REFERENCES

Anonymous, "Control System Lab: Line Following Robot" University of Engineering& Technology Lahore, viewed on 18 January 2015 at http://www.uet.edu.pk/pp/ee/~mtahir/EE340/Project_document.pdf

Aisyah, S. A. and Ya'umar, "*Penalaan Parameter Kontrol Pid Dengan Metode Heuristic, Aplikasi: Sistem Pengendalian Kecepatan Motor DC"*. Indonesia: Institute Teknologi 10 November, 2012.

Anirudh, S. N., Anirudh, K. R., and Malik,T "Implementation of PID Control to Reduce Wobbling in A Line Following Robot," in *2013 IJRET: International Journal of Research in Engineering and Technology*, 2013, pISSN 2321-7306.

Ardyani, F., "*Sistem Pengendalian Level Cairan Tinta Printer Epson C90 Sebagai Simulasi Pada Industri Percetakan Menggunakan Kontroler PID"*.Thesis. Universitas Brawijaya, 2013.

ChibiOS, ChibiOS Homepage", ChibiOS, viewed on 2nd January 2015 at http://chibios.org/dokuwiki/doku.php

Cook, D., Jet - the ultra-fast line following robot (2006), viewed on 5th December 2014 at www.robotroom.com/Jet.html

Dfrobot, 2014, "Micro Metal Gearmotor", Dfrobot, viewed on 2 January 2015, at http://www.dfrobot.com

Dudek, G., and Jenkin, M, "Computational Principles of Mobile Robotics", Cambridge University Press, 2000 (Chapter 1).

Fahmi, Z, "*MultipleX Robot Line Follower Dengan Kendali PID"*. Indonesia: Institute Teknologi 10 November, 2010.

Kumara. S, "*Sistem Pengendalian Motor"* , Journal Teknik Elektro Universitas Udayana, 2010.

Lee C.S., et all, *"A Hands-on Laboratory for Autonomous Mobile Robot Design Courses,"17th IFAC World Congress (IFAC`08),* Seoul, Korea, (2008).

Lewis, Paul H, and Yang, Chang, *"Basic Control Systems Engineering"*. New Jersey: Prentice Hall,1997.

Meshram, P. M. and Kanojiya, R. G., "Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor," in *2012 International Conference on Advances in Engineering, Science and Management (ICAESM)*, 2012, pp. 117–122.

Mu`izz, M, *"Banquet Serve Autonoous Robot"*. Malaysia: Universiti Tun Hussein Onn Malaysia,2011.

Ogata, K., *"Teknik Kontrol Automatik (Sistem Pengaturan)"*. Jakarta: Erlangga, 1997.

Pakdaman, M., Sanaatiyan, M., and Ghahroudi, M,R, "A line follower robot from design to implementation: Technical issues and problems," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 2010, vol. 1, pp. 5–9.

Philip, C. L. & Harbor, R. D, *"Feedback Control System"*. Prentice Hall. New Jersey: Prentice Hall, 1996.

Pise, S. J., *"ThinkQuest 2010: Proceedings of the First International Conference on Contours of Computing Technology"*. Springer Science & Business Media, 2011

Saidonr, M. S., Desa H., and Rudzuan, M. N., "A differential steering control with proportional controller for an autonomous mobile robot," in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications (CSPA)*, 2011, pp. 90–94.

Su J.H., et all, "An intelligent line-following robot project for introductory robot courses," *World Transactions on Engineering and Technology Education., Vol.8, No.4, 2010.*

Vishay Semiconductor, CNY70 - Reflective optical sensor with transistor output (2008), viewed on 31 December 2014 at www.vishay.com/docs/83751/cny70.pdf

Zakaria, M. F., "*Driver Information System Using On- Board Diagnostiv II Communication Protocol*". Malaysia: Universiti Putra Malaysia, 2007.

Zimmermann, H.-J., "*Fuzzy set theory - and its applications*", 4th Ed., Springer, 2001.

