LAMPIRAN A

FOTO ROBOT SWARM A & ROBOT SWARM B

LAMPIRAN A
FOTO ROBOT SWARM

FOTO ROBOT SWARM A TAMPAK DEPAN

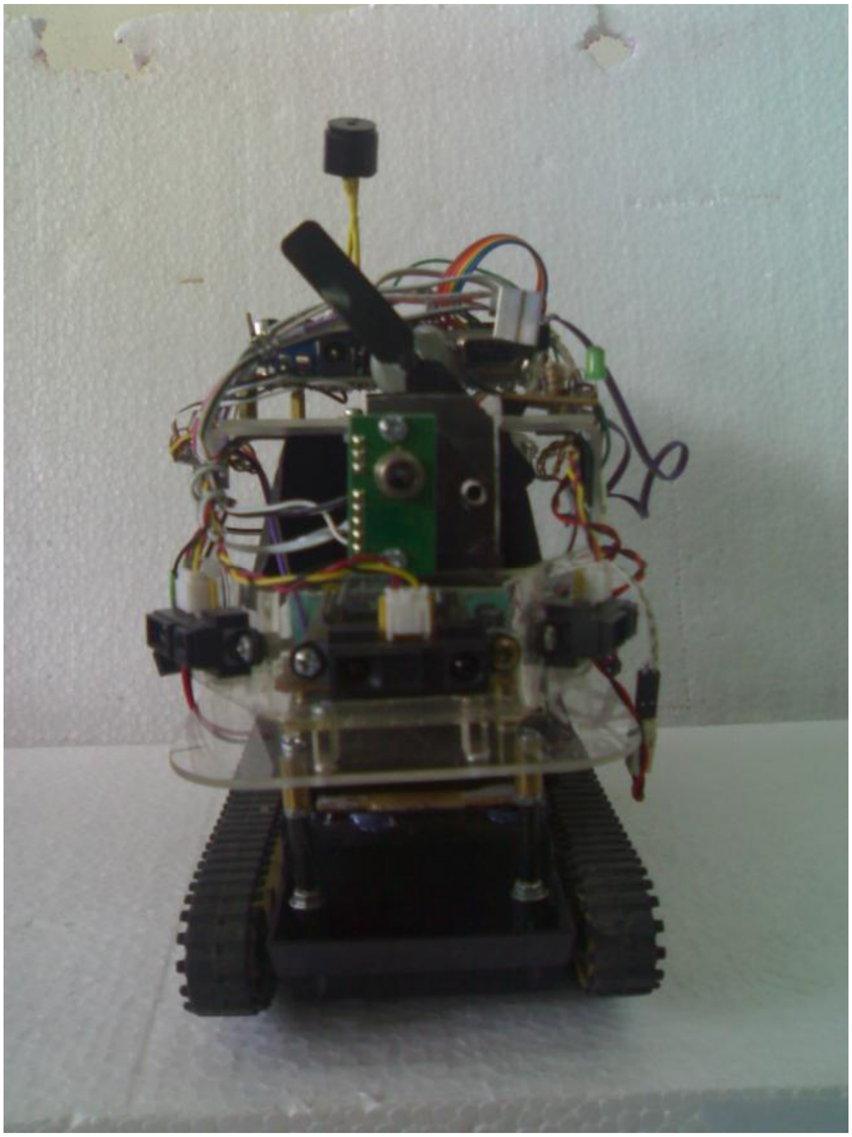
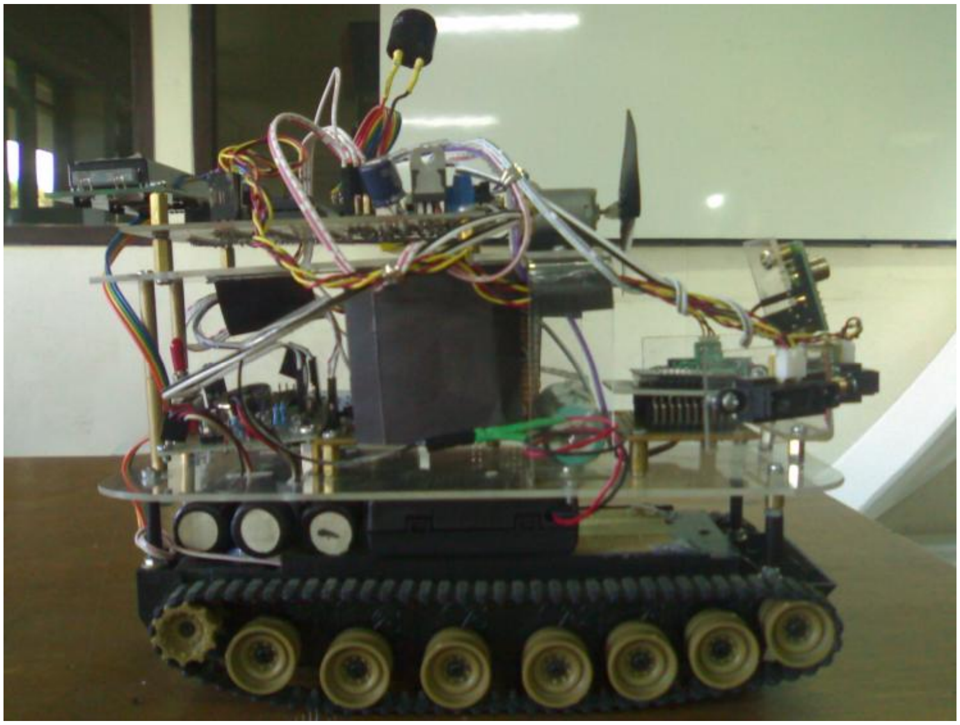


