
}

```

LAMPIRAN - B

Instruksi Program Visual Basic

```
Dim X, RPM_KIRI, RPM_KANAN, tekanan, mode As Integer
Dim var1, var2, var3, var4, var5 As String
```

```
Private Sub Command1_Click()
Text10.Text = 10
MSComm1.Output = "w"
End Sub
```

```
Private Sub Command10_Click()
End
End Sub
```

```
Private Sub Command2_Click()
Text10.Text = 10
MSComm1.Output = "a"
End Sub
```

```
Private Sub Command3_Click(Index As Integer)
Text10.Text = 10
MSComm1.Output = "x"
End Sub
```

```
Private Sub Command4_Click()
Text10.Text = 10
MSComm1.Output = "d"
End Sub
```

```
Private Sub Command5_Click()
MSComm1.Output = "s"
Text10.Text = 0
End Sub
```

```
Private Sub Command6_Click()
```

```
If mode = 0 Then
MSComm1.Output = "r"
Command6.Caption = "Lengan"
Frame1.Visible = False
Frame2.Visible = False
Frame3.Visible = True
Frame4.Visible = True
mode = 1
ElseIf mode = 1 Then
```

```
MSComm1.Output = "I"  
Command6.Caption = "Roda"  
Frame1.Visible = True  
Frame2.Visible = True  
Frame3.Visible = False  
Frame4.Visible = False  
mode = 0  
End If  
End Sub
```

```
Private Sub Command7_Click()  
MSComm1.Output = "+"  
Text10.Text = Text10 + 10  
If Text10.Text > 100 Then  
Text10.Text = 100  
  
End If
```

```
End Sub
```

```
Private Sub Command8_Click()  
MSComm1.Output = "-"  
bb = Text10.Text  
bb = bb - 10  
Text10.Text = bb
```

```
If Text10.Text < 0 Then  
Text10.Text = 0  
End If  
End Sub
```

```
Private Sub Command9_Click()
```

```
Text1.Text = 80  
Text2.Text = 90  
Text3.Text = 35  
Text4.Text = 165  
Text5.Text = 165  
Text6.Text = 90  
HScroll1.Value = 80  
HScroll2.Value = 90  
HScroll3.Value = 35  
HScroll4.Value = 165  
HScroll5.Value = 165  
HScroll6.Value = 90
```

```
MSComm1.Output = "a" + Chr$(Text1.Text)
MSComm1.Output = "b" + Chr$(Text2.Text)
MSComm1.Output = "c" + Chr$(Text3.Text)
MSComm1.Output = "d" + Chr$(Text4.Text)
MSComm1.Output = "e" + Chr$(Text5.Text)
MSComm1.Output = "f" + Chr$(Text6.Text)
```

End Sub

```
Private Sub Form_Load()
MSComm1.CommPort = 8
MSComm1.Settings = "9600, N, 8 ,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
Text10.Text = 0
Text1.Text = 80
Text2.Text = 90
Text3.Text = 35
Text4.Text = 165
Text5.Text = 165
Text6.Text = 90
```

```
Command6.Caption = "Roda"
Frame1.Visible = True
Frame2.Visible = True
Frame3.Visible = False
Frame4.Visible = False
```

End Sub

```
Private Sub HScroll1_Change()
X = HScroll1.Value
X = X / 5
HScroll1.Value = X * 5
Text1.Text = HScroll1.Value
MSComm1.Output = "a" + Chr$(Text1.Text)
End Sub
```

```
Private Sub HScroll2_Change()
X = HScroll2.Value
X = X / 5
HScroll2.Value = X * 5
Text2.Text = HScroll2.Value
MSComm1.Output = "b" + Chr$(Text2.Text)
End Sub
```

```
Private Sub HScroll3_Change()
```

```
X = HScroll3.Value
X = X / 5
HScroll3.Value = X * 5
Text3.Text = HScroll3.Value
MSComm1.Output = "c" + Chr$(Text3.Text)
End Sub
```

```
Private Sub HScroll4_Change()
X = HScroll4.Value
X = X / 5
HScroll4.Value = X * 5
Text4.Text = HScroll4.Value
MSComm1.Output = "d" + Chr$(Text4.Text)
End Sub
```

```
Private Sub HScroll5_Change()
X = HScroll5.Value
X = X / 5
HScroll5.Value = X * 5
Text5.Text = HScroll5.Value
MSComm1.Output = "e" + Chr$(Text5.Text)
End Sub
```