APPENDIX

SOURCE CODE

A1. Bang-bang Controller

```
#define  LEDL  7
#define   LEDM  4
#define  LEDR  8
#define  IRL   1
#define  IRM   2
#define  IRR   3
#define  LEDL_ON  digitalWrite(LEDL,HIGH)
#define  LEDM_ON  digitalWrite(LEDM,HIGH)
#define  LEDR_ON  digitalWrite(LEDR,HIGH)
#define  LEDL_OFF  digitalWrite(LEDL,LOW)
#define  LEDM_OFF  digitalWrite(LEDM,LOW)
#define  LEDR_OFF  digitalWrite(LEDR,LOW)
boolean IRL_STATUS, IRM_STATUS, IRR_STATUS;

double reading1, reading2,reading3;
int y=0;

void go_Forward(uint8_t speed1, uint8_t speed2) {
 speed1=map(speed1,0,100,0,150);
 speed2=map(speed2,0,100,0,150);
 //MOTOR1
 analogWrite(3,0);       //PWM1
 analogWrite(5,speed1);  //PWM2
 //MOTOR2
 analogWrite(6,speed2);  //PWM3
 analogWrite(9,0);       //PWM4
}

void go_Backward(uint8_t speed1, uint8_t speed2) {
 speed1=map(speed1,0,100,0,255);
 speed2=map(speed2,0,100,0,255);
 //MOTOR1
```

```
  analogWrite(3,speed1);  //PWM1
  analogWrite(5,0);       //PWM2
  //MOTOR2
  analogWrite(6,0);       //PWM3
  analogWrite(9,speed2);  //PWM4
 }

void go_Stop() {
  //MOTOR1
  analogWrite(3,0);     //PWM1
  analogWrite(5,0);     //PWM2
  //MOTOR2
  analogWrite(6,0);     //PWM3
  analogWrite(9,0);     //PWM4
}

void setup() {
  // put your setup code here, to run once:
  pinMode(LEDL,OUTPUT);
  pinMode(LEDM,OUTPUT);
  pinMode(LEDR,OUTPUT);
  pinMode(2,INPUT);

  pinMode(10,OUTPUT);pinMode(11,OUTPUT);pinMode(12,OUTPUT); pinMode(13,OUTPUT);
//DEBUG
   Serial.begin(9600);
  while(digitalRead(2)==HIGH)  //Wait for user button
  {
   reading1 = analogRead(IRL);
   reading2 = analogRead(IRM);
   reading3 = analogRead(IRR);

    Serial.print(reading1); Serial.print("\t");Serial.print(reading2);Serial.print("\t");
Serial.print(reading3); Serial.print("\t"); Serial.println(y);
  }
```

```
}

void loop() {
 //READ SENSOR=====================================
 digitalWrite(10,HIGH);
 uint8_t data_IRL = map(analogRead(IRL),0,1023,0,100);
 uint8_t data_IRM = map(analogRead(IRM),0,1023,0,100);
 uint8_t data_IRR = map(analogRead(IRR),0,1023,0,100);
 Serial.print(data_IRL); Serial.print("\t"); Serial.print(data_IRM); Serial.print("\t");
Serial.println(data_IRR);
 digitalWrite(10,LOW);
 digitalWrite(11,HIGH);
if(data_IRL >3) {IRL_STATUS=HIGH; LEDL_ON;} else {IRL_STATUS=LOW; LEDL_OFF;}
 if(data_IRM >3) {IRM_STATUS=HIGH; LEDM_ON;} else {IRM_STATUS=LOW;
LEDM_OFF;}
 if(data_IRR >3) {IRR_STATUS=HIGH; LEDR_ON;} else {IRR_STATUS=LOW; LEDR_OFF;}
 digitalWrite(11,LOW);
 //MOTOR CONTROL=====================================
 digitalWrite(12,HIGH);

if((!IRL_STATUS&&IRM_STATUS&&!IRR_STATUS)||(!IRL_STATUS&&IRM_STATUS&&!IR
R_STATUS)) go_Forward(100,70);  //STRAIGHT
 else if(IRL_STATUS&&IRM_STATUS&&!IRR_STATUS) go_Forward(80,100);  //MOVE LEFT
 else if(IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS) go_Forward(0,100);  //MOVE
HEAVY LEFT
 else if(!IRL_STATUS&&IRM_STATUS&&IRR_STATUS) go_Forward(100,80);  //MOVE
RIGHT
 else if(!IRL_STATUS&&!IRM_STATUS&&IRR_STATUS) go_Forward(100,0);  //MOVE
HEAVY RIGHT
 if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)
{ go_Stop();


} //STOP


digitalWrite(12,LOW);
}
```

