Name: Mhd Feras Aljoudi
SID: 200412916
Date: Mar. 25, 2024

Aljoudi Parking System Report

Table of Contents

List of Tables

List of Figures

1.0 Introduction

Aljoudi Parking System represents a significant step towards automating vehicle management and parking solutions. Developed as part of a university project for ENEL351 class, this system uses the microcontroller technology to simplify parking processes, enhance vehicle safety, and improve user experience in parking environments. The system is built on a practical, user-friendly platform that can accommodate up to four vehicles simultaneously. Incorporating different sensors and electronic components, Aljoudi Parking System aims to provide efficient, reliable, and secure parking services, focusing on areas with limited parking space. This report provides a comprehensive overview of the system's design, functionality, and operational testing, illustrating its potential benefits in modern parking management.

2.0 System Overview

Aljoudi Parking System is engineered to optimize parking space usage while providing users with real-time information regarding parking availability. Built around the STM32F103RB microcontroller, and it integrates several key components, including an LCD screen, potentiometer, servo motors, pressure sensor, IR sensor, and gas sensor, among others. This smart parking system allows for the parking of up to four vehicles at a time and includes an advanced fire alarm system for enhanced safety.

2.1 System Functions

Multiple key functions were designed to enhance user experience and improve parking lot management. These functions work together to ensure efficient operation, safety, and user convenience within the parking environment. The following list outlines the specific capabilities that the system provides:

- Display the number of available spots on the LCD.
- Weigh the vehicle at the entrance door.
- Provides the user the rate per hour displayed on the LCD screen based on the weight.
- Opening/closing the entrance door.
- Sensing the car at the exit door.
- Opening/closing the exit door.
- Sense any fire.
- Alarm if a fire occurred.
- Rejecting to open the entrance door if the parking is full.

2.2 Controls and Indicators

User interaction and feedback are facilitated through a set of controls and indicators. These elements are designed to provide the user with real-time system status and allow for easy adjustment and monitoring of system settings. Below is the detail of the primary controls available to the user and the indicators that provide essential system feedback.

The user has two main indicators which are:

- The LCD screen which will provide the users with different output.
- The digital buzzer which will alarm the user if a fire occurred.

The user has the following controls:

- Potentiometer to control the contrast on the LCD.
- Pressure of the car at the entrance door.
- Presentence of the car at the exit door.
- Gas or smoke which can be

2.3 Inputs and Outputs

The system operates through an interplay of inputs and outputs, each serving a unique function within the overall system. The inputs are responsible for gathering environmental and user data, while the outputs respond accordingly to ensure proper system performance and user interaction. Below is the list of the inputs and outputs we have.

Inputs	**Outputs**
Analog Pressure Sensor	LCD screen display
Analog Gas Sensor	2 Servo motors
Digital IR Sensor	Digital Buzzer
Potentiometer	-

Table 1: Inputs and Outputs

3.0 Electrical Connection

This section dives into the details of the electrical connections of Aljoudi Parking System, providing a comprehensive overview of how each component is interconnected to ensure seamless operation and functionality. By examining the block diagram, schematic diagram, detailed connection table and Components table, you will have a clear understanding of the system's electrical architecture and component interrelationships. This section is crucial for replicating the setup or diagnosing system issues, offering a foundation for both theoretical understanding and practical application.

3.1 Block Diagram

The Block Diagram provides a high-level overview of the system's main components and their primary connections. This visual representation serves as a foundational guide, showing the system's functional relationships and signal flow between components, helping in the overall understanding of the system's electrical design.

