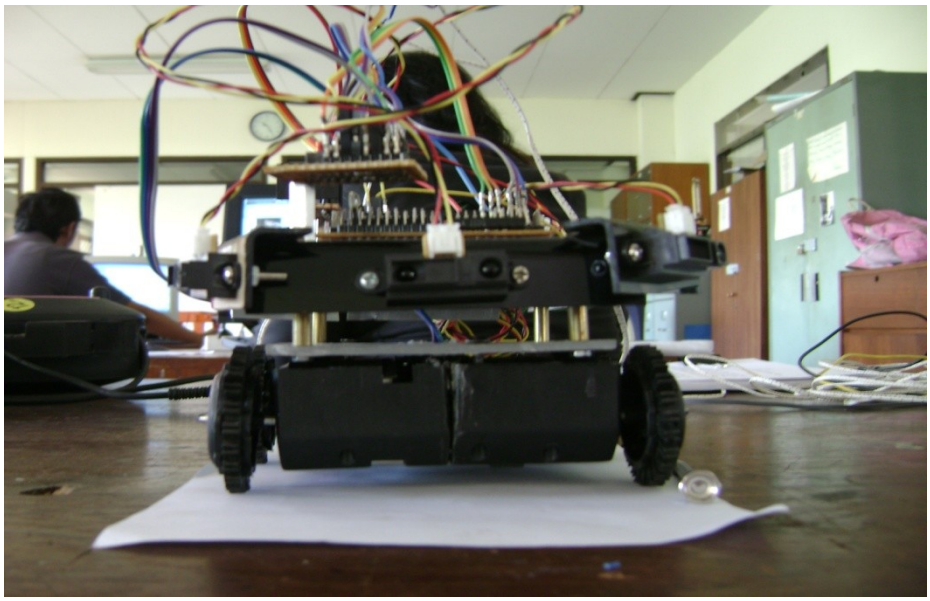
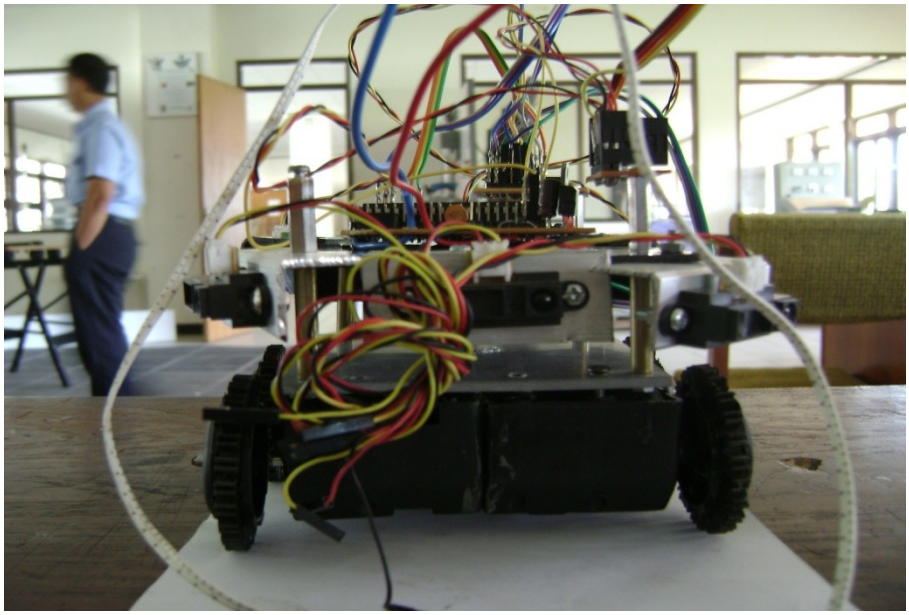
# LAMPIRAN A
# FOTO ROBOT MOBIL