A2. PID Controller Digital Concept Sensors

```
#define  LEDL  7
#define  LEDM  4
#define  LEDR  8
#define  LEDL_ON  digitalWrite(LEDL,HIGH)
#define  LEDM_ON  digitalWrite(LEDM,HIGH)
#define  LEDR_ON  digitalWrite(LEDR,HIGH)
#define  LEDL_OFF  digitalWrite(LEDL,LOW)
#define  LEDM_OFF  digitalWrite(LEDM,LOW)
#define  LEDR_OFF  digitalWrite(LEDR,LOW)

const int motorkiri = 3;
const int motorkanan = 9;
const int motorkiri2 = 5;
const int motorkanan2 = 6;
const int kiri     = A1;
const int tengah   = A2;
const int kanan    = A3;

int PV = 0;
int error = 0;
int setPoint = 0;

boolean IRL_STATUS, IRM_STATUS, IRR_STATUS,steady,left,stoped;
double reading1,reading2,
reading3,dError,sError,lError,dTime,PID,errorTop,errorBot,kP,kI,kD,y2,y1;

long pwmMotor;
unsigned long now,lTime;

void setup ()
{
        pinMode(motorkiri,OUTPUT);
        pinMode(motorkanan,OUTPUT);
```

```
    pinMode(LEDL,OUTPUT);
    pinMode(LEDM,OUTPUT);
    pinMode(LEDR,OUTPUT);

      pinMode(kiri,INPUT);
      pinMode(tengah,INPUT);
      pinMode(kanan,INPUT);
    pinMode(2,INPUT);

    pinMode(10,OUTPUT);pinMode(11,OUTPUT);pinMode(12,OUTPUT);
pinMode(13,OUTPUT); //DEBUG

    Serial.begin(9600);
    while(digitalRead(2)==HIGH)  //Wait for user button
    {
     Stop();
     reading1 = analogRead (kiri);
     reading2 = analogRead (tengah);
     reading3 = analogRead (kanan);
     Serial.print(reading1); Serial.print("\t"); Serial.print(reading2); Serial.print("\t");
Serial.println(reading3);
   LEDL_OFF; LEDM_OFF; LEDR_OFF;
  delay(100);
  LEDL_ON; LEDM_ON; LEDR_ON;
  delay(100);
  LEDL_OFF; LEDM_OFF; LEDR_ON;
  delay(100);
  LEDL_OFF; LEDM_ON; LEDR_OFF;
  delay(100);
  LEDL_ON; LEDM_OFF; LEDR_OFF;
  delay(100);
  LEDL_ON; LEDM_ON; LEDR_ON;
  delay(100);
  LEDL_OFF; LEDM_OFF; LEDR_OFF;
  delay(100);
   LEDL_ON; LEDM_ON; LEDR_ON;
  delay(100);
```

```
  LEDL_ON; LEDM_OFF; LEDR_OFF;
  delay(100);
  LEDL_OFF; LEDM_ON; LEDR_OFF;
  delay (100);
   LEDL_OFF; LEDM_OFF; LEDR_ON;
  delay(100);
   LEDL_ON; LEDM_ON; LEDR_ON;
   delay(100);
    }

          error = 0;
          dError = 0;
          sError = 0;
          lError = 0;
          dTime = 0;
          PID = 0;
          steady = false;
          left = false;
*****************************************************************************
                                parameter  set
*****************************************************************************
          setPoint = 0;
          kP = 0.0433;
          kI = 0.01;
          kD = 0.0473;

}
void loop ()
{
 digitalWrite(10,HIGH);

    Serial.print(PV);
    Serial.print("\n");
 readsensor ();
          compute ();
  if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)
```

```
  {

  Stop ();
  }
  else {


      if (steady){
            motor_lurus ();
      }

    else  {
            if (left){
                  setting_motor (1,pwmMotor);

            }
            else{
                  setting_motor (2,pwmMotor);


            }
