FINAL REPORT

SMART PARKING MANAGEMENT SYSTEM

SUBMITTED BY:

Arulpiruthiviraj Arunthavaraja	105212573
Baranidharan Pasupathi	110007279
Rishi Anand	105195887

SUBMITTED TO:

PROFESSOR ROOZBEH RAZAVI-FAR

Executive Summary

This project included developing and constructing a Smart Parking Management System-an intelligent vehicle parking management system designed to solve the ongoing parking crisis. With the world moving towards self-sufficiency, a self-parking system is in high demand all over the world. This main objective of this project was to develop and construct a smart parking management system, an intelligent vehicle parking management system designed to solve the ongoing parking crisis. The goal of this project is to recognize the solutions that could increase the efficiency of parking that enhances the user-experience. To achieve this, certain milestones were introduced into this project.

The first stage was a requirement analysis in which the prerequisites were analysed to produce an initial design. Then the individual functions of each component were designed separately on the Simulink environment. Further, each of these components were tested separately before being built together. Finally, all the hardware and software components were connected together to form the smart parking system.

This report outlines the approach to developing a smart parking system that locates the available parking spots in an area and is capable of positioning vehicles at those available spots in real time. This involves using low-cost sensors, collecting real time data of reservation of parking spots in advance and predicting the exact location of a spot. Therefore, introducing this system into the real world will eventually reduce parking and traffic congestion, resulting in a decrease in pollution and fuel emissions, thus allowing people to maintain a much healthier lifestyle.

Table of Contents

List of Figures

List of Tables

1. Preface

World industrialization, population increase, slow pace of city development and mismanagement of the available parking space has resulted in problems related to parking. There is a desperate need for a safe, knowledgeable, efficient and reliable system that can be used to manage the parking facilities properly [1]. According to a report, Smart Parking could result in 2,20,000 gallons of fuels saving till 2030 and approx. 3,00,000 gallons of fuels saved by 2050, if implemented successfully [2].

IoT (Internet of Things) devices is an essential aid to car parking. Already available in many corners of the world, it allows drivers to see if and where there are vacant spaces in a parking lot [3]. In this project, we implemented a similar system with some additional feature using Arduino UNO super starter kit and MATLAB Simulink. The below kit is effectively used to develop a simulated parking environment.

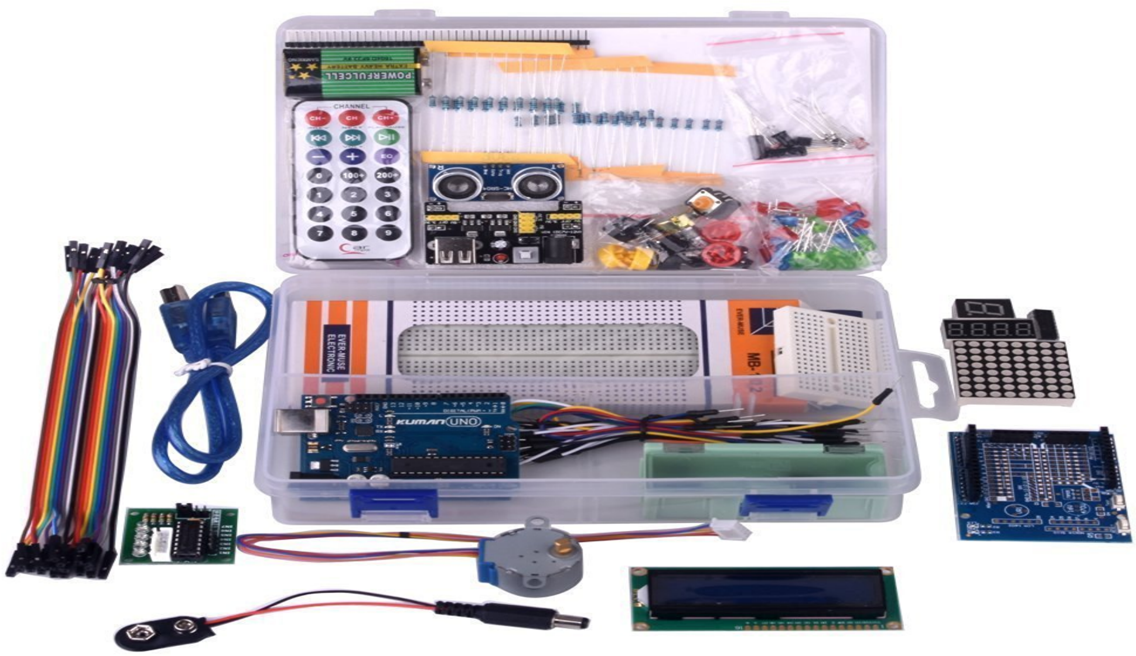


Figure 1: Arduino UNO super starter kit [3]