```
Private Sub HScroll6_Change()
X = HScroll6.Value
X = X / 5
HScroll6.Value = X * 5
Text6.Text = HScroll6.Value
MSComm1.Output = "f" + Chr$(Text6.Text)
End Sub
```

```
Private Sub Timer1_Timer()
Text11.Text = Text12.Text
Text12.Text = Text13.Text
Text13.Text = Text14.Text
Text14.Text = Text15.Text
Text15.Text = MSComm1.Input
```

```
If Text11.Text = "x" And Text12.Text = "x" Then
Text9.Text = ((Asc(Text13.Text) - 20) * 60) / 18
Text8.Text = ((Asc(Text14.Text) - 20) * 60) / 18
```

```
If (Asc(Text15.Text) - 20) > 54 And (Asc(Text15.Text) - 20) <= 72 Then
Text7.Text = 0.1
ElseIf (Asc(Text15.Text) - 20) > 72 And (Asc(Text15.Text) - 20) <= 104 Then
Text7.Text = 0.2
ElseIf (Asc(Text15.Text) - 20) > 104 And (Asc(Text15.Text) - 20) <= 120 Then
Text7.Text = 0.3
```

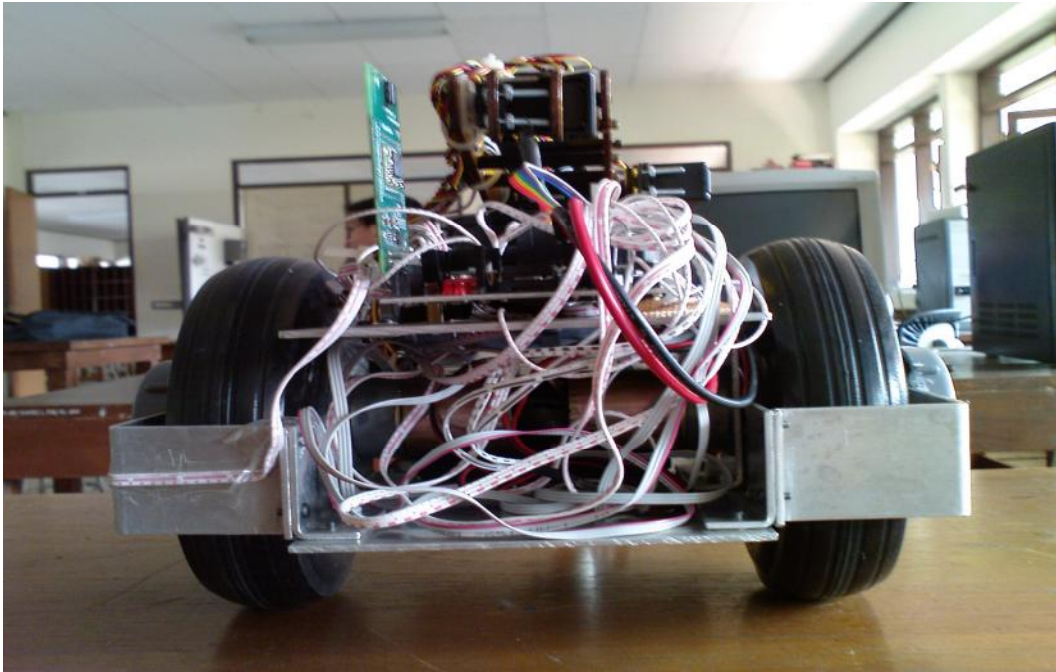