      }
  }
  digitalWrite(10,LOW);
delay(10);
 Serial.print(PV);
}
void Stop ()
{
      analogWrite (motorkiri, 0);
      analogWrite (motorkiri2, 0);
            analogWrite (motorkanan, 0);
      analogWrite (motorkanan2, 0);
}
void motor_lurus ()
{
      analogWrite (motorkiri, 0);
```

```
        analogWrite (motorkiri2, 150);
                analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 150);
}
void setting_motor (int motor,int pwm)
{
  digitalWrite(11,HIGH);

        if (motor==1){

        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 150);
                analogWrite (motorkanan, 0);
       analogWrite (motorkanan2, pwm);

        // analogWrite (motorkiri, 0);
       // analogWrite (motorkiri2, pwm);
                //analogWrite (motorkanan, pwm);
         //analogWrite (motorkanan2, 0);
        }
        if (motor==2){

        analogWrite (motorkiri,0);
        analogWrite (motorkiri2, pwm );
                analogWrite (motorkanan, 0);
       analogWrite (motorkanan2, 150);

        //analogWrite (motorkiri, pwm);
       // analogWrite (motorkiri2, 0 );
                //analogWrite (motorkanan, 0);
       // analogWrite (motorkanan2, pwm);

        }
        digitalWrite(11,LOW);
}
```

```
void readsensor()
{
 digitalWrite(12,HIGH);
        //data sensors read
        reading1 = analogRead (kiri);
     reading2 = analogRead (tengah);
     reading3 = analogRead (kanan);

        if(reading1 >30) {IRL_STATUS=HIGH; LEDL_ON;} else {IRL_STATUS=LOW;
LEDL_OFF;}
     if(reading2 >30) {IRM_STATUS=HIGH; LEDM_ON;} else {IRM_STATUS=LOW;
LEDM_OFF;}
    if(reading3 >30) {IRR_STATUS=HIGH; LEDR_ON;} else {IRR_STATUS=LOW;
LEDR_OFF;}

        if((IRL_STATUS&&IRM_STATUS&&IRR_STATUS)||(!IRL_STATUS&&IRM_STATU
S&&!IRR_STATUS)) PV = 0;  //STRAIGHT
      else if(IRL_STATUS&&IRM_STATUS&&!IRR_STATUS)  PV = 1;
      else if(IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS) PV = 2;
      else if(!IRL_STATUS&&IRM_STATUS&&IRR_STATUS)  PV =-1;
     else if(!IRL_STATUS&&!IRM_STATUS&&IRR_STATUS) PV =-2;
      if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)  stoped=true;

 digitalWrite(12,LOW);

}
void compute ()
{
 digitalWrite(13,HIGH);
        now = millis();
        dTime =(double) (now-lTime);

     //calculate error
     error = setPoint-PV;

     //condition now
```

```
if (error>setPoint){
        steady = false;
        left = true;

    }
    else if (error<setPoint){
        steady = false;
        left = false;

    }
    else {

        steady =true;

    }

//calculation PID signal
    sError =(sError+error);
    dError = (error-lError);
    PID = (kP*error)+((kI*sError)*(dTime/500))+((kD*dError)/(dTime/500));


if (PID>150)
        pwmMotor= 150;
    else if (PID<-150)
        pwmMotor = 150;
    else{
        if (PID>0)
            pwmMotor = (int) PID;
        else
            pwmMotor = (int)(PID*(-1));
    }

    lError = error;
    lTime = now;
digitalWrite(13,LOW);
}
```