FOTO ROBOT SWARM A TAMPAK SAMPING



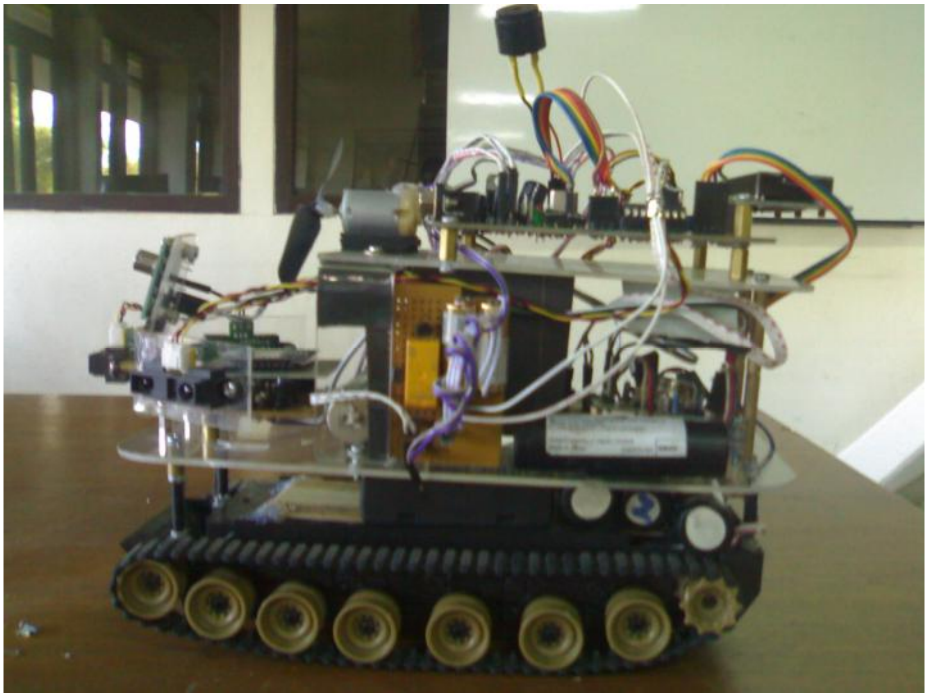


FOTO ROBOT SWARM A TAMPAK BELAKANG

