Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department


# Embedded Systems Final Project

Group 2

22442, 22340

Autonomous Mobile Robots Challenge



| Authors: | ID: | Major | Supervisor: |
|---|---|---|---|
| Albert Saman | 20220328 | Electronics Engineering | Dr. Esam Qaralleh |
| Ali Bani Bakkar | 20220766 | Power Engineering | Dr. Belal Sababha |
| Omar Khalil | 20220150 | Electronics Engineering | Dr. Belal Sababha |



January 14, 2026

# ABSTRACT

This project details the design and implementation of an autonomous mobile robot using the PIC16F877A microcontroller. The robot is programmed to navigate a multi-stage track consisting of a Line-Following zone, a Tunnel (darkness detection), an Obstacle Avoidance zone, and a Parking area. Strictly adhering to the "No Built-in Libraries" constraint, the firmware features custom drivers for Ultrasonic distance measurement using Timer1, a Software-based PWM system generated via Timer0 interrupts for motor speed control, and direct register manipulation for ADC and I/O handling. The final system demonstrates a robust Finite State Machine (FSM) architecture capable of real-time sensor fusion and precise navigation. This report covers the mechanical assembly, electrical circuit design, and the state-machine software architecture used to ensure precise timing, sensor data processing, and actuator control.

# TABLE OF CONTENTS

## TABLE OF FIGURES

# INTRODUCTION AND BACKGROUND

Autonomous mobile robots have become a cornerstone of modern embedded systems engineering. For navigation tasks, behavioral autonomy relies on sensor data, real-time actuation control, and the ability to adapt to structured and unstructured environments.

In this project, we were required to design a robot capable of completing a multi-zone course composed of the following:

1. **Start Zone:** Button-activated start with a precise 3-second delay.
2. **Line-Following Zone:** Stable tracking of a black line with straight, curved, and T-intersection patterns.
3. **Tunnel Zone:** Detection of reduced illumination, activation of a buzzer, and continued line tracking.
4. **Path Navigation Zone:** Free navigation without lines, using ultrasonic and IR sensors to avoid obstacles inside bordered walls.
5. **Bump Zone:** Differentiation between obstacles and a 30° ramp, following the line over the incline.
6. **Parking Zone:** Detection of black flooring, controlled approach to a wall with buzzer beeping, final stop, LED off, and raising a flag.

The robot must complete the course without human intervention. The PIC16F877A was selected due to its multipurpose I/O ports, integrated ADC, timers, and compatibility with PWM generation in software interrupts. The mikroC environment allowed the implementation of fine-grained timing control, ADC sampling, and multi-stage robotic behaviors.

**Objective:** The robot must complete all stages of the map smoothly and accurately in the shortest time possible. It must handle specific tasks, such as detecting light intensity changes in a tunnel, avoiding a randomly placed obstacle, and traversing a bump.

**Vehicle Description:** The robot utilizes a differential-drive mechanism, allowing it to move forward, backward, and rotate by independently controlling two powered wheels and their speeds using Pulse Width Modulation (PWM).

**Constraints:** The project mandates strict design rules, including a maximum vehicle size of 15 × 25 × 30 cm and the prohibition of specific coding shortcuts like built-in delays or bit-access unions.

The path that the robot will navigate through is shown in Figure 1.