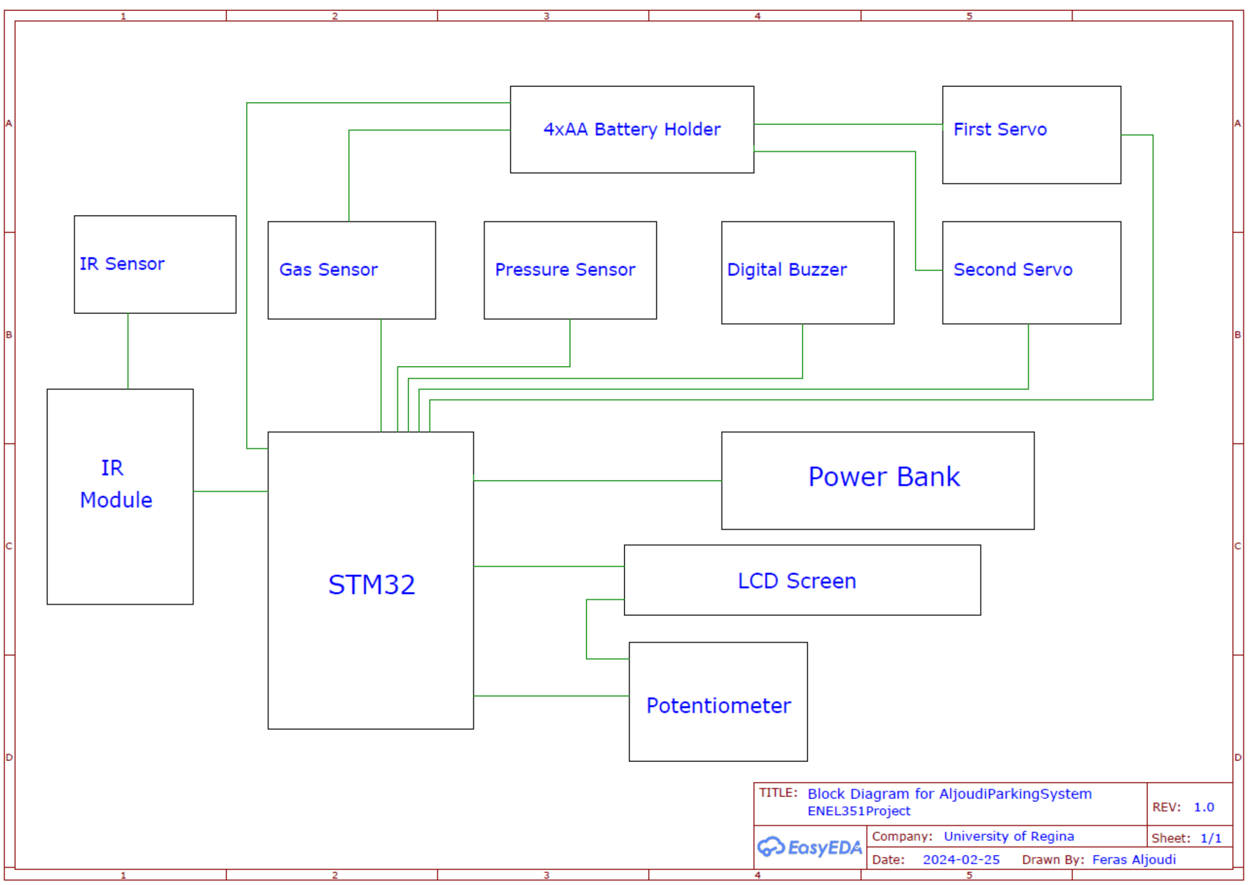


Figure 1: Block Diagram

3.2 Schematic Diagram

Following the block diagram, the [Schematic Diagram]() provides a detailed view of the electrical connections within the system. It includes specific connections, component values, and circuitry details essential for constructing the system. This diagram is essential for understanding the complex wiring and component integration as it acts as a map for the electrical design.

