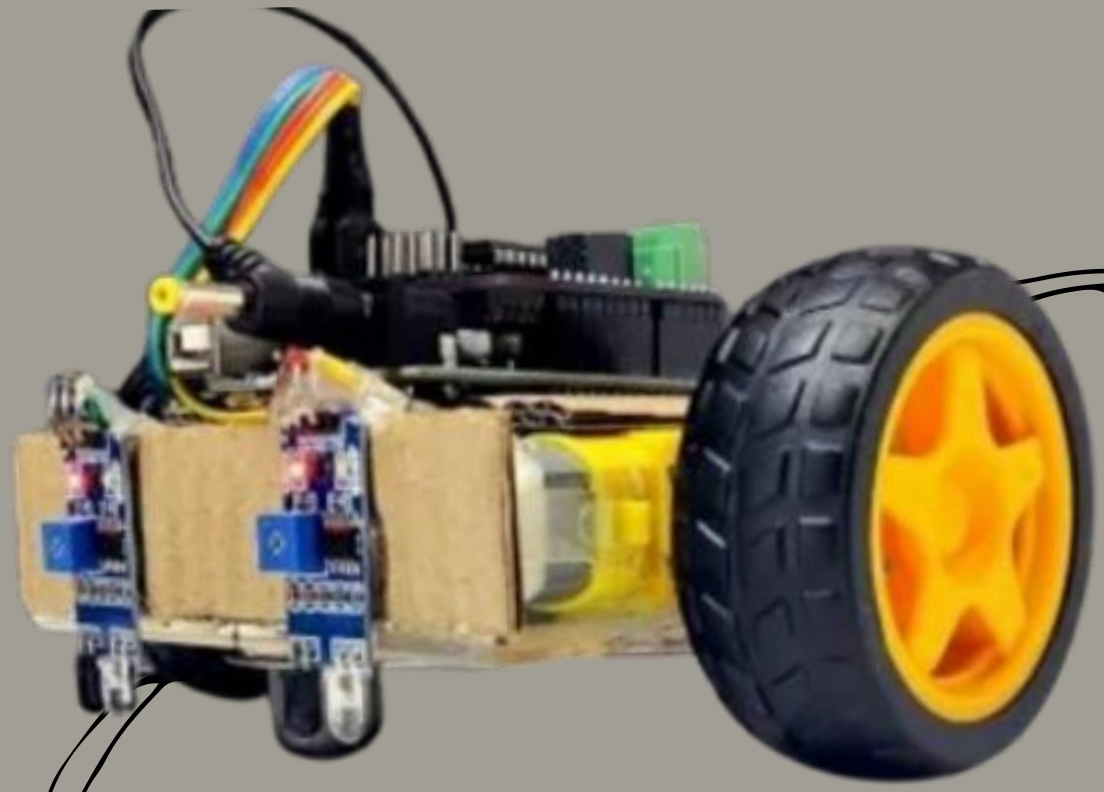
Line Follower Robot



Abstract

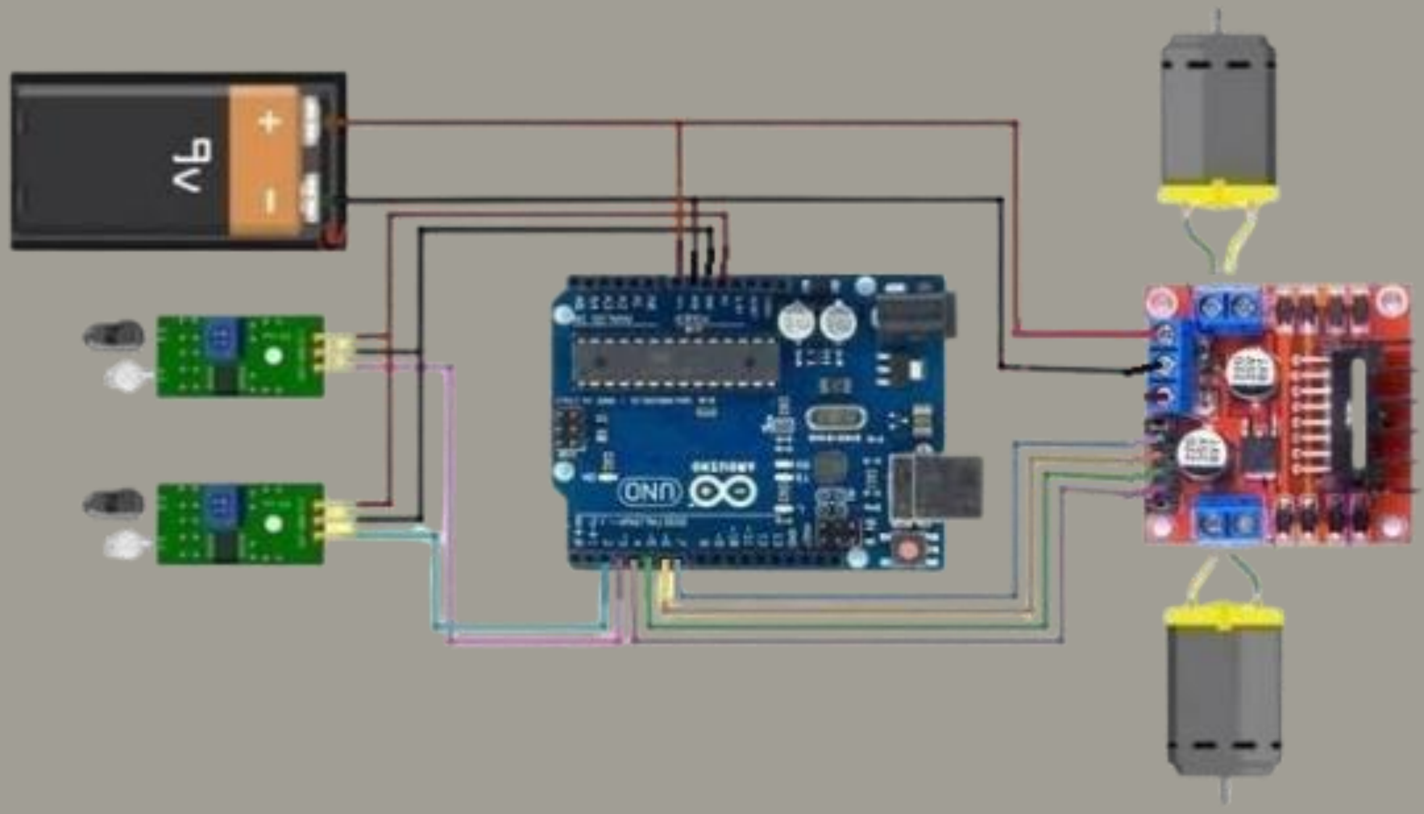
- A Line Following Robot is an autonomous robot which is able to follow either a black line that is drawn on the surface consisting of a contrasting color.

