COMSATS UNIVERSITY ISLAMABAD

Microprocessor Systems & Interfacing

LFR REPORT

NAME:

1. MUHAMMAD IRTAZA KARAMAT
2. MUHAMMAD HUZAIFA MUKHTAR
3. SHIEKH HASSAN TARIQ

REGISTRATION NO:

1. FA21-BEE-116
2. FA21-BEE-113
3. FA20-BEE-189

SUBMITTED TO:

Dr. JUNAID AHMED

Line Following Robot (LFR)

Introduction:

A Line Following Robot (LFR) is a specialized autonomous vehicle designed to traverse predefined paths by tracking lines on a surface. Employing infrared sensors, the robot detects the contrast between the path and its surroundings. By interpreting these sensor inputs, it makes real-time adjustments to its motor speeds, ensuring it stays on course. LFR robots are commonly used in educational settings to teach concepts of robotics, automation, and sensor integration. They provide a practical platform for enthusiasts to delve into the principles of line-following algorithms, enhancing understanding of both hardware and software aspects of robotic systems.

Objectives:

The primary goals of the line-following robot include:

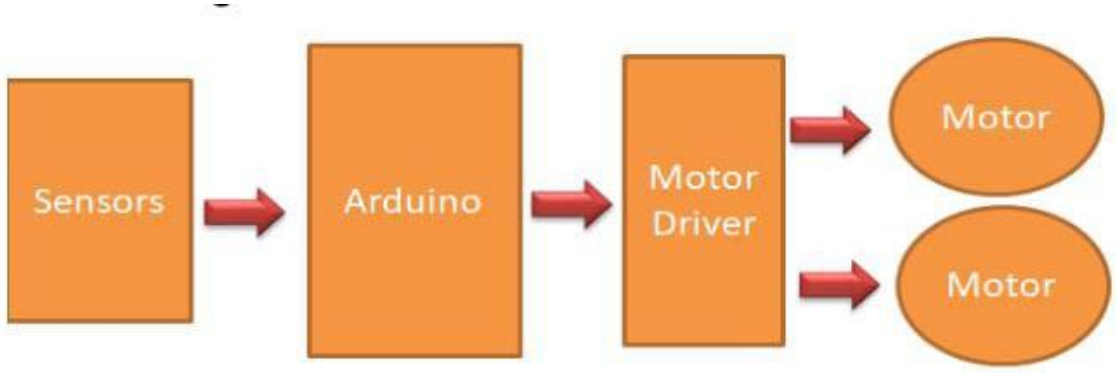
• To navigating along a designated path by utilizing IR sensors to detect both the line and its edges, influencing the operation of the wheels.

• Demonstrating the ability to execute turns of varying degrees.

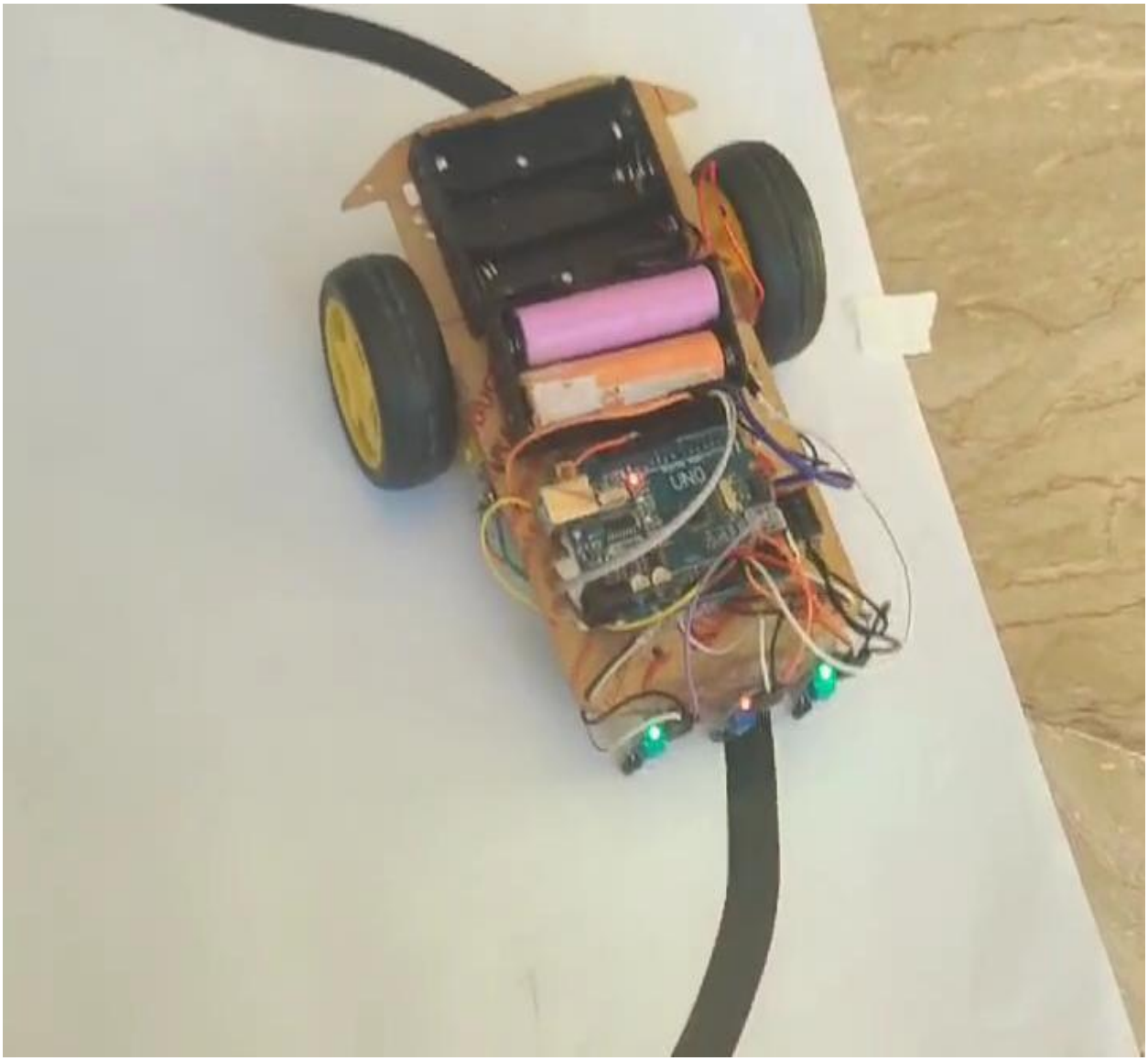
• Exhibiting resilience to external factors like intense illumination or extremely low-light conditions.

