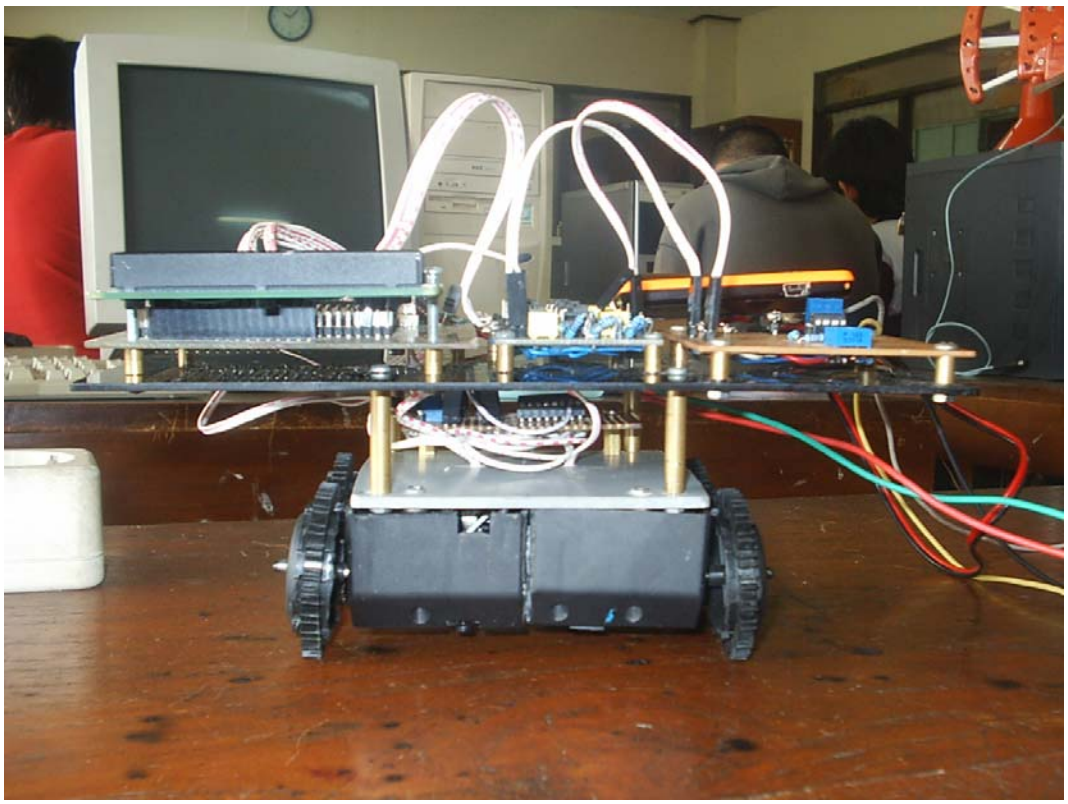
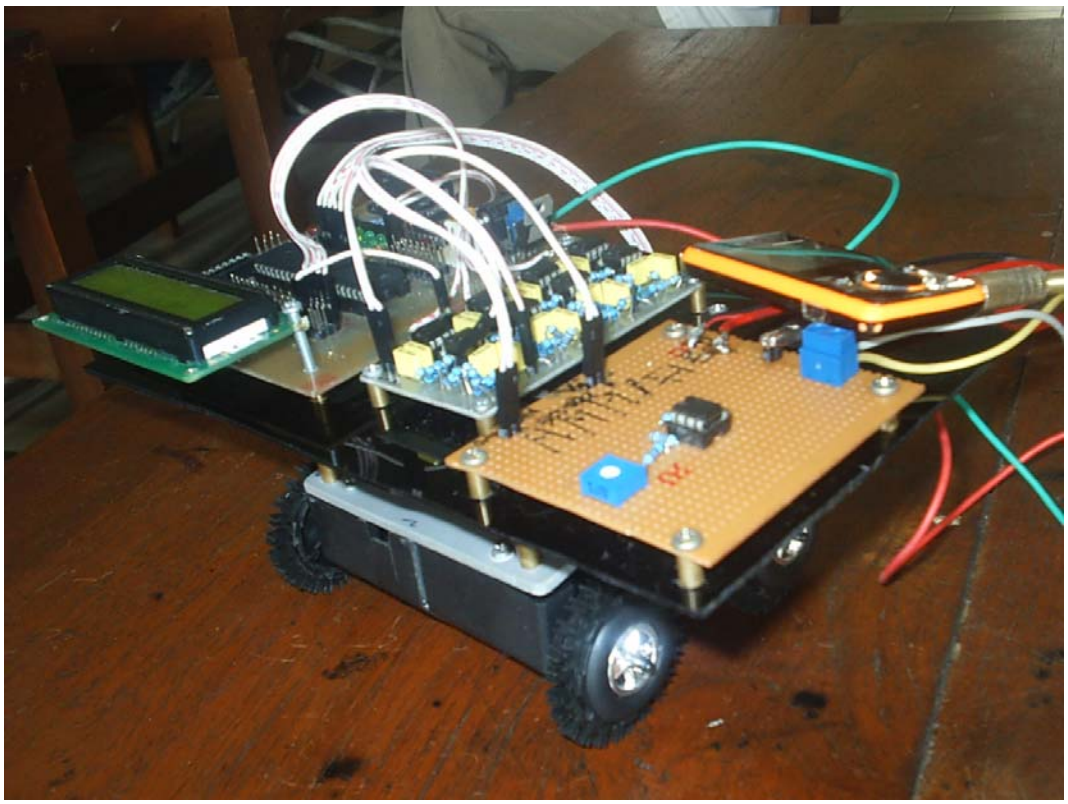
LAMPIRAN A
FOTO ROBOT BERODA

