Princess Sumaya University for Technology

King Abdullah II School of Engineering

Computer Engineering Department

EMBEDDED SYSTEMS
FINAL PROJECT
AUTONOMOUS CAR CHALLENGE

Authors:

Leen Salman	20220183	Communication Engineering & IoT
Sara Habash	20220757	Electrical Power and Energy Eng.
Jana Jaser	20220641	Communication Engineering & IoT

Supervisor:

Prof. Belal Sababha

January 14, 2026

Abstract

This project presents the design and implementation of an autonomous mobile car based on the PIC16F877A microcontroller. The vehicle is designed to autonomously navigate a multi-zone track consisting of a start zone, line following path, tunnel, obstacle avoidance path, inclined bump, and a final parking area.

The system integrates multiple sensors and actuators to perceive the environment and react accordingly, including the line detectors, line-intensity sensor, obstacle avoidance sensors, and motor control. The embedded software is written in C programming language, enabling the robot to switch reliably between operating modes corresponding to each track zone. Core embedded system concepts such as timers, analog-to-digital conversion (ADC), pulse-width modulation (PWM) for motor control, and voltage regulation are applied to achieve real-time performance and stability.

The implemented prototype successfully demonstrates the practical application of embedded systems in autonomous robotics and highlights the integration of hardware and software components to achieve reliable and efficient autonomous navigation.

TABLE OF CONTENTS

1 Introduction 2

1.1 Objectives 2

2 System Overview 3

3 Hardware Mechanical and Electrical Design 4

4 Software Design 7

5 Problems and Recommendations 8

6 Results 9

7 Conclusion 10

8 References 11

1 INTRODUCTION

Embedded systems are specialized computing systems designed to perform specific tasks within a larger electrical or mechanical system. Unlike general-purpose computers, embedded systems operate under strict constraints related to timing, power consumption, and reliability, and they are commonly used in applications such as robotics, automotive systems, industrial automation, and consumer electronics. These systems typically integrate a microcontroller, sensors, actuators, and dedicated software to interact with the physical environment in real time. Understanding embedded systems requires practical knowledge of hardware interfacing, low-level programming, real-time control, and efficient use of limited system resources.

This project focuses on the design and implementation of an autonomous mobile car using the PIC16F877A microcontroller as part of the Autonomous Mobile Robots Challenge. The system is designed as a fully standalone programmed vehicle capable of navigating a multi-zone track that includes a start zone, line-following section, tunnel, obstacle-based path navigation, an inclined bump, and finally a parking area. The mobile car relies on multiple sensors and actuators, along with a C software, to respond to different environmental conditions and transition between operating modes. The project highlights the integration of hardware and software components and demonstrates the application of embedded system concepts such as timers, analog-to-digital conversion (ADC), PWM motor control, voltage regulation, and state-based software design

1.1 OBJECTIVES

The specific objectives of the project are as follows:

- To design a self-driven mobile robot that satisfies the required size and mechanical constraints.
- To develop an embedded control system based on PIC16F877A microcontroller, including proper power supply, voltage regulation, and reset circuitry.
- To implement embedded software in C language using MikroC PRO without using built-in libraries to ensure full understanding and control of all components.
- To utilize core embedded system features such as timers, ADC , and PWM for real-time system operation.
- To achieve accurate and stable line-following behavior, including correct handling of straight paths, curves, and intersections.
- To detect and respond to reduced lightning conditions inside the tunnel by activating indicators, sensors, and ensuring timely exist .
- To perform autonomous path navigation and obstacle avoidance in areas without guiding lines.
- To execute precise parking behavior, including speed reduction, boundary detection, and visual indication during the parking process.
- To document the complete system design, implementation, challenges, and results in a clear and professional manner supported by source code, images, and demonstration video.