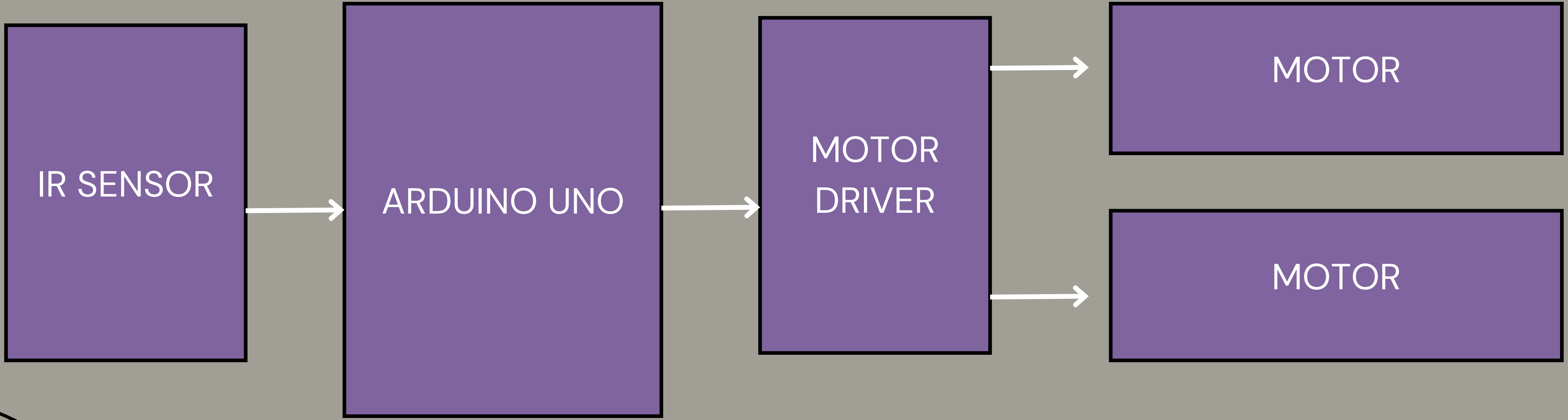
- It is designed to move automatically and follow the line. The robot uses arrays of IR sensors to identify the line, thus assisting the robot to stay on the track. The array of two sensor makes its movement precise and flexible. The robot is driven by DC gear motors to control the movement of the wheels.

- This project aims to implement the algorithm and control the movement of the robot by proper tuning of the control parameters and thus achieve better performance.

- It can be used industrial automated equipment carriers, small household applications, tour guides in museums and other similar applications, etc.

Block Diagram





IR SENSOR

ARDUINO UNO

MOTOR DRIVER

MOTOR

MOTOR

Methodology

Arduino Uno

Arduino Uno is an open-source microcontroller board . Arduino Uno plays a pivotal role in line following robot by acting as the central processing unit that reads sensor inputs, makes decisions based on those inputs, and controls the motors to navigate along the line.

L298N Motor Driver

Commonly used in line-following robots to control the movement of DC motors. It allows the Arduino to manage the direction and speed of the motors based on sensor inputs.