The proposed system gives information about available parking spaces. This in turn provide benefit to all drivers when they enter the parking facility. In addition, the parking gates and signal lights were synchronized to ease the entire process. Thus, this system helps to identify the number of car present as well as managing all physical entities in the confined parking facility.

2. Prerequisite Analysis

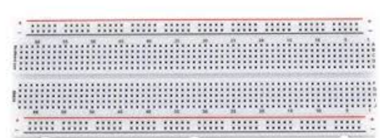
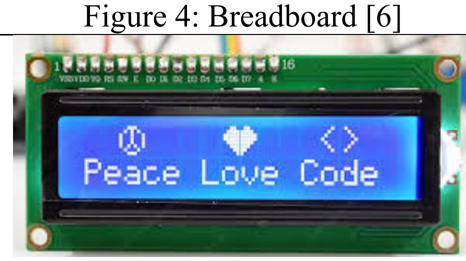
Prior to development of the project, the various requirements were decided upon and analyzed. A major prerequisite to the design and development of this system was a flowchart which identified the various actions to be performed by the system. Another major prerequisite identified was the connection details of the various hardware components used in this system.

3. Hardware and Software Used

To develop a model of a smart parking system effectively, the following hardware components and software were used.

3.1 Components Used

The below table shows the hardware elements used in this project.

Components	NO's	Figure
Arduino UNO R3	1	 Figure 2: Arduino board [4]
USB Cable	1	 Figure 3: Cable [5]
Breadboard	1	 Figure 4: Breadboard [6]
LCD 1602 module	1	 Figure 5: LCD [7]

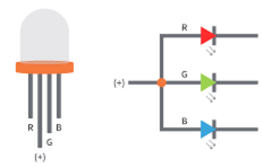
Servo motor SG90	1	 Figure 6: Servo motor [8]
RGB LED	1	 Figure 7: LED [9]
Push button	2	 Figure 8: Button [10]
Potentiometer 10k	1	 Figure 9: Potentiometer [11]
Resistor 1kohm	4	 Figure 10: Resistor [12]
Jumper wires	Few	 Figure 11: Wires [13]

Table 1: Components Used

3.2Software Used

MATLAB and Simulink.