2 SYSTEM OVERVIEW

The autonomous mobile car is designed as a self-operating embedded system that can complete the entire track without any human assistance. The system brings together mechanical, electrical, and software components to read the environment, process information in real time, and control the movement of the car. The vehicle moves using two independently controlled motors, which gives it the ability to move forward, adjust its direction smoothly, and perform turns as needed throughout the track.

The robot follows a structured sequence of operating zones arranged as:

Start → Line Following → Tunnel → Path Navigation → Bump → Parking

The path is illustrated in Figure 1. Each zone represents a distinct operating condition and requires different control strategies. As the robot moves through the track, it transitions between zones based on sensor readings and timing conditions. This design approach separates system functions clearly, improves system reliability, and simplifies software debugging and maintenance.

The PIC16F877A microcontroller is the brain of the system which handles sensor readings, decision making, and control of the motors and other output devices. The robot relies on 2 line follower sensors to follow the black line, a light dependent resistor (LDR) that detects low-light conditions inside the tunnel, 2 infrared radiation obstacle avoidance sensors (IR)s and one Ultrasonic sensor to identify obstacles in the navigation area. Output devices such as DC motors, LEDs, a buzzer, and a servo motor are driven using digital I/O pins, PWM signals, and timers. All system components are powered by a voltage regulated power supply, which allows the robot to operate fully on its own. Overall, the system is designed to be simple, reliable, and accurate, while meeting the project requirements and demonstrating effective integration between embedded hardware and software.

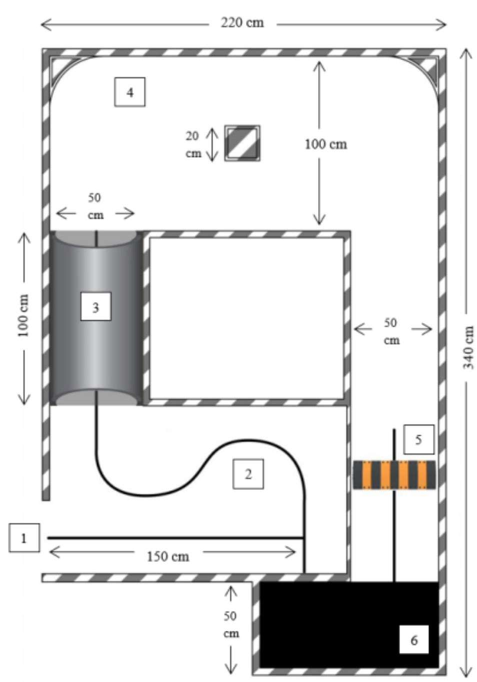


Figure 1: Obstacle Coursse

3 HARDWARE MECHANICAL AND ELECTRICAL DESIGN

The robot chassis used in this project was provided beforehand, and all mechanical and electrical components were assembled manually by the team. The assembly process started by testing then installing the DC motors onto the chassis securely and aligned properly. Several tests were performed to ensure that the motors were functioning correctly and rotating in the expected direction. Mechanical adjustments were then made to balance the motors and align the wheels so that the robot could move forward in a straight line without deviating to one side. Figure 2 shows the hardware schematic of the system.