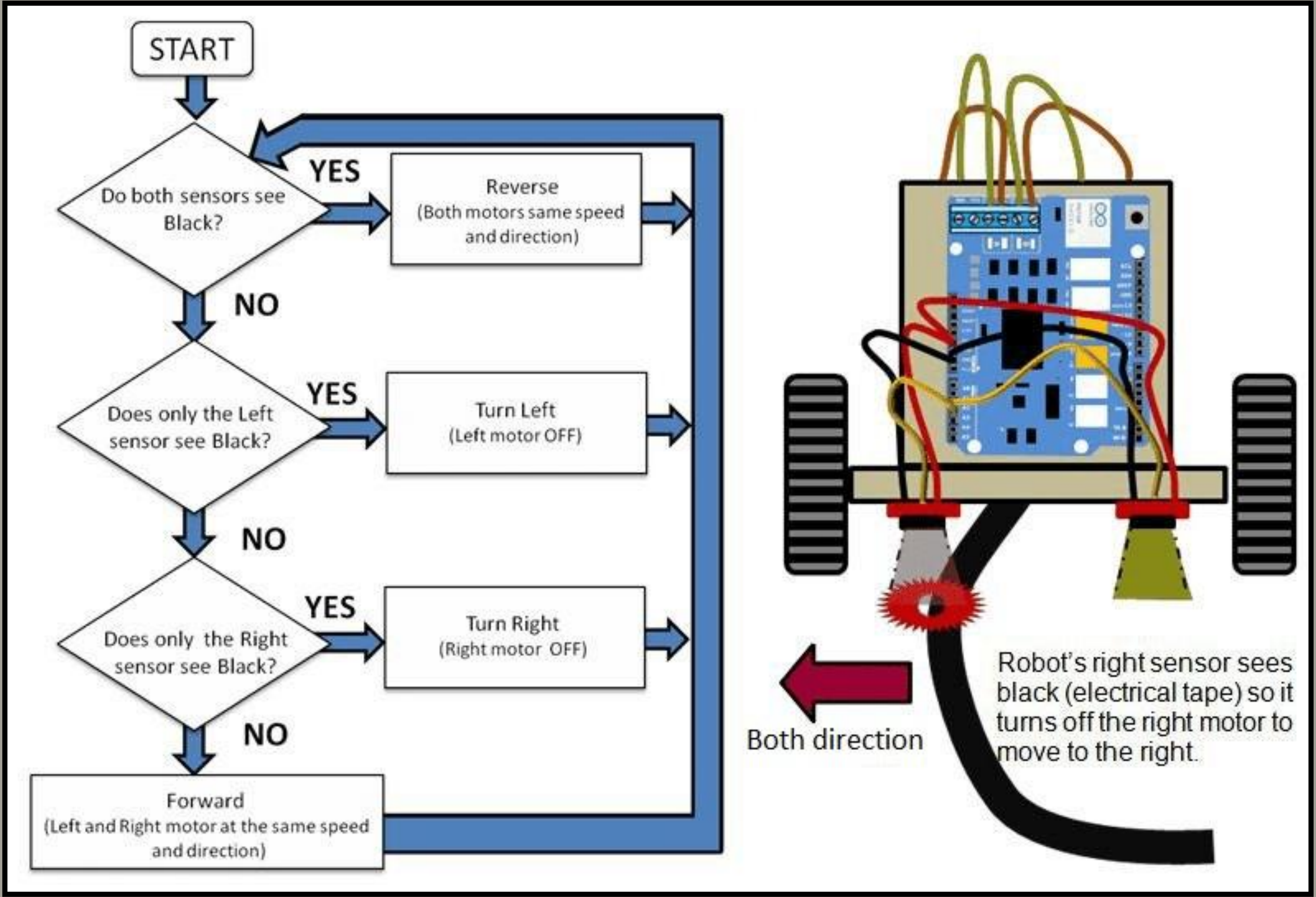
IR Sensors

An IR sensor can measure the heat of an object as well as detects the motion. Their primary role is to detect the presence or absence of a line (usually black) on a surface (usually white or a contrasting color). The sensors provide data to the Arduino to help the robot decide which direction to move to stay on the line.

The Body

Holds all the components and gives the robot structure whereas dc motors are used to rotate the wheels.

Methodology



START

Do both sensors see Black?

YES

Reverse (Both motors same speed and direction)

NO

Does only the Left sensor see Black?

YES

Turn Left (Left motor OFF)

NO

Does only the Right sensor see Black?

YES

Turn Right (Right motor OFF)