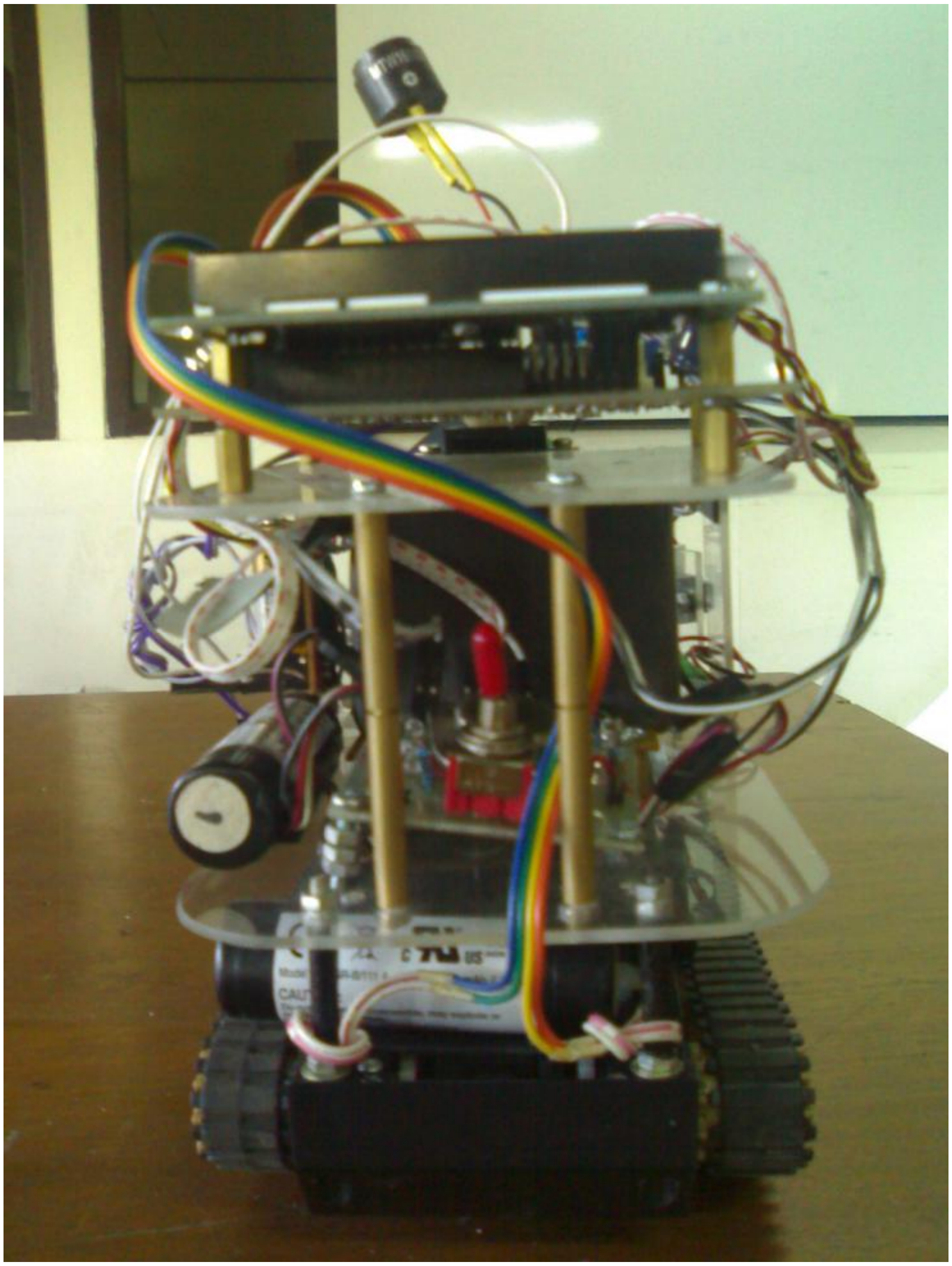


FOTO ROBOT SWARM A TAMPAK ATAS