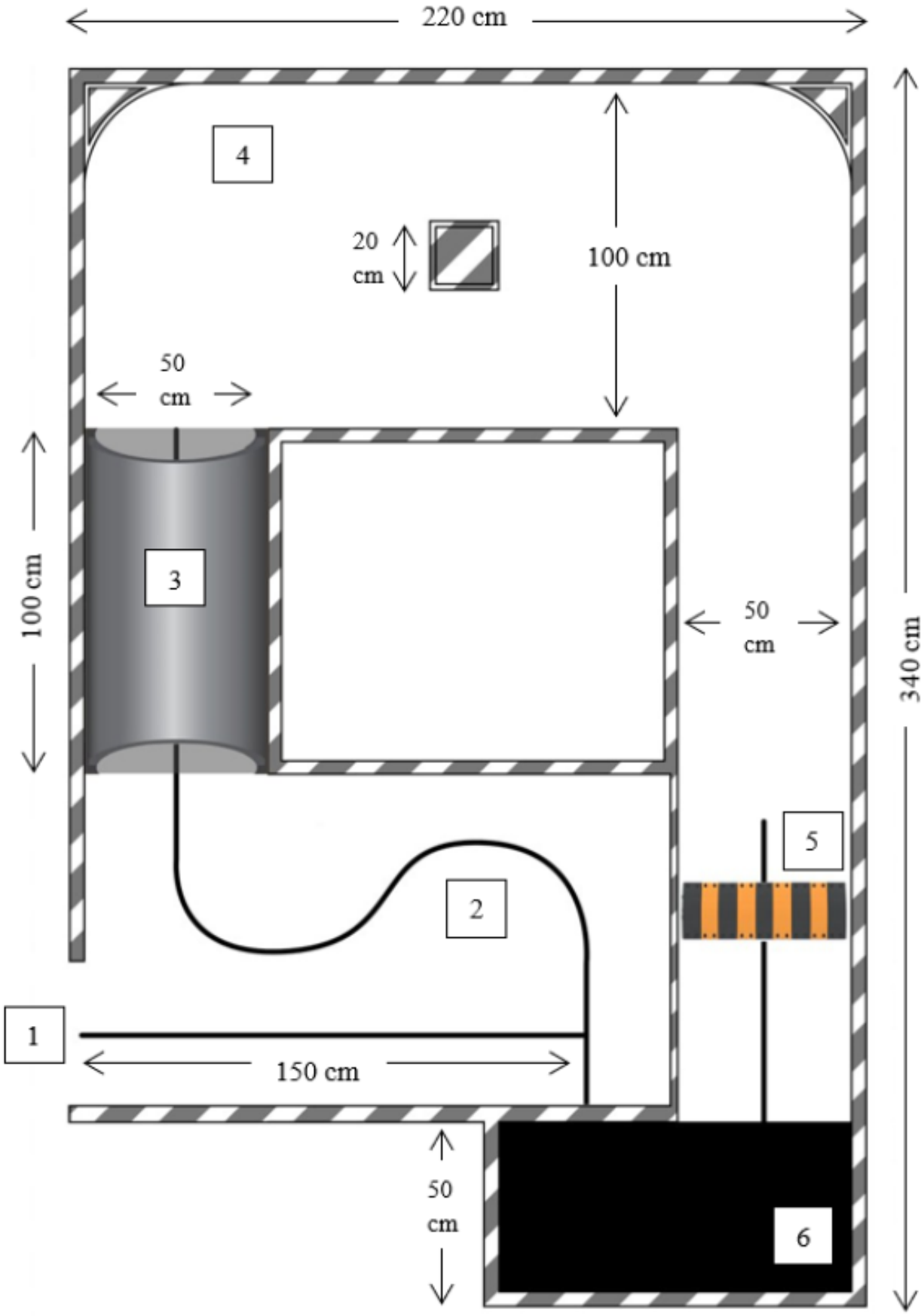


Figure 1: The Design of the Path

# DESIGN OVERVIEW

## MECHANICAL DESIGN

The system is built on a standalone embedded circuit board using the **PIC16F877A** microcontroller. The robot was made to comply with the maximum dimensional limits.

Sensors & Inputs:

- **Line Sensors:** Two IR sensors mounted at the front, underneath the body of the robot, facing downwards, to track the black line during the Line Following and Tunnel zones. These transducers output a logic high
- **Light Sensor:** An LDR sensor circuit positioned to detect the drop in ambient light when entering the tunnel.
- **Obstacle Avoidance Sensors:** An Ultrasonic (HC-SR04) at the front center and two IR sensors on either side, angled slightly outward to detect lateral obstacles, are mounted to scan the 100 cm wide path for detecting the randomly placed obstacle.
- **Start Button:** A Start pushbutton is placed with a pulldown resistor on pin RB0 to signal the start of the system.
- **Reset Button:** A Reset pushbutton is placed with a pull-up resistor on the Master-Clear ($\overline{MCLR}$) pin to signal a full reset of the system.

Actuators & Outputs:

- **Drive System:** Two DC motors are mounted at the front, controlled through an H-Bridge (L293D) module, and are provided with high-traction rubber tires to handle the 30-degree incline. A caster wheel at the rear provides stability.
- **Raising the Flag:** A Servomotor is mounted on the rear to act as a flag/indicator for the "Parking" completion signal. The input signal to the Servo, connected to pin RE2, is a PWM signal that controls the angle of motion.
- **LED Indicator:** A yellow LED in series with a current-limiting resistor shows that the system is in the active (driving) state.
- **Tunnel Buzzer:** This Piezo Buzzer is utilized during the tunnel and parking stages. In the dark tunnel, it beeps continuously to signal its presence. In the parking, it beeps every second to signal this stage.

**Power Management:** To optimize space and efficiency, the system utilizes a single 12 V power source. The high-current 12V rail powers the motors directly through the H-Bridge. The logic voltage (5V) is derived from the voltage regulator output, distributed to the microcontroller and sensors via a common 5V rail and ground.

## ELECTRICAL DESIGN

The MCU served as the centralized control unit responsible for:

- Analog sensor acquisition via ADC (LDR).
- Digital sensing via PORTB and PORTE inputs (IR sensors and Ultrasonic echo).
- Timing and delays (via Timer0 ISR).
- Actuator control (Motors, Buzzer, and LED).

Motor Control:

- **DC Motors:** An **L293D H-Bridge** driver interfaces the low-current microcontroller signals with the high-current motors. PORTD handles direction control (IN1-IN4), while RC1 and RC2 provide PWM signals for speed regulation.
- **Servo Control:** RE2 delivers repeated software-based 20 ms frames with 1 ms pulse width**.**

Sensor Interface:

- **Ultrasonic:** Trigger pin connected to RE0 (Output) and Echo pin to RE1 (Input).
- **Tunnel Sensor:** The LDR is configured in a voltage divider circuit connected to RA0 (Analog Input), allowing the ADC to measure light intensity.
- **Line-Following Sensors:** The left and right IR sensors are connected to pins RB1 and RB, respectively.
- **Obstacle Sensors:** The left and right obstacle IR sensors are connected to pins RB3 and RB4, respectively.