A3. PID Controller Analog Concept Sensors

```
#define  LEDL  7
#define  LEDM  4
#define  LEDR  8
#define  LEDL_ON  digitalWrite(LEDL,HIGH)
#define  LEDM_ON  digitalWrite(LEDM,HIGH)
#define  LEDR_ON  digitalWrite(LEDR,HIGH)
#define  LEDL_OFF  digitalWrite(LEDL,LOW)
#define  LEDM_OFF  digitalWrite(LEDM,LOW)
#define  LEDR_OFF  digitalWrite(LEDR,LOW)


const int motorkiri = 3;
const int motorkanan = 9;
const int motorkiri2 = 5;
const int motorkanan2 = 6;
const int kiri     = A1;
const int tengah   = A2;
const int kanan    = A3;


int PV = 0;
int error = 0;
int setPoint = 0;

boolean IRL_STATUS, IRM_STATUS, IRR_STATUS,steady,left,stoped;
double reading1,reading2, reading3,dError,sError,lError,dTime,PID,errorTop,errorBot,kP,kI,kD;

float y1,y2,x11,x12,x21,x22,z1,z2,zhasil1,zhasil2;

long pwmMotor;
unsigned long now,lTime;

void setup ()
```

```
{
        pinMode(motorkiri,OUTPUT);
        pinMode(motorkanan,OUTPUT);
      pinMode(LEDL,OUTPUT);
      pinMode(LEDM,OUTPUT);
      pinMode(LEDR,OUTPUT);

        pinMode(kiri,INPUT);
        pinMode(tengah,INPUT);
        pinMode(kanan,INPUT);
      pinMode(2,INPUT);

      pinMode(10,OUTPUT);pinMode(11,OUTPUT);pinMode(12,OUTPUT);
pinMode(13,OUTPUT); //DEBUG

      Serial.begin(9600);
      while(digitalRead(2)==HIGH)  //Wait for user button
      {
       Stop();
       reading1 = analogRead (kiri);
       reading2 = analogRead (tengah);
       reading3 = analogRead (kanan);
       Serial.print(reading1); Serial.print("\t"); Serial.print(reading2); Serial.print("\t");
Serial.println(reading3);
      }
        error = 0;
        dError = 0;
        sError = 0;
        lError = 0;
        dTime = 0;
        PID = 0;
        steady = false;
        left = false;
*******************************************************************************
                                parameter  set
*******************************************************************************
        setPoint = 0;
```

```
        kP = 0.0433;
        kI = 0.01;
        kD = 0.0473;


}
void loop ()
{
 digitalWrite(10,HIGH);

 readsensor ();
        compute ();
 if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)
 {

  Stop ();
  }
  else {


        if (steady){
                motor_lurus ();
        }


    else  {
                if (left){
                        setting_motor (1,pwmMotor);


                }
                else{
                        setting_motor (2,pwmMotor);


                }
        }
  }
  digitalWrite(10,LOW);
```

```
delay(10);
}
void Stop ()
{
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 0);
                analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 0);
}
void motor_lurus ()
{
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 255);
                analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 255);
}
void setting_motor (int motor,int pwm)
{
 digitalWrite(11,HIGH);

        if (motor==1){

        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 255);
                analogWrite (motorkanan, 0);
       analogWrite (motorkanan2, pwm);

        // analogWrite (motorkiri, 0);
       // analogWrite (motorkiri2, pwm);
                //analogWrite (motorkanan, pwm);
        //analogWrite (motorkanan2, 0);
