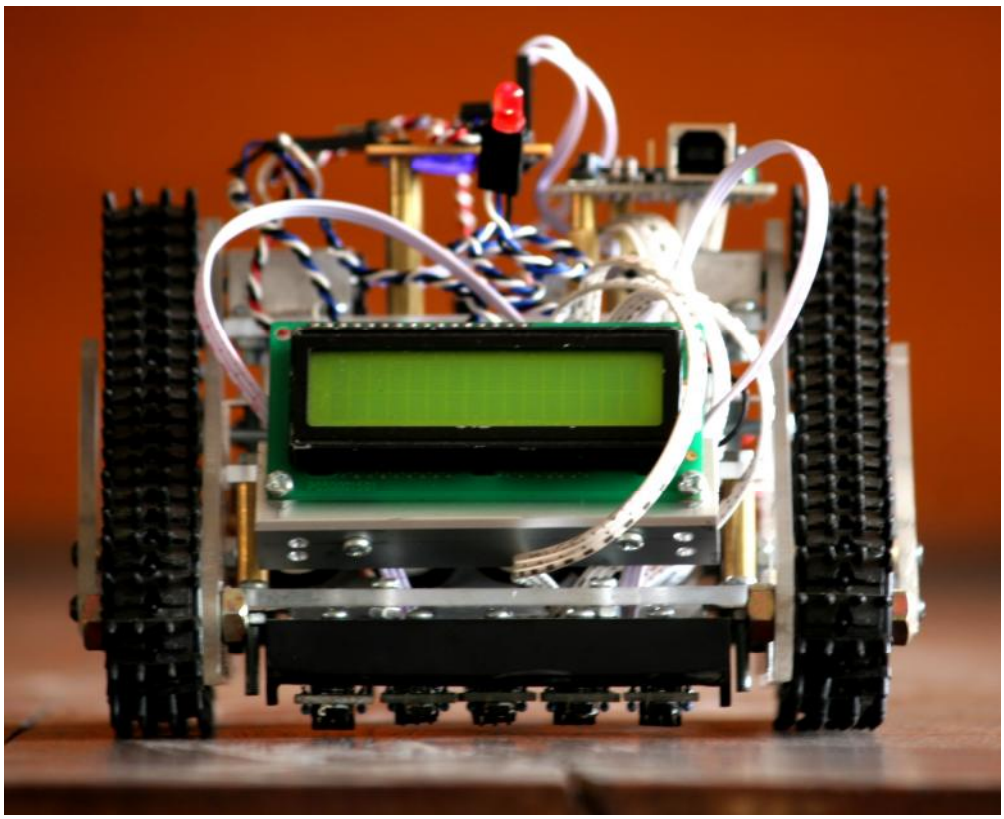
LAMPIRAN A

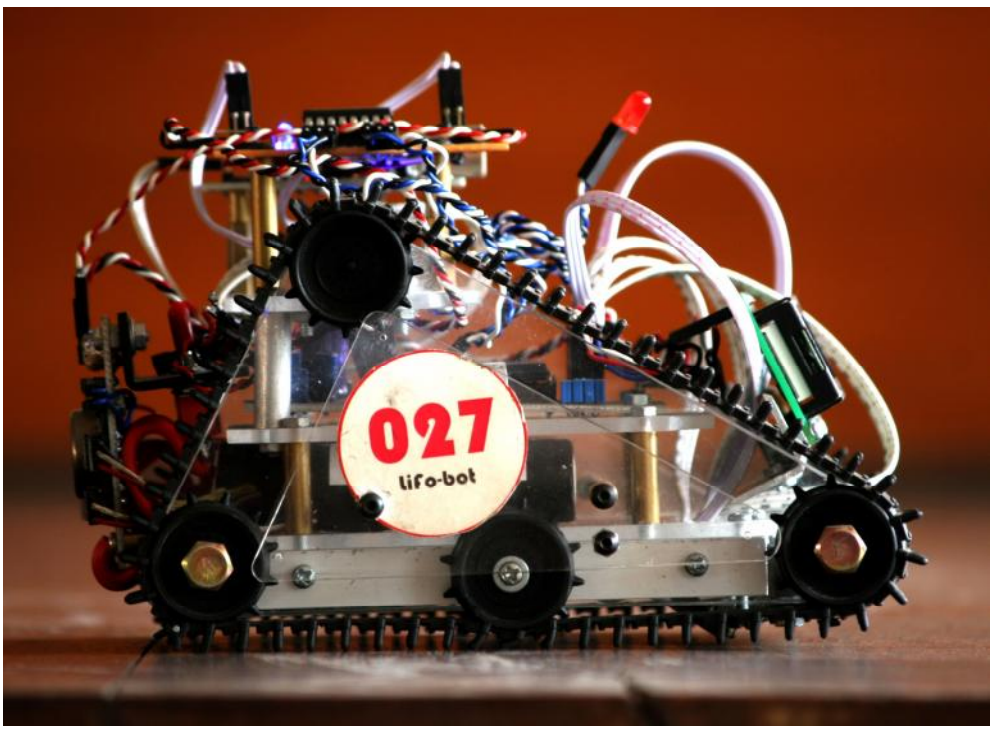
FOTO ROBOT