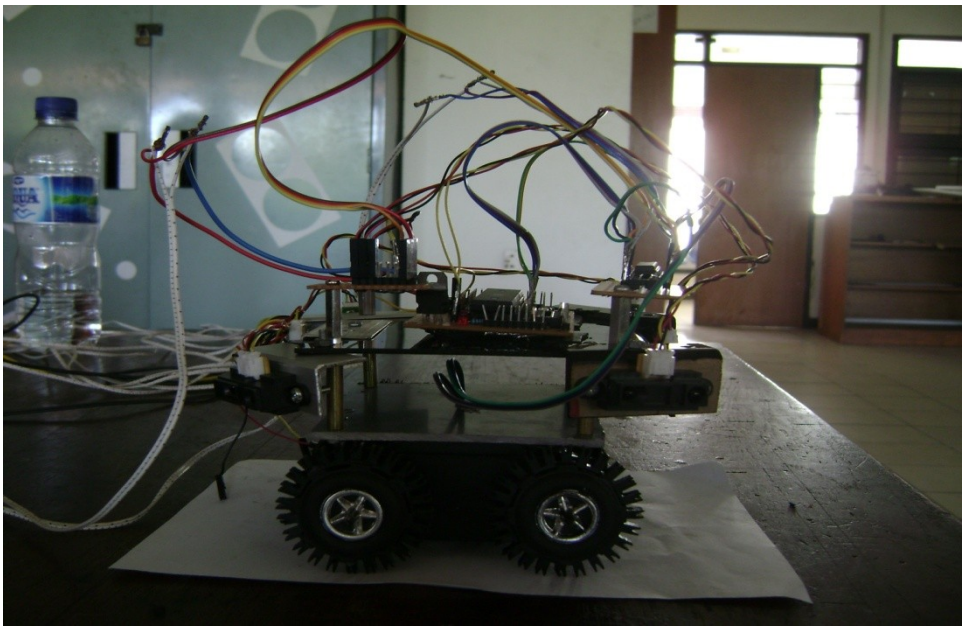
Tampak Depan



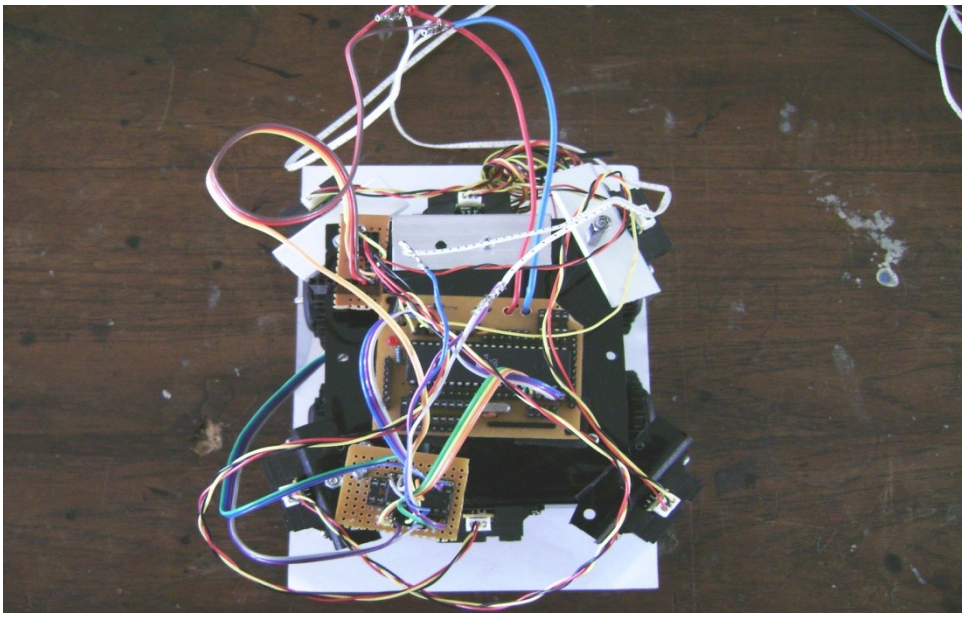
Tampak Belakang



Tampak Samping



Tampak Atas



# LAMPIRAN B

# PROGRAM PADA PENGONTROL

# ATMEGA16

## PROGRAM UTAMA

```
/*****************************************************
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 : wall f
Version :
Date    : 7/16/2009
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 <math.h>


unsigned int r1, l1, f1, r2, l2, f2;

bit depan_a_status, depan_b_status;

// 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

void kananmaju(void)
{
PORTD.3=1;//pwm
PORTD.2=0;//kanan
PORTD.1=1;//kanan
}
void kananmundur(void)
{
PORTD.3=1;//pwm
PORTD.2=1;//kanan
PORTD.1=0;//kanan
}
void kirimaju(void)
{
PORTD.6=1;//pwm
PORTD.5=0;
PORTD.4=1;
}
void kirimundur(void)
{
PORTD.6=1;//pwm
PORTD.5=1;
PORTD.4=0;
}

void belok_kanan(){
 kirimaju();
 kananmundur();
}

void belok_kiri(){
 kananmaju();
 kirimundur();
}
```

```
void maju_1(){
 kananmaju();
 kirimaju();
}

void maju_2(){
 kananmundur();
 kirimundur();
}

void baca_sensor(){

 l1 = read_adc(0);
 l1 = 2141.72055 * (pow(l1,-1.078867)); // kiri 1
 f1 = read_adc(1);
 f1 = 2141.72055 * (pow(f1,-1.078867)); // depan 1
 r1 = read_adc(2);
 r1 = 2141.72055 * (pow(r1,-1.078867)); // kanan 1
 l2 = read_adc(3);
 l2 = 2141.72055 * (pow(l2,-1.078867)); // kiri 2
 f2 = read_adc(4);
 f2 = 2141.72055 * (pow(f2,-1.078867)); // depan 2
 r2 = read_adc(5);
 r2 = 2141.72055 * (pow(r2,-1.078867)); // kanan 2


}




   /**********
     depan a
   ***********/

void depan_a(){

  baca_sensor();

   if ((r1<15) && (10<=r1)){
     maju_1();
     } else if (r1>15) {
        belok_kanan();
       } else if(r1<10){
         belok_kiri();
```

```
      }else if ((r1<15) && (f1<20) && (l1>20) ){
          while (f1<30){
                belok_kiri();
                baca_sensor();
                  }
            }else {
                depan_a_status=0;
                depan_b_status=1;
            }
}



   /**********
    depan b
   ***********/

void depan_b(){
  baca_sensor();
   if ((r2<15) && (10<=r2)){
    maju_2();
    } else if (r2>15) {
      belok_kanan();
     }else if(r2<10){
      belok_kiri();
      } else if ((r2<15) && (f2<20) && (l2>20) ){
          while (f2<30){
