**T.C.**

**BAHÇEŞEHİR UNIVERSITY**

**FACULTY OF ENGINEERING AND NATURAL SCIENCES**

**DEPARTMENT OF ARTIFICAL INTELLIGENCE ENGINEERING**

**PROJECT FINAL REPORT**

**LIGHT FOLLOWING ROBOT**

**CMP3010 Course Project**

**İSTANBUL, MAY 2025**

# TABLE OF CONTENTS (right-click and select "Update Field")

TABLE OF CONTENTS (right-click and select "Update Field") ii

1 OVERVIEW 1

1.1 Problem Statement and Objectives 1

1.2 Background Information 1

2 METHODOLOGY 2

2.1 Project design 2

2.2 Project components 2

3 WORK PLAN 3

3.1 Tasks and Time Line 3

3.2 Cost Proposal 3

4 REFERENCES 5

5 PHOTOS 6

6 APPENDIX 7

# OVERVIEW

## Problem Statement and Objectives

This project addresses the challenges of creating a robot capable of detecting and responding to light sources while also incorporating user interaction, safety shutdown features, and wireless control. One liability we found is that the robot works perfectly in dark places when you aim a flashlight at it however testing it in open well lit spaces was a struggle due to the robot’s sensitivity to light. It had a hard time figuring out what light it was supposed to follow. The general goal was to build a fully functioning robot that can follow light and perform multiple other functions alongside that. The final product should be able to perform 6 specific functions. It should follow light, and avoid it with a press of a button, led should light up either red or green to represent whether it’s in a follow or avoid mode. It is also capable of sounding off a buzzer before shutting down when no light is detected for a certain period of time. One more thing is that it can be controlled manually using Bluetooth .

## Background Information

Light-following robots are a common topic in robotics education and research due to their simplicity and their ability to demonstrate sensor-based navigation, control, and environmental interaction. These robots typically use Light Dependent Resistors (LDRs) to detect light intensity and adjust their movement accordingly.

Similar systems have been used in solar tracking applications, where devices automatically follow sunlight to maximize energy absorption, such as in solar panels or solar-powered vehicles [1]. Several projects in the public domain use LDRs. For example, robots like the "BEAM" (Biology, Electronics, Aesthetics, and Mechanics) light-seeking robots use analog circuits to follow light sources in real time. In short, they don’t use microcontrollers like Arduino, instead, they rely on basic electronic to produce reactive behaviors [2]. However, such systems often lack modularity, interactivity, and safety protocols. By combining basic robotic functionality with interactivity, shutdown safety, and wireless control, the project bridges the gap between simple reactive machines and more intelligent, multi-modal robotic systems.

# METHODOLOGY

## Project design

The light-following robot is built around the Arduino Uno microcontroller, which serves as the central control unit. It integrates several components, including LDRs (Light Dependent Resistors) for detecting light intensity, an HC-06 Bluetooth module for wireless control via smartphone, and an ultrasonic sensor positioned at the end to detect obstacles behind the robot. Motor control is handled through an L298N motor driver. The robot includes visual indicators with two LEDs (green for follow mode, red for avoid mode), a buzzer for alerts and shutdown signals, and a manual button to toggle between light-following and light-avoiding modes. The entire system is powered by a 12V battery. One of the main technical challenges during development was the robot's inconsistent behavior in well-lit environments. The LDR sensors, while effective in darker settings, were overly sensitive to brightly lit rooms. To solve this behaviour, a tunnel-like shield made from dark paper was folded and mounted over each LDR. This limits the sensor’s field of view, allowing it to respond only to direct light coming from the front.

