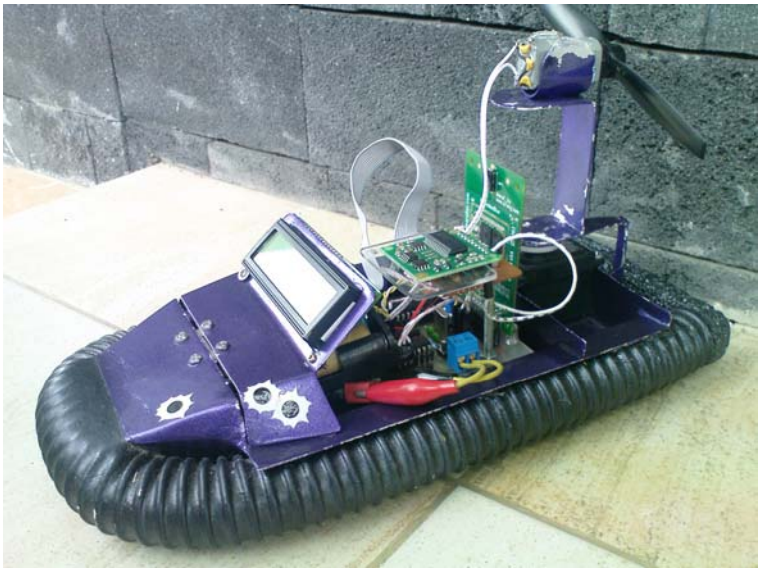
LAMPIRAN A

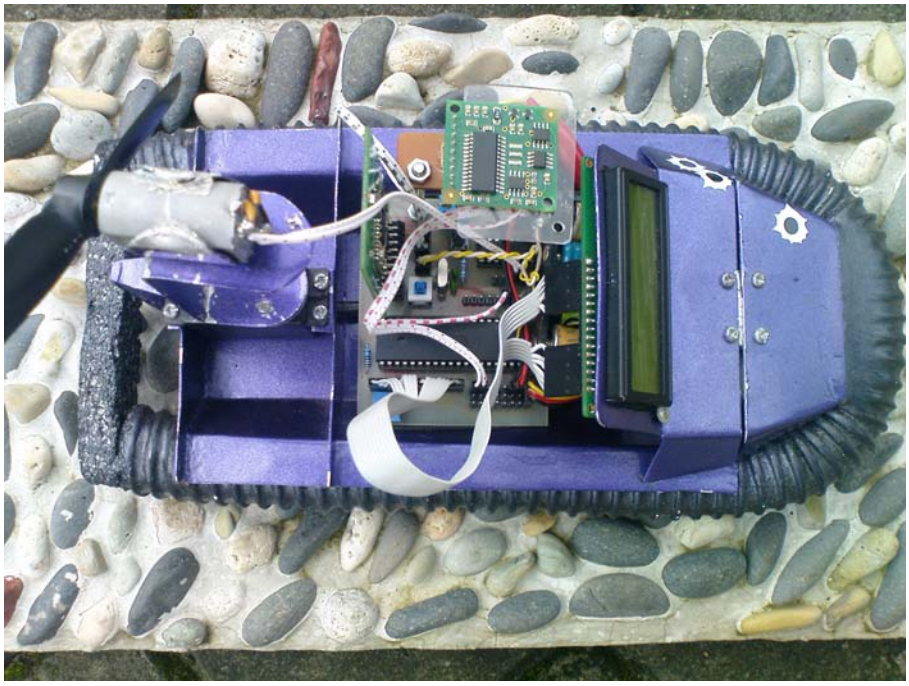
FOTO MODEL KENDARAAN *HOVERCRAFT*

TAMPAK DEPAN

TAMPAK SAMPING



TAMPAK ATAS



TAMPAK BELAKANG

LAMPIRAN B
PROGRAM PADA PENGONTROL MIKRO ATMEGA16

```
/*****************************************************
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 : Final Assignment
Version : 2009
Date    : 2/2/2009
Author  : Erwin Surianto
Company : Electrical Engineering 2005 UKM
Comments: Realization Hovercraft Vehicle Model Using AVR


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 16.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

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

int ton,tim,detik;
char temp,data1,data2,text[16],seperlimapuluh,seperlimapuluh2,terima;
float data3,ref;
bit ack,set,uart;


// I2C Bus functions
#asm
   .equ __i2c_port=0x1B ;PORTA
   .equ __sda_bit=2
   .equ __scl_bit=1
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.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;
      };
   };
   uart=1;
   terima=data;
}

#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

// 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>

void servo(int sudut)
{
ton=(sudut*10)+600;
OCR1A=ton*2;
}

void dc(int cycle)
{
int hasil;
hasil=cycle*255/100;
OCR0=hasil;
}

void kompas(void)
{
i2c_start();
i2c_write(0xC0);
i2c_write(0x02);
i2c_start();
i2c_write(0xC1);
data1=i2c_read(1);
data2=i2c_read(0);
i2c_stop();

data3=((float)data1*256+data2)/10;
}


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

seperlimapuluh++;

if(seperlimapuluh==tim)
{
kompas();
lcd_gotoxy(4,1);lcd_putsf("derajat");
lcd_gotoxy(4,0);
sprintf(text,"%5.1f",data3);
lcd_puts(text);
puts(text);putchar(13);
seperlimapuluh=0;
}


seperlimapuluh2++;

if(seperlimapuluh2==50)
```

```
{
detik++;
seperlimapuluh2=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=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x08;

// 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=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=0 State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x20;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 15.625 kHz
// Mode: Fast PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x6D;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 2000.000 kHz
// Mode: Fast PWM top=ICR1
// OC1A output: Non-Inv.
// OC1B output: Discon.
// 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=0x82;
TCCR1B=0x1A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x9C;
ICR1L=0x40;
OCR1AH=0x0B;
OCR1AL=0xB8;
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=0x20;

// 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=0x67;

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

// I2C Bus initialization
i2c_init();

// LCD module initialization
lcd_init(16);

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

PORTA=0x40;

lcd_clear();
lcd_putsf("tunggu perintah");

while (1)
    {
    if(uart==1)
    {
    uart=0;
    switch (terima)
    {
    case 'a':
    case 'A':
                        lcd_clear();
            lcd_putsf("A");

            // Utara (open loop)

            while(data3>0.5 && data3<359.5)
            {
```

```
        tim=1;

        if(data3>0 && data3<=180)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }
        }

        dc(0);
        servo(90);
        delay_ms(4000);
        tim=50;
        dc(70);
        delay_ms(10000);
        dc(0);

break;

case 'b':
case 'B':
        lcd_clear();
        lcd_putsf("B");

        // Utara (close loop)

        while(data3>0.5 && data3<359.5)
        {
        tim=1;
        if(data3>0 && data3<=180)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }
        }

        dc(0);
        servo(90);
        delay_ms(4000);

        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;

        if(data3>0 && data3<=180)
        {
        servo(95);
        dc(70);
        }
        else
        {
        servo(85);
        dc(70);
        }
        }
        dc(0);

break;
```

```
case 'c':
case 'C':
        lcd_clear();
        lcd_putsf("C");

        // Timur (open loop)

        while(data3<89.5 || data3>90.5)
        {
        tim=1;

        if(data3>90 && data3<=270)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);
        tim=50;
        dc(70);
        delay_ms(10000);
        dc(0);

 break;


 case 'd':
 case 'D':
        lcd_clear();
        lcd_putsf("D");

        // Timur (close loop)

        while(data3<89.5 || data3>90.5)
        {
        tim=1;

        if(data3>90 && data3<=270)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);

        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;
```

```
        if(data3>90 && data3<=270)
        {
        servo(95);
        dc(70);
        }
        else
        {
        servo(85);
        dc(70);
        }
        }
        dc(0);

break;


case 'e':
case 'E':
        lcd_clear();
        lcd_putsf("E");

        // Selatan (open loop)

        while(data3<179.5 || data3>180.5)
        {
        tim=1;

        if(data3>180 || data3<=0)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);
        tim=50;
        dc(70);
        delay_ms(10000);
        dc(0);

break;


case 'f':
case 'F':
        lcd_clear();
        lcd_putsf("F");

        // Selatan (close loop)

        while(data3<179.5 || data3>180.5)
        {
        tim=1;

        if(data3>180 || data3<=0)
        {
        servo(110);
        dc(70);
        }
        else
```

```
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);

        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;

        if(data3>180 || data3<=0)
        {
        servo(95);
        dc(70);
        }
        else
        {
        servo(85);
        dc(70);
        }
        }
        dc(0);

break;


case 'g':
case 'G':
        lcd_clear();
        lcd_putsf("G");

        // Barat (open loop)

        while(data3<269.5 || data3>270.5)
        {
        tim=1;

        if(data3>270 || data3<=90)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);
        tim=50;
        dc(70);
        delay_ms(10000);
        dc(0);

 break;


 case 'h':
 case 'H':
```

```
        lcd_clear();
        lcd_putsf("H");

        // Barat (close loop)

        while(data3<269.5 || data3>270.5)
        {
        tim=1;

        if(data3>270 || data3<=90)
        {
        servo(110);
        dc(70);
        }
        else
        {
        servo(70);
        dc(70);
        }

        }

        dc(0);
        servo(90);
        delay_ms(4000);


        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;

        if(data3>270 || data3<=90)
        {
        servo(95);
        dc(70);
        }
        else
        {
        servo(85);
        dc(70);
        }
        }
        dc(0);

break;


case 'i':
case 'I':
        lcd_clear();
        lcd_putsf("I");

        // Melingkar Kecil (open loop)

        servo(50);
        dc(0);
        delay_ms(5000);

        tim=50;
        for(temp=20;temp<=70;temp++)
        {
        dc(temp);
        delay_ms(100);
        }

        for(temp=40;temp<=60;temp++)
        {
        dc(temp);
```

```
        delay_ms(100);
        }

        for(temp=40;temp<=60;temp++)
        {
        dc(temp);
        delay_ms(100);
        }

        for(temp=40;temp<=60;temp++)
        {
        dc(temp);
        delay_ms(100);
        }

        for(temp=40;temp<=60;temp++)
        {
        dc(temp);
        delay_ms(100);
        }

        dc(0);

break;


case 'j':
case 'J':
        lcd_clear();
        lcd_putsf("J");

        // Melingkar Kecil (close loop)

        ref=data3;
        servo(55);
        dc(0);
        delay_ms(5000);

        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;

        ref=ref+90;
        if(ref>=360)
        {ref=ref-360;ack=!ack;set=1;}

        if(data3>=0 && data3<30 && set==1)
        {ack=!ack;set=0;}

        delay_ms(100);

        if(ack==0)
        {
                if(data3<(ref-2))
                {
                servo(50);
                dc(60);
                }
                if(data3>(ref+2))
                {
                servo(60);
                dc(60);
                }
                if(data3>=(ref-2) && data3<=(ref+2))
                {
                servo(55);
                dc(70);
                }
```

```
        }


        if(ack==1)
        {
              if(data3<(ref-2))
              {
              servo(60);
              dc(60);
              }
              if(data3>(ref+2))
              {
              servo(50);
              dc(60);
              }
              if(data3>=(ref-2) && data3<=(ref+2))
              {
              servo(55);
              dc(70);
              }
        }
        }

break;


case 'k':
case 'K':
        lcd_clear();
        lcd_putsf("K");

        // Melingkar Besar (open loop)

        servo(70);
        dc(0);
        delay_ms(5000);

        tim=50;
        dc(70);
        delay_ms(10000);

        dc(0);

break;


case 'l':
case 'L':
        lcd_clear();
        lcd_putsf("L");

        // Melingkar Besar (close loop)

        ref=data3;
        servo(70);
        dc(0);
        delay_ms(5000);

        detik=0;seperlimapuluh2=0;
        while(detik<=10)
        {
        tim=50;

        ref=ref+36;
        if(ref>=360)
        {ref=ref-360;ack=!ack;set=1;}

        if(data3>=0 && data3<30 && set==1)
        {ack=!ack;set=0;}
```

```
delay_ms(100);

if(ack==0)
{
      if(data3<(ref-2))
      {
      servo(65);
      for(temp=20;temp<=80;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }
      }

      if(data3>(ref+2))
      {
      servo(75);
      for(temp=20;temp<=80;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }

      for(temp=50;temp<=70;temp++)
      {
      dc(temp);
      }
      }

      if(data3>=(ref-2) && data3<=(ref+2))
      {
      servo(70);
      for(temp=20;temp<=80;temp++)
      {
      dc(temp);
```

```
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }
        }
}


if(ack==1)
{
        if(data3<(ref-2))
        {
        servo(75);
        for(temp=20;temp<=80;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }
        }

        if(data3>(ref+2))
        {
        servo(65);
        for(temp=20;temp<=80;temp++)
        {
        dc(temp);
        }

        for(temp=50;temp<=70;temp++)
        {
        dc(temp);
        }
```

```
                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }
                }

                if(data3>=(ref-2) && data3<=(ref+2))
                {
                servo(70);
                for(temp=20;temp<=80;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }

                for(temp=50;temp<=70;temp++)
                {
                dc(temp);
                }
                }
            }
            }

      break;


    } // yg switch

    } // yg if

    delay_ms(200);
    lcd_clear();
    lcd_putsf("tunggu perintah");


    }; // yg while(1)
}
```