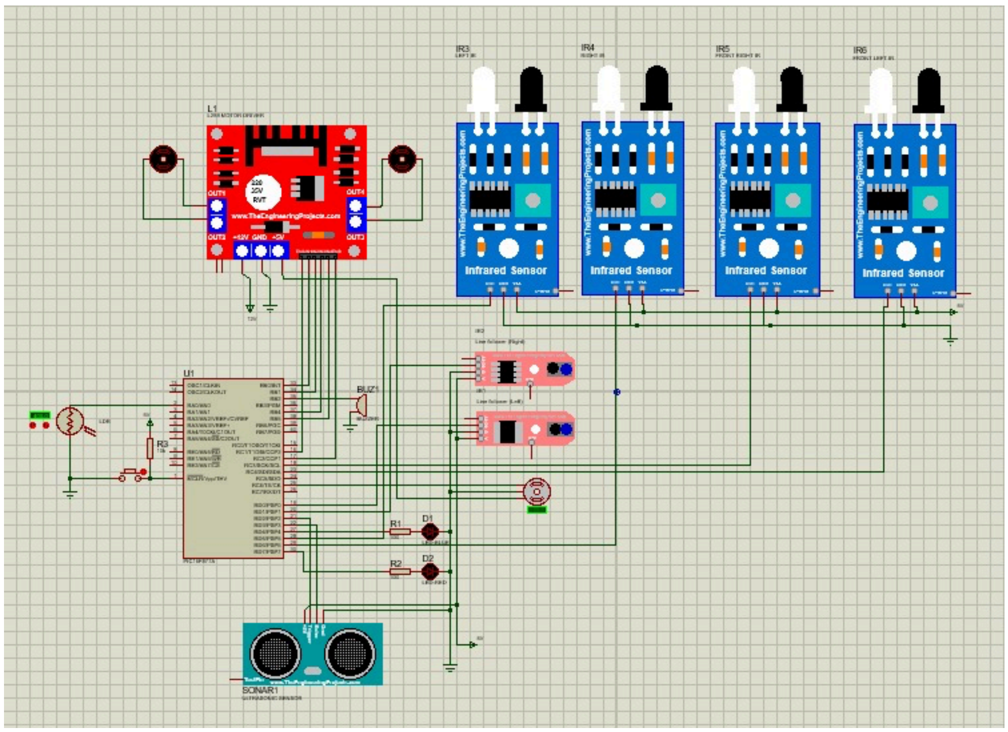


Figure 2: Hardware System Schematic

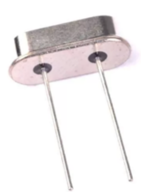
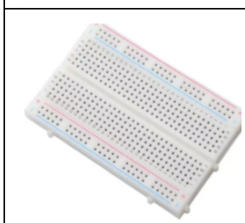
After confirming stable motor movement, additional sensing and output components were gradually added to the robot. Two line-following sensors were placed at the front edges of the car to detect the black line against the white surface. Their placement allowed the robot to adjust its direction by steering toward the sensor that detected the black line, ensuring accurate tracking along straight paths, curves, and intersections. For tunnel detection, an LDR was installed to monitor changes in ambient light intensity, while a buzzer was used to provide an audible indication when the robot entered the tunnel.

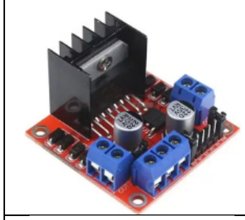
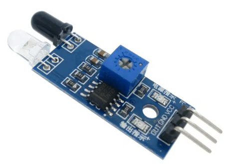
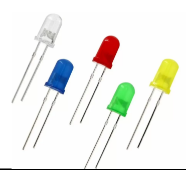
An ultrasonic sensor was then tested and placed at the front of the car, along with four IR sensors, two on each side of the robot, to enable obstacle detection during the path navigation zone. Additional output devices, such as LEDs and a servo motor used to raise a flag, were added during the final stages of the project to indicate successful parking. All components were positioned in a way to avoid mechanical interference, ensure reliable sensor readings, and maintain organized wiring throughout the system. To further improve organization and simplify debugging, all wires were color-coded, allowing easy identification of power, ground, and signal connections for each component. All components used can be seen in Table 1.

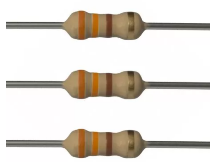
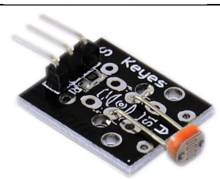
On the electrical side, a breadboard was used throughout the development process to prototype and test all circuit connections. The DC motors were connected to the PIC microcontroller through an H-bridge motor driver, which allowed the motors to be controlled in both direction and speed. A power switch was added between the battery supply and the circuit to allow controlled startup of the system. The batteries were the main source of power which provided sufficient current for the motors and the rest of the system while the servo motor took power directly from the H-bridge.

