Princess Sumaya University for Technology

King Abdullah II School for Electrical Engineering

Computer Engineering Department

Supervisor: Prof. Esam Qaralleh

Section: 2
Embedded Systems
22442

Name	Student ID	Major
Hala Al - Zaben	20221066	NIS
Ruwaya Al -Mahadin	20220314	NIS
Zeyad Al -Shweikh	20220337	NIS

Table of Contents

Abstract 3

Introduction 3

Components Used…….....4

Mechanical Design ……...6

Electrical design……8

Software design ……..…9

Pins connections ……11

Problems Encountered and Solutions Applied……...…14

Results 15

Conclusion.....16

Table of Figures

Figure 1 : Two Wheel Rover 6

Figure 2 : Two Wheel Rover 7

Figurec3 : Two Wheel Rover ……..……...…….…….....7

Figure 4 : Electrical Design ……..……..……...8

Figure 5 : Software Design ……...……9

Abstract

This project presents the design and implementation of an autonomous mobile robot using the **PIC16F877A microcontroller**. The robot is capable of navigating a multi-zone track including line-following, tunnel traversal, obstacle avoidance, bump climbing, and precise parking. Using multiple sensors—line sensors, light sensors, and distance sensors—the PIC16F877A processes inputs in real time and controls the differential-drive wheels to move autonomously.

A push-button starts the system with a precise delay, after which LEDs indicate active driving. The robot follows the black line, signals tunnel entry via a buzzer, avoids obstacles in unmarked areas, climbs ramps smoothly, and executes controlled parking with audio and visual feedback. This project demonstrates the effectiveness of embedded systems and low-level PIC programming in creating an intelligent, responsive, and reliable autonomous vehicle.

Introduction

Autonomous mobile robots are robotic systems capable of navigating and performing tasks without direct human intervention. These robots rely on sensors, control algorithms, and embedded systems to perceive their environment, make decisions, and execute movements accurately. The increasing use of autonomous robots in industries, research, and education highlights the importance of real-time control and intelligent navigation.

This project focuses on developing an embedded system-based autonomous robot using the **PIC16F877A microcontroller**. The robot is designed to navigate a multi-zone track, including line-following, tunnel detection, obstacle avoidance, ramp climbing, and precise parking. By integrating multiple sensors and real-time control logic, the system demonstrates how low-level embedded programming and hardware-software integration can create a reliable and responsive mobile robot capable of completing complex tasks autonomously.

Components Used

1. Processing and Control

Component	Quantity	Description
PIC16F877A Microcontroller	1	Central control unit responsible for sensor processing, state management, timing, PWM generation, and motor control

2. Motion and Drive System

Component	Quantity	Description
DC Gear Motors	2	Provide independent left and right wheel motion for differential-drive navigation
Motor Driver IC (e.g., L298N / L293D)	1	Interfaces between PIC16F877A and DC motors to control speed and direction
Wheels	2	Attached to DC motors
Caster Wheel	1	Provides balance and smooth turning

3. Indicators and Alerts

Component	Quantity	Description
Buzzer	1	Activated during tunnel traversal and system alerts (RB6)
LEDs	2-3	Indicate active system state and operational feedback

4. Sensors

Sensor	Quantity	Code Evidence / Function
Line-Following IR Sensors	2	Used on RB3 and RB4 to detect black line (`IR_LEFT_MASK`, `IR_RIGHT_MASK`)
Front IR Obstacle Sensor	1	Connected to RB0 (`IR_SENSOR_MASK`) for forward obstacle detection
Left IR Obstacle Sensor	1	Connected to RB5 (`IR_LEFT_OBJ_MASK`)
Right IR Obstacle Sensor	1	Connected to RB7 (`IR_RIGHT_OBJ_MASK`)
Light Dependent Resistor (LDR)	1	Connected to RA0 (AN0) to detect tunnel lighting conditions
Ultrasonic Sensor (HC-SR04)	1	Used in PATH state for distance-based obstacle detection (RC4 TRIG, RC5 ECHO)

5. User Interface and Control

Component	Quantity	Description
Push Button	1	Starts the robot operation (used with software delay and state transition)
Reset Button	1	Resets the microcontroller
Power ON/OFF Switch	1	Controls main system power