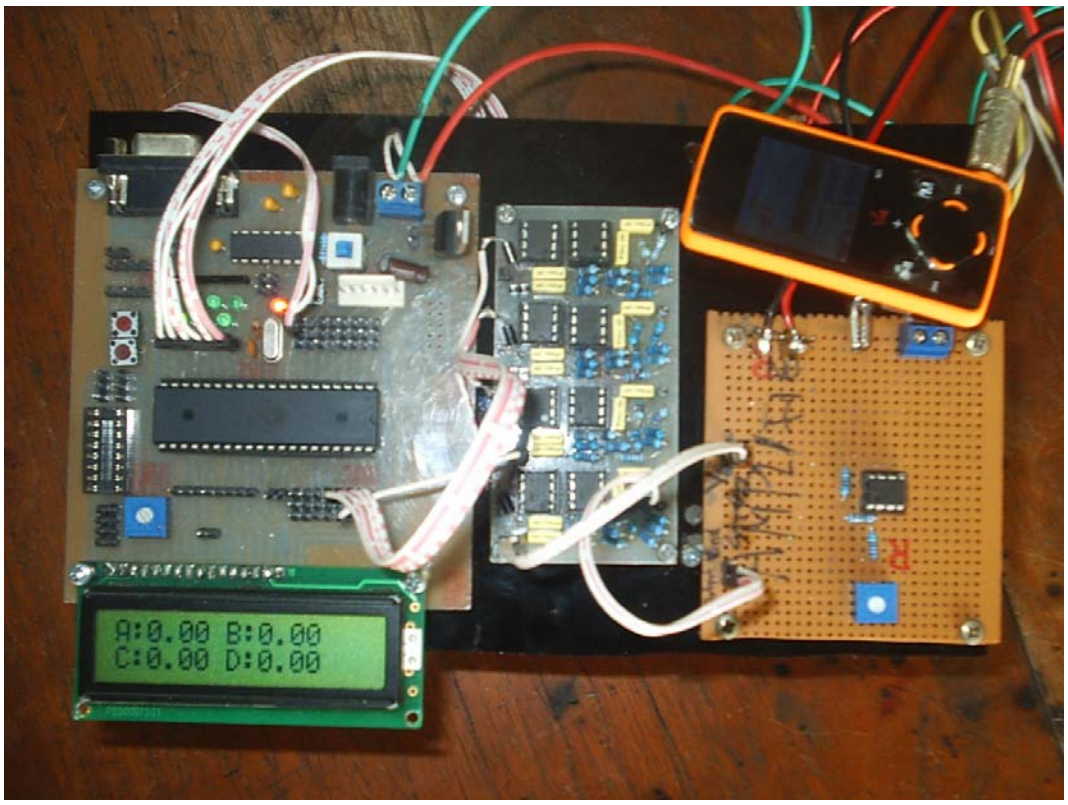
TAMPAK DEPAN



TAMPAK SAMPING DEPAN



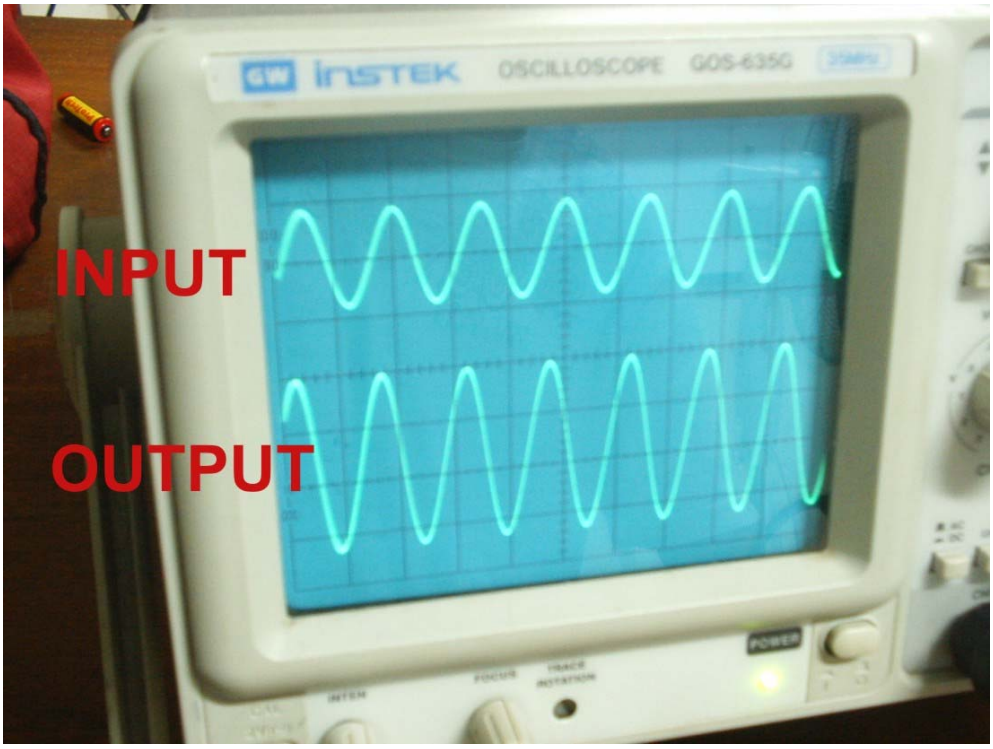
TAMPAK ATAS



LAMPIRAN B

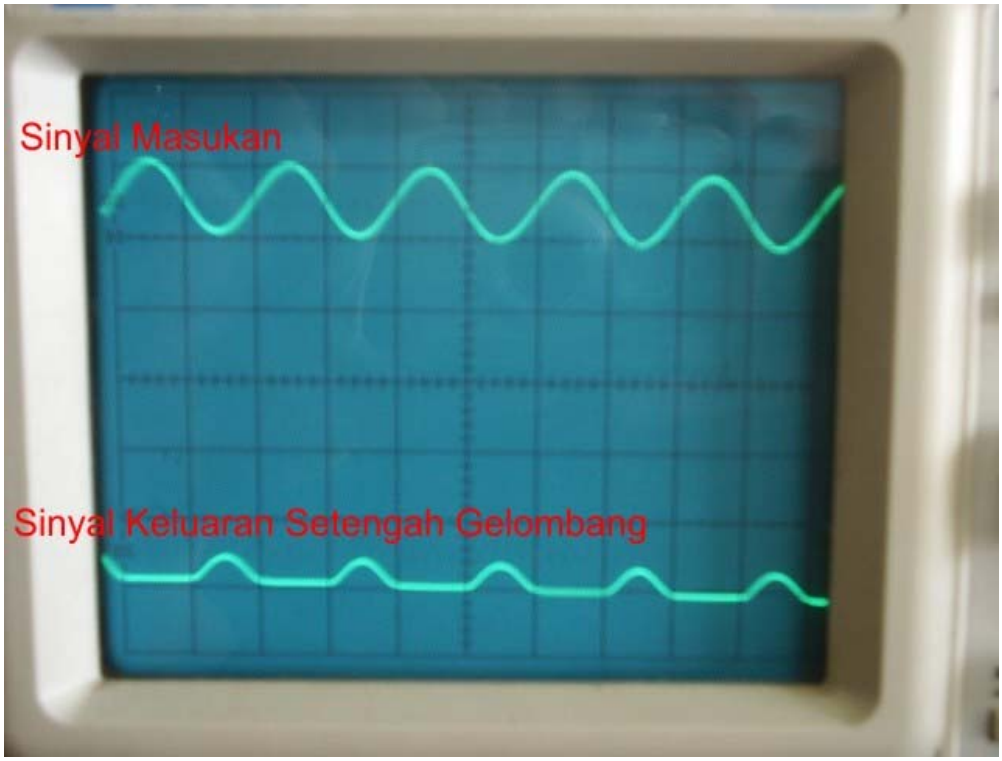
FOTO ANALISA SINYAL PADA *OSCILLOSCOPE*

TAMPILAN SINYAL INPUT DAN OUTPUT



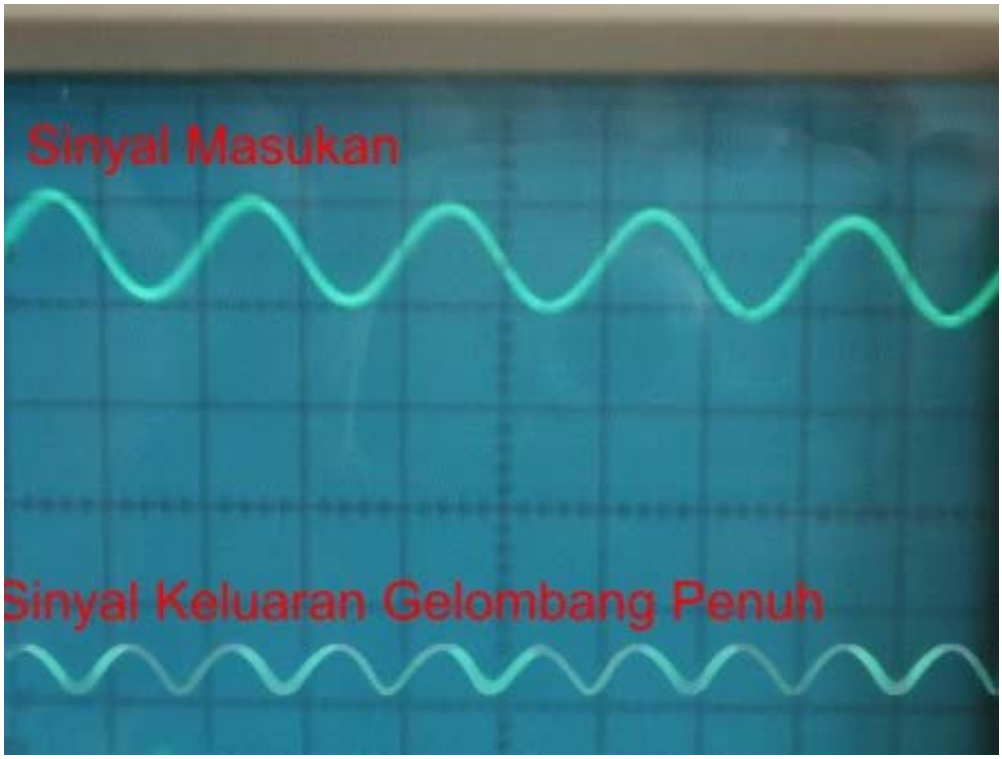
SINYAL MASUKAN DAN SINYAL

KELUARAN SETENGAH GELOMBANG



SINYAL MASUKAN DAN SINYAL

KELUARAN GELOMBANG PENUH



LAMPIRAN C

PROGRAM PADA PENGONTROL MIKRO ATMEGA16

PROGRAM UTAMA

```
/*****************************************************
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    : 6/11/2009
Author  : Lab Instrumentasi
Company : UKM
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 <math.h>
#include <delay.h>

void baca_data(void);

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

// Declare your global variables here
int temp1,temp2,temp3,temp4;
float vin1,vin2,vin3,vin4;