To ensure stable operation of the microcontroller and sensors, a voltage regulator was added to supply a regulated 5V output to the PIC. This helped protect sensitive components and improve overall system reliability. Wiring was initially done using jumper wires connected to a power supply for testing, but soldering was later applied where needed to reduce loose connections and improve durability. Overall, the mechanical and electrical design process was much iterative, involving continuous testing and adjustments to ensure that all components worked together reliably as a complete autonomous system.

Table 1: Overview of each component.

COMPONENT	**Component Name**	**Input/Output**	**Digital / Analog**	**Functionality**
	PIC16F877A [1]	-	Digital & Analog	Acts as the system controller, reading sensor inputs, processing them, and driving outputs such as motors and indicators.
	Clock Oscillator [2]	-	Digital	Provides a digital clock signal to the microcontroller and acts as an internal timing input
	2WD Robot Car Body [3]	-	-	Provides the mechanical structure for all robot com ponents and supports movement.
	Breadboard 400 Tie-points [4]	-	-	Used for testing and connecting all circuit connections without soldering.

	L298N Dual H-Bridge Module [5]	Input & Output	Digital	Drives and controls the direction and speed of the DC motors using control signals from the microcontroller.
	HC-SR04 Ultrasonic Sensor [6]	Input	Analog	Measures distance to obstacles using ultrasonic waves for path navigation and obstacle avoidance.
	IR Infrared Obstacle Avoidance Module [7]	Input	Digital	Detects nearby objects and boundaries to assist with obstacle avoidance during navigation.
	Active Buzzer 5V [8]	Output	Digital	Generates an audible sound to indicate tunnel entry and system status.
	5mm LED Light (Different Colors) [9]	Output	Digital	Provides visual indication for system states such as parking and active operation.
	Voltage regulator [10]	-	-	used to provide a stable and safe voltage to the circuit components,
	TCRT5000 Infrared Reflective Sensor Module [11]	Input	Analog	Detects black and white surfaces for accurate line-following behavior.
	Mini Servo Motor SG90 [12]	Output	Digital	Raises a flag to indicate successful parking at the end of the track.

	Push Button Switch [13]	Input	Digital	Starts the system when pressed by the user in the start zone.
	10k Ohm Resistor 1W [14]	-	-	Used for pull-up or pull-down configurations and voltage control in sensor circuits.
	330 Ohm Resistor 1/4W [15]	-	-	Limits current to LEDs to protect them from damage.
	KY-018 Photosensitive Sensor Module (LDR) [16]	Input	Analog	Detects changes in light intensity for tunnel detection.
	Wires [17]	-	-	Used to connect components
	3x Lithium Battery 18650 3.7v [18]	Input	Analog	An analog power input that supplies a continuous 12V DC approximately.

4 SOFTWARE DESIGN

The software for the autonomous mobile car was developed using the C programming language and uploaded to the PIC16F877A microcontroller, using the EasyPic V7. The development process followed a gradual approach, where each component was implemented and tested individually with a test code before being added into the full code. This approach helped identify errors early and made debugging more manageable as the system complexity increased. Figure 3 shows the system software through a flow diagram.

The final software follows a state-based structure, where each part of the track represents a different operating state. The robot switches between states based on sensor readings and timing conditions, allowing each task to be handled independently. Motor speeds are adjusted through PWM to maintain smooth movement, while timers are used for delays and time-based decisions. Analog values from sensors are read using the ADC, and simple conditional logic is used to control movement and system responses. Throughout development, the code was continuously refined by adjusting values, conditions, and control flow until stable and reliable behavior was achieved across all track sections.