&

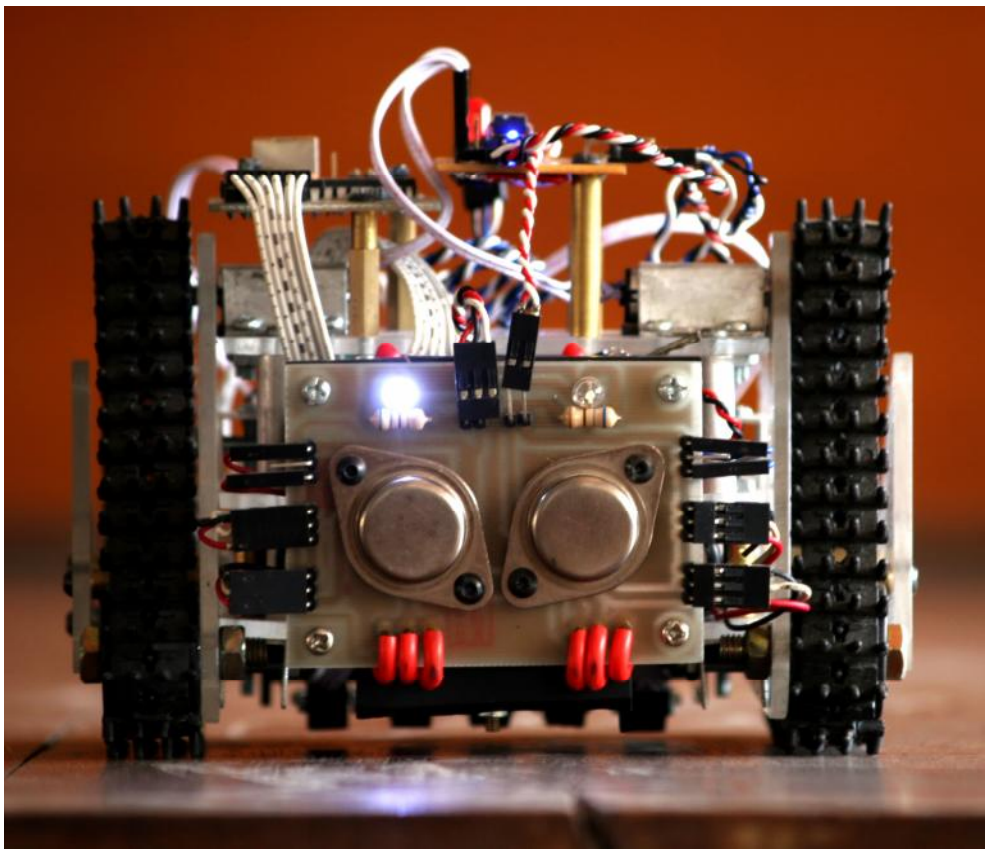
HARDWARE



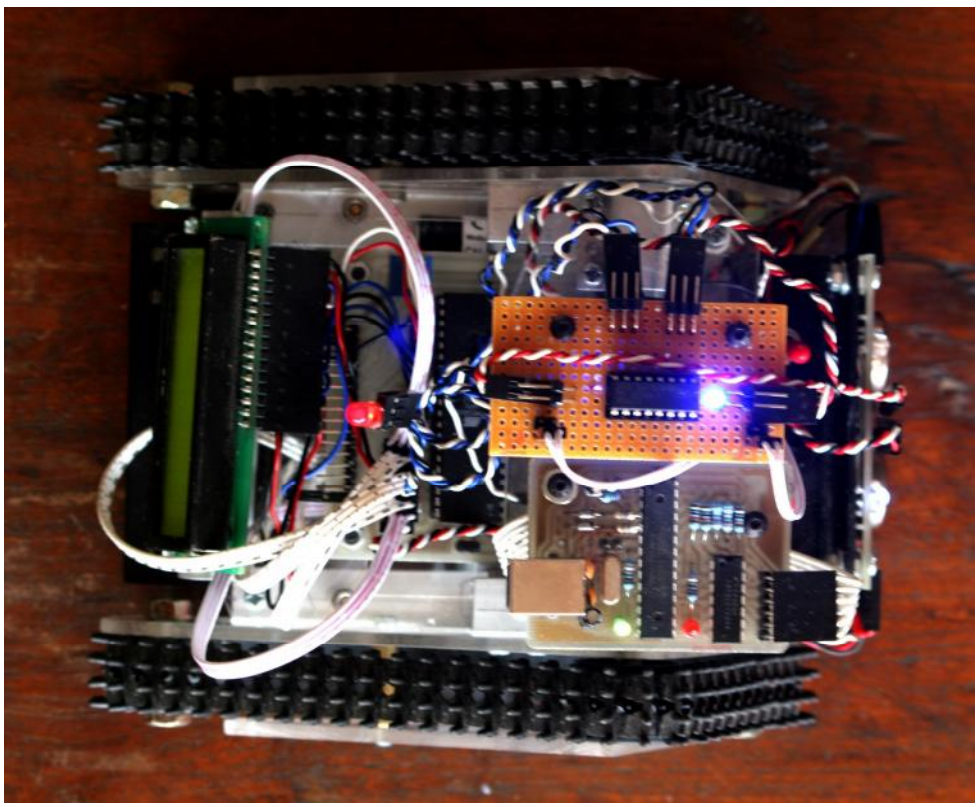