6. Power Supply and Circuitry

Component	Quantity	Description
Battery Pack	1	Provides portable power to motors and control circuitry
Voltage Regulator (7805 or equivalent)	1	Regulates voltage to stable 5V for PIC and sensors
Standalone Embedded PCB	1	Hosts the PIC16F877A and all peripheral connections

7. Mechanical Structure

Component	Quantity	Description
Robot Chassis (2WD or Custom)	1	Mechanical frame meeting size constraints
Mounting Hardware (screws, brackets)	As needed	Secures motors, sensors, and PCB

Mechanical Design

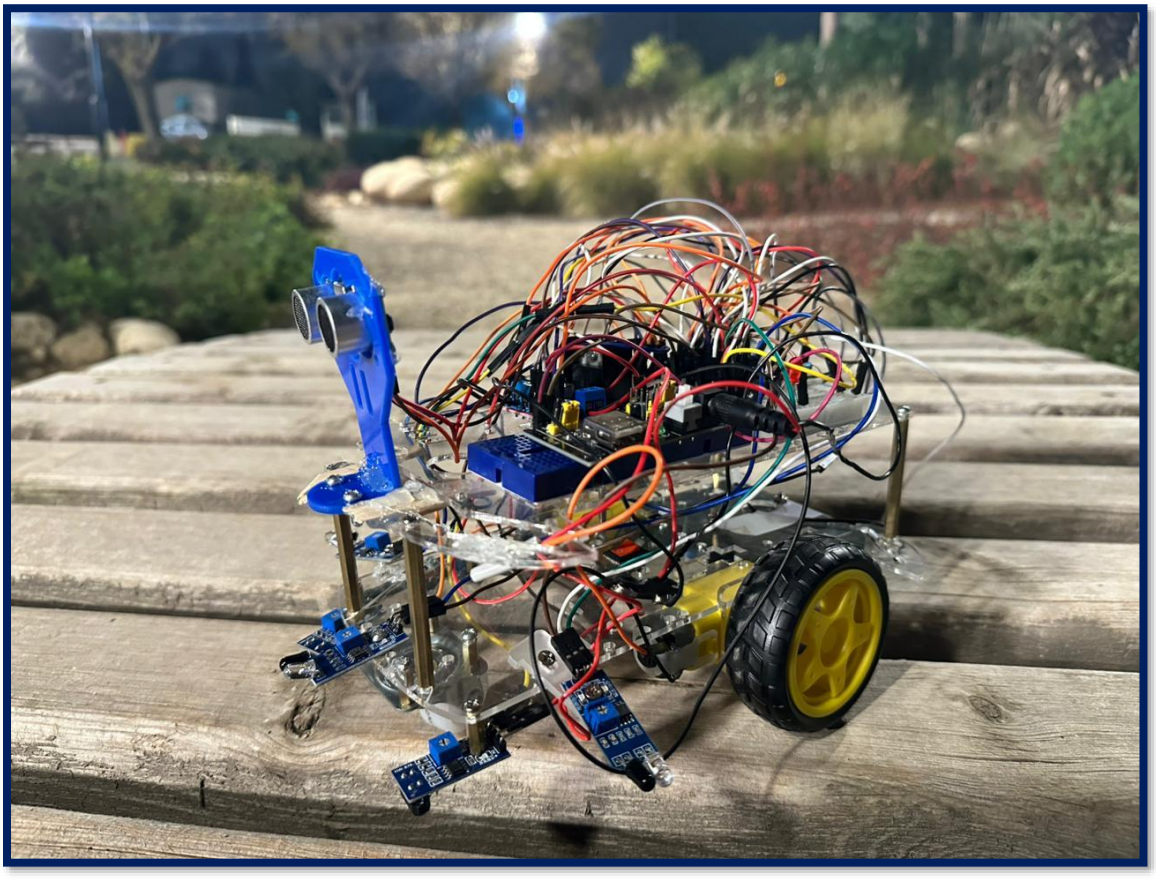

Figure 1 : Two Wheel Rover

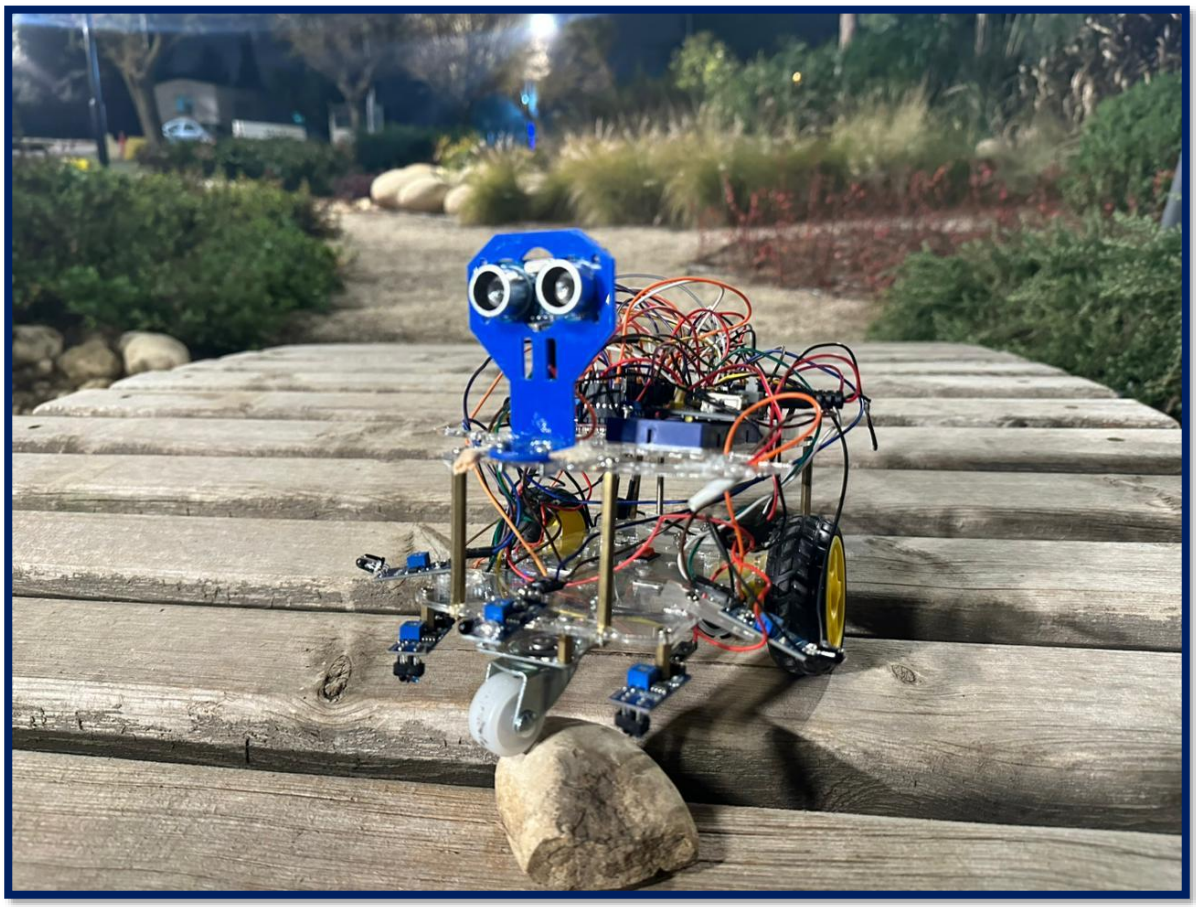

Figure 2 : Two Wheel Rover