char satu[16];
char dua[16];
char tiga[16];
char empat[16];
```

```
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=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: 43.200 kHz
// Mode: Ph. correct PWM top=01FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// 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=0xA2;
TCCR1B=0x04;
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: 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;

// LCD module initialization
lcd_init(16);

lcd_putsf("astRon adRian'04:: maju doonk ::");
delay_ms(1000);
DDRD.4=1;
DDRD.5=1;
OCR1A=255;   // =PORTD.5  enable
OCR1B=230;   // =PORTD.4  enable

while (1)
     {
     // Place your code here
       baca_data();
       delay_ms(50);


if    ( (vin1>=0.53) && (vin1<=0.69) &&    (vin2>=0.25) &&    (vin2<=0.31) &&    (vin3>=0.08) &&
(vin3<=0.14) && (vin4>=0.08) &&  (vin4<=0.12) )
 { //maju
       DDRD=255;

       PORTD.0=0; // kanan
       PORTD.1=1;
       PORTD.2=0; // kiri
       PORTD.3=1;
       }
```

```
if    ( (vin1>=1.00) && (vin1<=1.04) &&    (vin2>=0.55) &&    (vin2<=0.61) &&    (vin3>=0.16) &&
(vin3<=0.25) && (vin4>=0.00) &&  (vin4<=0.02) )
 { //mundur
       DDRD=255;
       PORTD.0=1; // kanan
       PORTD.1=0;


       PORTD.2=1; // kiri
       PORTD.3=0;
       }

if    ( (vin1>=0.55) && (vin1<=0.57) &&    (vin2>=0.12) &&    (vin2<=0.23) &&    (vin3>=0.00) &&
(vin3<=0.08) && (vin4>=0.24) &&  (vin4<=0.35) )
{ //kiri
       delay_ms(800);
       DDRD=255;
       PORTD.0=1; // kanan
       PORTD.1=0;


       PORTD.2=0; // kiri
       PORTD.3=1;
       }

if    ( (vin1>=0.35) && (vin1<=0.37) &&    (vin2>=0.21) &&    (vin2<=0.24) &&    (vin3>=0.06) &&
(vin3<=0.18) && (vin4>=0.22) &&  (vin4<=0.23) )
{ //kanan
       DDRD=255;
       PORTD.0=0; // kanan
       PORTD.1=1;