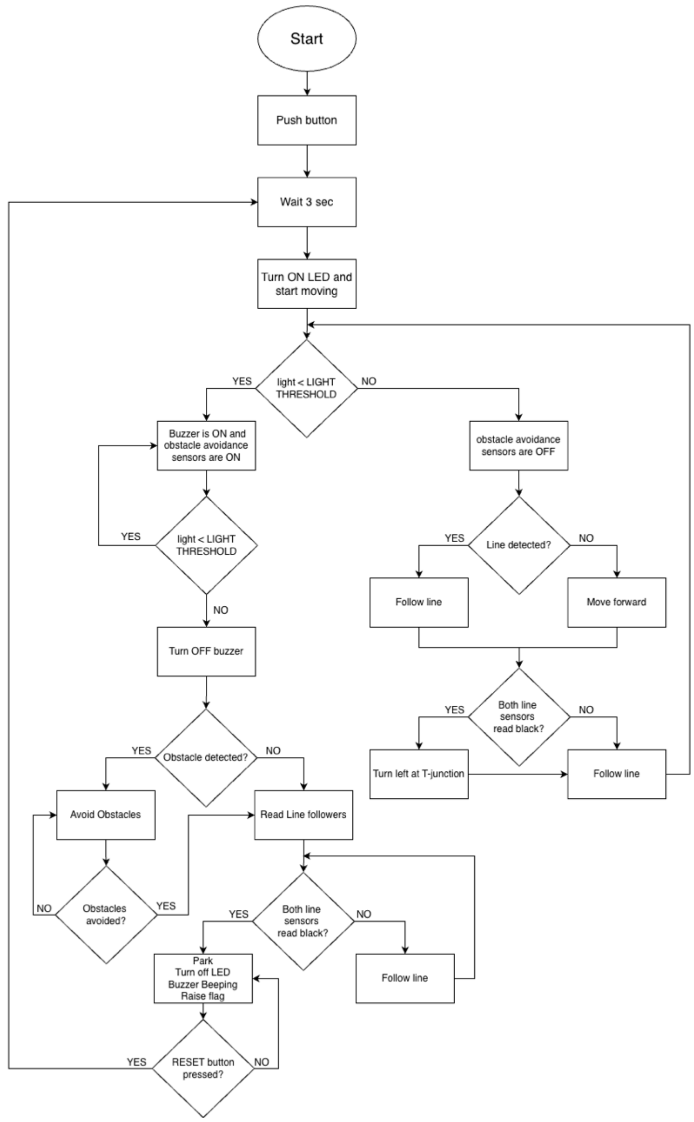


Figure 3: System Software Flow Diagram

5 PROBLEMS AND RECOMMENDATIONS

Throughout the development of the autonomous mobile car, several hardware and software challenges were encountered. These challenges required extensive testing, debugging, and repeated modifications to both the mechanical setup and the embedded software. Addressing these issues played a major role in achieving a reliable and stable final system.

One of the main challenges faced was hardware failure during testing. The H-bridge motor driver was burnt multiple times, and several components such as infrared sensors and DC motors stopped functioning and had to be replaced. In addition, the microcontroller and the EasyPIC board malfunctioned during development, so they needed to be replaced. These failures highlighted the importance of correct wiring, proper power management, and careful handling of components.

Over time, the team became more attentive to wiring polarity, pin placement, and voltage levels to prevent further damage.
Additionally, the microcontroller experienced random resets during operation due to voltage drops caused by the motors. This issue was resolved by disabling the brown-out detection feature, which allowed the system to operate continuously without unintended resets.

Motor imbalance was another recurring issue, as the DC motors did not operate at the same speed even when supplied with identical signals. This caused the robot to drift and deviate from the intended path. To solve this problem, dedicated motor testing was performed, and motor speeds were adjusted in software until straight and stable movement was achieved. These adjustments were repeated multiple times to compensate for changes in battery voltage and load conditions.