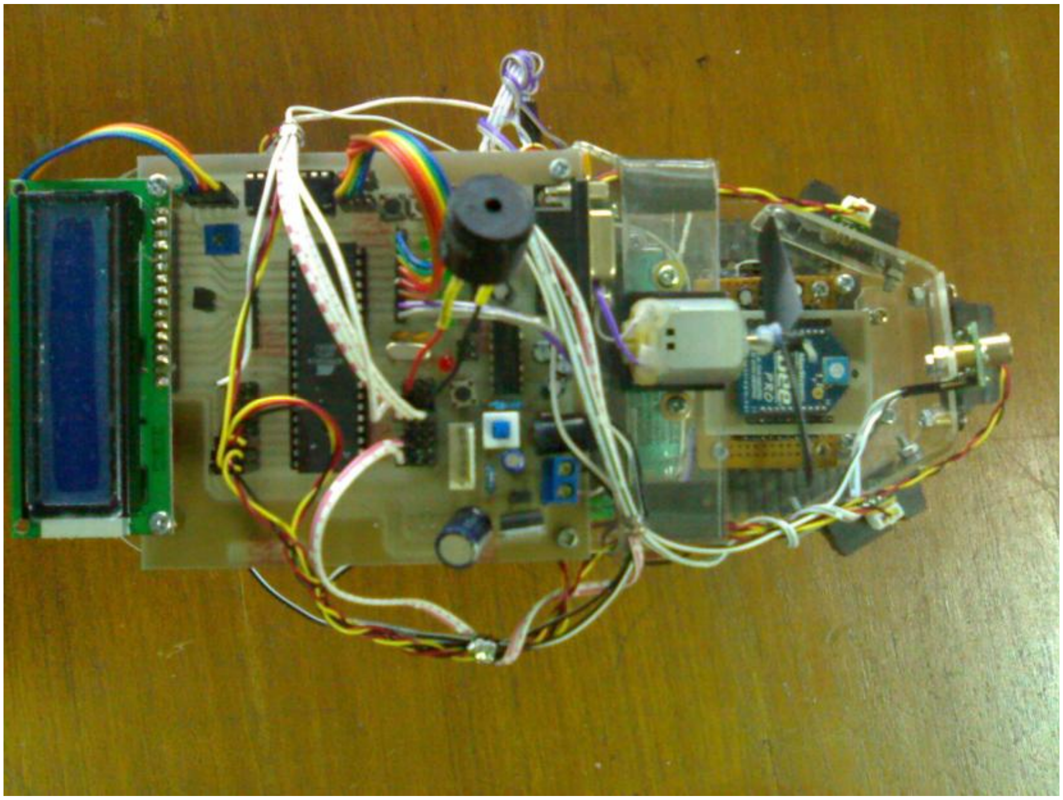


FOTO ROBOT SWARM B TAMPAK DEPAN

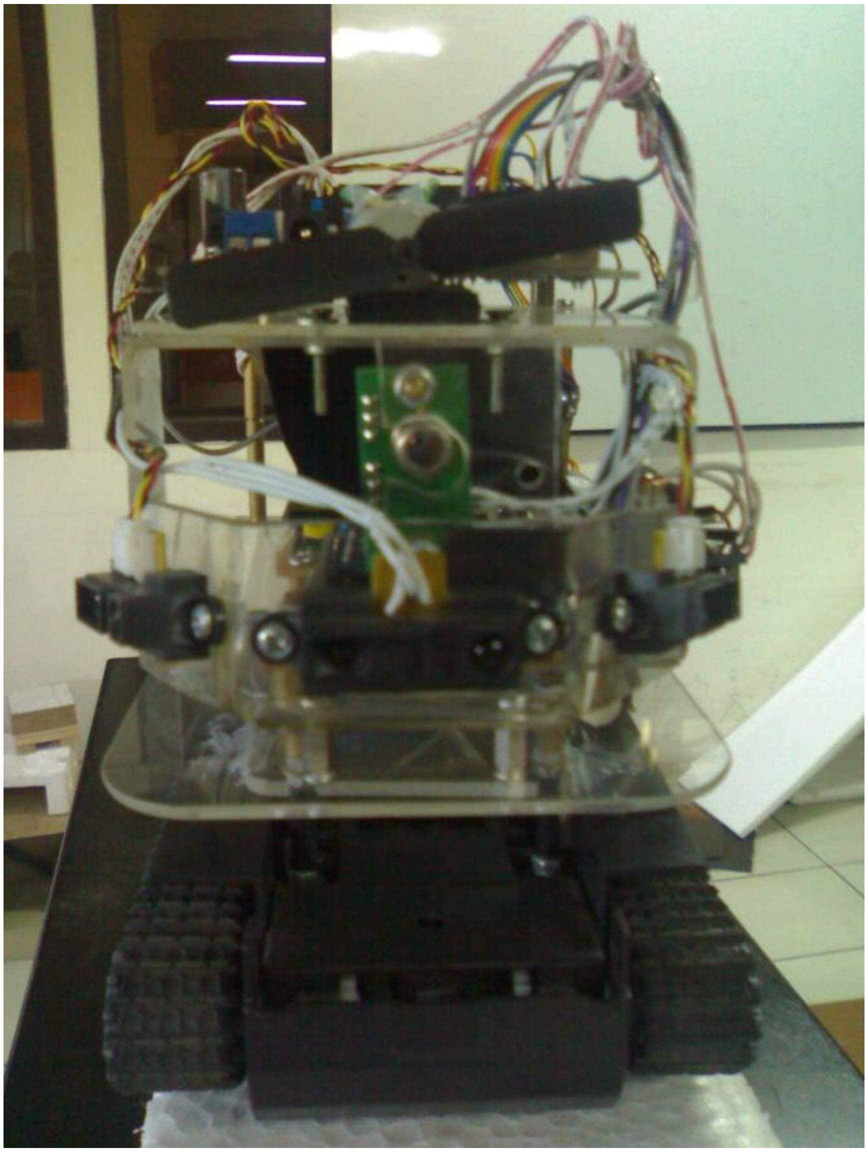