```
ElseIf (Asc(Text15.Text) - 20) > 120 And (Asc(Text15.Text) - 20) <= 125 Then  
Text7.Text = 0.4  
ElseIf (Asc(Text15.Text) - 20) > 125 And (Asc(Text15.Text) - 20) <= 130 Then  
Text7.Text = 0.5  
ElseIf (Asc(Text15.Text) - 20) > 130 And (Asc(Text15.Text) - 20) <= 133 Then  
Text7.Text = 0.6  
ElseIf (Asc(Text15.Text) - 20) > 133 And (Asc(Text15.Text) - 20) <= 140 Then  
Text7.Text = 0.7  
ElseIf (Asc(Text15.Text) - 20) > 140 And (Asc(Text15.Text) - 20) <= 146 Then  
Text7.Text = 0.8  
ElseIf (Asc(Text15.Text) - 20) > 146 And (Asc(Text15.Text) - 20) <= 151 Then  
Text7.Text = 0.9  
ElseIf (Asc(Text15.Text) - 20) > 151 And (Asc(Text15.Text) - 20) <= 155 Then  
Text7.Text = 1  
ElseIf (Asc(Text15.Text) - 20) > 155 And (Asc(Text15.Text) - 20) <= 156 Then  
Text7.Text = 1.1  
ElseIf (Asc(Text15.Text) - 20) > 156 And (Asc(Text15.Text) - 20) <= 157 Then  
Text7.Text = 1.2  
ElseIf (Asc(Text15.Text) - 20) > 157 And (Asc(Text15.Text) - 20) <= 158 Then  
Text7.Text = 1.3  
ElseIf (Asc(Text15.Text) - 20) > 158 And (Asc(Text15.Text) - 20) <= 160 Then  
Text7.Text = 1.4  
ElseIf (Asc(Text15.Text) - 20) > 160 And (Asc(Text15.Text) - 20) <= 162 Then  
Text7.Text = 1.5  
ElseIf (Asc(Text15.Text) - 20) > 162 And (Asc(Text15.Text) - 20) <= 165 Then  
Text7.Text = 1.6  
ElseIf (Asc(Text15.Text) - 20) > 165 And (Asc(Text15.Text) - 20) <= 167 Then  
Text7.Text = 1.7  
ElseIf (Asc(Text15.Text) - 20) > 167 And (Asc(Text15.Text) - 20) <= 170 Then  
Text7.Text = 1.8  
ElseIf (Asc(Text15.Text) - 20) > 170 And (Asc(Text15.Text) - 20) <= 173 Then  
Text7.Text = 1.9  
ElseIf (Asc(Text15.Text) - 20) > 173 Then  
Text7.Text = 2  
End If  
  
End If  
End Sub
```

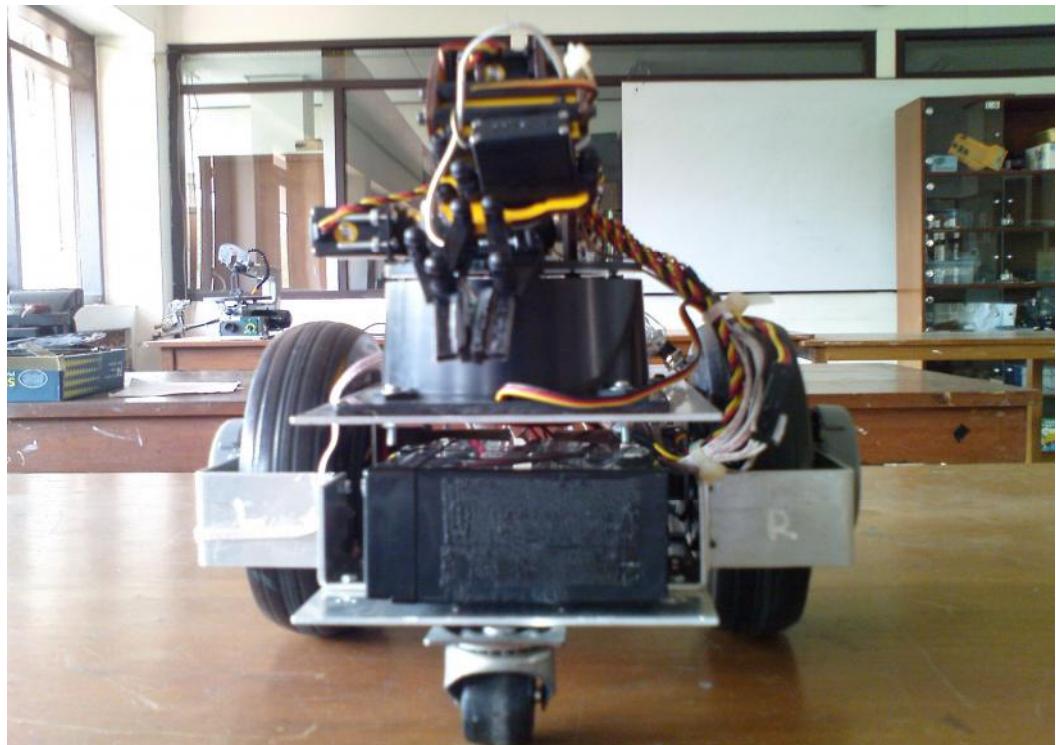
LAMPIRAN - C

Foto Alat

Tampak Depan



Tampak Belakang



Tampak Samping Kiri



Tampak Samping Kanan



Tampak Atas