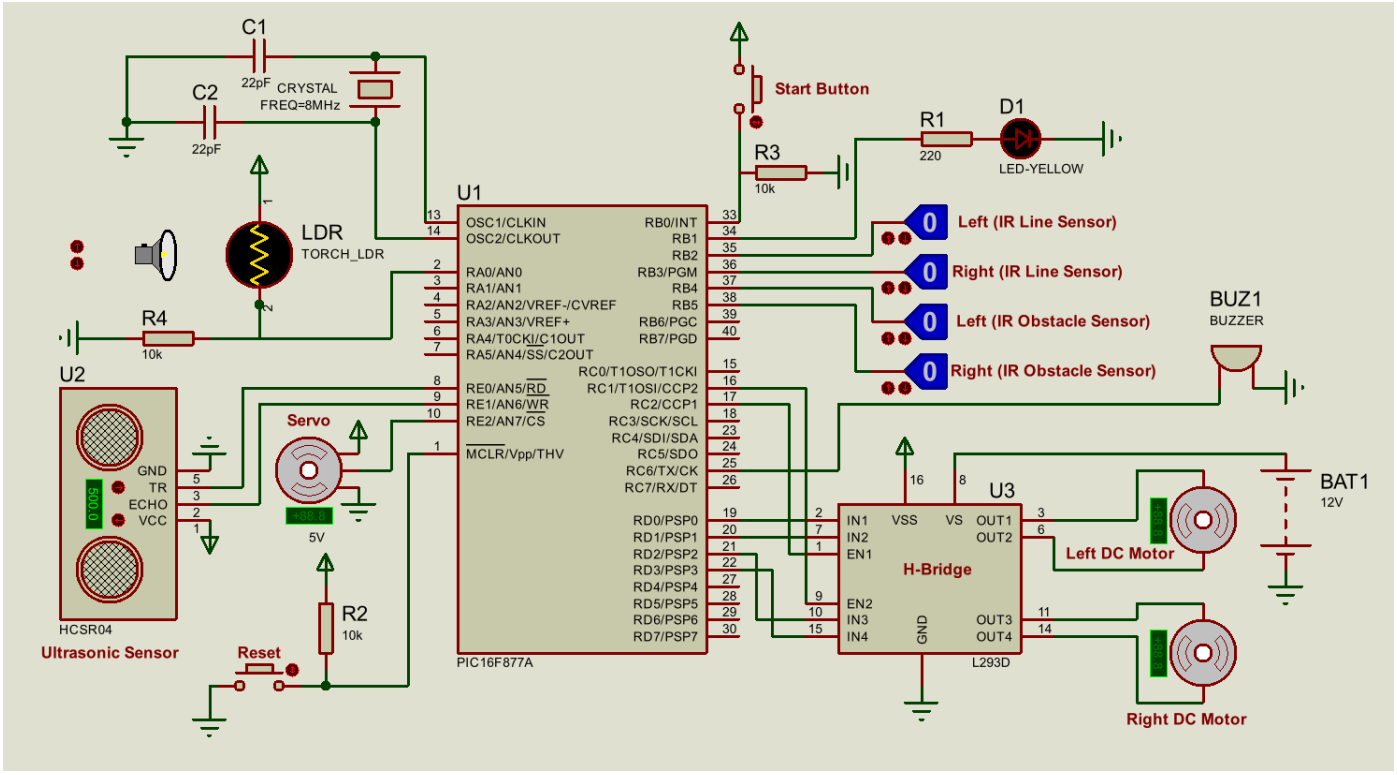


Figure 2: Hardware Design in Proteus

The electric connections of the sensors and actuators to the pins are shown in Figure 2. This design was implemented using the software tool Proteus. The IR sensors were modeled as logic states instead of the actual sensor for an easier view. However, in the physical implementation, the sensor outputs were connected to the pins, while the logic high power line and logic low ground were connected in their appropriate slots.

## SOFTWARE DESIGN

The firmware is written in C (MikroC environment) and is structured as a Finite State Machine (FSM) and a Timer0 Interrupt Service Routine (ISR). Bitwise operations (masking) were used for all register access instead of unions, as one of the restrictions of the challenge.

The robot was designed to successfully transition through the stages listed below in this specific sequence.

**Stages of Transition:** Start → Follow Line → Tunnel → Obstacle → Bump → Park

### Global Timing & Software PWM (The ISR)

Instead of using the hardware CCP PWM module, we implemented a **Software PWM** system using **Timer0**.

- **Interrupt Logic:** Every time TMR0 overflows, the ISR executes. It increments a global tick_ms counter (for timekeeping) and a pwm_counter (0-100).
- **Motor Control:** The ISR compares the pwm_counter against two global speed request variables (left & right motors). If the counter is lower, the pin is set HIGH; otherwise, it is LOW. This allows for independent speed control of both motors on standard digital pins.
- **Buzzer:** The ISR also handles the "beeping" frequency (peep flag), toggling RC6 to generate a toggling tone every second for the parking sequence.