A computer circuit board with many wires

AI-generated content may be incorrect.Figure . Overall design of the system and circuit schematic (PROTEUS)

## Project components

1. Arduino uno: controls the logic

1. DC Motors and wheels: provides the robot with the ability to move
2. Breadboard: helps make connections
3. L298N Motor Driver: It controls the DC motors. Another alternative Motor Driver was L293D however the L298N is better suited for mid-sized robots like ours that require more current and simplicity during assembly
4. Screwdriver : Used to assemble and tighten hardware components
5. Jumper wires: connect various components to the arduino
6. 12V battery: powers motors and arduino
7. Battery holder: holds and orgnaizes the 12V battery safely
8. Buzzer: Makes a sound like an alert if no light detected for a period of time
9. Hc-06 bluetooth module: Enables manual control of the robot via a smartphone

# WORK PLAN

## Cost Proposal

Table 1. Components and their estimated costs.

|  |  |
| --- | --- |
| **Component** | **Cost (TL)** |
| Arduino Uno | 0 |
| Chasis and 4 wheels | 0 |
| 4 DC Motors | 0 |
| Breadboard, jumper wires | 0 |
| L298N Motor Driver | 0 |
| Button | 0 |
| Buzzer | 0 |
| Hc-06 Bluetooth Module | 0 |
| Green and Red LEDS | 0 |
| Battery Holder | 45 |
| 12V Batteries | 620 |
| LDR Module | 85 |
| **Total** | **750** |

Table 2. The project Gantt chart.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | WEEKS | | | | | | |  |  |  |  |
| TASK LIST | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **1. Research and Design** |  |  |  |  |  |  |  |  |  |  |  |
| 1.1 Define the goals, and perform a literature search. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 Prepare an overall design and order parts. |  |  |  |  |  |  |  |  |  |  |  |
| **2. Construction and Coding** |  |  |  |  |  |  |  |  |  |  |  |
| 2.1 Set up base system (motors, wiring sensors) |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 Add Sensors, LED, Button and set up code |  |  |  |  |  |  |  |  |  |  |  |
| 2.3 Assemble Bluetooth system and buzzer |  |  |  |  |  |  |  |  |  |  |  |
| 2.4 Integrate all systems, update code |  |  |  |  |  |  |  |  |  |  |  |
| **3. Testing and Documentation** |  |  |  |  |  |  |  |  |  |  |  |
| 3.1 Prepare the proposal |  |  |  |  |  |  |  |  |  |  |  |
| 3.2 Test and Verify everything is working |  |  |  |  |  |  |  |  |  |  |  |
| 3.3 Prepare the report |  |  |  |  |  |  |  |  |  |  |  |
| 3.4 Presentation |  |  |  |  |  |  |  |  |  |  |  |

# REFERENCES

1. A. S. M. Shihavuddin, M. A. H. Akhand, and M. A. H. Chowdhury, "Design and implementation of an automatic solar tracking system for efficient solar energy harvesting," *Materials Today: Proceedings*, vol. 33, pp. 1560–1565, 2020
2. DigiKey, "What are BEAM Bots?", DigiKey Electronics, May 2019.

# PHOTOS

A green and red light on a robot

AI-generated content may be incorrect.A person working on a toy car

AI-generated content may be incorrect.

A toy car with wires and wheels

AI-generated content may be incorrect. A hand holding a toy car

AI-generated content may be incorrect.

A device with wires on it

AI-generated content may be incorrect. A machine with wires on wheels

AI-generated content may be incorrect.

# APPENDIX

You may put any formulas calculations here. **It is a must to put your full code here with explanations.**

// Pin Definitions

int RMotor\_1 = 2;

int RMotor\_2 = 3;

int LMotor\_1 = 4;

int LMotor\_2 = 5;

int REnable = 10;

int LEnable = 11;

int LDR\_Right = A0;

int LDR\_Left = A1;

int modeButton = 9;

int greenLED = 6;

int redLED = 13;

int buzzerPin = A2;

bool avoidMode = false;

bool buttonPreviouslyPressed = false;

bool shutDownTriggered = false;

bool noLightWarningActive = false;

unsigned long noLightStartTime = 0;

int lightThreshold = 600;

// Movement Functions

void move\_forward() {

digitalWrite(RMotor\_1, LOW); digitalWrite(RMotor\_2, HIGH);

digitalWrite(LMotor\_1, HIGH); digitalWrite(LMotor\_2, LOW);

}

void move\_backward() {

digitalWrite(RMotor\_1, HIGH); digitalWrite(RMotor\_2, LOW);

digitalWrite(LMotor\_1, LOW); digitalWrite(LMotor\_2, HIGH);