4. Hardware Model

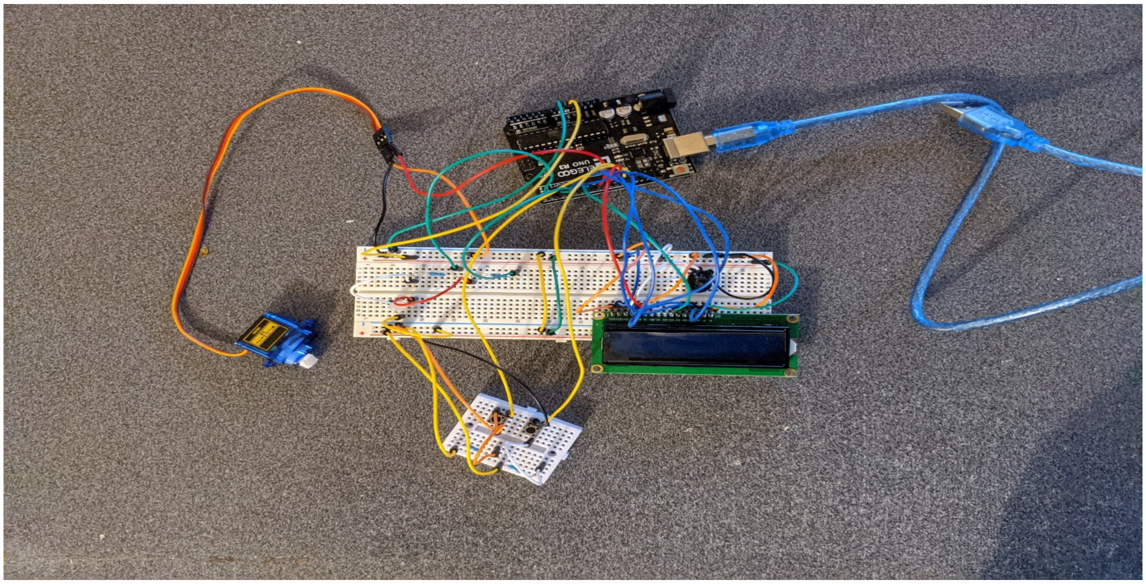


Figure 12: Overall circuit design

Figure 12 shows the overall hardware model of this project. The four major components used are an RGB LED (Red Blue and Green LED), an LCD 1602 module, 2 push buttons and a servo motor.

- The RGB LED has four pins, a red, green and blue pin and a negative pin. The negative is connected to ground whereas the red pin is connected through a 1kohm resistor to pin 2 of the Arduino board and the green pin in the LED is connected through a 1kohm resistor to pin 3 of the Arduino board.

- The LCD module has 16 pins with pins 1 and 2 corresponding to the supply of the LCD and pins 15 and 16 corresponding to the backlight of the module. So, pins 1 and 16 are connected to ground while pins 2 and 15 are connected to 5V. Pin 3 is connected to the potentiometer for varying the contrast of the screen and pin 4, which is the register select pin is connected to pin 10 of the Arduino board. Pin 5 of the LCD is the data read/write pin and is connected to ground, corresponding to write mode, while pin 6 is the data enable pin and is connected to pin 11 of the Arduino. Then data pins D4, D5, D6 and D7 of the LCD are connected to the pins 7, 6, 5 and 4 of the Arduino to facilitate 4 bytes of data input to the LCD module.

- Two push buttons are also used to simulate an ENTER and an EXIT button used by the driver to enter or exit the parking structure. These are connected such that one end goes through a resistor to 5V and is regarded as the positive pin and the other end is connected to ground and is considered the negative end. The positive end of the ENTER button is connected to pin 12 of the Arduino board whereas the positive end of the EXIT button is connected to pin 13 of the Arduino board.

- The servo motor has 3 wires, a red wire which is the positive end, a brown wire which is the negative end and an orange wire which goes to the Arduino board. So, the red wire is connected to 5V, the brown wire to ground and the orange wire to pin 9 of the Arduino board. Also, a plastic arm is attached to the motor to facilitate the motion of a gate barrier arm. Thus, the maximum angle of motion for the servo motor is maintained at 90 degrees. This ensures that the gate barrier arm moves up and down when required, thus providing a more realistic version of the project.

5. Implementation and Working

- The main objective of this project is to simulate a smart parking system, which can identify parking spots effectively, thus reducing traffic congestion while scouting for spots, which in turn enables a reduction in fuel emissions and pollution levels.

- This is achieved by connecting all the required hardware components together and preparing a model of the hardware on the software Simulink. All necessary requirements and user-expectations are to be kept in mind while designing this system.

- After the process of designing, the model must be tested for all the provided test cases, to ensure maximum efficiency.

5.1 Flowchart

Figure 13 depicts the flowchart of the simulated smart parking system.