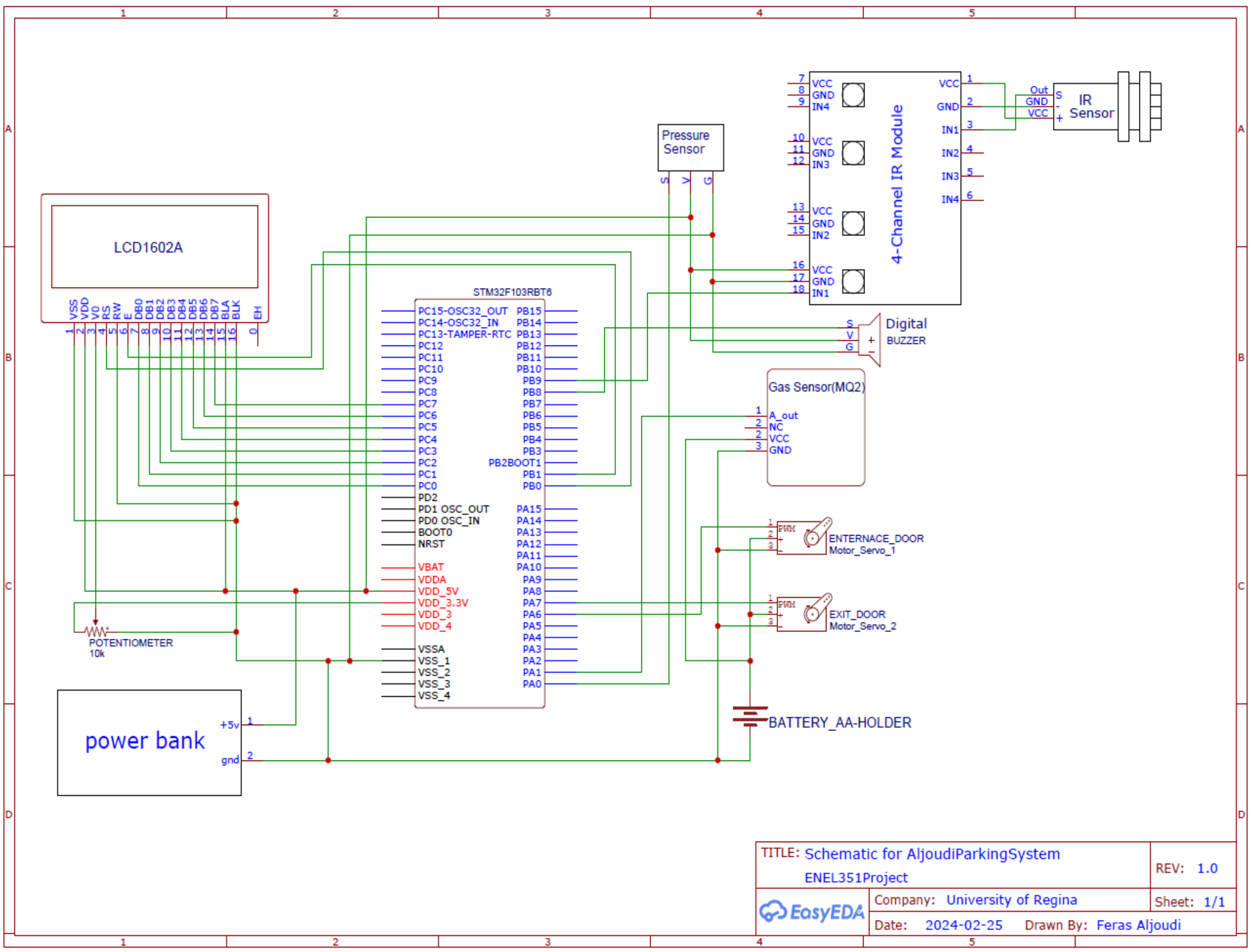


Figure 2: Schematic Diagram

3.3 Connection Details for I/O Components

This table provides detailed information on the input/output connections specific to the system. It lists each component, its corresponding pin, and how it is connected to the microcontroller and other elements. These details are essential for ensuring accurate connections for proper functionality.

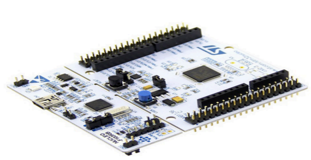
Pin in the Module	Connected to
STM32 Microcontroller	
STM32-PA0	Gas Sensor-S
STM32-PA1	Pressure Sensor-S
STM32-PA6	Servo Motor_1-S
STM32-PA7	Servo Motor_2-S
STM32-PB0	LCD-RS
STM32-PB1	LCD-E
STM32-PB8	Digital Buzzer-S
STM32-PB9	IR Sensor-Out
STM32-PC0 to PC7	LCD-D0 to D7
LCD Display	
1_Vss	GND
2_Vdd	5V
3_V0	10K Potentiometer
4_RS	STM32-PB0
5_R/W	GND
6_E	STM32-PB1
7_DB0 to 14_DB7	STM32-PC0 to PC7
15_A	5V
16_K	GND
Analog Gas Sensor	
A_out	STM32-PA1
V	5V
G	GND
Digital Buzzer Module	
S	STM32-PB8

V	5V
G	GND
Analog Pressure Sensor	
S	STM32-PA0
V	5V
G	GND
Servo Motors	
S from the 1st Servo	STM32-PA6
S from the 2nd Servo	STM32-PA7
V	6V
G	GND
Digital IR Sensor	
Out	STM32-PB9
VCC	5V
GND	GND

Table 2: Connection Details

3.4 Components Used

The final table in this section lists all components used in the system, along with their specifications and quantities. This table is essential for accurate component sourcing and ensuring compatibility within the system.