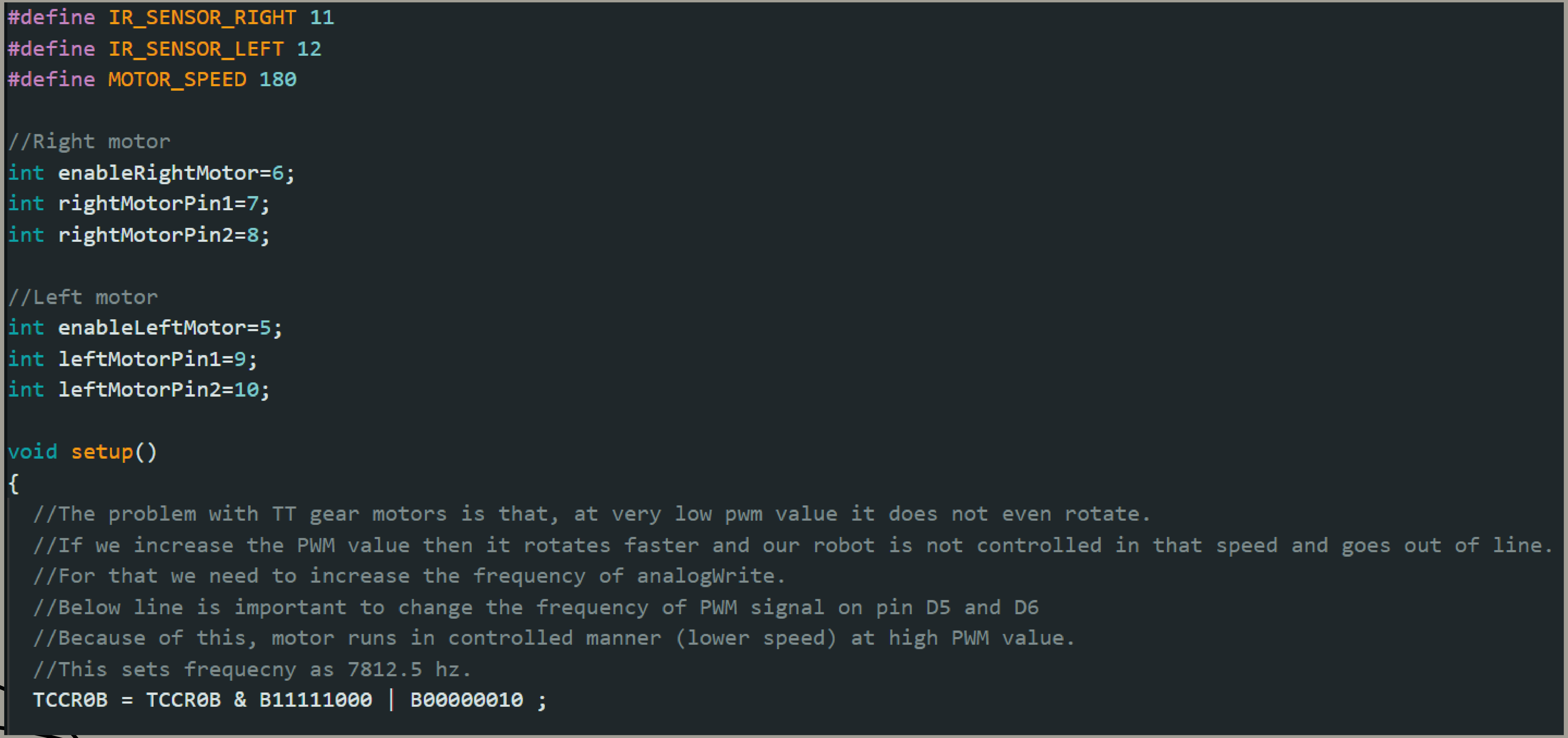
NO

Forward (Left and Right motor at the same speed and direction)

Both direction

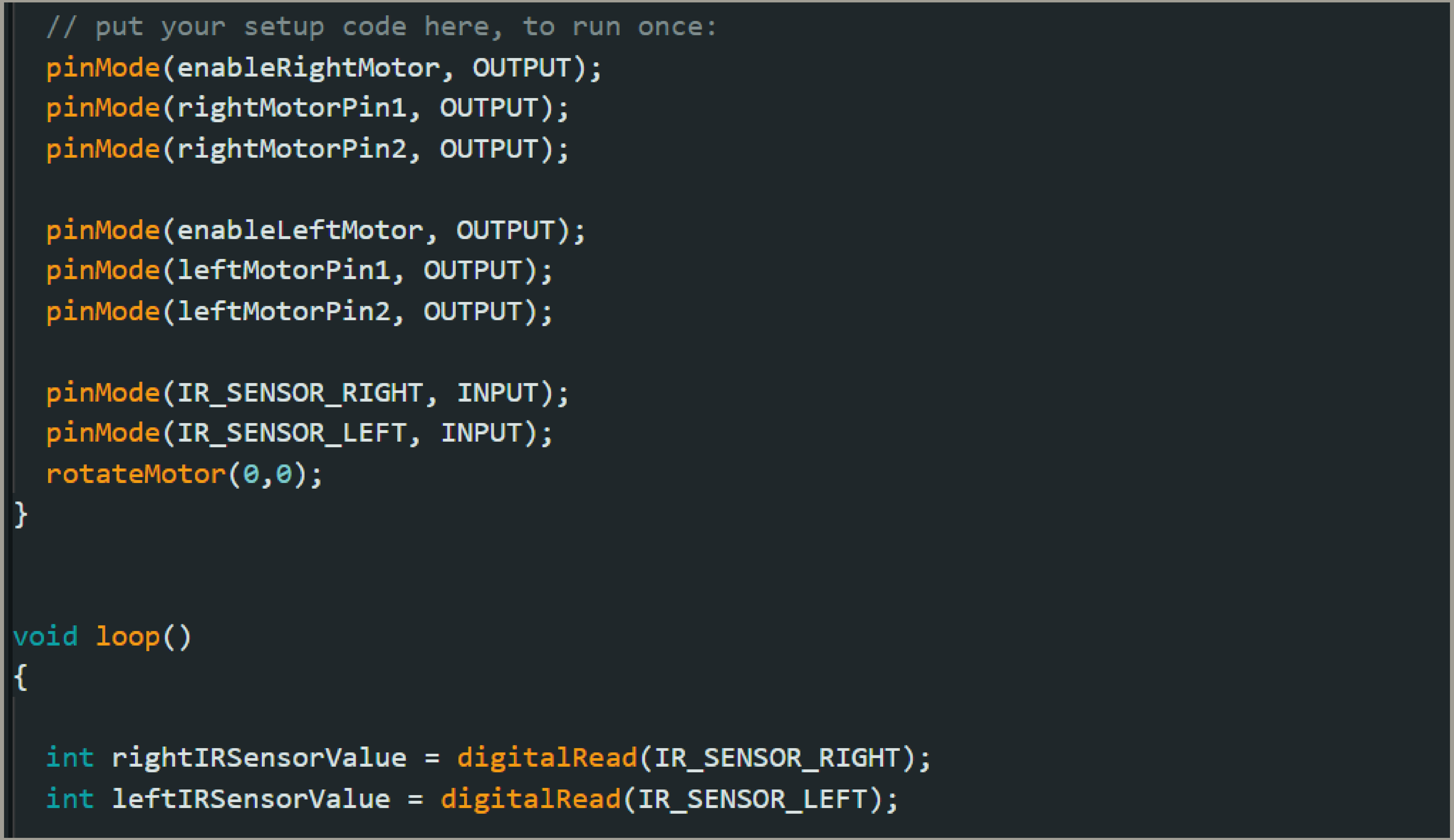
Robot's right sensor sees black (electrical tape) so it turns off the right motor to move to the right.