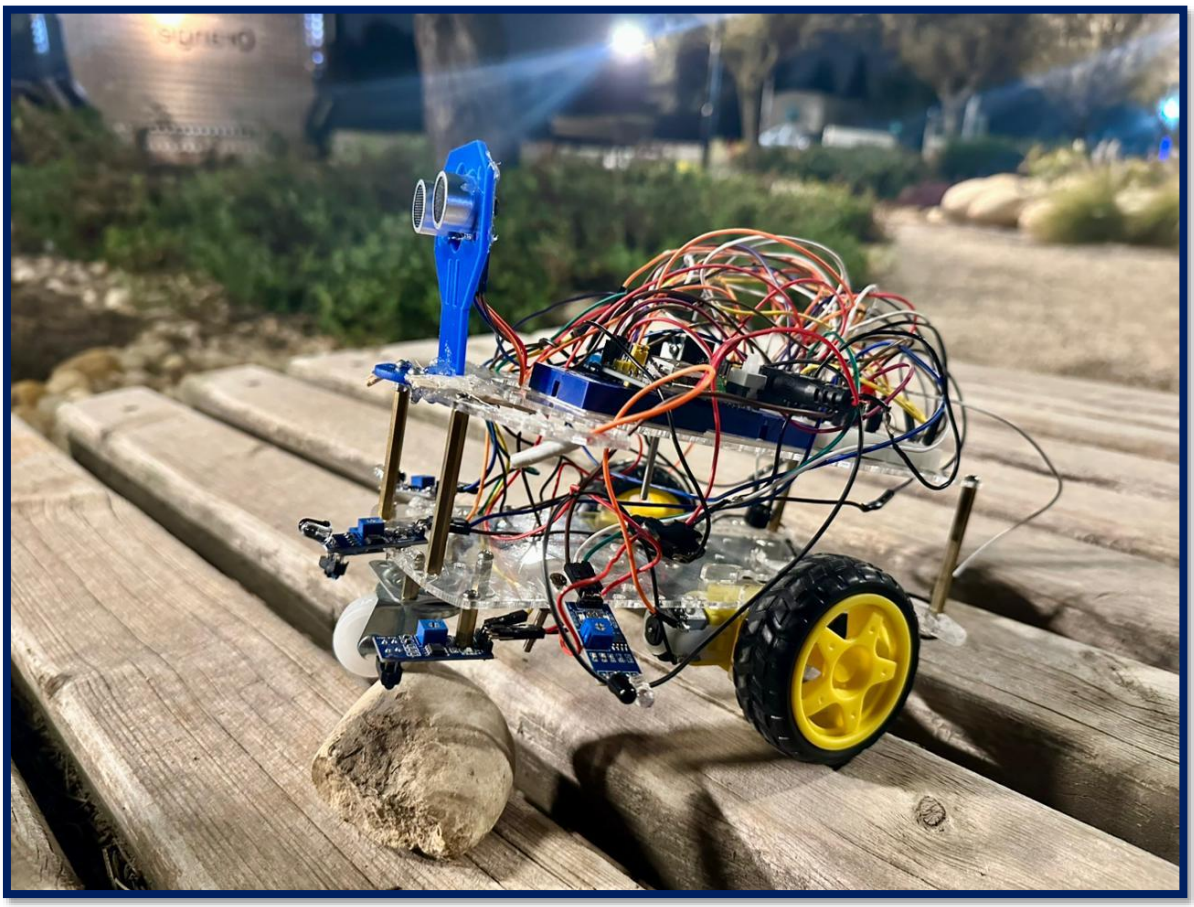

Figure 3 : Two Wheel Rover

Electrical design

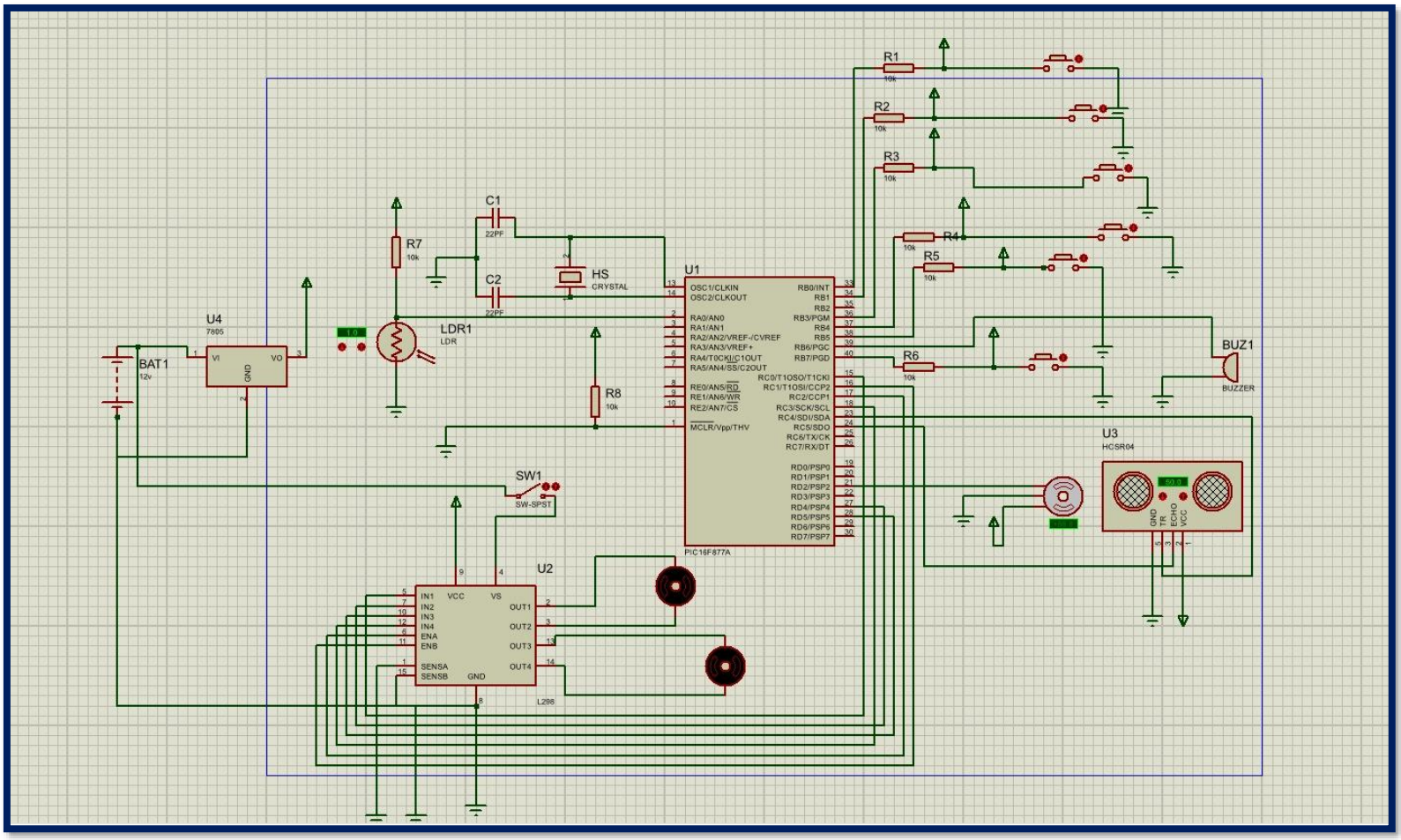

Figure 4: Electrical design

Software design

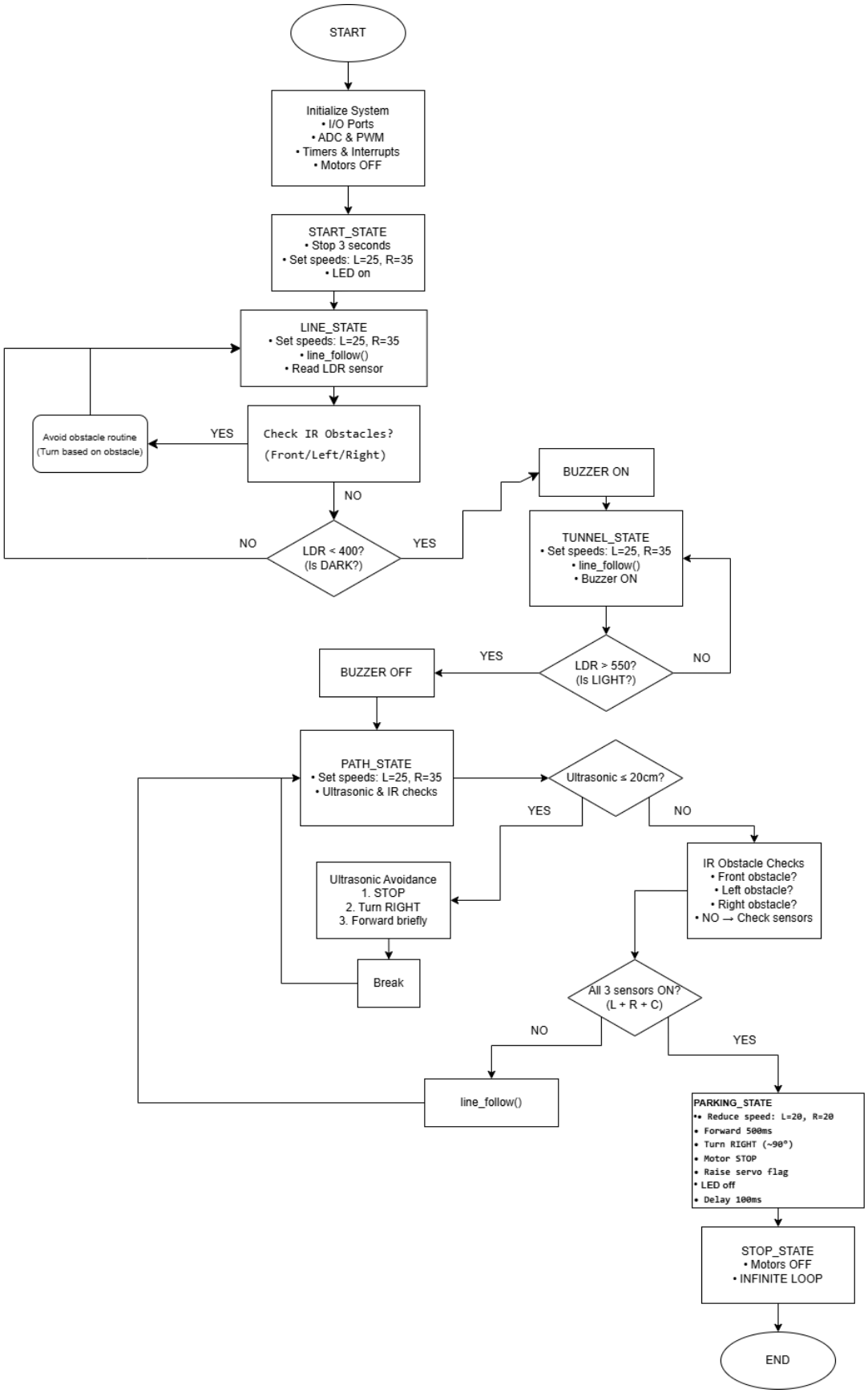

Figure 5 : Software Design

Flow chart explanation :

1. Initialization begins with configuring I/O ports, ADC, PWM modules, timers, and interrupts while keeping motors inactive.

2. Start-up procedure includes a 3-second pause followed by setting initial motor speeds to Left = 25 and Right = 35.

3. Line-following phase continuously executes the line-following algorithm while monitoring the LDR sensor; if darkness is detected (LDR < 400), the system transitions into tunnel navigation with the buzzer activated.

