Lab 4: Hybrid Robot Control Using Dabble and Ultrasonic Sensor

Course: Robotics / Embedded Systems

Submission: Group Work

Total Marks: 100

1. Introduction

In this lab, students will extend their previous robot control system by integrating Bluetooth control through the Dabble application and implementing an automatic obstacle avoidance mode using an ultrasonic sensor. The goal is to design a robot that can operate in two modes: Manual Mode and Automatic Mode.

The robot must remain idle at startup. It should not move until the user selects a mode. The SELECT button activates Manual Mode, while the START button activates Automatic Mode.

2. Learning Outcomes

By completing this lab, students should be able to:

- Implement Bluetooth-based robot control using Dabble.

- Control servo position using incremental and preset commands.

- Design a simple obstacle avoidance algorithm.

- Implement mode selection logic safely and correctly.

- Apply structured logic and avoid blocking code where possible.

3. Required Hardware

- ESP32 development board

- Motor driver and 4 DC motors

- Servo motor

- Ultrasonic sensor (fixed at the front)

- Battery pack

- Smartphone with Dabble app

4. Manual Mode Requirements

When SELECT is pressed, the robot enters Manual Mode. In this mode, the joystick controls movement:

- Up: Move forward

- Down: Move backward

- Left: Rotate left

- Right: Rotate right

Servo control in Manual Mode:

- Square: Increase angle by 10 degrees

- Circle: Decrease angle by 10 degrees

- Cross: Move servo to 0 degrees

- Triangle: Move servo to 180 degrees

The servo angle must always remain within the range of 0 to 180 degrees. Students must ensure the angle does not exceed these limits.

5. Automatic Mode Requirements

When START is pressed, the robot switches to Automatic Mode. In this mode, the robot ignores joystick commands and uses the ultrasonic sensor to detect obstacles.

If the ultrasonic sensor detects an object closer than 10 cm, the robot must rotate left by approximately 90 degrees and then continue moving forward.

If an obstacle is still detected after turning, the robot should continue rotating left in similar increments until a clear path is found.

6. System Logic Requirements

Students must implement clear mode-switching logic. The robot should operate in only one mode at a time. The program should continuously check inputs and sensor readings inside the main loop.

Long blocking delays should be minimized to ensure smooth operation.

7. Flowchart Requirement

Students must design a flowchart showing the overall system structure, including:

1. System initialization

2. Waiting for mode selection

3. Manual control logic

4. Automatic avoidance logic

5. Continuous looping

8. Demonstration

Each group must demonstrate:

- Mode selection before movement

- Full joystick movement control

- Servo incremental movement

- Servo preset positions

- Successful obstacle avoidance

9. Submission Requirements

- Source code (.ino file)

- Flowchart diagram

- Short written explanation (approximately one page)

- Video demonstration of the robot working