FOTO ROBOT SWARM B TAMPAK SAMPING

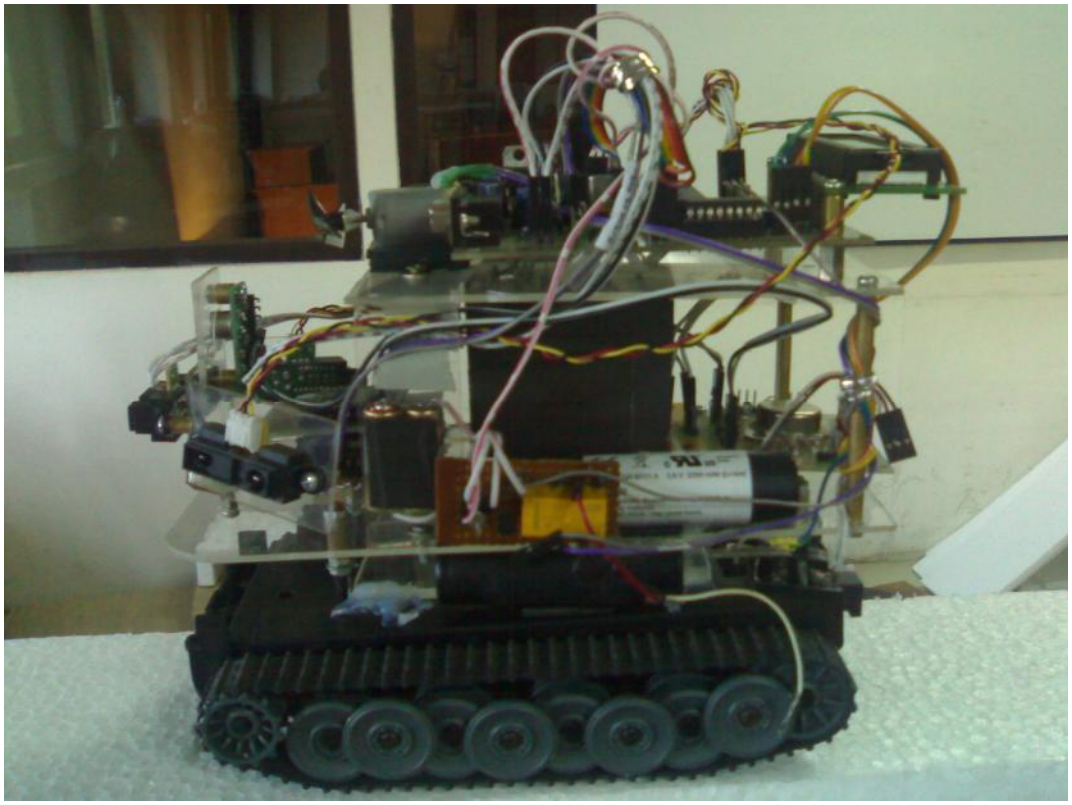


FOTO ROBOT SWARM B