4. Tunnel navigation maintains line tracking until the LDR detects sufficient light (LDR > 550), at which point the buzzer is turned off and the system proceeds.

5. Path navigation with obstacle detection utilizes ultrasonic sensing to avoid obstacles within 20 cm by stopping, turning right, and moving forward briefly, while IR sensors simultaneously scan for a parking marker.

6. Parking routine is triggered when all three IR sensors are activated, leading to reduced speed, a full stop after 90 ms, and motor shutdown.

7. System termination places the robot in an infinite idle loop, concluding the autonomous cycle.

Pins connections

1.IR LINE FOLLOWING SENSORS

Sensor	Function	PIC Port	Pin	Mask
Left Line IR	Line detection	PORTB	RB3	`IR_LEFT_MASK (0x08)`
Center Line IR	Line detection	PORTB	RB1	`IR_CENTER_MASK (0x02)`
Right Line IR	Line detection	PORTB	RB4	`IR_RIGHT_MASK (0x10)`

- Used in: LINE_STATE, TUNNEL_STATE, PATH_STATE

2. IR OBSTACLE SENSORS

Sensor	Direction	PIC Port	Pin	Mask
Front IR	Obstacle detection	PORTB	RB0	`IR_SENSOR_MASK (0x01)`
Left IR	Wall / object	PORTB	RB5	`IR_LEFT_OBJ_MASK (0x20)`
Right IR	Wall / object	PORTB	RB7	`IR_RIGHT_OBJ_MASK (0x80)`

- Used in: `PATH_STATE`

3. ULTRASONIC SENSOR (HC-SR04)

Signal	PIC Port	Pin	Mask
TRIG	PORTC	RC4	`US_TRIG_MASK (0x10)`
ECHO	PORTC	RC5	`US_ECHO_MASK (0x20)`

- Used **only in `PATH_STATE`**
- Distance threshold: **≤ 20 cm**

4. LDR (LIGHT SENSOR) – ADC Channel : AN0

Component	PIC Port	Pin	Function
LDR (Analog)	PORTA	RA0 / AN0	Tunnel detection

- Dark: `< 400`
- Light exit: `> 550`
- Used in: `LINE_STATE`, `TUNNEL_STATE`

5.BUZZER

Component	PIC Port	Pin	Mask
Buzzer	PORTB	RB6	`BUZZER_MASK (0x40)`

- ON inside tunnel
- OFF outside tunnel

6. MOTOR DRIVER CONNECTIONS (L298 / L293 / H-Bridge)

I. Direction Control Pins

Motor	Signal	PIC Port	Pin
Left Motor	IN1	PORTC	RC0
Left Motor	IN2	PORTD	RD4
Right Motor	IN3	PORTD	RD5
Right Motor	IN4	PORTC	RC3