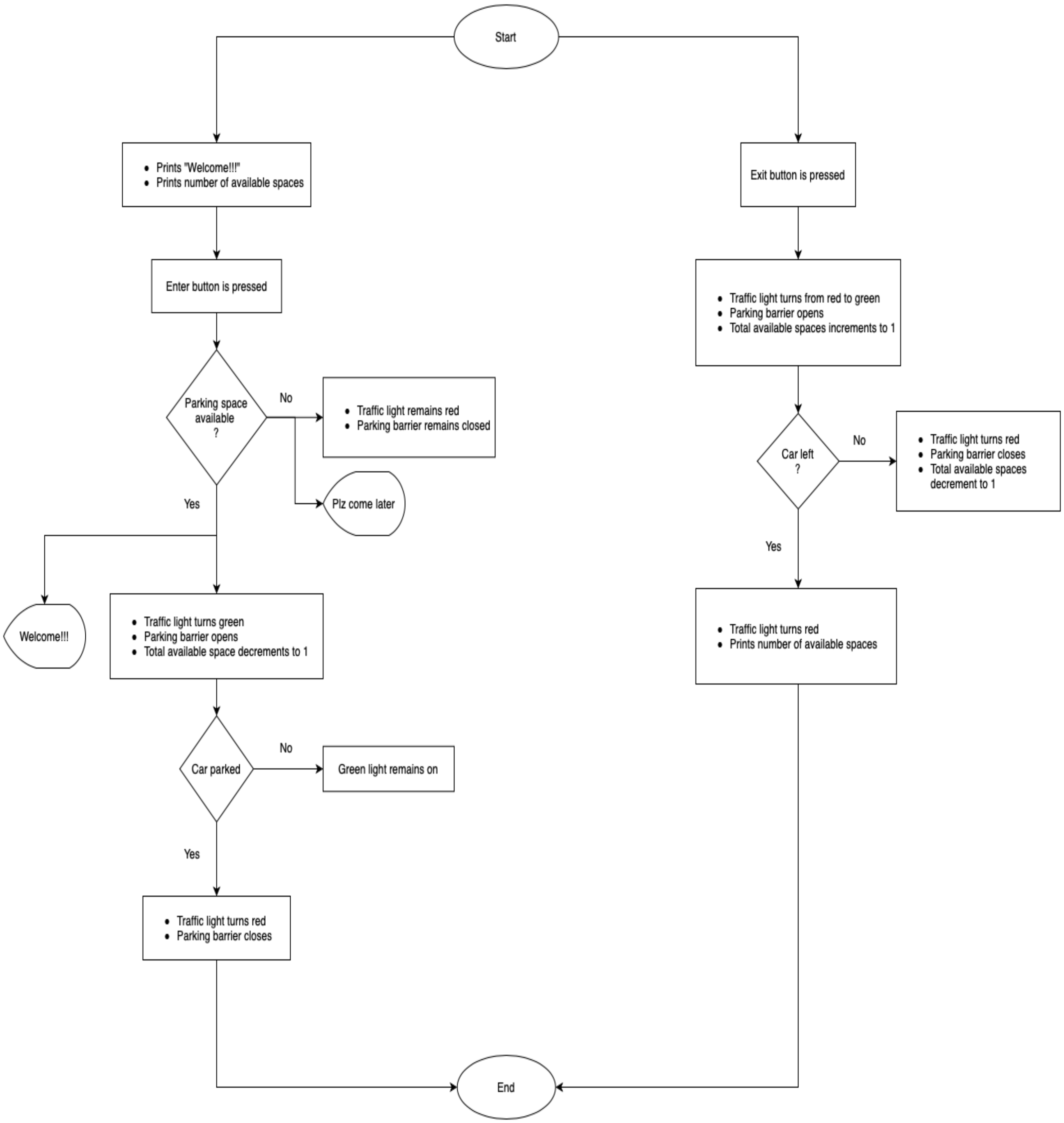


Figure 13: Flowchart of the smart parking project

5.2 Design and Development

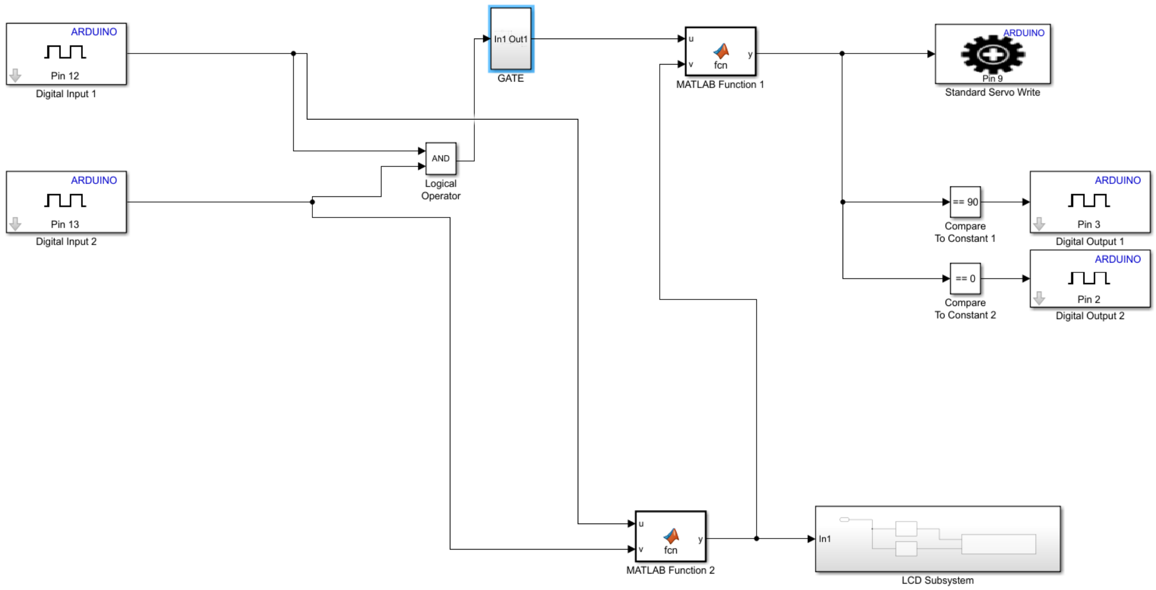


Figure 14: Overall Simulink model

Figure 14 shows the model of the smart parking project simulated using Simulink. It contains two subsystems representing the LCD module and the Gate system for the servo motor, and two MATLAB function blocks, one to represent a counter and the other for checking the counter value before sending input to the servo motor and the RGB LED.

- Digital Input 1:
  This block is used in the model to receive the input from the ENTER push button. If the button is pressed, the input to the block is 0, else value passed remains 1.

- Digital Input 2:
  This block is used to receive input from the EXIT push button. If the button is pushed, the value passed becomes 0, and it remains 1 at all other times.

- GATE Subsystem:
  This subsystem performs the function of regulating the input to the servo motor by passing 90 to the motor when either of the push buttons are pressed and 0 is passed