       PORTD.2=1; // kiri
       PORTD.3=0;
       }

if    ( (vin1>=0.02) && (vin1<=0.04) &&    (vin2>=0.01) &&    (vin2<=0.17) &&    (vin3>=0.05) &&
(vin3<=0.27) && (vin4>=0.06) &&  (vin4<=0.21) )
{ //stop
       DDRD=255;
       PORTD.0=0; // kanan
       PORTD.1=0;


       PORTD.2=0; // kiri
       PORTD.3=0;
       }


     };

}

void baca_data(void)
{
lcd_clear();
temp1=read_adc(0);
temp2=read_adc(1);
temp3=read_adc(2);
temp4=read_adc(3);
```

```
vin1=((float)temp1*5/255);
vin2=((float)temp2*5/255);
vin3=((float)temp3*5/255);
vin4=((float)temp4*5/255);

sprintf(satu,"A:%0.2f ",vin1);
sprintf(dua,"B:%0.2f ",vin2);
sprintf(tiga,"C:%0.2f ",vin3);
sprintf(empat,"D:%0.2f ",vin4);

lcd_gotoxy(0,0);
lcd_puts(satu);
lcd_puts(dua);
lcd_gotoxy(0,1);
lcd_puts(tiga);
lcd_puts(empat);

printf("A:%0.2f ",vin1);
printf("B:%0.2f ",vin2);
printf("C:%0.2f ",vin3);
printf("D:%0.2f ",vin4);

}
```