TAMPAK DEPAN – *FRONT VIEW*



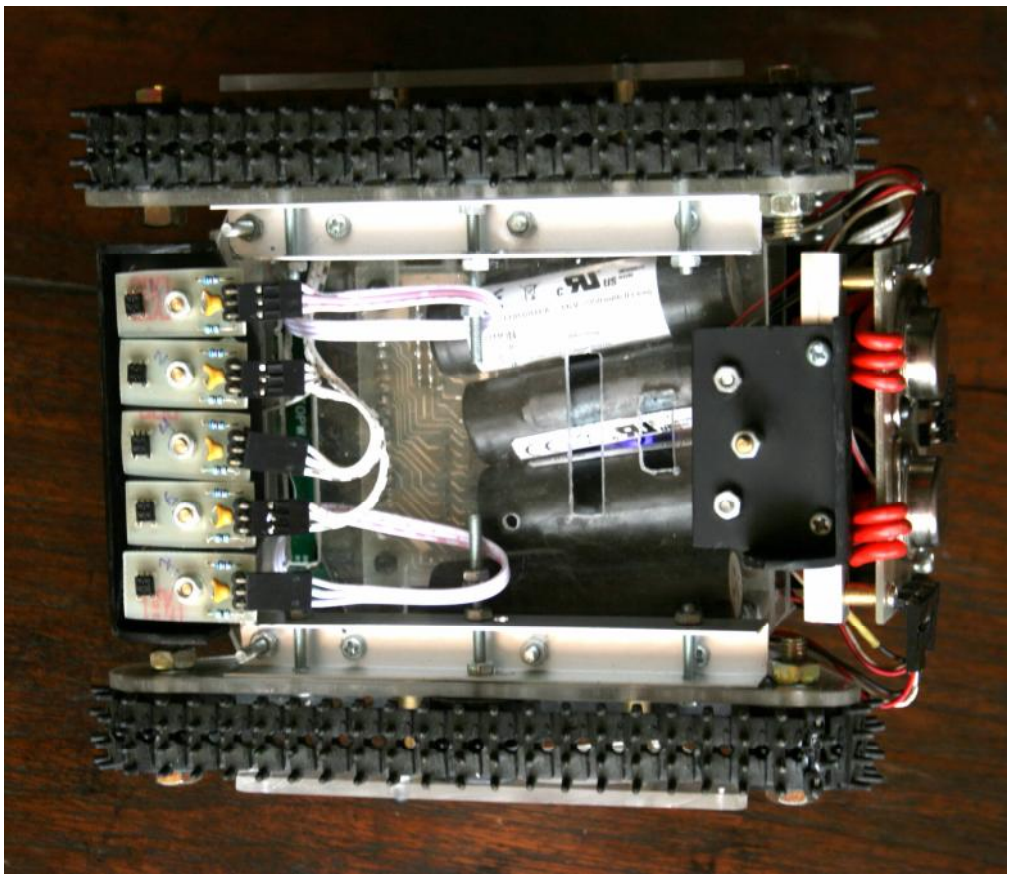
TAMPAK SAMPING KANAN – *RIGHT SIDE VIEW*