        }
        if (motor==2){

        analogWrite (motorkiri,0);
        analogWrite (motorkiri2, pwm );
```

```
                analogWrite (motorkanan, 0);
            analogWrite (motorkanan2, 255);

             //analogWrite (motorkiri, pwm);
            // analogWrite (motorkiri2, 0 );
                    //analogWrite (motorkanan, 0);
            // analogWrite (motorkanan2, pwm);

            }
            digitalWrite(11,LOW);
}

void readsensor()
{
 digitalWrite(12,HIGH);
            //data sensors read
            reading1 = analogRead (kiri);
       reading2 = analogRead (tengah);
       reading3 = analogRead (kanan);

            if(reading1 >30) {IRL_STATUS=HIGH; LEDL_ON;} else {IRL_STATUS=LOW;
LEDL_OFF;}
       if(reading2 >30) {IRM_STATUS=HIGH; LEDM_ON;} else {IRM_STATUS=LOW;
LEDM_OFF;}
      if(reading3 >30) {IRR_STATUS=HIGH; LEDR_ON;} else {IRR_STATUS=LOW;
LEDR_OFF;}

            if((IRL_STATUS&&IRM_STATUS&&IRR_STATUS)||(!IRL_STATUS&&IRM_STATU
S&&!IRR_STATUS)) PV = 0;  //STRAIGHT
       else if(IRL_STATUS&&IRM_STATUS&&!IRR_STATUS)  PV = 1;
       else if(IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS) PV = 2;
       else if(!IRL_STATUS&&IRM_STATUS&&IRR_STATUS)  PV =-1;
      else if(!IRL_STATUS&&!IRM_STATUS&&IRR_STATUS) PV =-2;
       if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)  stoped=true;

       {
```

```
 if (reading1 <30) reading1 = 30;
 else if (reading1 >180) reading1=180;
 else  {reading1=reading1;}

 if (reading2 <30) reading2 = 30;
 else if (reading2 >180) reading2=180;
 else  {reading2=reading2;}

 if (reading3 <30) reading3 = 30;
 else if (reading3 >180) reading3=180;
 else  {reading3=reading3;}
 }

 if ((reading1=30)&&(reading3>30))
 {

   y1 = reading2-reading1;
   z1= 2302+(34.12*(120+y1));
   zhasil1 = sqrt(z1);
   x11= (47.98+z1)/17.06;
   x12= (47.98-z1)/17.06;


}
else if ((reading3=30)&&(reading1>30))
 {

   y2 = reading2-reading3;
   z2= 2302+(34.12*(120+y2));
   zhasil2 = sqrt(z2);
   x21= (47.98+z1)/17.06;
   x22= (47.98-z1)/17.06;

}
 digitalWrite(12,LOW);
```

```
}

void compute ()
{
 digitalWrite(13,HIGH);
      now = millis();
      dTime =(double) (now-lTime);

    //calculate error
    error = setPoint-PV;

    //condition now
    if (error>setPoint){
            steady = false;
            left = true;

    }
    else if (error<setPoint){
            steady = false;
            left = false;

    }
    else {

            steady =true;

    }

    //calculation PID signal
      sError =(sError+error);
      dError = (error-lError);
      PID = (kP*error)+((kI*sError)*(dTime/500))+((kD*dError)/(dTime/500));

    if (PID>255)
            pwmMotor= 255;
```

```
        else if (PID<-255)
                pwmMotor = 255;
        else{
                if (PID>0)
                        pwmMotor = (int) PID;
                else
                        pwmMotor = (int)(PID*(-1));
        }
      Serial.print(PID);
       lError = error;
        lTime = now;
Serial.print("\t");
      Serial.print("PID=(");
      Serial.print(PID);
      Serial.print(")");
Serial.print("\t");
       Serial.print("PWM=(");
      Serial.print(pwmMotor);
      Serial.print(")");
      Serial.print("\n");

digitalWrite(13,LOW);
}
```