to the motor when the buttons are not pushed. The value 90 is passed because we want the barrier arm connected to the servo motor to move up to 90 degrees and to come back to 0, and this is performed using a sequence of logical operations. Figure (no) shows the Simulink system of the gate subsystem used to regulate the input to the servo motor.

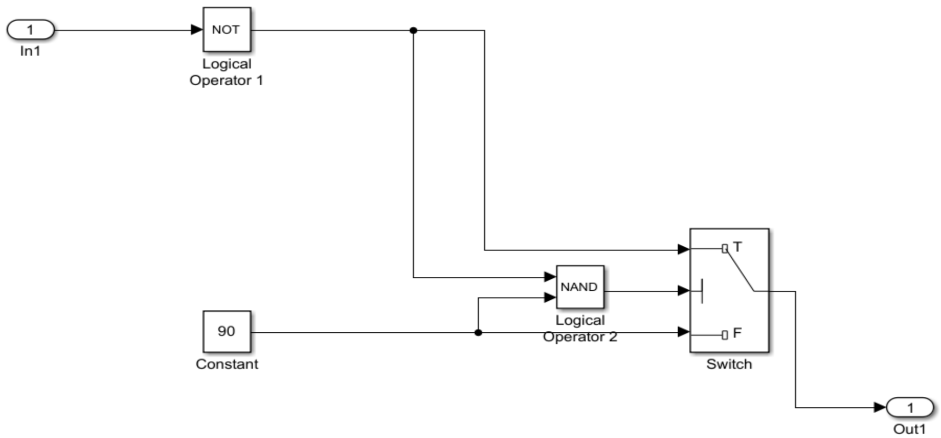


Figure 15: GATE subsystem

- <u>MATLAB Function 1:</u>

    This function is used to check the number of spots available which is received as output from the counter function.

    % Create a MATLAB function y = fcn(u,v)
    % Variable u represents the input to servo motor from the gate subsystem
    % Variable v represents the output from the counter subsystem
    % Check whether v is within the range
    If v<=13 and v>=0, y = u
    Else, y = 0
    End if
    % y is the input to the servo motor and the RGB LED

- <u>MATLAB Function 2:</u>

    This function performs the operations of a counter, to maintain a count of the number of parking spots available to print it in the LCD screen.