Image	Description	Quantity
	**STM32F103RB** Microcontroller to control the system.	1
	**LCD Display** Output to provide the user with data such as rate per hour, number of available spots and fire alarm.	1

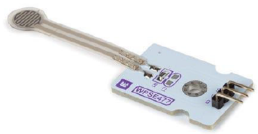
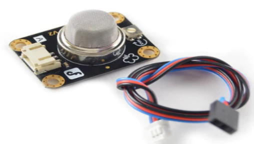
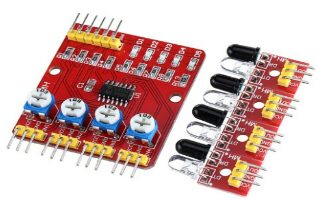
	**Potentiometer** 10K Potentiometer to control the contrast of the LCD	1
	**Analog Pressure Sensor** Used to weigh the car before entering the lot. The rate is determined based on the weight.	1
	**Analog Gas Sensor** Used to sense if a fire occurred in the parking lot.	1
	**Digital Buzzer** Used to buzz if the Gas Sensor sensed a fire.	1
	**Digital IR Sensors & Module** One IR sensor used to sense the car at the exit door.	1
	**Servo Motors** Two servos to open and close the entrance and the exit doors	2
	**Power Bank** Power bank to power the STM32.	1
	**Battery Holder** 4-AA battery holder as a power source support.	1

Table 3: Components Used

Here is the links for the datasheets I used for the above components:

- **STM32F103RB:**
  - STM Manual
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/STM32F103ReferenceManual.pdf
  - STM Schematic
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/STM32F103Schematic.pdf
- **LCD Display:**
  - LCD 16x2 Datasheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/16x2LCD_Datasheet.pdf
- **Potentiometer:**
  - Potentiometer Datasheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/potentiometerDatasheet.pdf
- **Analog Pressure Sensor:**
  - Pressure Sensor Manual
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/PressureSensorManual.pdf
- **Analog Gas Sensor:**
  - Gas Sensor Datasheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/AnalogGasSensorDatasheet.pdf
- **Digital Buzzer:**
  - Digital Buzzer Datasheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/DigitalBuzzerDatasheet.pdf
- **Digital IR Sensors & Module:**
  - IR Sensor Datasheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/IR_sensorDatasheet.pdf
- **Servo Motors:**
  - Servo Specification Sheet
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/HRC31050ServoSpecSheet.pdf
  - Servo Manual
    https://github.com/ferasaljoudi/AljoudiParkingSystem/blob/main/Documentation/Datasheets/HRC31050S-Servo_Manual.pdf

4.0 Program Listings

This section provides an overview of each code file utilized within the system. The descriptions aim to outline the purpose and functionality embedded in each file, offering insights into how the various components of the system are controlled and managed through the software. These listings will help understand the structural and operational aspects of the codebase, contributing to a comprehensive view of the system's software architecture.

- **Servo.h & Servo.c:** These files are crucial for controlling the servo motors in the system. Servo.h includes the necessary function declarations related to servo operation. Servo.c includes configurations for the GPIO pins used by the servo motors and sets up PWM (Pulse Width Modulation) for controlling the servo angles. Servo_init() function prepares the GPIO settings for PWM output, and pwm_init() configures the timers for generating the PWM signals. The convertAngle() function converts a specified angle to a corresponding PWM signal width. updateServoAngle() is used to apply these PWM settings to move the servo to a designated angle.
- **GasSensor.h & GasSensor.c:** These files are crucial for integrating and utilizing the gas sensor within the system. GasSensor.h includes declarations for the functions related to the operation of the gas sensor. GasSensor.c includes configurations for initializing the ADC (Analog-to-Digital Converter) that interacts with the gas sensor. The gasADC_init() function configures the relevant GPIO pins and the ADC settings to properly read analog signals from the gas sensor. The gasADC_acquire() function initiates an ADC conversion, waits for it to complete, and then returns the converted analog value, which corresponds to the gas concentration level detected.
- **PressureSensor.h & PressureSensor.c:** These files are crucial for the integration and operation of the pressure sensor within the system. PressureSensor.h includes declarations for the necessary functions related to the pressure sensor's activities. PressureSensor.c includes configurations for initializing the ADC that interacts with the pressure sensor. The pressureADC_init() function sets up the required GPIO pins and ADC settings to properly read analog signals from the pressure sensor. The pressureADC_acquire() function then initiates an ADC conversion process, waits for it to complete, and then returns the converted analog value, reflecting the pressure level detected.
- **Timer.h & Timer.c:** These files are central to the timing functions within the system, since they facilitate precise control over time-dependent operations. timer.h outlines the

