```
#include <QTRSensors.h>

void calculateIntegral();
void calculateProportional();
void readValues();

#define NUM_SENSORS 8
#define TIMEOUT 2500
#define EMITTER_PIN 0
#define avgSpeed 255

int time = 0;

int pwmA = 3;
int pwmB = 11;
int dirA = 12;
int dirB = 13;

int kp = 1;
int kd = 1;
int ki = 1;

int error = 0;
int lastError = 0;

int proportional = 0;
int derivative = 0;
int integral = 0;

QTRSensorsRC qtrrc((unsigned char[]) {2, 4, 5, 6, 7, 8, 9, 10},
  NUM_SENSORS, TIMEOUT, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];


void setup(){
  Serial.begin(9600);

  pinMode(pwmA, OUTPUT);
  pinMode(pwmB, OUTPUT);
  pinMode(dirA, OUTPUT);
  pinMode(dirB, OUTPUT);

  pinMode(1, OUTPUT);
  for (int i=0; i<5; i++){
      digitalWrite(1, HIGH);
      delay(50);
      digitalWrite(1, LOW);
      delay(950);
    }

  analogWrite(pwmA, avgSpeed);
  analogWrite(pwmB, avgSpeed);
```

```
}

void loop(){

  readValues();
  calculateProportional();

  derivative = error-lastError;
  integral += proportional;

  if(integral > 255){
    integral=255;
  };
  if(integral < -255){
    integral=-255;
  };

  int turn = proportional*kp + derivative*kd + integral*ki;

  if(turn>=255)
    turn=255;
  if(turn<=-255)
    turn=-255;

  int speedA=0;
  int speedB=0;

  if(turn>=0){
    speedA=avgSpeed;
    speedB=avgSpeed-turn;
  }
  else{
    speedA=avgSpeed+turn;
    speedB=avgSpeed;
  }

  Serial.print("P=");
  Serial.print(proportional);
  Serial.print('\t');
  Serial.print("I=");
  Serial.print(integral);
  Serial.print('\t');
  Serial.print("D=");
  Serial.print(derivative);
  Serial.print('\t');
  Serial.print("Turn=");
  Serial.print(turn);
  Serial.print('\t');
```

```
  Serial.print("speedA=");
  Serial.print(speedA);
  Serial.print('\t');
  Serial.print("speedB=");
  Serial.print(speedB);
  Serial.print('\t');

  analogWrite(pwmA, speedA);
  analogWrite(pwmB, speedB);

  lastError=error;

  Serial.println();
}

void readValues(){
  qtrrc.read(sensorValues);
  for (int i=0; i<NUM_SENSORS; i++){
//    Serial.print(sensorValues[i]);
//    Serial.print('\t');
    if(sensorValues[i]>400)
      sensorValues[i]=1;
      else
      sensorValues[i]=0;
  }
}

void calculateProportional(){
   int sum = 0;
   int posLeft = 10;
   int posRight = 10;

   for (int i=0; i<NUM_SENSORS/2; i++){
     sum=sum+sensorValues[i];
     if(sensorValues[i]==1){
       posRight=i-3;
     }
   }
   for (int i=NUM_SENSORS-1; i>=NUM_SENSORS/2; i--){
     sum=sum+sensorValues[i];
     if(sensorValues[i]==1){
       posLeft=i-4;
     }
   }

   if(sum>=3){
     sum=2;
   }

   if(sum==0){
     if(lastError<0){
       error=-8;
```

```
      }
      else{
        error=8;
      }
    }
    else if((posLeft!=10)&&(posRight!=10)){
      error=0;
    }
      else if(posLeft!=10){
        error=posLeft*2+sum;
      }
        else if(posRight!=10){
          error=posRight*2-sum;
        }

    proportional = error;

}
```