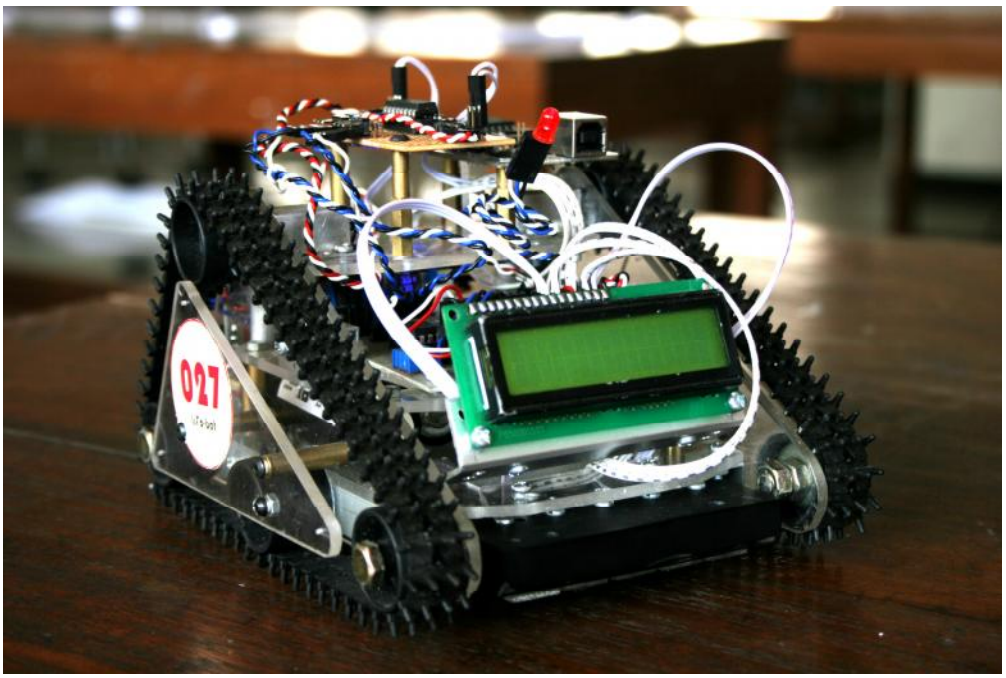
TAMPAK BELAKANG – *BACK VIEW*



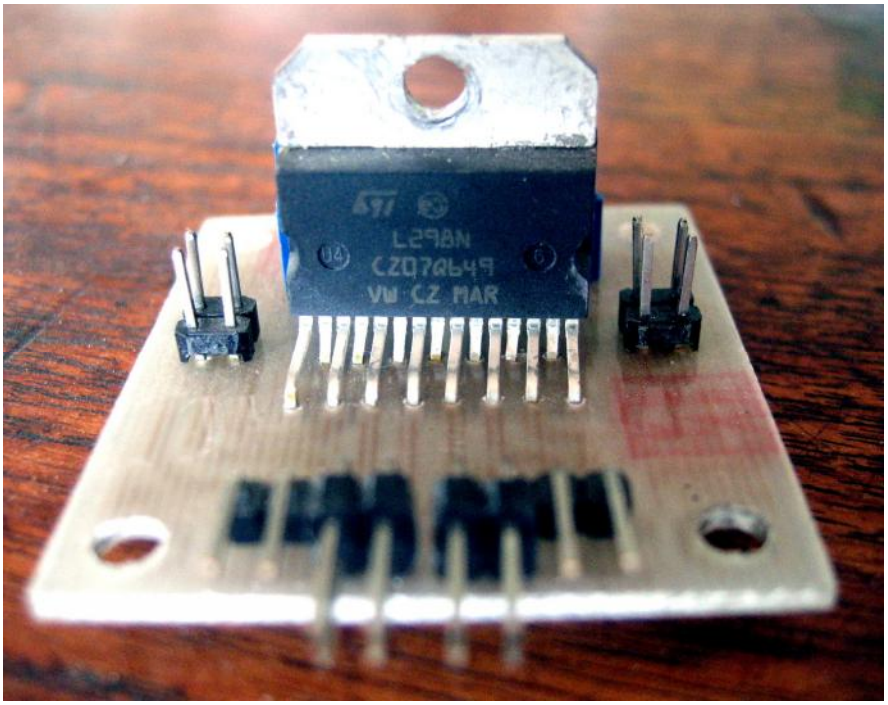
TAMPAK ATAS – *TOP VIEW*



TAMPAK BAWAH – *BOTTOM VIEW*



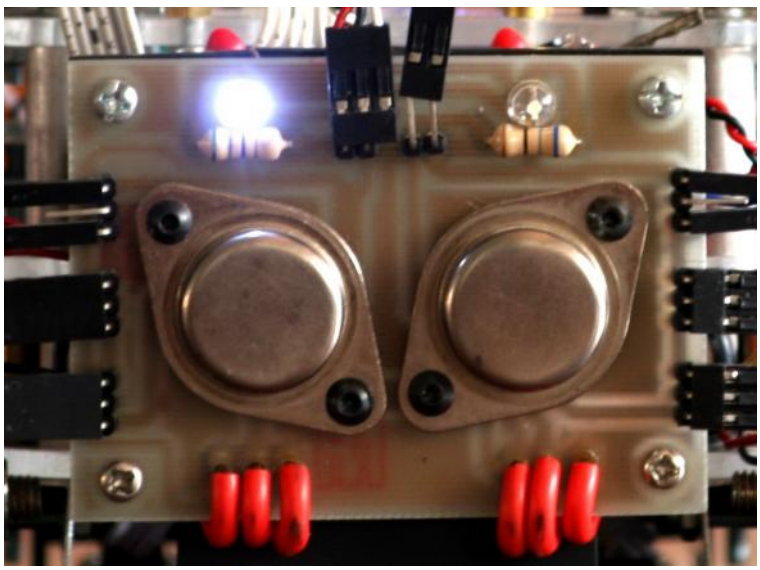
TAMPAK PERSPEKTIF – *PERSPECTIVE VIEW*



Realisasi *Hardware* L298N

Realisasi *Hardware* Sensor Hamamatsu P5587

Susunan Posisi Sensor Hamamatsu P5587



Realisasi *Hardware Regulator Power Supply*

Baterai *Lithium-Ion* 3.6 Volt 2350 mAh

LAMPIRAN B

LISTING PROGRAM

```
/*****************************************************
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    : 2/15/2009
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>
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

unsigned char text[32];
unsigned int a;

// OCR2 = kanan
// OCR0 = kiri

void majukiri(void)
{
OCR0=225;
PORTD.2=0;
PORTD.3=1;
}

void majukanan(void)
```

```
{
OCR2=225;
PORTD.4=0;
PORTD.5=1;
}
void mundurkiri(void)
{
OCR0=225;
PORTD.2=1;
PORTD.3=0;
}
void mundurkanan(void)
{
OCR2=225;
PORTD.4=1;
PORTD.5=0;
}
