

```

#include<NewPing.h>
#define TRIGGER_Pin 2
#define ECHO_Pin 3
#define ena 10
#define inA 9
#define inB 8
#define inC 7
#define inD 6
#define enb 5
#define MAX_DISTANCE 20
//NewPing sonar(TRIGGER_Pin, ECHO_Pin, MAX_DISTANCE);

const int irPins[5] = {A0,A1,A2,A3,A4};

void wheel(int lm, int rm);
int s[5] = {0,0,0,0,0};
int i, lastSensor,lastError;
int uturn = -100;
int base_L= -120;
int base_R=-120;
float kp=3.1416;
float kd=1.2255;

void setup()
{
  //Serial.begin(9600);
  mot_init();
  other_init();
  ustad_samne_plastic();
}
void loop()
{
  //void stop();
  line_follow();
  //wheel(200,200);

  //unsigned int distance = sonar.ping_cm();

  //Serial.print("Distance: ");
  // Serial.print(distance);
  // Serial.println(" cm");

  /*if (distance < MAX_DISTANCE) {

```

```

    ustad_samne_plastic();
    Serial.println("Obstacle detected! Motors stopped.");
} else {

    line_follow();
}*/

    line_follow();
};

```

```

void other_init()
{
    lastSensor=0;
    lastError=0;
    //Serial.begin(9600);
}

```

```

void mot_init()
{
    pinMode(inA,OUTPUT);
    pinMode(inB,OUTPUT);
    pinMode(inC,OUTPUT);
    pinMode(inD,OUTPUT);
    pinMode(ena,OUTPUT);
    pinMode(enb,OUTPUT);

}

```

```

void wheel(int lm, int rm)
{
    if(lm==0)
    {

        digitalWrite(inC,HIGH);
        digitalWrite(inD,HIGH);

    }
    if(lm>0)
    {
        digitalWrite(inC,HIGH);

```

```

    digitalWrite(inD,LOW);

}
else if(lm<0)
{
    digitalWrite(inC,LOW);
    digitalWrite(inD,HIGH);
}
if(rm==0)
{
    digitalWrite(inA,HIGH);
    digitalWrite(inB,HIGH);
}
if(rm>0)
{
    digitalWrite(inA,HIGH);
    digitalWrite(inB,LOW);
}
else if(rm<0)
{
    digitalWrite(inA,LOW);
    digitalWrite(inB,HIGH);
}
if(lm>254) lm=254;
if(lm<-254) lm=-254;
if(rm>254) rm=254;
if(rm<-254) rm=-254;

analogWrite(ena,abs(rm));
analogWrite(enb,abs(lm));

}

int readSensor()
{

    s[0]=digitalRead(irPins[0]);
    s[1]=digitalRead(irPins[1]);
    s[2]=digitalRead(irPins[2]);
    s[3]=digitalRead(irPins[3]);
    s[4]=digitalRead(irPins[4]);
    s[0]=1-s[0];
    s[1]=1-s[1];
    s[2]=1-s[2];

```

```

    s[3]=1-s[3];
    s[4]=1-s[4];
    int error,sum;
    sum=s[0]+s[1]+s[2]+s[3]+s[4];
    if(sum!=0)
    {
        error=(s[0]*10+s[1]*20+s[2]*30+s[3]*40+s[4]*50)/sum - 45; //set point 45 is for 10 array
        sensor. Make sure to change the setpoint according to your sensor size.
    }
    else
    {
        error=420;
    }

    if(s[0]==1) lastSensor=1;

    else if(s[4]==1) lastSensor=2;
    //Serial.print(error);
    //Serial.print(" ");
    return error;
}

void ustad_samne_plastic()
{

    digitalWrite(inA, LOW);
    digitalWrite(inB, LOW);
    digitalWrite(inC, LOW);
    digitalWrite(inD, LOW);

    analogWrite(ena, -255);
    analogWrite(ena, 255);
}

void line_follow()
{
    int error,corr;
    float p,d;
    error=readSensor();
    if(error==420)
    {
        if(lastSensor==1) wheel(-uturn,uturn);
        else if(lastSensor==2) wheel(uturn,-uturn);
    }
    else

```

```
{  
  p=kp*error;  
  d=kd*(error-lastError);  
  corr=p+d;  
  // Serial.println(corr);  
  wheel(base_L+corr,base_R-corr);  
  if((error-lastError)!=0) delay(5);  
  lastError=error;  
}  
}
```