The integration of sensors also presented several challenges. Testing individual components before integrating them into the full system took lots of time and effort, but it was necessary to ensure reliable operation. When multiple sensors were connected simultaneously, previously working components would sometimes malfunction due to electrical noise, incorrect grounding, or sensitivity issues. To overcome this, each sensor was tested independently using separate test programs, and sensitivity thresholds were manually adjusted through trial and error. Sensor placement was also carefully reconsidered and modified to ensure accurate detection and reduce interference.

Moreover, some sensors, such as the IR obstacle avoidance sensors, experienced difficulties when detecting dark surfaces. Dark surfaces tend to absorb more of the emitted infrared light, resulting in weaker reflections and reduced detection range, which can affect the sensor's reliability in certain conditions.

From the software perspective, a large portion of the challenges were resolved through continuous code refinement. Adjustments to motor speeds, sensor thresholds, pin assignments, conditional logic, and loop structures were made repeatedly. The robot often worked correctly in one trial and failed in subsequent trials without apparent changes, which required patience and systematic debugging. Incremental testing, such as running the ultrasonic sensor alone for path navigation or isolating specific software modules, proved essential in identifying and fixing these issues.

Overall, the combination of extensive testing, gradual integration, careful hardware adjustments, and continuous software modification allowed the team to overcome these challenges and achieve a reliable autonomous system

6 RESULTS

The final design and implementation of the autonomous mobile robot are shown in Figure 4. The robot incorporates the complete mechanical, electrical, and embedded control systems and demonstrates the features described throughout this project.