void maju(void)
{
majukiri();
majukanan();
}
void mundur(void)
{
mundurkiri();
mundurkanan();
}
void belok_kiri(void)
{
majukanan();
mundurkiri();
}
void belok_kanan(void)
{
majukiri();
mundurkanan();
}
void putar_kiri(void)
{
OCR0 = 225;
OCR2 = 50;
majukiri();
majukanan();
```

```
}
void putar_kanan(void)
{
OCR0 = 50;
OCR2 = 225;
majukiri();
majukanan();
}

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

// 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=In Func5=Out Func4=Out Func3=Out Func2=Out Func1=In Func0=In
// State7=0 State6=T State5=0 State4=0 State3=0 State2=0 State1=T State0=T
PORTD=0x00;
DDRD=0xBC;



// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
```

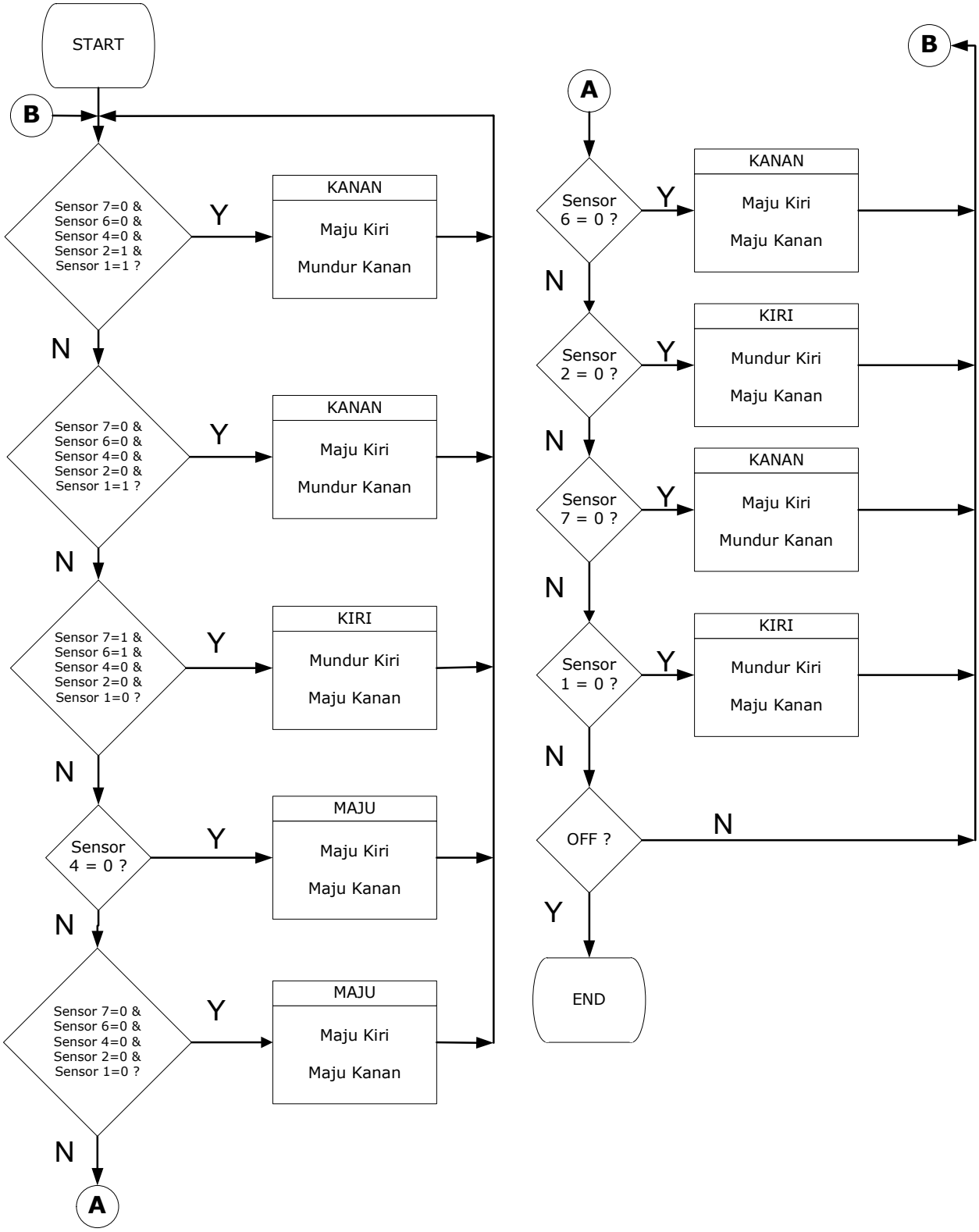
```
// Mode: Phase correct PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x65;
TCNT0=0x00;
OCR0=0x00;

// 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=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;

// LCD module initialization
lcd_init(16);

while (1)
    {
      PORTB = PINA;