}

void turn\_right() {

digitalWrite(RMotor\_1, HIGH); digitalWrite(RMotor\_2, LOW);

digitalWrite(LMotor\_1, HIGH); digitalWrite(LMotor\_2, LOW);

}

void turn\_left() {

digitalWrite(RMotor\_1, LOW); digitalWrite(RMotor\_2, HIGH);

digitalWrite(LMotor\_1, LOW); digitalWrite(LMotor\_2, HIGH);

}

void move\_stop() {

digitalWrite(RMotor\_1, LOW); digitalWrite(RMotor\_2, LOW);

digitalWrite(LMotor\_1, LOW); digitalWrite(LMotor\_2, LOW);

}

void setup() {

Serial.begin(9600);

pinMode(RMotor\_1, OUTPUT); pinMode(RMotor\_2, OUTPUT);

pinMode(LMotor\_1, OUTPUT); pinMode(LMotor\_2, OUTPUT);

pinMode(REnable, OUTPUT); pinMode(LEnable, OUTPUT);

analogWrite(REnable, 210); analogWrite(LEnable, 210);

pinMode(modeButton, INPUT\_PULLUP);

pinMode(greenLED, OUTPUT);

pinMode(redLED, OUTPUT);

pinMode(buzzerPin, OUTPUT);

digitalWrite(buzzerPin, LOW);

}

void loop() {

if (shutDownTriggered) {

move\_stop();

digitalWrite(greenLED, LOW);

digitalWrite(redLED, LOW);

return;

}

// Bluetooth Commands

if (Serial.available()) {

char command = Serial.read();

switch (command) {

case 'F': move\_forward(); break;

case 'B': move\_backward(); break;

case 'L': turn\_left(); break;

case 'R': turn\_right(); break;

case 'S': move\_stop(); break;

case 'D': dance(); break;

default: move\_stop(); break;

}

delay(100);

return;

}

// Toggle Mode with Button

bool buttonPressed = digitalRead(modeButton) == LOW;

if (buttonPressed && !buttonPreviouslyPressed) {

avoidMode = !avoidMode;

}

buttonPreviouslyPressed = buttonPressed;

// LED Indicator

digitalWrite(greenLED, avoidMode ? LOW : HIGH);

digitalWrite(redLED, avoidMode ? HIGH : LOW);

// LDR Readings

int ldrright = analogRead(LDR\_Right);

int ldrleft = analogRead(LDR\_Left);

bool noLight = (ldrleft >= lightThreshold && ldrright >= lightThreshold);

if (noLight) {

if (!noLightWarningActive) {

noLightStartTime = millis();

noLightWarningActive = true;

} else if (!shutDownTriggered && millis() - noLightStartTime >= 40000) {

move\_stop();

digitalWrite(buzzerPin, HIGH);

delay(2000);

digitalWrite(buzzerPin, LOW);

shutDownTriggered = true;

}

} else {

noLightWarningActive = false;

shutDownTriggered = false;

digitalWrite(buzzerPin, LOW);

}

// Follow or Avoid Light Mode

if (!avoidMode) {

if (ldrright < lightThreshold && ldrleft < lightThreshold) {

move\_forward();

} else if (ldrright < lightThreshold && ldrleft >= lightThreshold) {

turn\_right();

} else if (ldrright >= lightThreshold && ldrleft < lightThreshold) {

turn\_left();

} else {

move\_stop();

}

} else {

if (ldrright < lightThreshold && ldrleft < lightThreshold) {

move\_backward();

} else if (ldrright < lightThreshold && ldrleft >= lightThreshold) {

turn\_right();

} else if (ldrright >= lightThreshold && ldrleft < lightThreshold) {

turn\_left();

} else {

move\_stop();

}

}

delay(100);

}

void dance() {

for (int i = 0; i < 3; i++) {

move\_forward(); flashLights(); delay(400);

move\_backward(); flashLights(); delay(400);

turn\_left(); flashLights(); delay(300);

turn\_right(); flashLights(); delay(300);

}

move\_stop();

}

void flashLights() {

digitalWrite(greenLED, HIGH);

digitalWrite(redLED, HIGH);

tone(buzzerPin, 1000, 100);

delay(100);

digitalWrite(greenLED, LOW);

digitalWrite(redLED, LOW);

}