declarations for initializing system timers and for creating microsecond and millisecond delays. Timer.c implements these functionalities where time_init() sets up the hardware timer, while delay_us() and delay_ms() provide delay functions based on microseconds and milliseconds, respectively. It is important to note that the timer.h and timer.c files were provided by **Dave Duguid**.

- **LCD.h & LCD.c:** These files are crucial for interacting with the LCD display in the system. LCD.h defines a set of constants representing specific LCD commands and settings such as turning on the display, moving the cursor, and clearing the screen. It also declares essential functions for initializing the LCD. LCD.c implements the functions declared in LCD.h.
  - The LCD_init() function activates the necessary GPIO ports and sets them for output, preparing the system for LCD interaction. It then sends a series of initialization commands to the LCD to configure it for use.
  - LCD_sendCommand() and LCD_sendData() manage the sending of control commands and data to the LCD, enabling the display of text and characters.
  - LCD_sendString() and LCD_sendInteger() are used to allow for displaying strings and integer values.
  - The clearLCD() function clears the display.
- **Operation.h & Operation.c:** These files contain the core functionalities and user interface logic of the system. operation.h provides the function prototypes that control the system's operations, from displaying messages on the LCD to managing the entrance and exit processes.
  - welcomeMessage(): Displays a welcoming message on the LCD.
  - displaySpots(int spots, int carsCount): Calculates and shows the number of available parking spots by subtracting the number of parked cars from the total spots.
  - fireAlarm(int readGas): Activates the fire alarm routine by displaying a warning on the LCD and activating a buzzer in a blinking pattern if dangerous gas levels are detected.
  - entranceDoor(int spots, int carsCount): Manages the logic for the entrance door, including vehicle weight assessment and space availability. It updates the count of parked cars and controls the servo motor to open or close the entrance door accordingly.

    - exitDoor(int carsCount): Controls the exit door operations, allowing cars to leave and updating the total count of parked vehicles.

  Operation.c implements these functions, interfacing directly with the hardware (LCD, sensors, and servo motors) to perform the specific actions required for each operation within the parking system.

- **Main.c:** This file serves as the heart of the system, where all individual components and functionalities converge to create a cohesive operation. It begins with system initialization, setting up the necessary hardware components and configurations:
    - System and Timer Initialization.
    - LCD, Servo, Pressure Sensor, and Gas Sensor Initialization.

  In the main operational loop, the program continuously monitors sensor inputs to manage parking availability and safety measures:
    - Pressure Sensor Monitoring: The system checks the pressure sensor to determine if a vehicle is present at the entrance and if so, processes the vehicle through the entranceDoor function, then updates the parking availability using displaySpots function.
    - Gas Sensor Monitoring: It regularly reads the gas sensor values to detect hazardous conditions. If dangerous gas levels are detected, the fireAlarm function is triggered.
    - Infrared Sensor Checking: Monitors the exit for vehicles attempting to leave. When a vehicle is detected, the exitDoor function is called, and the parking availability is updated.

5.0 Physical System and Operation

This section provides an in-depth look at the physical layout and operational dynamics of the system. The design and functionality are highlighted through the examination of key areas within the system: the whole system setup, the entrance door mechanism, and the parking lot interior.

5.1 Whole system

The comprehensive setup of the system is illustrated in figure 3 below. This overview provides a snapshot of the system's entirety, showcasing its integrated components and general layout.