    % Create a MATLAB function y = fcn(u,v)
    % Variable u represents the input from ENTER button
    % Variable v represents the input from the EXIT button
    Declare a persistent variable n
    % n represents the no. of spots occupied
    Initialize a variable total = 13
    If n is empty, initialize n = 0
    End if block
    % If ENTER button is pushed
    If u == 0, add 1 to n
    % If EXIT button is pushed
    If v == 0, subtract 1 from n
    End if
    % Set output variable with total spots left

Update value of y = total - n
% If value of y exceeds the range
If y>13, set n = 0
Else if y<0, set n = 13
End if
% Output value is y

- Standard Servo Write:
  This block is used to input the value to the servo motor using pin 9 of the Arduino board and is given the input of 90 or 0, which represents the angle of rotation for the servo motor.

- Digital Output 1:
  The input to the servo motor is compared with the constant 90, and if it is true, then 1 is passed to the RGB LED through the green pin using pin 3 of the Arduino board. For this a digital output block is used with the pin number set at 3.

- Digital Output 2:
  The input to the servo motor is interceded and compared with 0 using a compare to constant block. If it is 0 then 1 is passed to the RGB LED through the red pin using pin 2 of the Arduino board.

- LCD Subsystem:
  This subsystem deals with the LCD component of the project. Its main operation is to accept the number of spots remaining in the parking lot from the counter function as input and print the corresponding output on the LCD screen. The Simulink model of the LCD subsystem used in this project is depicted in figure (no). The subsystem contains two functions, function 3 is used to check the range of the input value and to print either 'Welcome' or 'Plz. come later' depending on the input. Function 4 is used to verify the value and to print the number of spots remaining in the parking lot in line 2 of the LCD module.

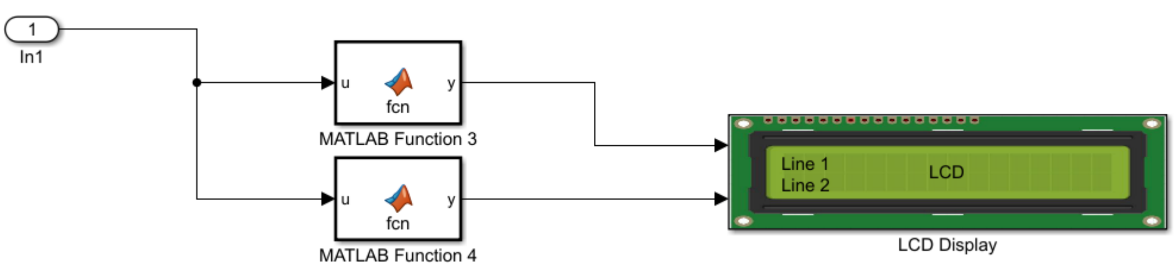


Figure 16: LCD subsystem

```
% MATLAB Function 3
% Create a function y = fcn(u)
% Variable u represents the number of spots available
If u>0, substitute y = ‘Welcome’
Else, substitute y = ‘Plz. come later’
End if
% y is the output from the function which is printed in line 1 of the LCD
% The values must be uint8 values as used in Simulink
```

```
% MATLAB Function 4
% Create a function y = fcn(u)
% Variable u represents the number of spots available
% Verify range of the variable u
If u<0, u = 0
Else if u>13, u = 13
End if
% Print the message corresponding to the value of u
y = ‘No of spots: xx’
% xx is replaced by the number stored in u
% The values must be in uint8 as used in Simulink
% y is the output from the function which is printed in line 2 of the LCD.
```

6. Test Procedure

(A) When the parking lot is empty, 6 cars enter the parking lot sequentially.