### The Finite State Machine (Main Loop)

**Initialization & Start:** The robot waits for the START button press. A hardware timer (Timer0) interrupt counts exactly 3 seconds. Upon completion, the yellow LED is activated, and the robot moves to the Line-Following zone of mode 1 in the main loop.

The main() function operates in a continuous loop, switching between two primary modes:

- **Mode 1: Line Following & Tunnel**
  - ✓ **Line Logic:** Reads RB2 and RB3. If Left = 1, the robot turns Left. If Right = 1, it turns Right. If both are 1 (Intersection), it executes a 90-degree Pivot Turn.
  - ✓ **Tunnel Logic:** The ADC reads the LDR value. If it drops below a certain threshold, the InTunnel flag is set, and the buzzer activates and remains on until exiting the tunnel. The robot's speed is boosted in the tunnel to ensure it exits within 3 seconds, thereby avoiding penalties.
  - ✓ **Transition:** Upon exiting the tunnel (detecting light after darkness), the system automatically switches to mode = 2 (Obstacle Mode).

- **Mode 2: Obstacle Avoidance**

Upon losing the line, the robot enters the walled path. Sensors detect the randomly placed obstacle and the walls. The algorithm steers the robot around the object and realigns it toward the Bump.

- ✓ **Ultrasonic Fusion:** The robot polls the Ultrasonic sensor. If the distance < 55cm, it stops for a while before carrying on the proper obstacle-avoidance technique.
- ✓ **Decision Making:** It checks the side IR sensors (RB4, RB5). If the Left side is blocked, it pivots Right, and vice versa. If it is a dead end, it performs a U-turn.
- ✓ **Return to Line:** If the line sensors detect the track again, the robot switches back to mode = 1 and the intersection counter is set to 1. (The second "intersection" that will be detected is the bump).

## Parking Routine

When a third "intersection" is detected, the Parking_Routine() is triggered:

1. **Wall Approach:** Robot creeps forward until distance < 20 cm.
2. **Maneuver:** Executes a precise 90-degree turn.
3. **Flag Raise:** A specific loop commands the servomotor on RE2 to rotate the servo to 180°. (1 ms High, 19 ms Low).
4. **Finish:** Detects the black parking floor, stops, and disables the buzzer and LED.