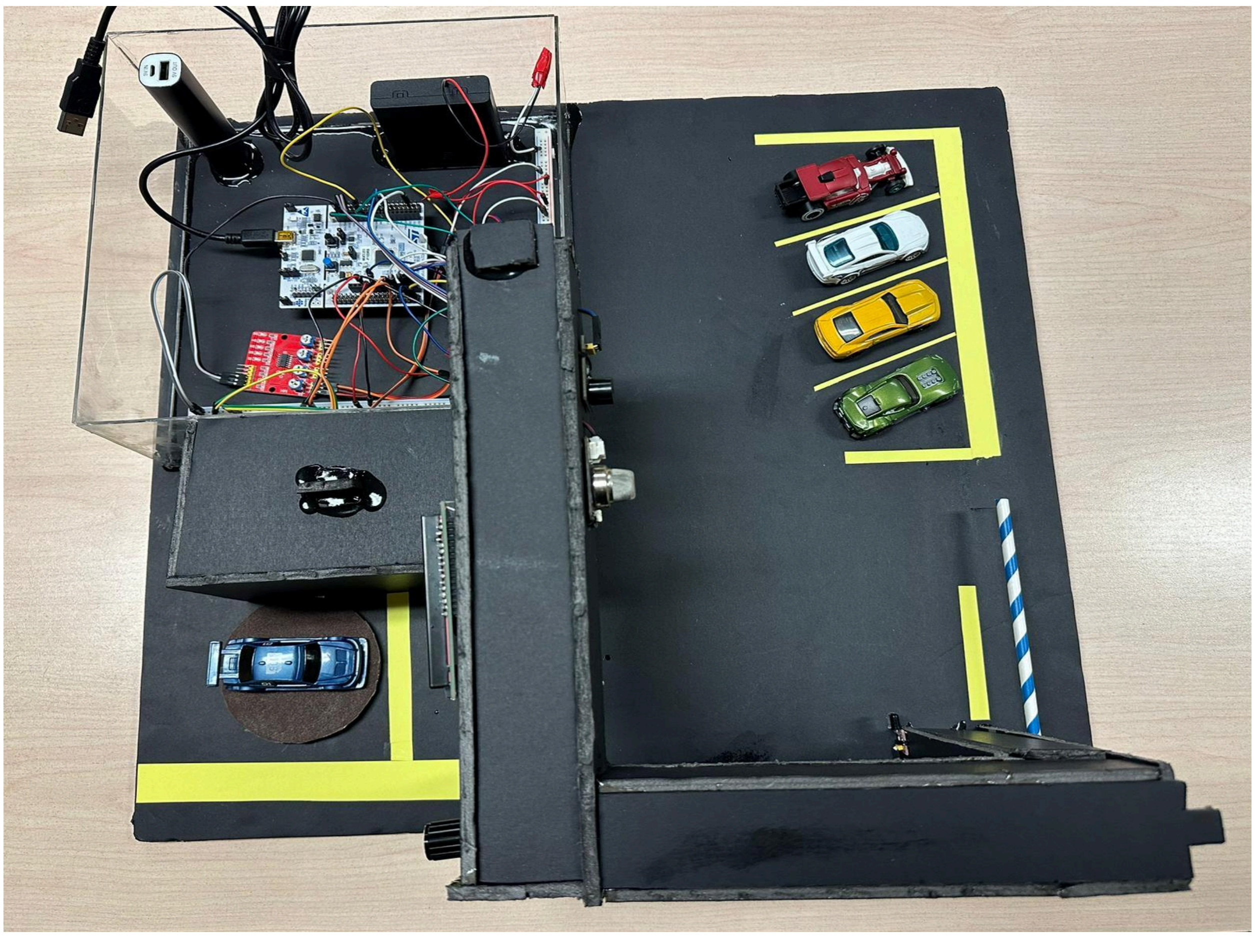


Figure 3: Whole System

5.2 Entrance Door

At the heart of the entrance mechanism is a pressure sensor embedded within a plate where vehicles halt upon arrival. Adjacent to this, a potentiometer is strategically positioned to the right of the LCD screen, enabling the adjustment of the contrast on the

LCD for optimal visibility. This arrangement is captured in figure 4 below, detailing the precise setup that vehicles encounter as they approach the system.