• Equipped with infrared sensors, the robot identifies the line, sending signals to the Arduino Uno to maintain its alignment with the path and recognize checkpoints. Each time the robot passes a checkpoint, it triggers a buzzer sound or activates an LED.

Block Diagram:



ROBOT DIAGRAM:



ALGORITHM:

1. Begin by configuring the Arduino Uno with essential components for the robot, including sensors, gear motors, motor drivers, and wheels.

2. Upload the code to the Arduino Uno and verify its effectiveness in keeping the robot aligned with the line.

3. Utilize the sensors to assess the robot's position concerning the line.

4. Adjusting the IR sensor values using a potentiometer based on environmental variables such as bright or extremely dark conditions.

5. Repeat the procedure to achieve accurate responses from the robot, ensuring its consistent alignment with the designated line.

QUESTION 02

Pseudocode or flow diagram showing how the robot will maneuver the maze.

INITIALIZATION OF MOTOR PINS

```
int S_A = 6;   // Speed motor A
int M_A1 = 2;   // Motor A = +
int M_A2 = 3;   // Motor A = -
int M_B1 = 4;   // Motor B = -
int M_B2 = 5;   // Motor B = +
int S_B = 7;    // Speed motor B
```

Here I have only initialize the pins for the motors where we can do the connection our motors.

INTIALIZATION OF SENSORS

```
int R_S = A0;   // Sensor R
int S_S = A1;   // Sensor S
int L_S = A2;   // Sensor L
```

Here I have only initialize the pins for the motors where we can do the connection for our sensors motors.

SETTING OVERALL SPEED FOR THE SYSTEM

```
int maxSpeed = 160;  // speed value
```

Setting the speed for overall system

INITIALIZATION OF INPUT AND OUTPUT PINS

```
void setup() {
  pinMode(S_B, OUTPUT);
  pinMode(S_A, OUTPUT);


  pinMode(L_S, INPUT);
  pinMode(S_S, INPUT);
```

```
pinMode(R_S, INPUT);


digitalWrite(S_A, HIGH);
digitalWrite(S_B, HIGH);
delay(1000);}
```

EXEMPLIFYING WHAT WILL HAPPEN IF THE SENSORS WILL DETECT THE BLACK LINE

```
void loop() {
 int currentSpeed = maxSpeed;  // Set the initial speed


 if ((digitalRead(L_S) == 0) && (digitalRead(S_S) == 1) && (digitalRead(R_S) == 0)) {
  forward(currentSpeed);
 }


 if ((digitalRead(L_S) == 1) && (digitalRead(S_S) == 1) && (digitalRead(R_S) == 0)) {
  turnLeft(currentSpeed);
 }
 if ((digitalRead(L_S) == 1) && (digitalRead(S_S) == 0) && (digitalRead(R_S) == 0)) {
  turnLeft(currentSpeed);
 }


 if ((digitalRead(L_S) == 0) && (digitalRead(S_S) == 1) && (digitalRead(R_S) == 1)) {
  turnRight(currentSpeed);
 }
 if ((digitalRead(L_S) == 0) && (digitalRead(S_S) == 0) && (digitalRead(R_S) == 1)) {
  turnRight(currentSpeed);
 }


 if ((digitalRead(L_S) == 1) && (digitalRead(S_S) == 1) && (digitalRead(R_S) == 1)) {
```

```
   stopMotors();
  }
}
```