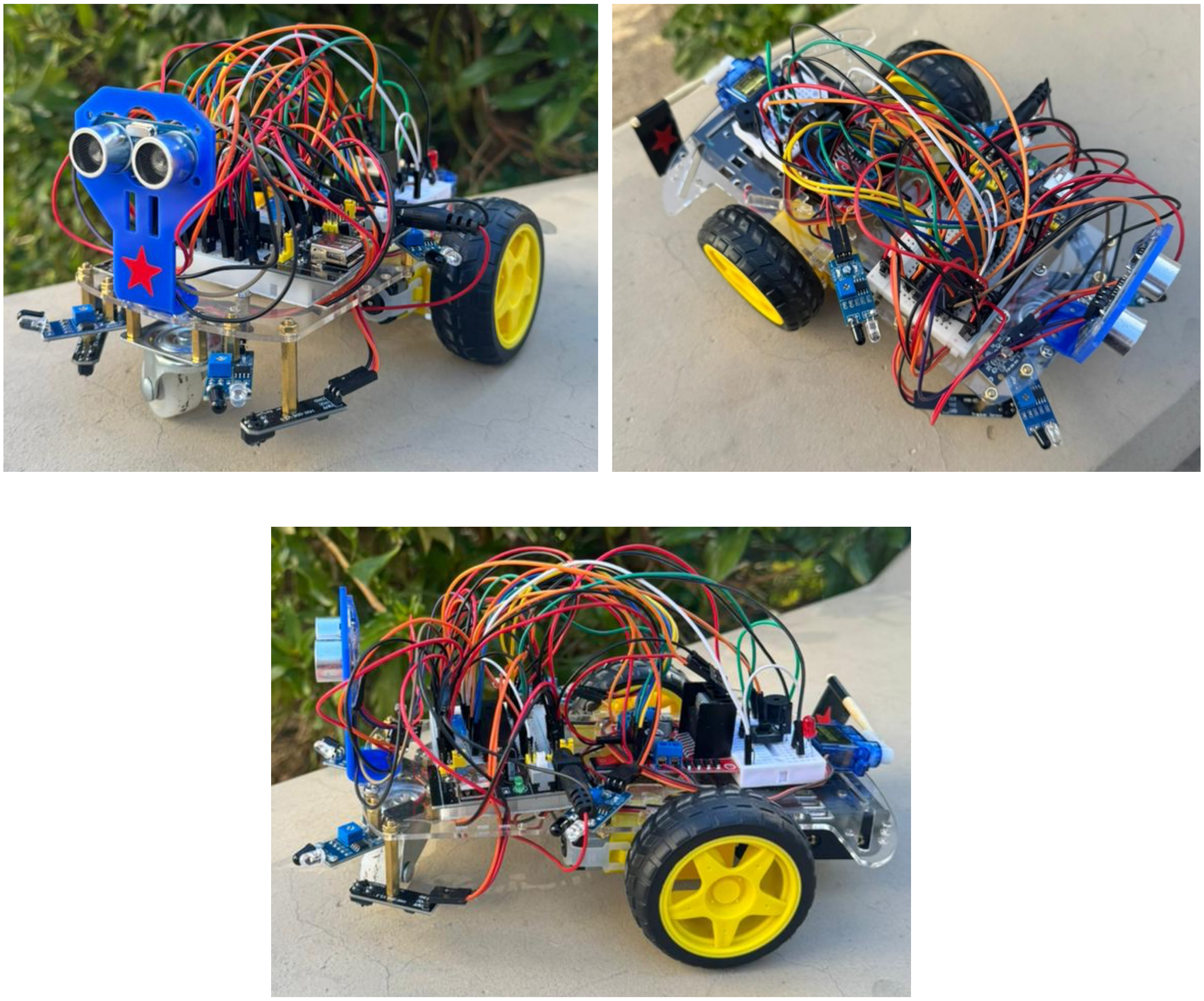


Figure 4: Final Implementation of the autonomous robot

7 CONCLUSION

This project focused on designing and building an autonomous mobile car using the PIC16F877A microcontroller that is capable of navigating a multi-zone track without human intervention. By integrating multiple sensors, actuators, and embedded software, the robot was able to follow lines, detect changes in light inside the tunnel, avoid obstacles, handle the bump, and park successfully at the end of the track. Throughout the project, continuous testing and adjustments were required on both the hardware and software sides, which helped strengthen understanding of embedded systems, power management, and sensor behavior. Despite the challenges faced during development, the final system operated reliably and demonstrated effective integration between mechanical, electrical, and software components. Overall, this project provided valuable hands-on experience in autonomous robotics and highlighted the importance of patience, careful testing, and iterative design when working with real embedded systems.

8 REFERENCES

[1] https://waslleh.com/product/pic-16f877a-original/

[2] https://waslleh.com/product/crystal-oscillator-8m/

[3] https://waslleh.com/product/2wd-robot-car-body/

[4] https://waslleh.com/product/2wd-robot-car-body/

[5] https://waslleh.com/product/l298n-dual-h-bridge-module/

[6] https://waslleh.com/product/hc-sr04-ultrasonic-sensor/

[7] https://waslleh.com/product/ir-infrared-obstacle-avoidance-module/

[8] https://waslleh.com/product/active-buzzer-5v/

[9] https://waslleh.com/product/5mm-led-light-2pin-different-colors/

[10] https://waslleh.com/?s=voltage+regultor&post_type=product&dgwt_wcas=1

[11]https://mikroelectron.com/product/tcrt5000-infrared-reflective-photoelectric-switch-line-tracking-sensor-module-for-line-follower

[12] https://mikroelectron.com/product/mini-servo-motor-sg90-9g-servo-180-degree

[13] https://mikroelectron.com/product/push-button-switch-666mm

[14] https://waslleh.com/product/10k-ohm-resistor-1w/

[15] https://waslleh.com/product/330-ohm-resistor-1-4w/

[16] https://waslleh.com/product/ky-018-photosensitive-sensor-module-light-module/

[17] https://waslleh.com/product/40cm-male-to-female-40pin-jumper-wires/

[18] https://waslleh.com/product/lithium-battery-18650-3-7v-3800mah-golden-power/

[19] https://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf