University of Windsor

Computational Methods and Modeling for Engineering Applications

(GENG 8030 - W2023)

Final Project Report - Smart Parking System

Date – 03/30/2023

Group - 10

Name	Student Number
Altamash Yar Khan	110104393
Tehreem Hussain Qureshi	110105443
Sabian Coutinho	110098311

Table of Contents

Contents

1. Introduction

1.1 Project Objective

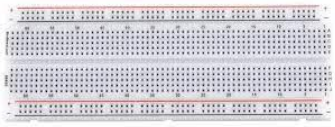
The objective of this project is to address the growing demand for secure parking spaces due to the increasing urbanization and car ownership [1]. Traditional parking systems are inefficient and result in wasted time, money, and productivity for drivers. Additionally, they contribute to traffic congestion, which accounts for 35% of overall commute time [2]. To manage and reduce traffic caused by drivers searching for parking on city streets, municipalities can use smart parking solutions. These technologies improve parking safety and convenience, while also decreasing the number of cars circling the streets searching for a spot. The aim of our project is to develop a smart parking system that offers a faster, more convenient, and hassle-free parking experience for drivers, while also reducing traffic congestion and promoting smoother traffic flow [2][3].

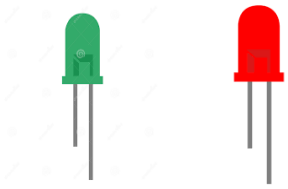
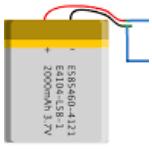
1.2 Project Overview

In this project we have developed a simulation of a parking lot with several features, including divided parking spaces, an automated lifting bar, indication traffic signals, and a display showing the number of available parking spots. The system will show the total number of available parking spaces and welcome messages on an LCD display when there are open spots. The Arduino is programmed to respond to the driver pressing the ‘Enter’ button, causing the entrance barrier to lift, the RED traffic signal to switch to GREEN, and allow the car to enter the lot, reducing the number of available parking spaces by one. The barrier arm will close once the car is parked, and the light will turn back to RED. When the driver presses the ‘Exit’ button, the barrier arm at exit will open, and the traffic signal will change to GREEN, allowing the car to exit. The LCD display will show the number of available parking spots increasing by one and change the traffic signal to RED as the car exits the parking space. When there are no available parking spaces or the lot reaches its maximum capacity of 13 slots, in this situation if the driver presses the ‘Enter’ button, the barrier arm will not open, and the LCD will display the message "Plz come later."

2. Methodology

2.1 Hardware Components:

Components	Amount	Images	Functionality
Arduino Uno	1		• The Arduino Uno is a microcontroller board that is used to control the various components in your project. • It is programmed using the MATLAB code and receives input from the PCs port which controls the components connected to it.
Breadboard	2		• The breadboard is used to connect the various components together. It provides a convenient way to create electrical connections without having to solder anything.
Servo motor	1		• The servo motor acts as the barrier arm that allows vehicles to enter and exit the parking lot. • The motor rotates to lift or lower the arm when it receives a signal from the Arduino.
Push Buttons	2		• Enter Button: used to signal that a car wants to enter the parking lot. • Exit button: is used to signal that a car wants to leave the parking lot.

LCD display	1		• Used to display information about the number of available parking spots, and other messages when parking spots are full.
common cathode LED module (Red & Green)	2		• The Red LED indicates that the barrier is closed. • The Green LED indicates that the barrier is open.
Connecting Wires	N/A		• The jumper wires are used to connect the different components on the breadboard to the microcontroller.
Potentiometer	1		• The variable resistor is used to control the contrast of the LCD display.
Battery	1		• The battery can be connected to the system to provide power to the components when the main power source from arduino is not available.

2.2 Software Requirements

1. **MATLAB**: a high-level programming language software used for numerical computing, data analysis, and visualization. We use MATLAB to write the code for our car parking system project. This code includes all the essential conditions and functionalities required to successfully run our project. We download packages such as MATLAB Support Package for Arduino Hardware and Simulink Support Package for Arduino Hardware to program the Arduino Uno, which is the microcontroller used in our project.
2. **PROTEUS 8:** is a software tool used for circuit simulation and PCB design. We use Proteus 8 to design the software prototype of our project. We implement all the necessary components and connections in Proteus 8 to simulate the car parking system. This allows us to validate our prototype design before implementing it physically. We can test the functionality of our project and make changes if needed.
3. **ARDUINO IDE:** Proteus only allows ".hex files" to run simulations on the designed prototype. However, we write our code in a .mlx file. Therefore, to allow Proteus to read our MATLAB code, we first generate a standalone executable file using the "deploytool" function in MATLAB. This executable file can be read by the Arduino IDE. Then, we upload the hex file generated from the Arduino IDE to Proteus for simulation.

2.3 Block Diagram/ Software Implementation

Case 1: Below is the software simulation of our prototype designed in **Proteus 8**.

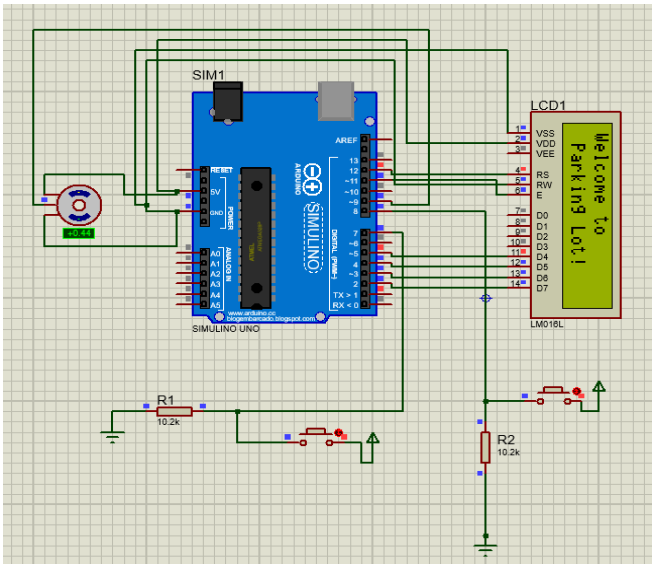


Figure 2.1: Initializing the prototype.

- Initially LCD Displays “Welcome to Parking Lot”.
- Servo Motor is closed. (At an angle of 0 degrees)
- Enter button is the one connected to Resistor R1.
- Exit button is the one connected to Resistor R2.
- Due to limited space and pin availability in Proteus simulation, we were unable to connect the LEDs. However, they will be connected in the hardware implementation of the design where we will have access to more pins and space.

Case-2: **When a driver enters a parking lot, number of available space decreases by 1.**

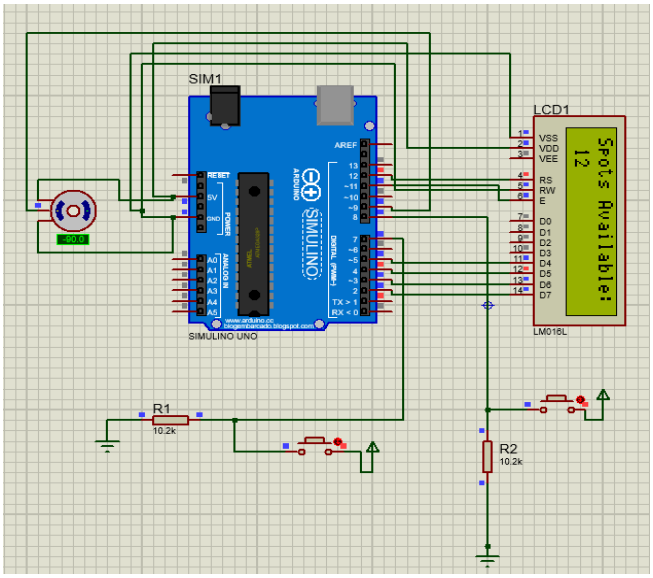


Figure 2.2: Driver Entering the Parking lot.

- When the driver enters the parking spot, the arms of the servo motor rotates to 90 degrees, indicating that the barrier is open for the driver to enter.
- Spots available decreases by 1.

Case-3: **When Driver exits the parking lot, space increases by 1**.

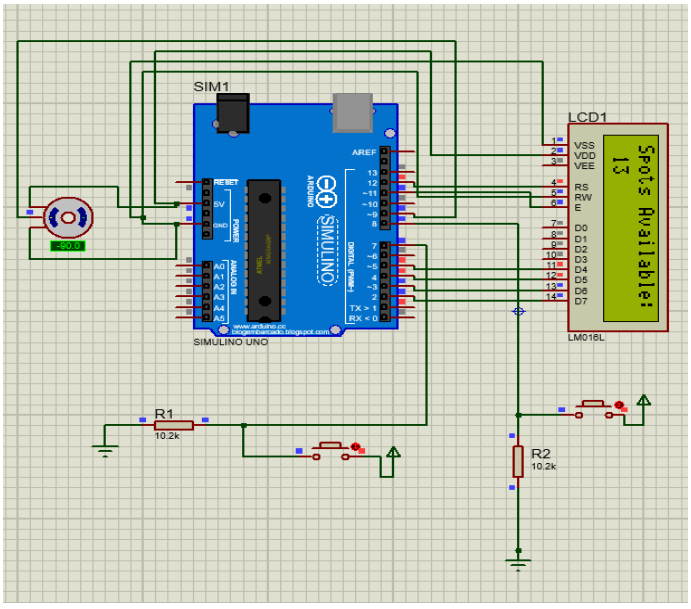


Figure 2.3: Driver Exits the Parking Lot.

- Barrier opens up for the driver to exit (Servo Motor rotates 90 degrees.)
- Space increases by 1.

2.4 Hardware Implementation

This is the hardware design of our project. The pictures below depicts the smart parking system that displays the number of parking spots available in the parking lot through LCD and changes the LED's from red to green when the vehicle presses the Enter/Exit buttons and simultaneously lifts the barrier.

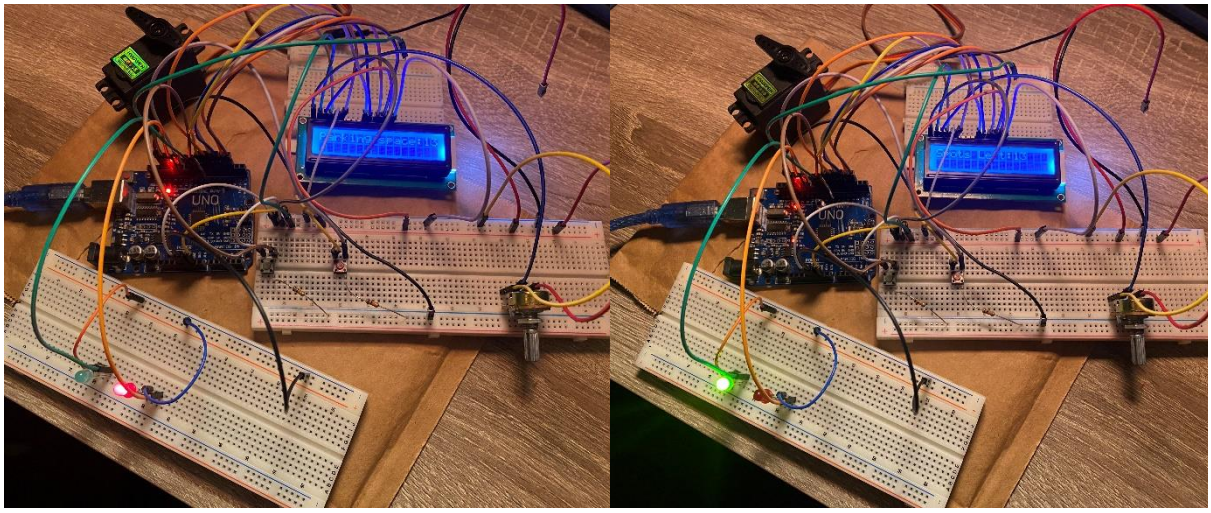


Figure 2.4: Hardware Prototype

2.5 Flow-Chart (Pseudo Code)

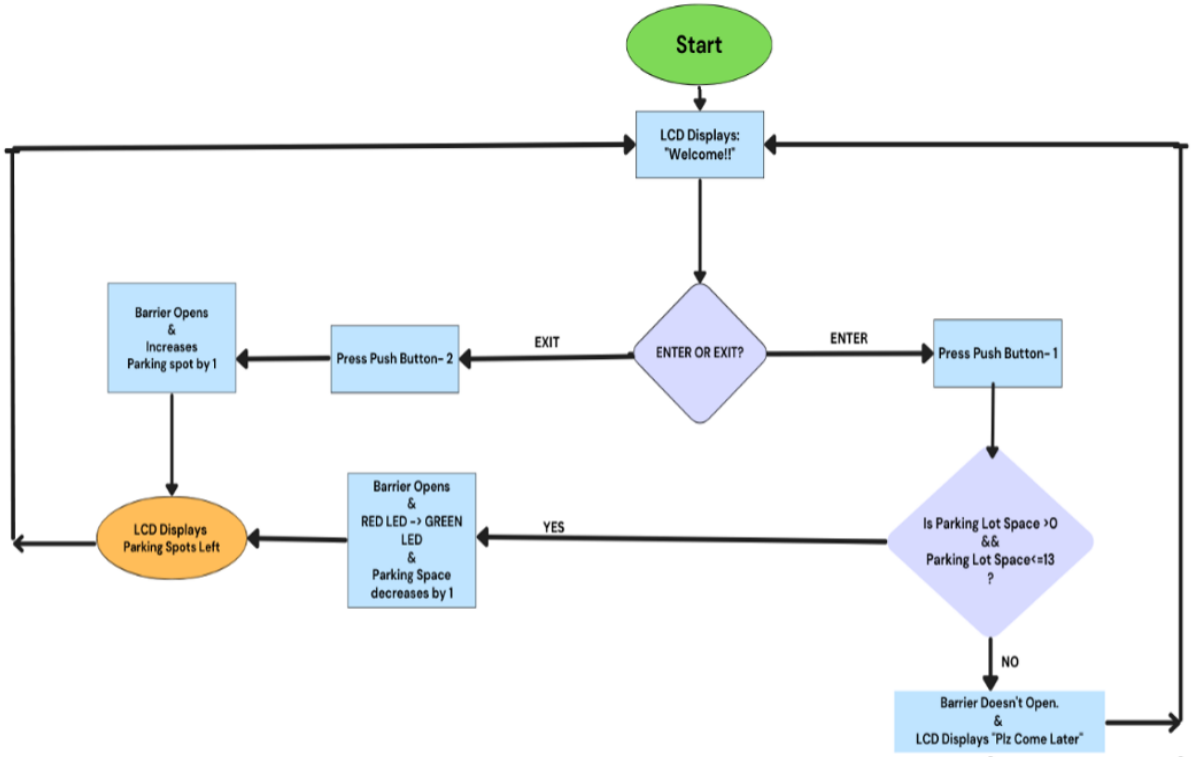


Figure 2.5: Flow Chart

3. Timeline and Limitations

3.1 Timeline:

Task	Start Date	End Date	Duration	
1. Buying Components	2023-02-01	2023-02-15	15 days	✓
2. Software Simulation	2023-02-15	2023-02-25	10 days	✓
3. Writing Report	2023-02-20	2023-02-27	7 days	✓
4. MATLAB Programming	2023-03-01	2023-03-15	15 days	✓
5. Hardware Implementation	2023-03-16	2023-05-26	10 days	✓
6. Testing and Troubleshooting	2023-03-27	2023-03-31	4 days	✓
7. Finalizing the Project	2023-04-01	2023-04-02		

Table 3.0: TimeLine

3.2 Limitations:

1. **Technical limitations:** There may be technical limitations associated with the use of the Arduino board, LCD indicator, and other components. The components might turn out to be faulty which would affect the functionality of the project.
2. **Design limitations:** The parking control system may have design limitations that need to be taken into consideration, such as the availability of space to install the system.
3. **Time constraints:** Developing and implementing a parking control system requires time, effort, and resources. Group members may have limited time and resources to devote to the project, which may affect the quality and functionality of the system.
4. **Communication issues:** Communication is key in group projects. Any miscommunication between group members can lead to delays, misunderstandings, and ultimately, project failure.

3.3 Roles and Responsibilities

Roles and Responsibilities	**Altamash**	**Tehreem**	**Sabian**
Organizing meetings			✓
Taking the minutes of the meetings		✓	✓
Reviewing Resources	✓	✓	✓
Ordering the hardware kit		✓	
Working on the flowchart	✓	✓	
Working on the preliminary report	✓		✓
Software implementation	✓		
Hardware Implementation	✓	✓	✓

Table 3.1: Roles and Responsibility

3.4 Tasks Assigned:

Member 1 (Altamash): Responsible for working on project description, methodology, creating flowchart, developing initial code, finalizing the code, and installing hardware setup and working on final report.

Member 2 (Tehreem): Responsible for taking notes of the meeting, reviewing available resources, ordering hardware kit required, reviewing flowchart, revising the initial code, installing final hardware setup.

Member 3 (Sabian): Responsibilities include organizing group meetings, noting down steps to follow, reviewing resources, working on project objective and final report, revising the code, and installing final hardware setup.

4. Conclusion

To sum up, advanced technology used in smart parking systems helps monitor and manage parking spaces, leading to benefits such as reducing the time drivers spend circling for parking spots, improving overall parking efficiency, and providing real-time information about available spots. This results in a better user experience, reduced traffic congestion, pollution, and increased revenue. As technology continues to progress, the potential for even more advanced smart parking systems exists, which could make parking easier and more efficient for drivers around the world.

5. MATLAB Script:

```
clear all


ard = arduino('com6','uno','libraries',{'ExampleLCD/LCDAddon' 'Servo'});
lcd=addon(ard,'ExampleLCD/LCDAddon','RegisterSelectPin','D7','EnablePin','D6',
'DataPins',{'D5','D4','D3','D2'});
initializeLCD(lcd);
s=servo(ard,'D9')
enterbtn = 'D8';
exitbtn ='D11'
redLedPin = 'D12';
greenLedPin= 'D13';
configurePin(ard, enterbtn, 'DigitalInput');
configurePin(ard, exitbtn, 'DigitalInput');
count=13;


printLCD(lcd,'Welcome!');
pause(1)
clearLCD(lcd);
printLCD(lcd,['Parking space:',num2str(count)]);
pause(1)




while true
EnterbuttonState = readDigitalPin(ard, enterbtn);
ExitbuttonState = readDigitalPin(ard, exitbtn);


if(EnterbuttonState==0 && ExitbuttonState==0)
writePosition(s,0)
while EnterbuttonState == 0 && ExitbuttonState==0
EnterbuttonState = readDigitalPin(ard,enterbtn);
ExitbuttonState= readDigitalPin(ard,exitbtn);
writeDigitalPin(ard, redLedPin, 1);
writeDigitalPin(ard, greenLedPin, 0);
```

```
end


else
while EnterbuttonState == 1 || ExitbuttonState==1
EnterbuttonState = readDigitalPin(ard,enterbtn);
ExitbuttonState= readDigitalPin(ard,exitbtn);


if(count>0 && count <=13)
writeDigitalPin(ard, redLedPin, 0);
writeDigitalPin(ard, greenLedPin, 1);
writePosition(s,0.5);
clearLCD(lcd);
printLCD(lcd,['Spots Left:',num2str(count)]);
pause(1)


elseif(count==0)
writeDigitalPin(ard, redLedPin, 1);
writeDigitalPin(ard, greenLedPin, 0);
writePosition(s,0)
clearLCD(lcd)
printLCD(lcd,'Spots are full');
pause(1)
printLCD(lcd,'Plz Come Later:');
pause(1)


end
if(ExitbuttonState==1 && count<13)
count=count+1;
elseif(EnterbuttonState==1&&count>=0)
count=count-1;
end
end
end
end
```

6. References

[1] P. MacCallum, “Parking management challenges,” *OperationsCommander*, 22-Jun2021. [Accessed: 23-Jun-2022].

[2] “Smart Parking Solutions- Top 5 Benefits for the Drivers &Ndash; Conure.” *Conure*, 23 Feb. 2022, www.conurets.com/smart-parking-solutions-top-5-benefits-for-the-drivers

[3] S. Adablanu, “Review on automatic smart car parking system,” *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 8, pp. 79–82, 2022.