Here I have explained every condition what will happen when one sensor will be on, two sensors will be on or three sensors will be on.

DEFINING HOW THE MOTOR WILL MOVE FORWARD

```
void forward(int speed) {
  analogWrite(M_A1, 0);
  analogWrite(M_A2, speed);
  analogWrite(M_B1, speed);
  analogWrite(M_B2, 0);
}
```

Now here I have configure A2 and B1 pin to high so it will move in forward direction by setting others pin 0 so they will be off.

DEFINING HOW THE MOTOR WILL TAKE A RIGHT TURN

```
void turnRight(int speed) {
  analogWrite(M_A1, 0);
  analogWrite(M_A2, speed);
  analogWrite(M_B1, 0);
  analogWrite(M_B2, 0);
}
```

Now here I have configure A2 pin to high so it will move in right direction by setting others pin 0 so they will be off.

DEFINING HOW THE MOTOR WILL TAKE A LEFT TURN

```
void turnLeft(int speed) {


  analogWrite(M_A1, 0);
  analogWrite(M_A2, 0);
  analogWrite(M_B1, speed);
```

```
  analogWrite(M_B2, 0);
}
```

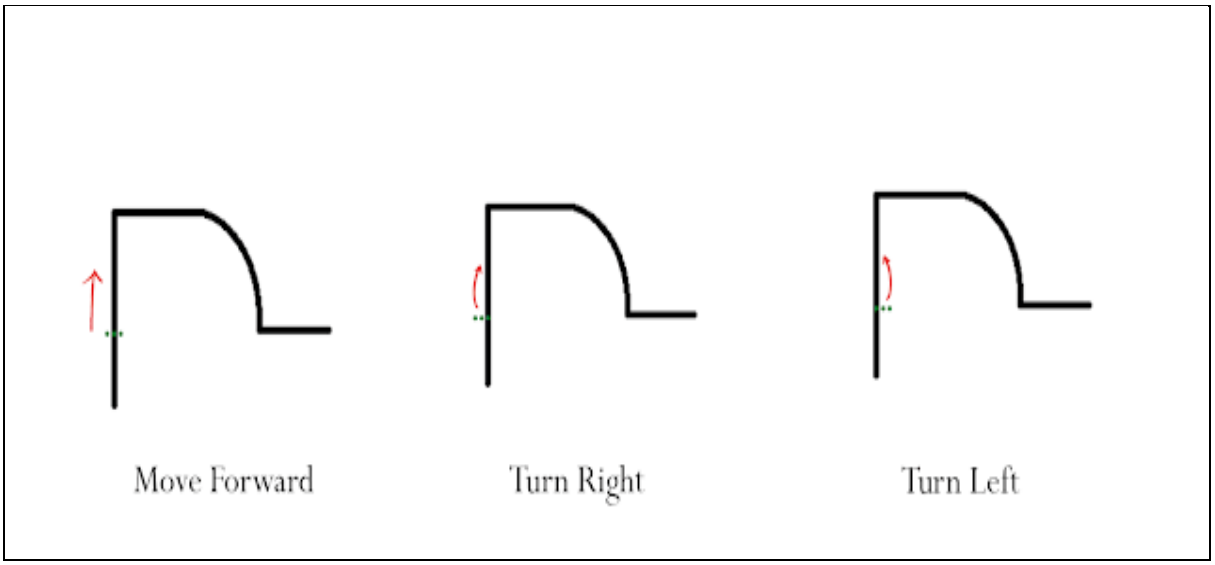
Now here I have configure B1 pin to high so it will move in right direction by setting others pin 0 so they will be off.

DEFINING HOW THE MOTOR WILL STOP

```
void stopMotors() {
  analogWrite(M_A1, 0);
  analogWrite(M_A2, 0);
  analogWrite(M_B1, 0);
  analogWrite(M_B2, 0);
}
```

Now here I have configure every pin to 0 so it will stop.

Dots show the position of the Sensors



1. When the center sensor detect the line it will move forward.
2. When the center sensor and right sensor will detect the line it will move right.
3. When the center sensor and left sensor will detect the line it will move left.