A4. PID ChibiOS Real-Time (Digital Concept Sensors)

```
#include <ChibiOS_AVR.h>

#define  LEDL       7
#define  LEDM       4
#define  LEDR       8
#define  LEDL_ON     digitalWrite(LEDL,HIGH)
#define  LEDM_ON     digitalWrite(LEDM,HIGH)
#define  LEDR_ON     digitalWrite(LEDR,HIGH)
#define  LEDL_OFF    digitalWrite(LEDL,LOW)
#define  LEDM_OFF    digitalWrite(LEDM,LOW)
#define  LEDR_OFF    digitalWrite(LEDR,LOW)
#define  motorkiri   3
#define  motorkanan  9
#define  motorkiri2  5
#define  motorkanan2 6
#define  kiri        A1
#define  tengah      A2
#define  kanan       A3


int PV = 0;
int error = 0;
int setPoint = 0;

boolean IRL_STATUS, IRM_STATUS, IRR_STATUS,steady,left,stoped;
double reading1,reading2,
reading3,dError,sError,lError,dTime,PID,errorTop,errorBot,kP,kI,kD,y2,y1;

long pwmMotor;
unsigned long now,lTime;


void Stop ()
{
```

```
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 0);
                analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 0);
}
void motor_lurus ()
{
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 150);
                analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 150);
}
void setting_motor (int motor,int pwm)
{
      if (motor==1){

       analogWrite (motorkiri, 0);
       analogWrite (motorkiri2, 150);
                analogWrite (motorkanan, 0);
      analogWrite (motorkanan2, pwm);

       // analogWrite (motorkiri, 0);
      // analogWrite (motorkiri2, pwm);
                //analogWrite (motorkanan, pwm);
       //analogWrite (motorkanan2, 0);
      }
      if (motor==2){

       analogWrite (motorkiri,0);
       analogWrite (motorkiri2, pwm );
                analogWrite (motorkanan, 0);
      analogWrite (motorkanan2, 150);

       //analogWrite (motorkiri, pwm);
      // analogWrite (motorkiri2, 0 );
                //analogWrite (motorkanan, 0);
```

```
        // analogWrite (motorkanan2, pwm);

        }

}

void readsensor()
{
        //data sensors read
        reading1 = analogRead (kiri);
      reading2 = analogRead (tengah);
      reading3 = analogRead (kanan);

        if(reading1 >30) {IRL_STATUS=HIGH; } else {IRL_STATUS=LOW; }
     if(reading2 >30) {IRM_STATUS=HIGH; } else {IRM_STATUS=LOW; }
    if(reading3 >30) {IRR_STATUS=HIGH; } else {IRR_STATUS=LOW; }


        if((IRL_STATUS&&IRM_STATUS&&IRR_STATUS)||(!IRL_STATUS&&IRM_STATU
S&&!IRR_STATUS)) PV = 0;  //STRAIGHT
      else if(IRL_STATUS&&IRM_STATUS&&!IRR_STATUS)  PV = 1;
      else if(IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS) PV = 2;
      else if(!IRL_STATUS&&IRM_STATUS&&IRR_STATUS)  PV =-1;
     else if(!IRL_STATUS&&!IRM_STATUS&&IRR_STATUS) PV =-2;
      if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)  stoped=true;



}

void compute ()
{
        now = millis();
        dTime =(double) (now-lTime);
```

```
//calculate error
error = setPoint-PV;

//condition now
if (error>setPoint){
        steady = false;
        left = true;

    }
    else if (error<setPoint){
        steady = false;
        left = false;

    }
    else {

        steady =true;

    }

//calculation PID signal
    sError =(sError+error);
    dError = (error-lError);
    PID = (kP*error)+((kI*sError)*(dTime/1000))+((kD*dError)/(dTime/1000));

if (PID>255)
        pwmMotor= 255;
    else if (PID<-255)
        pwmMotor = 255;
    else{
        if (PID>0)
            pwmMotor = (int) PID;
        else