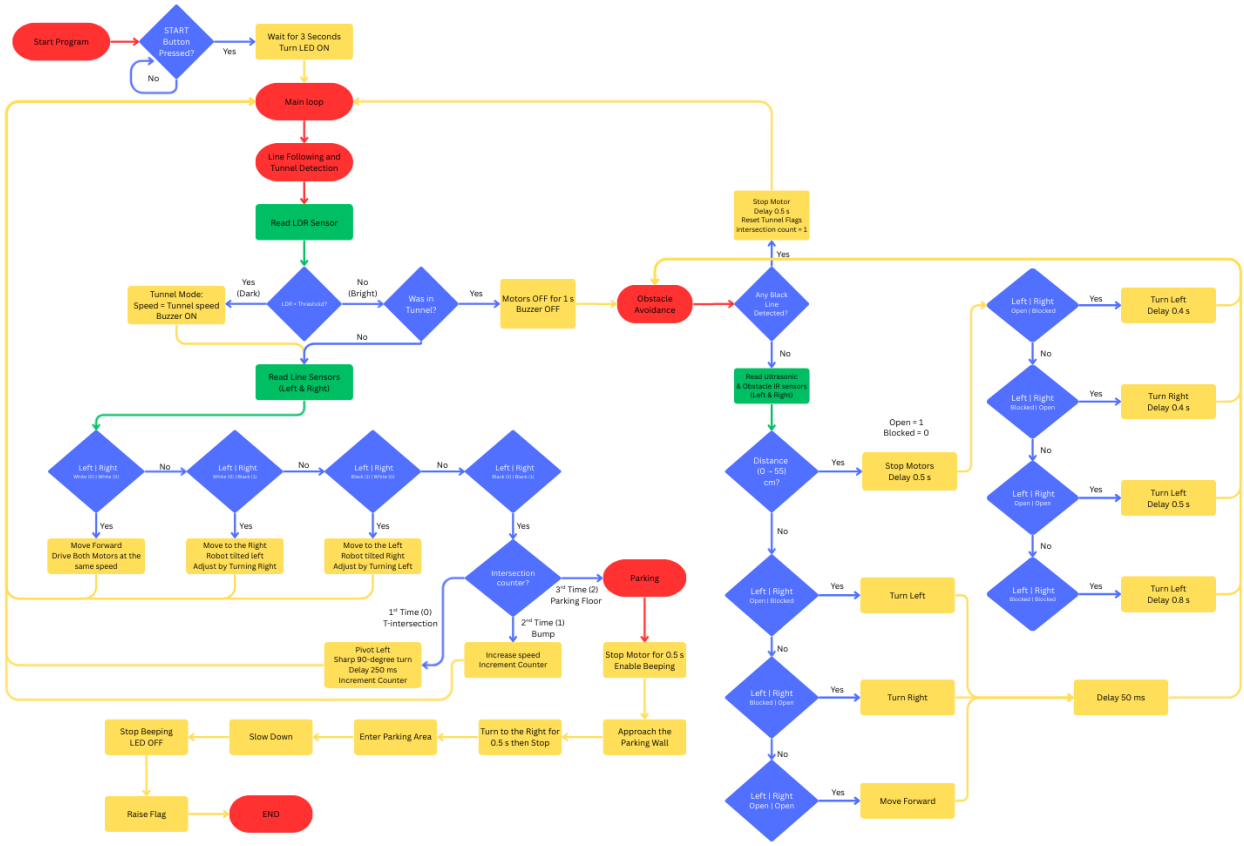


Figure 3: Flowchart Diagram

# GALLERY

In this section, we provide high-quality photographs of the robot we have built. All the critical parts are labeled.