TAMPAK BELAKANG

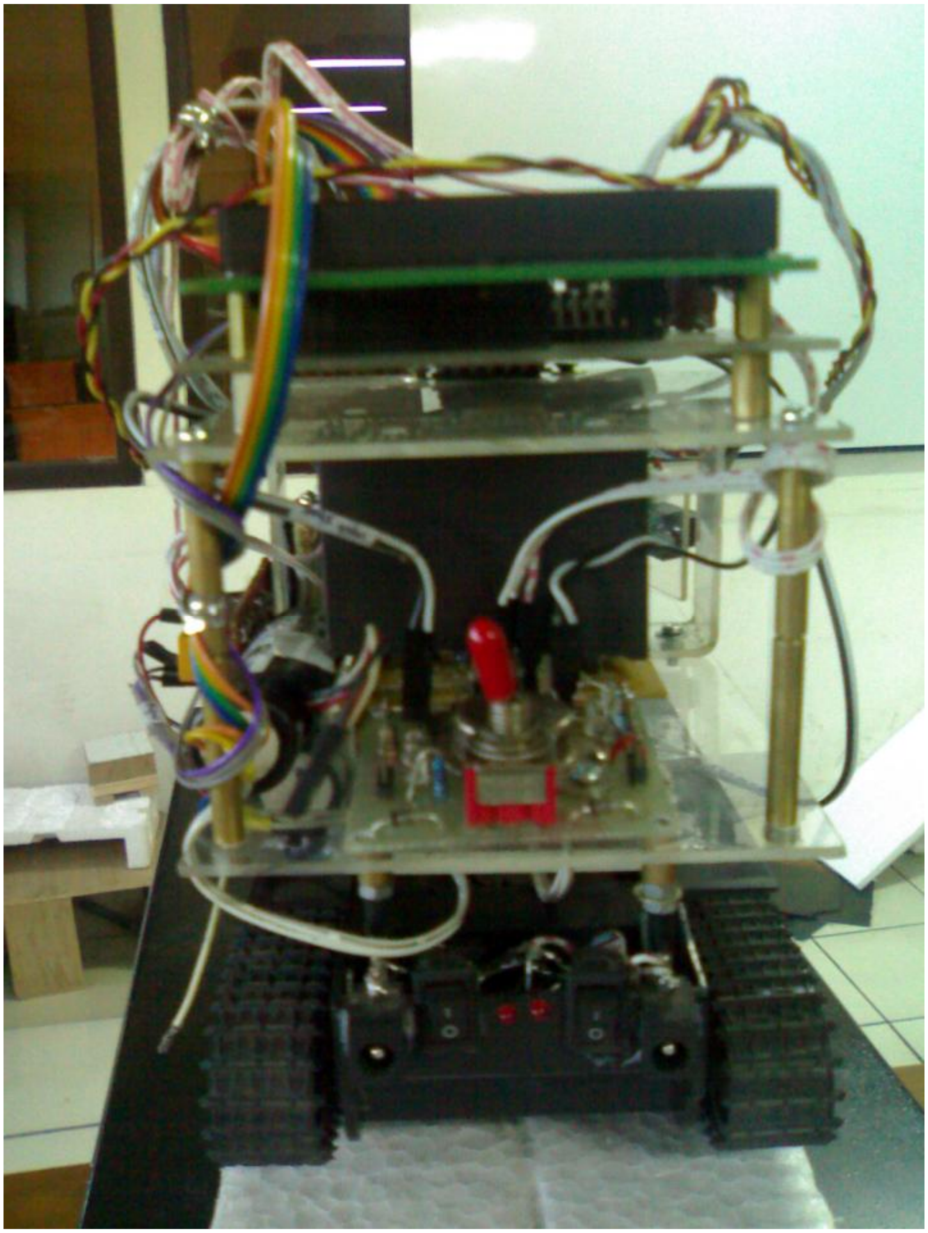
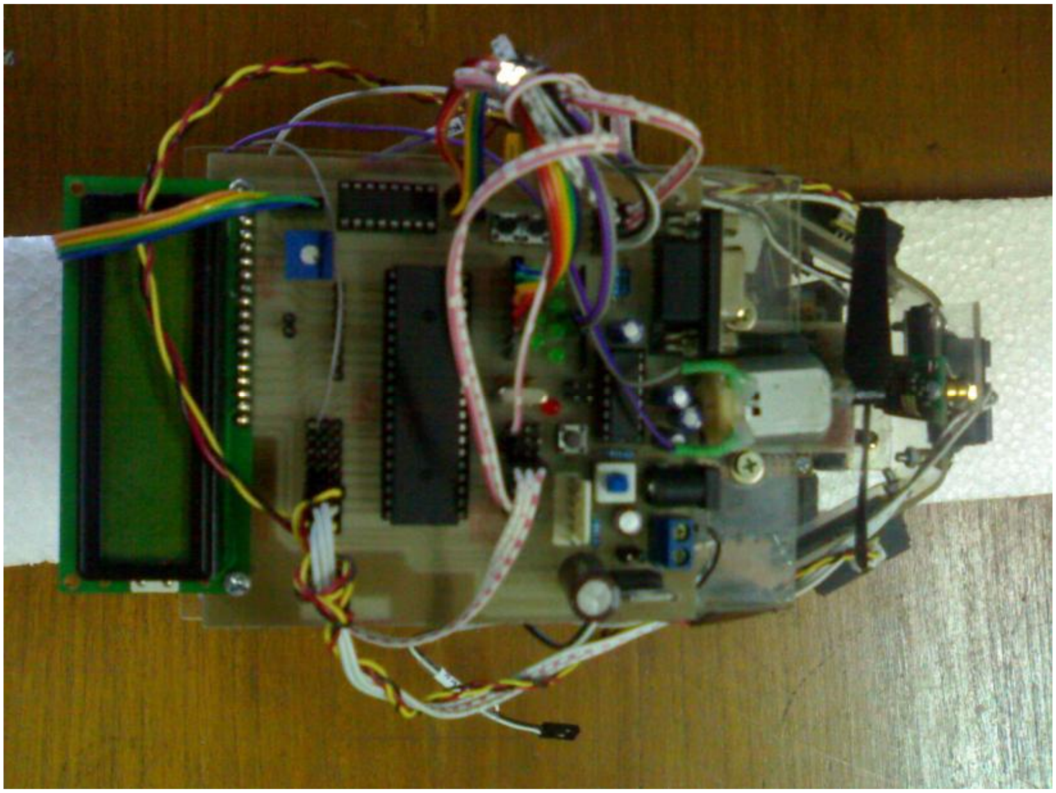