                belok_kiri();
                baca_sensor();
                  }
            }else {
                depan_a_status=1;
                depan_b_status=0;
            }
}


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
// Func=out
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xFF;

// 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: 10.800 kHz
// Mode: Phase correct PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x65;
TCNT0=0x00;
OCR0=0x70;


// 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: Phase correct PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x67;
TCNT2=0x00;
OCR2=0x70;

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

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x82;


// 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);

depan_a_status=1;

while (1)
{


 while(depan_a_status){
    depan_a();
     }

 while(depan_b_status){
    depan_b();
    }



// lcd_clear();
// sprintf(text," Sensor %3d %3d %3d ",a,b,c);
// lcd_puts(text);
// delay_ms(200);
// kananmaju();
// kirimaju();
// delay_ms(1000);
// kananmundur();
// kirimundur();
// delay_ms(1000);

      };
}
```

# LAMPIRAN C

# DATASHEET GP2D12

# GP2D12/GP2D15

## General Purpose Type Distance Measuring Sensors

### ■ Features

1. Less influence on the color of reflective objects, reflectivity
2. Line-up of distance output/distance judgement type
   Distance output type (analog voltage) : **GP2D12**
   Detecting distance : 10 to 80cm
   Distance judgement type : **GP2D15**
   Judgement distance : 24cm
   (Adjustable within the range of 10 to 80cm)
3. External control circuit is unnecessary
4. Low cost

### ■ Applications

1. TVs
2. Personal computers
3. Cars
4. Copiers

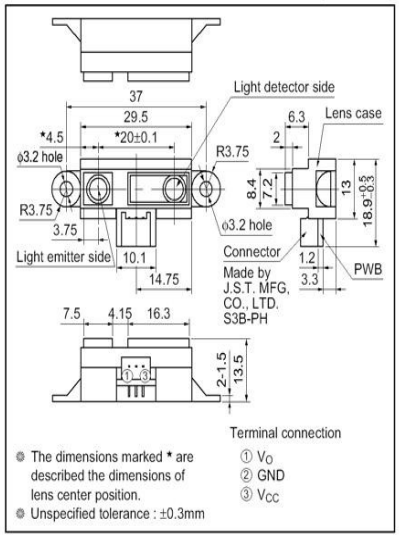
### ■ Absolute Maximum Ratings

(Ta=25°C, Vcc=5V)

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Supply voltage | $V_{CC}$ | −0.3 to +7 | V |
| Output terminal voltage | $V_O$ | -0.3 to Vcc +0.3 | V |
| Operating temperature | $T_{opr}$ | −10 to +60 | °C |
| Storage temperature | $T_{stg}$ | −40 to +70 | °C |

### ■ Outline Dimensions

(Unit : mm)



## ■ Recommended Operating Conditions

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Operating supply voltage | $V_{CC}$ | 4.5 to +5.5 | V |