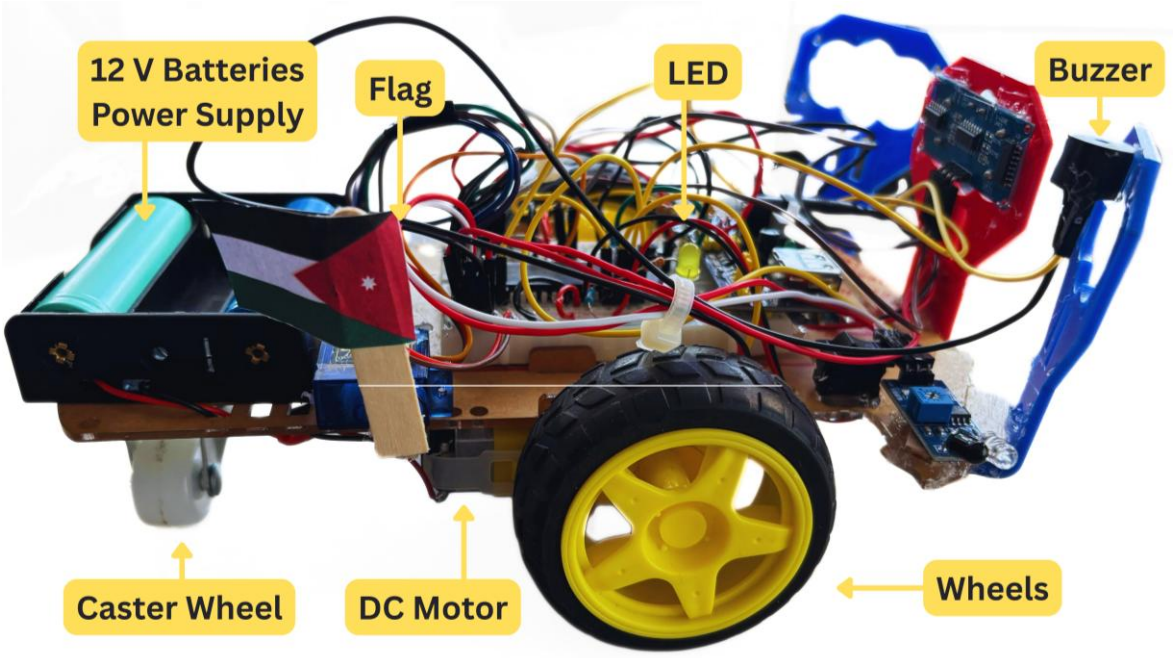


Figure 4: Side View

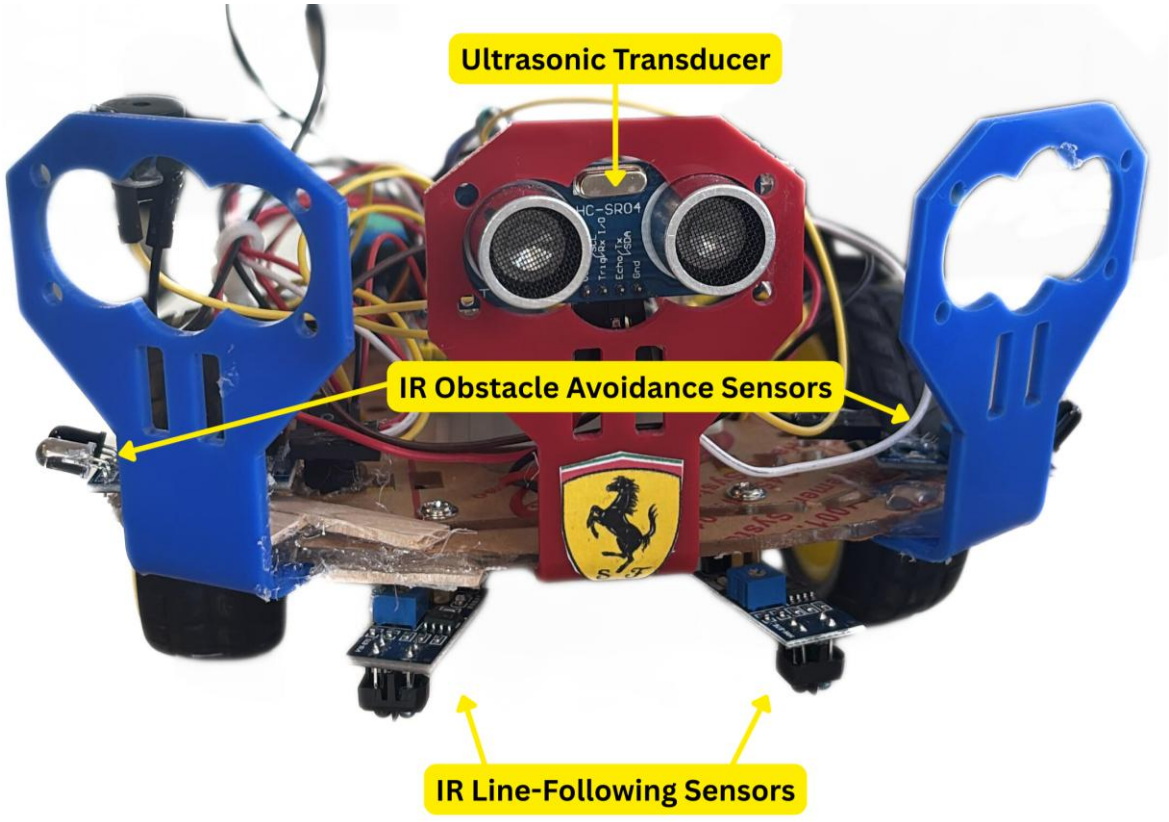


Figure 5: Front View

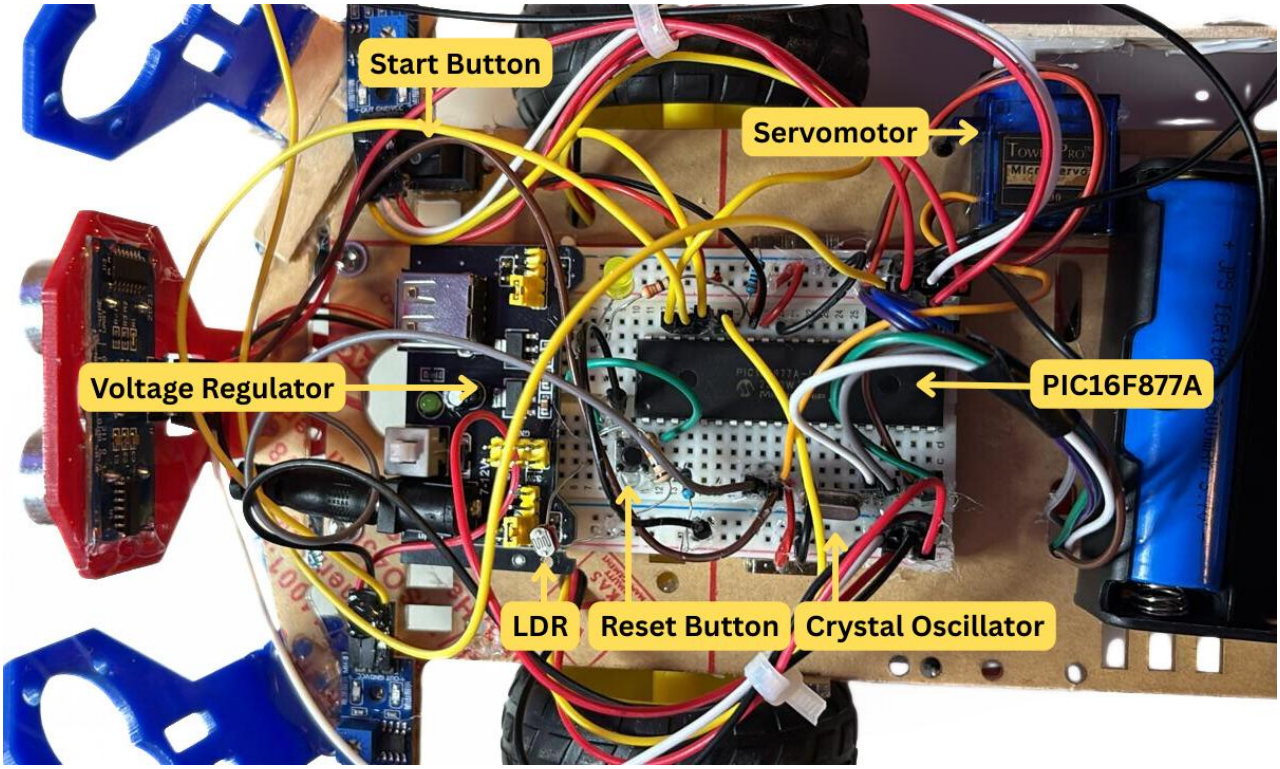


Figure 6: Top View

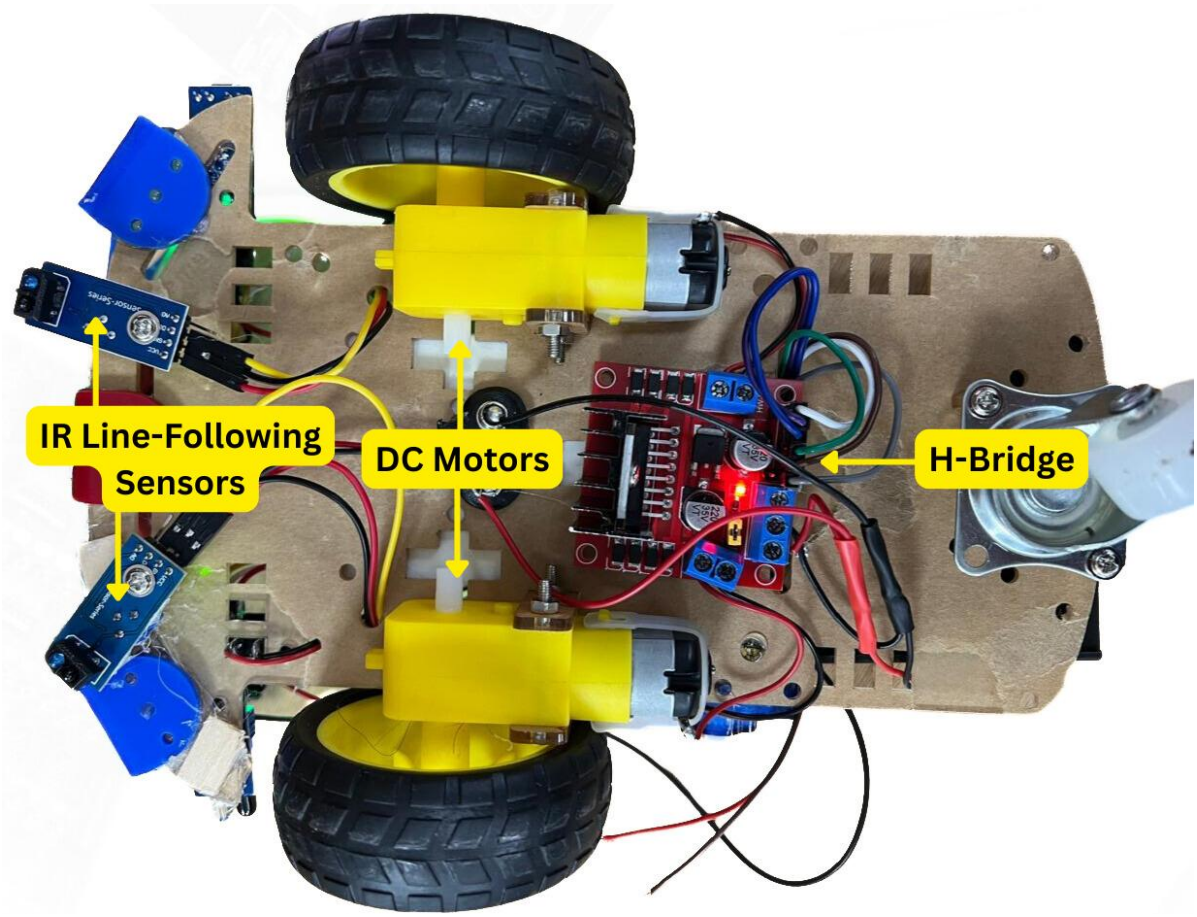


Figure 7: Bottom View

# PROBLEMS ENCOUNTERED AND RECOMMENDATIONS

While testing the robot in real life, we came across plenty of challenges and fine-tuning. Coding the MCU required a series of trial-and-error to gain feedback on whether or not our algorithms, threshold limits, and speeds were appropriate and adequate.

## Hardware Integrity and Component Troubleshooting

During the integration phase, we encountered significant hardware failures with the provided kit components.

**LDR Module Failure:** The stock Light Dependent Resistor module exhibited erratic behavior and inconsistent sensitivity. To resolve this, we bypassed the module and designed a custom **discrete voltage divider circuit**. This solution provided a linear and reliable analog voltage swing for the ADC, essential for accurate tunnel detection.

**H-Bridge Malfunction:** The initial tests yielded zero motor response despite correct logic signals. Extensive debugging revealed that the provided H-Bridge module was internally defective. This was a critical setback that initially halted progress; however, through resilient teamwork and rigorous troubleshooting, the component was replaced, and the drive system was successfully restored.

## False Positive Detection at the Incline (Bump)

**Problem:** The 30-degree incline presented a unique challenge for the IR line sensors. Due to the incline, the sensors interpreted the ramp as a "Black Line" (Intersection), causing the Finite State Machine (FSM) to prematurely trigger the **Parking Routine**.

**Solution:** We modified the software logic to treat this specific sensor pattern not as a stopping point, but as a "Momentum Stage." When this specific intersection type is detected after the Obstacle Zone, the software overrides the stop command and instead **increases the Motor PWM duty cycle**. This ensures the robot has sufficient torque and momentum to traverse the incline without stalling.