Code



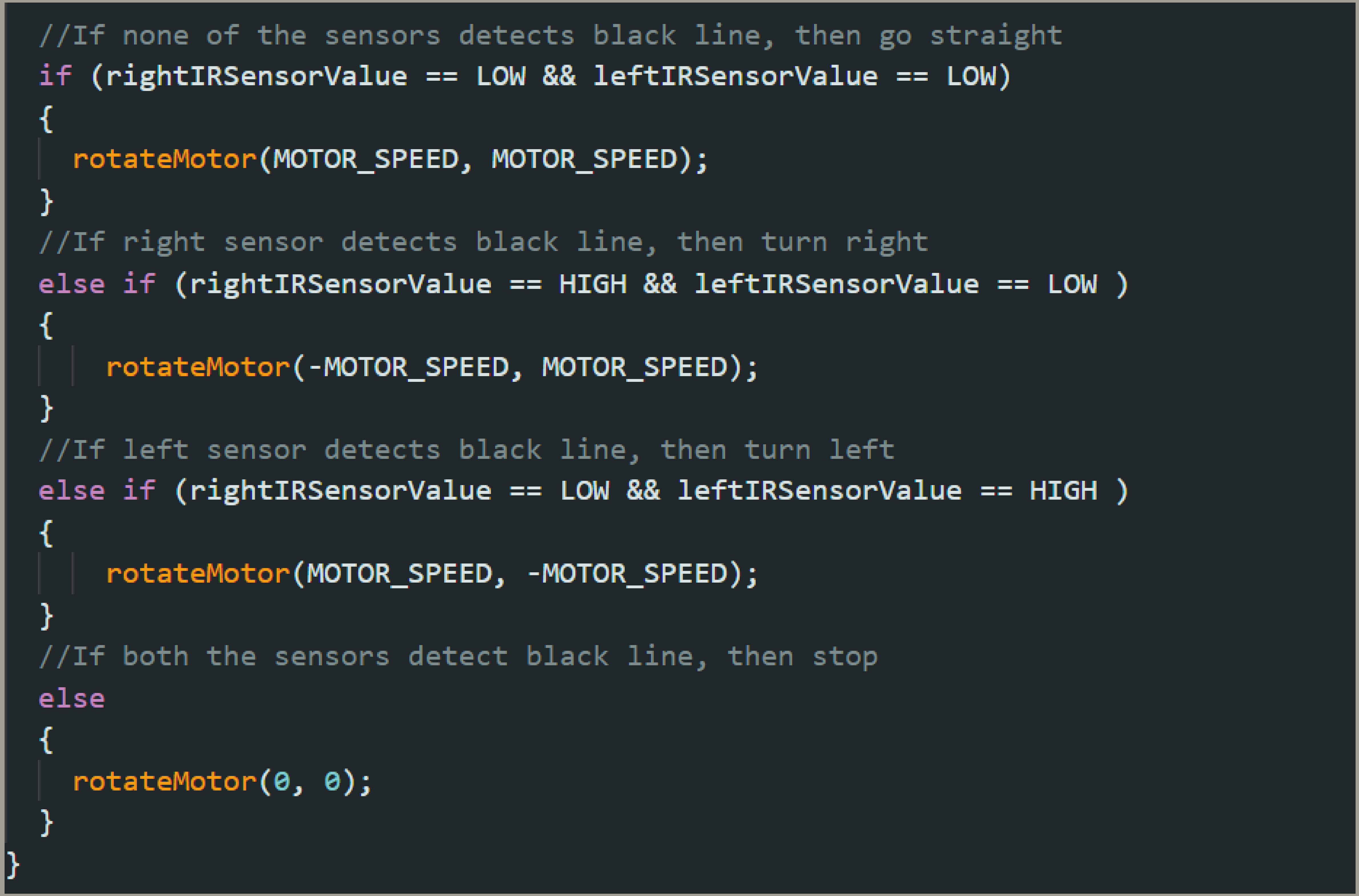
```
#define IR_SENSOR_RIGHT 11
#define IR_SENSOR_LEFT 12
#define MOTOR_SPEED 180

//Right motor
int enableRightMotor=6;
int rightMotorPin1=7;
int rightMotorPin2=8;

//Left motor
int enableLeftMotor=5;
int leftMotorPin1=9;
int leftMotorPin2=10;

void setup()
{
  //The problem with TT gear motors is that, at very low pwm value it does not even rotate.
  //If we increase the PWM value then it rotates faster and our robot is not controlled in that speed and goes out of line.
  //For that we need to increase the frequency of analogWrite.
  //Below line is important to change the frequency of PWM signal on pin D5 and D6
  //Because of this, motor runs in controlled manner (lower speed) at high PWM value.
  //This sets frequecny as 7812.5 hz.
  TCCR0B = TCCR0B & B11111000 | B00000010 ;
```