PROGRAM TAMBAHAN
(SINYAL MASUKAN BERASAL DARI *OSCILLATOR)*

```
/*****************************************************
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    : 6/24/2009
Author  : Lab Instrumentasi
Company : UKM
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 <math.h>
#include <delay.h>

void baca_data(void);

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

// Declare your global variables here
int temp1,temp2,temp3,temp4;
float vin1,vin2,vin3,Vin4;
char satu[16];
char dua[16];
char tiga[16];
char empat[16];
```

```
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=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: 43.200 kHz
// Mode: Ph. correct PWM top=01FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// 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=0xA2;
TCCR1B=0x04;
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: 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;

// LCD module initialization
lcd_init(16);

lcd_putsf("astRon adRian'04:: maju doonk ::");
delay_ms(1000);
DDRD.4=1;
DDRD.5=1;
OCR1A=255;   // =PORTD.5  enable
OCR1B=230;   // =PORTD.4  enable

while (1)
     {
     // Place your code here

       baca_data();
       delay_ms(50);

       if   ( (vin>=4.00) && (vin1>=0.00) &&   (vin1<=3.00) &&   (vin2>=0.00) &&   (vin2<=3.00) &&
(vin3>=0.00) &&  (vin3<=3.00))
       { //maju
       DDRD=255;

       PORTD.0=0; // kanan
       PORTD.1=1;
       PORTD.2=0; // kiri
       PORTD.3=1;
       }
```

```
       if   ( (vin>=0.00) && (vin<=2.00) && (vin1>=4.00) &&   (vin2>=0.00) &&   (vin2<=2.00) &&
(vin3>=0.00) &&  (vin3<=2.00))
       { //mundur
       DDRD=255;

       PORTD.0=1; // kanan
       PORTD.1=0;
       PORTD.2=1; // kiri
       PORTD.3=0;
             }

     };
}

void baca_data(void)
{
lcd_clear();
temp1=read_adc(0);
temp2=read_adc(1);
temp3=read_adc(2);
temp4=read_adc(3);

vin=((float)temp1*5/255);
vin1=((float)temp2*5/255);
vin2=((float)temp3*5/255);
vin3=((float)temp4*5/255);

lcd_gotoxy(0,0);

sprintf(satu,"A:%0.2f ",vin1);
sprintf(dua,"B:%0.2f ",vin2);
sprintf(tiga,"C:%0.2f ",vin3);
sprintf(empat,"D:%0.2f ",vin4);

lcd_puts(satu);
lcd_puts(dua);
lcd_gotoxy(0,1);
lcd_puts(tiga);
lcd_puts(empat);
                      }
```

LAMPIRAN D

DATASHEET