    if (PINA.7==0 && PINA.6==0 && PINA.4==0 && PINA.2==1 && PINA.1==1) // kanan hitam
    {
            belok_kanan();
            delay_ms(150);
    }
    if (PINA.7==0 && PINA.6==0 && PINA.4==0 && PINA.2==0 && PINA.1==1) // kanan hitam
    {
            belok_kanan();
            delay_ms(150);
    }
    if (PINA.7==1 && PINA.6==1 && PINA.4==0 && PINA.2==0 && PINA.1==0) // kiri hitam
     {
             belok_kiri();
             delay_ms(150);
     }
    if (PINA.4==0) // tengah hitam
    {
            maju();
    }
    else if (PINA.7==0 && PINA.6 ==0 && PINA.4==0 && PINA.2==0 && PINA.1==0) //
    {
            maju();
    }
     else if (PINA.6==0) // kanan hitam
```

```
    {
        belok_kanan();
        delay_ms(100);
    }

    else if (PINA.2==0) // kiri hitam
    {
        belok_kiri();
        delay_ms(100);
    }
    else if (PINA.7==0) // kanan luar hitam
    {
        belok_kanan();
    }
    else if (PINA.1==0) // kiri luar hitam
    {
        belok_kiri();
    }
    else
    {
        maju();
    }
    };
}
```