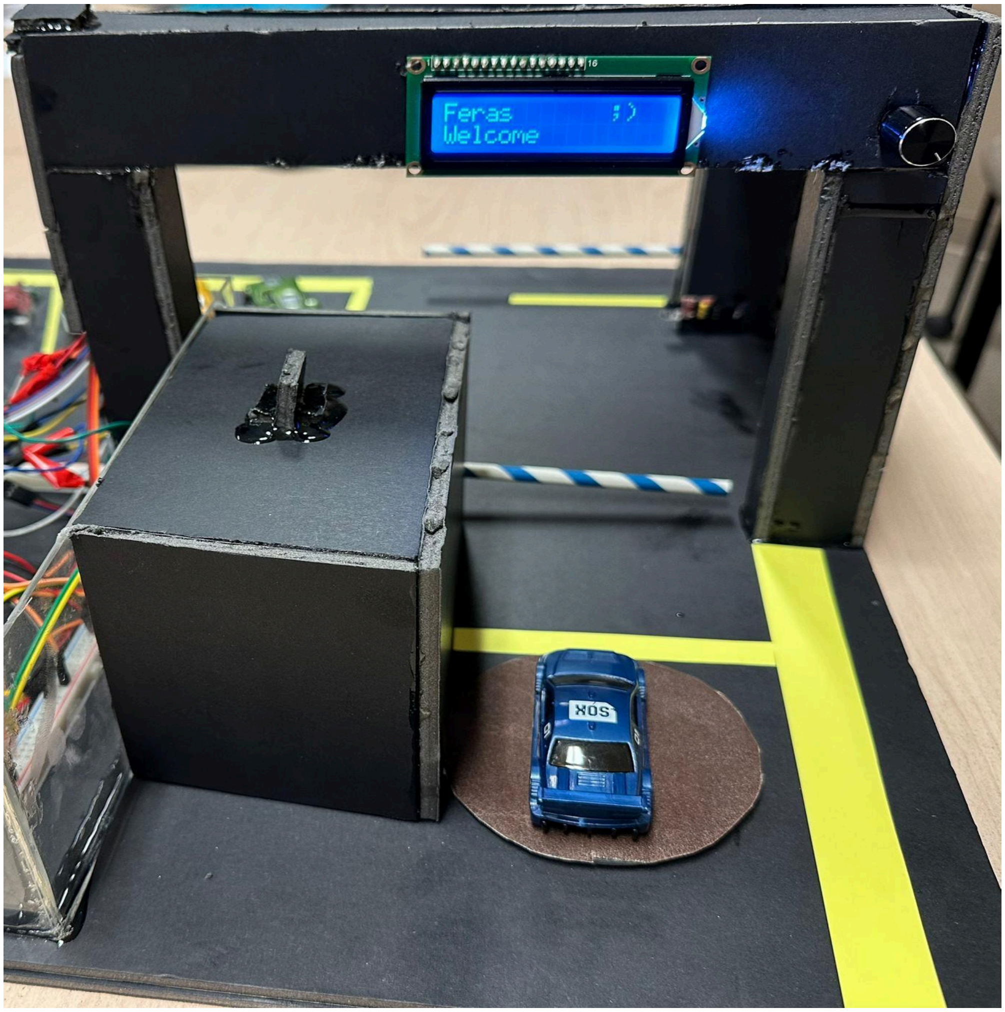


Figure 4: Entrance Door

Upon the detection of an empty spot by the system, the pressure sensor activates to weigh the incoming vehicle. The vehicle's weight then determines the parking rate, which is promptly displayed on the LCD. Subsequently, the servo motor initiates the opening of the entrance door, allowing the vehicle to proceed. Concurrently, the LCD screen updates to reflect the new vehicle count within the parking lot. Conversely, if no spots are available, the LCD screen will display a clear message: “Sorry… No available spots”.

5.3 Parking Lot Interior

Within the interior of the parking lot, safety and monitoring features are prominently installed. Atop a central pole, a gas sensor coupled with a digital buzzer forms a crucial safety mechanism. This setup, shown in figure 5 below, is designed to detect smoke or fire, with the buzzer emitting an alert in the event of smoke detection, thereby signaling a potential fire hazard.