## ■ Electro-optical Characteristics

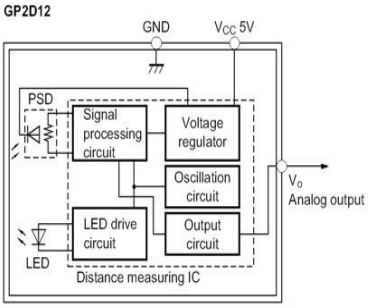
(Ta=25°C, $V_{CC}$=5V)

| Parameter | | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Distance measuring range | | ΔL | *1 *3 | 10 | – | 80 | cm |
| Output terminal voltage | GP2D12 | $V_O$ | L=80cm *1 | 0.25 | 0.4 | 0.55 | V |
| | GP2D15 | $V_{OH}$ | Output voltage at High *1 | Vcc −0.3 | – | – | V |
| | | $V_{OL}$ | Output voltage at Low *1 | – | – | 0.6 | V |
| Difference of output voltage | GP2D12 | $\Delta V_O$ | Output change at L=80cm to 10cm *1 | 1.75 | 2.0 | 2.25 | V |
| Distance characteristics of output | GP2D15 | $V_O$ | *1 *2 *4 | 21 | 24 | 27 | cm |
| Average Dissipation current | | $I_{CC}$ | L=80cm *1 | – | 33 | 50 | mA |

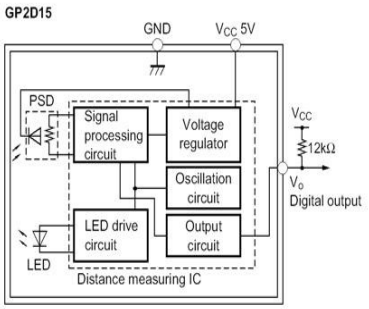
Note) L : Distance to reflective object.

*1 Using reflective object : White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio ; 90%).
*2 We ship the device after the following adjustment : Output switching distance L=24cm±3cm must be measured by the sensor.
*3 Distance measuring range of the optical sensor system.
*4 Output switching has a hysteresis width. The distance specified by Vo should be the one with which the output L switches to the output H.

### Fig.1 Internal Block Diagram




### Fig.2 Internal Block Diagram




### Fig.3 Timing Chart

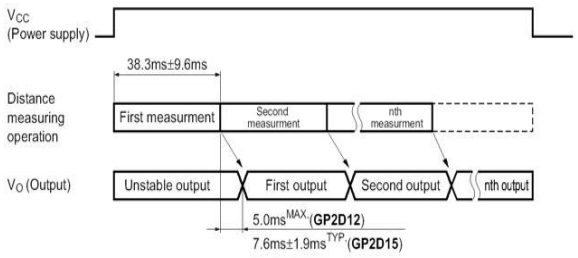


**Fig.4 Distance Characteristics**

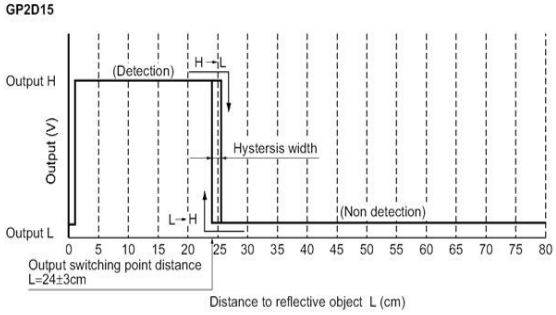



**Fig.5 Analog Output Voltage vs. Surface Illuminance of Reflective Object**

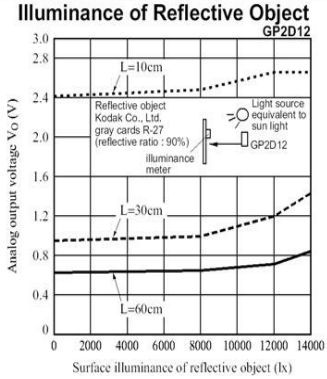



**Fig.6 Analog Output Voltage vs.Distance to Reflective Object**

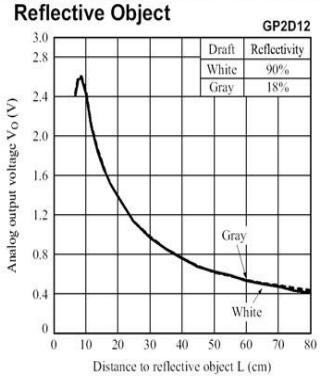



**Fig.7 Analog Output Voltage vs.Ambient Temperature**

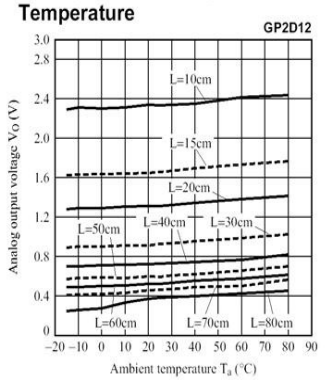



**Fig.8 Analog Output Voltage vs.Detection Distance**