FOTO ROBOT SWARM B TAMPAK ATAS



LAMPIRAN B

PROGRAM PADA PENGONTROL

ATMEGA16

LAMPIRAN B

PEMROGRAMAN PADA PENGONTROL MIKRO ATMEGA16

```
/****************************************************
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         : Robot Swarm
Version         :
Date            : 2/6/2007
Author          : Dita Kostian Malahayati
Company          : Universitas Kristen Maranatha
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 <math.h>
#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long
#define kipas PORTB.5
```

```
// I2C Bus functions
#asm
   .equ __i2c_port=0x18 ;PORTB
   .equ __sda_bit=1
   .equ __scl_bit=0
#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

bit bayi_ditemukan;
uchar jumlah_lilin_yang_dipadamkan;



interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;

if(data=='b'){
    data=0;
    bayi_ditemukan=1;                                    // terima berita bayi sudah ketemu
}
if(data=='l'){
    data=0;
    jumlah_lilin_yang_dipadamkan++;           // terima berita bahwa lilin sudah di padamkan
}



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

// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
#define ADC_VREF_TYPE 0x60
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

//===============================================stop
void stop(void)
{
    PORTD.2=0;
    PORTD.3=0;
    PORTD.6=0;
    PORTD.7=0;
}

//===============================================maju
void maju(void)
{
    PORTD.2=0;
    PORTD.3=1;
    PORTD.4=1;
    PORTD.5=1;
    PORTD.6=0;
    PORTD.7=1;
    delay_ms(100);
}

//===============================================belok_kanan
void belok_kanan(void)
{
    PORTD.2=1;
    PORTD.3=0;
    PORTD.4=0;
```

```
    PORTD.5=1;
    PORTD.6=0;
    PORTD.7=1;
    delay_ms(100);
}

//===============================================belok_kiri
void belok_kiri(void)
{
    PORTD.2=0;
    PORTD.3=1;
    PORTD.4=1;
    PORTD.5=0;
    PORTD.6=1;
    PORTD.7=0;
    delay_ms(100);
}


//===============================================baca_IR
uchar IR_kanan;
uchar IR_kiri;
uchar IR_depan;
void baca_IR(void)
{
    uchar kanan;
    uchar kiri;
    uchar depan;

    kiri=read_adc(0);
    IR_kiri=2141.72055 * (pow(kiri,-1.078867));      // agar dalam cm

    depan=read_adc(1);
    IR_depan=2141.72055 * (pow(depan,-1.078867));     // agar dalam cm
    kanan=read_adc(2);
    IR_kanan=2141.72055 * (pow(kanan,-1.078867));     // agar dalam cm
}


//==========================================telusur_dinding_kiri
#define beeper PORTB.7
void telusur_dinding_kiri(void)
{

       baca_IR();
       if(IR_depan>20)
       {
            if(IR_kiri<=14) belok_kanan();
            else if(IR_kiri>15 && IR_kiri<19) maju();
            else if(IR_kiri>=19)belok_kiri();
       }
       else
       {
            if(IR_depan<23 || IR_kiri<19)
            belok_kanan();
       }
  }


//==========================================telusur_dinding_kanan

void telusur_dinding_kanan(void)
{

       baca_IR();
       if(IR_depan>20)
       {
```

```
       if(IR_kanan<=14) belok_kiri();
       else if(IR_kanan>15 && IR_kanan<19) maju();
       else if(IR_kanan>=19)belok_kanan();
    }
    else
    {
       if(IR_depan<23 || IR_kanan<19)
       belok_kiri();
    }
}


//=============================================TPA_read
uchar data_temperatur[8];
void TPA_read(void)
{
uchar i;
  for (i=0;i<8;i++)
  {
          delay_ms(40);
          i2c_start();
          i2c_write(0xD0);
          i2c_write(i+2);
          i2c_start();
          i2c_write(0xD1);
          data_temperatur[i]=i2c_read(0);
          i2c_stop();

        }
}
```

```
//===============================================deteksi_panas
#define uvtron PORTB.4

bit lilin_ditemukan;
bit deteksi_lilin;
bit deteksi_bayi;
void deteksi_panas(void){
uchar temperatur;
uchar k;
    TPA_read();
    for (k=0;k<8;k++){
        temperatur =data_temperatur[k];
        if (temperatur>100)
        {
            if(PINB.4==1)deteksi_lilin=1;
            if(PINB.4==0)deteksi_bayi=1;
        }

    }


}

//===============================================main

void main(void)
{
DDRB=0x70;
DDRD=0xFC;

UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
```

```
UBRRL=0x47;

ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

i2c_init();
lcd_init(16);
#asm("sei")
bayi_ditemukan=0;
DDRB.4=0;
beeper=0;
jumlah_lilin_yang_dipadamkan=0;
lilin_ditemukan=0;
kipas=0;
while (1)
      {



            if(bayi_ditemukan==0)
            {
                telusur_dinding_kiri();
                deteksi_panas();
                if(deteksi_bayi)

                {
                    bayi_ditemukan=1;
                    stop();
                    putchar('b');
                    beeper=1;
                    delay_ms(2000);
                    beeper=0;
                    deteksi_lilin=0;
                }
```

```
    else if(deteksi_lilin)
    {
        lilin_ditemukan=1;
        deteksi_lilin=0;
    }
}
else if(bayi_ditemukan==1)

{
    if(lilin_ditemukan)telusur_dinding_kanan();
    else if(!lilin_ditemukan)telusur_dinding_kiri();
    deteksi_panas();
    if(deteksi_lilin)
    {
        stop();
        putchar('l');
        kipas=1;
        delay_ms(3000);
        kipas=0;
        jumlah_lilin_yang_dipadamkan++;
        deteksi_lilin=0;
        if( jumlah_lilin_yang_dipadamkan>1)
        {
            stop();
            for(;;){}
        }
    }
}
```