## Software Architecture and Future-Proofing

**PWM Implementation:** The current firmware utilizes a robust **Software PWM** generated via the Timer0 Interrupt Service Routine (ISR). This approach works effectively; however, during the PCB design phase, we strategically mapped the Motor Enable pins to **RC1 (CCP2)** and **RC2 (CCP1)**. This design choice ensures forward compatibility, allowing the system to be easily upgraded to hardware-based PWM in future iterations without altering the physical circuit.

**Servo Control:** Similarly, the Servo motor is currently controlled via bit-banging in the Timer0 ISR on pin **RE2**. While a CCP pin could have been used, we reserved the hardware PWM resources for the motors, demonstrating efficient resource management.

## System Timing and Mathematical Validation

Strict compliance with the "No Built-in Libraries" rule required manual calculation of all system timings based on the **8MHz Crystal Oscillator**.

**ADC & Timers:** The ADC conversion clock (Fosc/16) and Timer prescalers were calculated to ensure data integrity and correct overflow rates. Timer0 prescaler is (1:8) while Timer1 has a prescaler of (1:2).

**Ultrasonic Equation:** The distance calculation algorithm was derived manually as follows:

$$Distance = \frac{1}{2}(Time \times speed\ of\ sound)$$

$$Distance = \frac{1}{2}(Ticks \times PrescalerFactor \times instruction\ period \times speed\ of\ sound)$$

$$Distance\ (cm) = \frac{1}{2}\left(Timer1\ value \times 2 \times 0.5\ \mu s \times 0.034\frac{cm}{\mu s}\right) = 0.017 \times Timer1\ value$$

$$\boldsymbol{Distance\ (cm) \sim \frac{Timer1\ value}{58}}$$

Calibration and Compliance

**Trial and Error:** extensive field-testing was conducted to determine the optimal PWM duty cycles for different zones (e.g., low speed for parking, high speed for the tunnel). Threshold values for the LDR and Ultrasonic sensors were empirically tuned to filter out environmental noise.

**Regulatory Compliance:** All solutions were implemented strictly adhering to the challenge rules. No built-in delay functions, bit-shifting operations, or union data structures were used. All timing is interrupt-driven, and bit manipulation is performed via explicit logical masking (AND/OR operations).

# CONCLUSIONS

The project successfully demonstrated the capabilities of a differential-drive robot to operate autonomously in a multi-zone environment. The team gained deep insight into low-level embedded programming, specifically regarding bitwise port manipulation, ADC configuration, and interrupt-based timing.

The final prototype met all design requirements, navigating from the start line to the parking zone while effectively managing obstacles and environmental changes, and parks autonomously using the custom-coded algorithms. The project effectively combines mechanical design, electronic subsystems, and software control strategies using the PIC16F877A microcontroller.

Future improvements can integrate hardware PWM, enhanced sensing modalities, and more advanced navigation algorithms. Nonetheless, the current design demonstrates a robust end-to-end autonomous robotic system suitable for academic and competition environments.

# REFERENCES


[1] Princess Sumaya University for Technology, "Course 22442: Autonomous Mobile Robots Challenge Project Document," Fall 2025.

[2] **T. Wilmshurst**, *Designing Embedded Systems with PIC Microcontrollers: Principles and Applications*, 2nd ed. Oxford, U.K.: Newnes, 2010. ISBN: 978-1-85617-750-4.

[3] **Microchip Technology Inc.**, "PIC16F877A Data Sheet: 28/40/44-Pin Enhanced Flash Microcontrollers," Chandler, AZ, 2013. [Online]. Available: https://www.microchip.com.

[4] **STMicroelectronics**, "L293D Push-Pull Four Channel Driver with Diodes," Datasheet, 2000. [Online]. Available: https://www.st.com/resource/en/datasheet/l293d.pdf.

[5] **Vishay Semiconductors**, "TCRT5000 Reflective Optical Sensor with Transistor Output," Rev. 1.9, 2023. [Online]. Available: https://www.vishay.com/docs/83760/tcrt5000.pdf.

[6] **HandsOn Technology**, "IR Obstacle Avoidance Sensor Module User Guide," Version 1.0. [Online]. Available: https://www.handsontec.com/dataspecs/sensor/IR%20Obstacle%20Detector.pdf.

[7] **Components101**, "Light Dependent Resistor (LDR) Datasheet." [Online]. Available: https://components101.com/sites/default/files/component_datasheet/LDR%20Datasheet.pdf.

[8] **Multicomp Pro**, "MCKP Series Piezo Buzzer Datasheet," Farnell, 2016. [Online]. Available: https://www.farnell.com/datasheets/2171929.pdf.

[9] **Kingbright**, "L-53YD 5mm Yellow LED Datasheet," Farnell, 2013. [Online]. Available: https://www.farnell.com/datasheets/1660999.pdf.

[10] **ElecFreaks**, "HC-SR04 Ultrasonic Ranging Module User Guide," 2011. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[11] **Tower Pro**, "SG90 Micro Servo Data Sheet," 2014. [Online]. Available: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf.

[12] **MikroElektronika**, "MikroC PRO for PIC Library Reference Manual," 2024.