Code



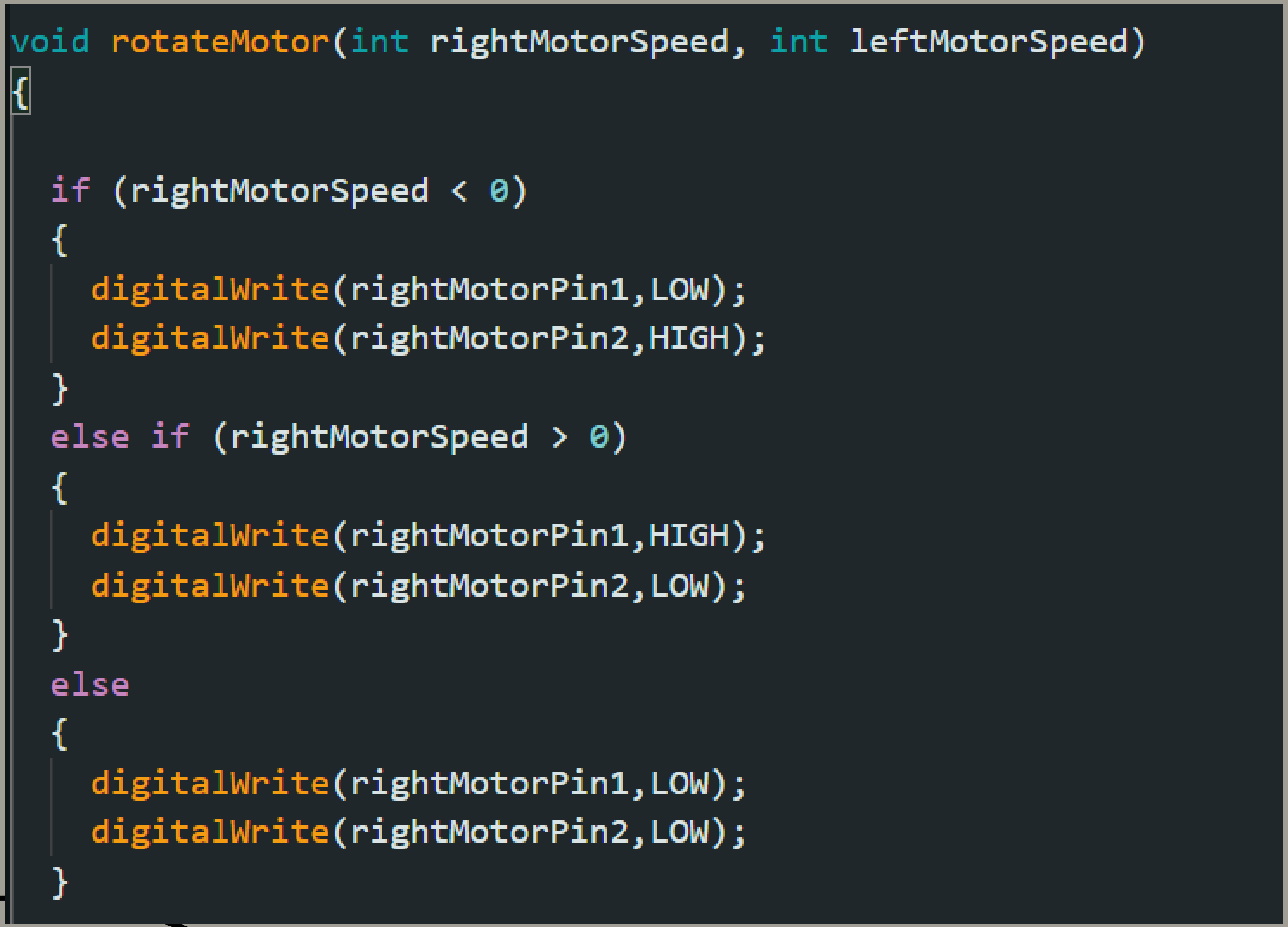
```
  // put your setup code here, to run once:
  pinMode(enableRightMotor, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);

  pinMode(enableLeftMotor, OUTPUT);
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);

  pinMode(IR_SENSOR_RIGHT, INPUT);
  pinMode(IR_SENSOR_LEFT, INPUT);
  rotateMotor(0,0);
}


void loop()
{

  int rightIRSensorValue = digitalRead(IR_SENSOR_RIGHT);
  int leftIRSensorValue = digitalRead(IR_SENSOR_LEFT);
```