            pwmMotor = (int)(PID*(-1));
    }
```

```
    lError = error;
        lTime = now;


}


//-------------------------------------------------------------------
//TASK1: READ LINE SENSOR
static WORKING_AREA(waTask1, 64);  //SIZE OF WORKING MEMORY AREA

static msg_t Task1(void *arg) {
  while (1) {
    digitalWrite(10,HIGH);
    Serial.print("\n");
    Serial.print(PV);
    Serial.print("\n");

    readsensor ();
    digitalWrite(10,LOW);
    Serial.print(PV);
    chThdSleepMilliseconds(1);  //DELAY 1ms
  }
  return 0;
}
//---------------------------------------------------------------------------
//TASK2: PID CALCULATION
static WORKING_AREA(waTask2, 128);

static msg_t Task2(void *arg) {
  while (1) {
    digitalWrite(11,HIGH);
    compute ();
```

```
  digitalWrite(11,LOW);
  chThdSleepMilliseconds(10);
 }
 return 0;
}
//------------------------------------------------------------------------
//TASK3: ROBOT MOTION
static WORKING_AREA(waTask3, 128);

static msg_t Task3(void *arg) {
 while (1) {
  digitalWrite(12,HIGH);
  if(!IRL_STATUS&&!IRM_STATUS&&!IRR_STATUS)  Stop ();
  else{
      if (steady) motor_lurus ();
      else{
           if (left) setting_motor (1,pwmMotor);
              else setting_motor (2,pwmMotor);
             }
     }
  digitalWrite(12,LOW);
  chThdSleepMilliseconds(1);
 }
 return 0;
}

//------------------------------------------------------------------------
//TASK4: LIVE LED
static WORKING_AREA(waTask4, 64);

static msg_t Task4(void *arg) {
 while (1) {
  digitalWrite(13,HIGH);
  LEDL_OFF; LEDM_OFF; LEDR_OFF;
  chThdSleepMilliseconds(300);
  LEDL_ON; LEDM_ON; LEDR_ON;
```

```
  digitalWrite(13,LOW);
  chThdSleepMilliseconds(300);
 }
 return 0;
}

//------------------------------------------------------------------------
void setup() {

 pinMode(10,OUTPUT);pinMode(11,OUTPUT);pinMode(12,OUTPUT); pinMode(13,OUTPUT);
//DEBUG
        pinMode(motorkiri,OUTPUT);
        pinMode(motorkanan,OUTPUT);
    pinMode(LEDL,OUTPUT);
    pinMode(LEDM,OUTPUT);
    pinMode(LEDR,OUTPUT);

        pinMode(kiri,INPUT);
        pinMode(tengah,INPUT);
        pinMode(kanan,INPUT);
    pinMode(2,INPUT);

    Serial.begin(9600);
    while(digitalRead(2)==HIGH)  //Wait for user button
    {
     Stop();
     LEDL_ON; LEDM_ON; LEDR_ON;
    }

        error = 0;
        dError = 0;
        sError = 0;
        lError = 0;
        dTime = 0;
        PID = 0;
        steady = false;
```

```
        left = false;
**************************************************************************
                        parameter  set
**************************************************************************
        setPoint = 0;
        kP = 0.0433;
        kI = 0.01;
        kD = 0.0473;

 chBegin(chSetup);
 // chBegin never returns, main thread continues with mainThread()
 while(1);
}

//------------------------------------------------------------------------
// main thread runs at NORMALPRIO
void chSetup() {

 // start blink thread
 chThdCreateStatic(waTask1, sizeof(waTask1),
  NORMALPRIO + 3, Task1, NULL);

 chThdCreateStatic(waTask2, sizeof(waTask2),
  NORMALPRIO + 2, Task2, NULL);

 chThdCreateStatic(waTask3, sizeof(waTask3),
  NORMALPRIO + 1, Task3, NULL);

 chThdCreateStatic(waTask4, sizeof(waTask4),
  NORMALPRIO , Task4, NULL);
}
//------------------------------------------------------------------------
void loop() {
 // not used
}
```