II. PWM Speed Control

Motor	Enable Pin	PIC Pin	CCP
Left Motor	ENA	RC2	CCP1
Right Motor	ENB	RC1	CCP2

- PWM Frequency ≈ **4 kHz**
- Speed controlled via `CCPR1L`, `CCPR2L`

7.TIMER & INTERRUPTS

Module	Purpose
Timer0	Software delay (`delay_ms`)
Timer1	Ultrasonic echo timing
CCP1 / CCP2	PWM motor control

9.Servo

Component	PIC Port	Pin	Function
Servo	PORTD	RD2	Raise / lower flag at PARK_STATE

8. UNUSED / FREE PINS (AVAILABLE)

Port	Pins
PORTA	RA1–RA5 (except RA0)
PORTC	RC6, RC7
PORTD	RD0,RD1,RD3, RD6, RD7
PORTE	All

Problems Encountered and Solutions Applied

1. In general, PIC16F877A is a pin-sensitive microcontroller, we faced problems of pins getting broken by getting it in and out of the kit and breadboard. Also, when connecting an unsuitable voltage source, the PIC will be burnt and this was also one of our problems. Also, we faced connecting the pic upside down which got us to connect the VDD pin to one of the pin ports.

Our way to deal with such an issue was replacing the pic with another new one.

2. At the first we place the line following sensors modules too closely together, the problem was that the sensors interfere with each other's readings, leading to inaccurate results.

Our approach to addressing this issue was to ensure adequate spacing between the sensors to minimize interference between sensors.

3. One of the issues encountered was related to the buzzer connected to pin RD2. The buzzer drew excessive current, which caused system instability and occasionally shut down the car.

To resolve this issue, the buzzer was reassigned to pin RB6, which provided a more suitable connection and ensured reliable operation without affecting the rest of the system.

Results

The autonomous mobile robot was tested on the complete challenge track consisting of line-following, tunnel traversal, path navigation with obstacles, bump handling, and parking. The robot successfully completed all required zones in the correct sequence without manual intervention.

At system start-up, the robot remained stationary for a fixed three-second delay, after which it transitioned smoothly into the line-following mode. During the line-following zone, the robot accurately followed straight lines, curves, and intersections using the IR line sensors, maintaining stable motion through controlled PWM motor speeds.

When entering the tunnel, the LDR sensor detected the reduction in light intensity, triggering the buzzer as an audible indicator. The robot continued line-following inside the tunnel and exited within the allowed time. Upon detecting normal lighting conditions, the buzzer was automatically deactivated and the robot transitioned to the next operational state.

In the path navigation zone, where no guiding line was present, the robot relied on ultrasonic and IR obstacle sensors to detect nearby obstacles. The system correctly identified obstacles within the defined distance threshold and executed avoidance maneuvers by stopping, turning away from the obstacle, and proceeding forward. These maneuvers were performed once per obstacle, preventing repeated or unstable reactions.

Upon reaching the parking area, the robot detected the black surface using its IR sensors, reduced its speed, and executed a controlled parking sequence. The robot stopped near the boundary and successfully raised the flag using the servo motor, indicating task completion. Finally, the system entered a stop state and remained idle.

Conclusion

This project successfully demonstrated the design and implementation of an autonomous mobile robot using the PIC16F877A microcontroller. By integrating multiple sensors, actuators, and real-time control logic, the robot was able to navigate a multi-zone environment autonomously and complete all required tasks.

The use of a state-based control architecture simplified system behavior and improved reliability by clearly separating the functionality of each operational phase. Low-level programming techniques, including direct register manipulation, custom timing routines, and interrupt handling, ensured compliance with course constraints and provided precise control over system behavior.

The project highlights the importance of sensor calibration, hardware-software integration, and structured embedded design in autonomous systems. Despite minor limitations related to environmental variations and fixed threshold values, the overall system performed effectively and consistently during testing.

In conclusion, the project met its objectives and provided valuable hands-on experience in embedded systems design, real-time programming, and autonomous robot control. The developed system demonstrates a solid foundation that can be extended in future work to include more advanced control strategies and adaptive behaviors.