Code

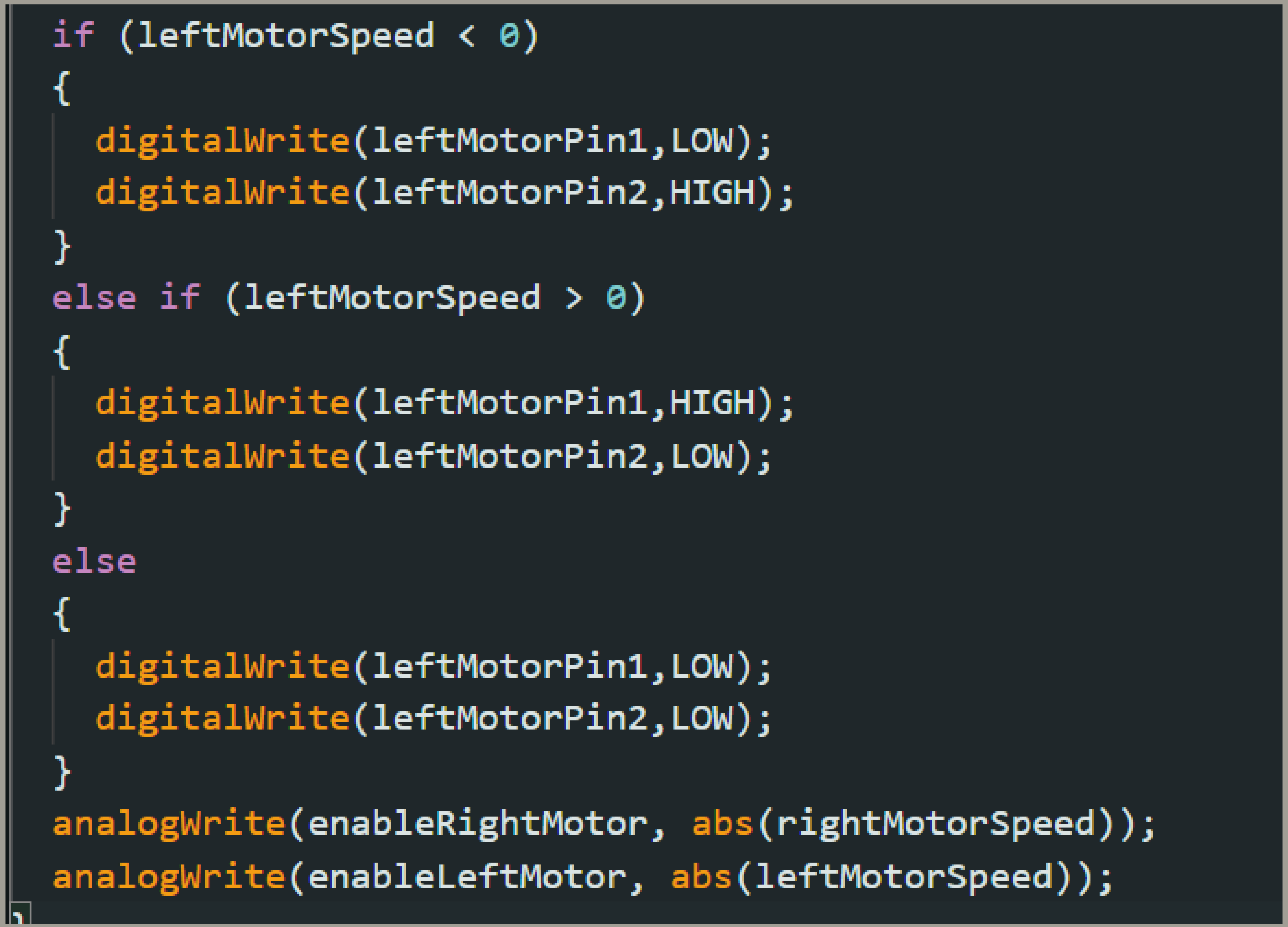


```
  //If none of the sensors detects black line, then go straight
  if (rightIRSensorValue == LOW && leftIRSensorValue == LOW)
  {
    rotateMotor(MOTOR_SPEED, MOTOR_SPEED);
  }
  //If right sensor detects black line, then turn right
  else if (rightIRSensorValue == HIGH && leftIRSensorValue == LOW )
  {
      rotateMotor(-MOTOR_SPEED, MOTOR_SPEED);
  }
  //If left sensor detects black line, then turn left
  else if (rightIRSensorValue == LOW && leftIRSensorValue == HIGH )
  {
      rotateMotor(MOTOR_SPEED, -MOTOR_SPEED);
  }
  //If both the sensors detect black line, then stop
  else
  {
    rotateMotor(0, 0);
  }
}
```