LAMPIRAN C

FLOWCHART


START
B
Sensor 7=0 &
Sensor 6=0 &
Sensor 4=0 &
Sensor 2=1 &
Sensor 1=1 ?
Y
KANAN
Maju Kiri
Mundur Kanan
N
Sensor 7=0 &
Sensor 6=0 &
Sensor 4=0 &
Sensor 2=0 &
Sensor 1=1 ?
Y
KANAN
Maju Kiri
Mundur Kanan
N
Sensor 7=1 &
Sensor 6=1 &
Sensor 4=0 &
Sensor 2=0 &
Sensor 1=0 ?
Y
KIRI
Mundur Kiri
Maju Kanan
N
Sensor
4 = 0 ?
Y
MAJU
Maju Kiri
Maju Kanan
N
Sensor 7=0 &
Sensor 6=0 &
Sensor 4=0 &
Sensor 2=0 &
Sensor 1=0 ?
Y
MAJU
Maju Kiri
Maju Kanan
N
A
A
Sensor
6 = 0 ?
Y
KANAN
Maju Kiri
Maju Kanan
N
Sensor
2 = 0 ?
Y
KIRI
Mundur Kiri
Maju Kanan
N
Sensor
7 = 0 ?
Y
KANAN
Maju Kiri
Mundur Kanan
N
Sensor
1 = 0 ?
Y
KIRI
Mundur Kiri
Maju Kanan
N
OFF ?
N
B
Y
END

LAMPIRAN D

DATASHEET