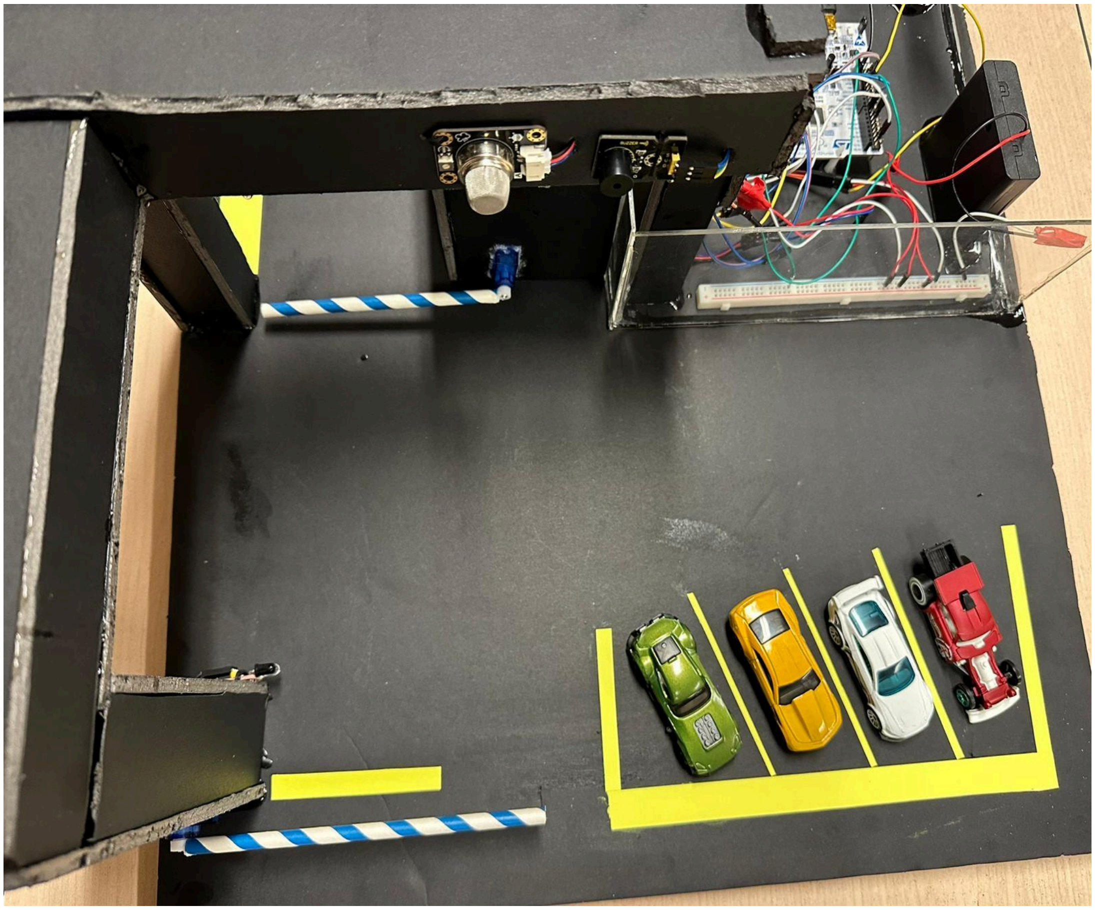


Figure 5: Parking Lot Interior

Additionally, the exit mechanism uses an IR sensor strategically placed to identify vehicles preparing to leave. Upon detection, the servo motor activates to open the exit door, facilitating the vehicle's departure. Concurrently, the LCD display is updated to accurately represent the current count of parked vehicles, ensuring real-time tracking of the parking lot's occupancy.

Through these descriptions and accompanying figures, a clear understanding of the system’s physical layout and operational protocols is presented, demonstrating the system's efficiency and user-centric design.

6.0 Conclusion

The development and implementation of Aljoudi Parking System represent a significant step forward in smart parking solutions. Throughout the project, I successfully integrated various technologies, including microcontrollers, sensors, and actuators, to create a system that improves parking efficiency and safety. The system's ability to display available parking spots, weigh vehicles, and adjust parking rates accordingly, combined with its fire detection and alert capabilities, showcases the potential of integrating technology into everyday buildings.

The project was a comprehensive test of my technical abilities, pushing the limits of my understanding of electronics, programming, and system design. It provided a learning experience, deepening my expertise in practical engineering applications and highlighting the importance of meticulous planning, problem-solving, and iterative testing.

In conclusion, Aljoudi Parking System project has been a pivotal step in my journey as an engineering student, offering a solid groundwork for further innovation in parking management systems. The insights gained from this project are invaluable, and I am eager to apply them to future challenges, aiming to contribute meaningfully to the evolution of urban infrastructure solutions.

7.0 Additional Resources

For further details and to explore the codebase of Aljoudi Parking System, please visit my GitHub repository https://github.com/ferasaljoudi/AljoudiParkingSystem.

To see the Parking System in action and understand its operational flow, watch my demonstration on YouTube https://www.youtube.com/watch?v=ReGvdZkHqbI.