Code



```
void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{

  if (rightMotorSpeed < 0)
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,HIGH);
  }
  else if (rightMotorSpeed > 0)
  {
    digitalWrite(rightMotorPin1,HIGH);
    digitalWrite(rightMotorPin2,LOW);
  }
  else
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,LOW);
  }
```



```
  if (leftMotorSpeed < 0)
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,HIGH);
  }
  else if (leftMotorSpeed > 0)
  {
    digitalWrite(leftMotorPin1,HIGH);
    digitalWrite(leftMotorPin2,LOW);
  }
  else
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,LOW);
  }
  analogWrite(enableRightMotor, abs(rightMotorSpeed));
  analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```

Outcome

Accurate Line Tracking

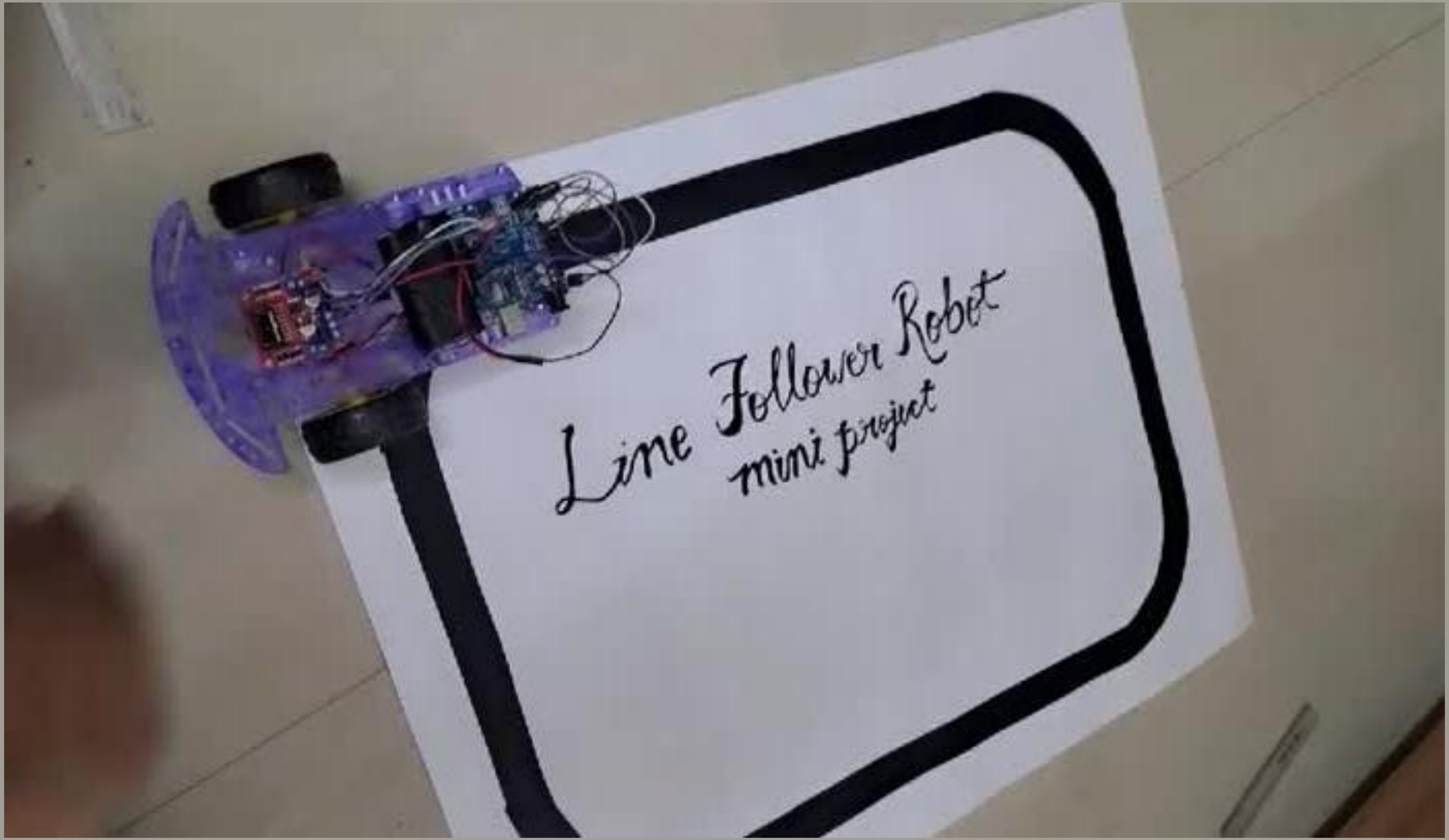
The robot should be able to detect and follow the designated path (usually a black line on a white surface or vice versa) with precision.

Smooth Navigation

The robot should move smoothly without excessive jittering or abrupt changes in direction, even around curves or when there are small deviations in the line.

Efficient Use of Sensors

The robot’s sensors (IR sensors) should work optimally to detect the line in various lighting conditions and at different speeds.


Line Follower Robot
mini project