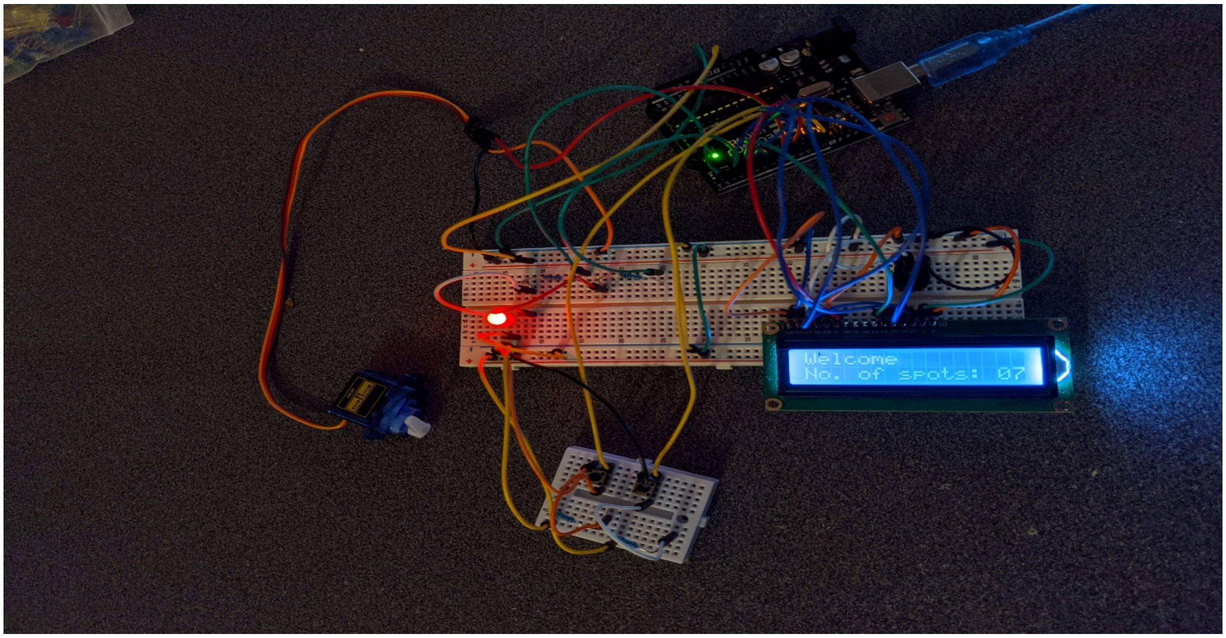


Figure 17: Output after test case 1

(B) 1 car leaves the parking lot.

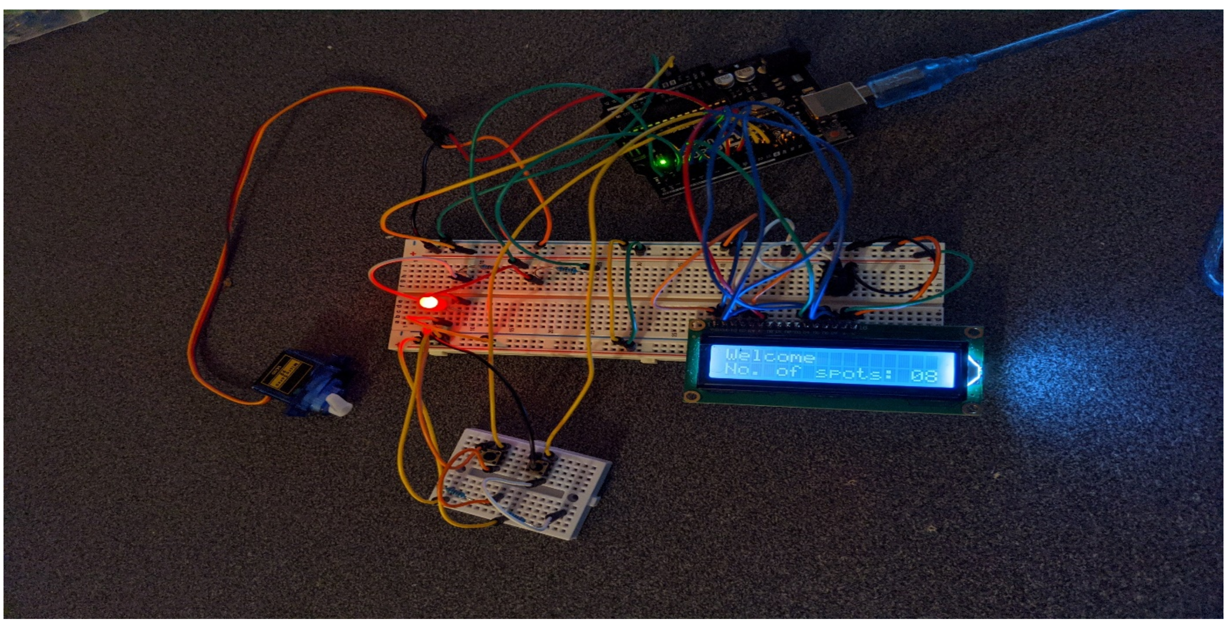


Figure 18: Output after test case 2

(C) 8 cars enter the parking lot sequentially.

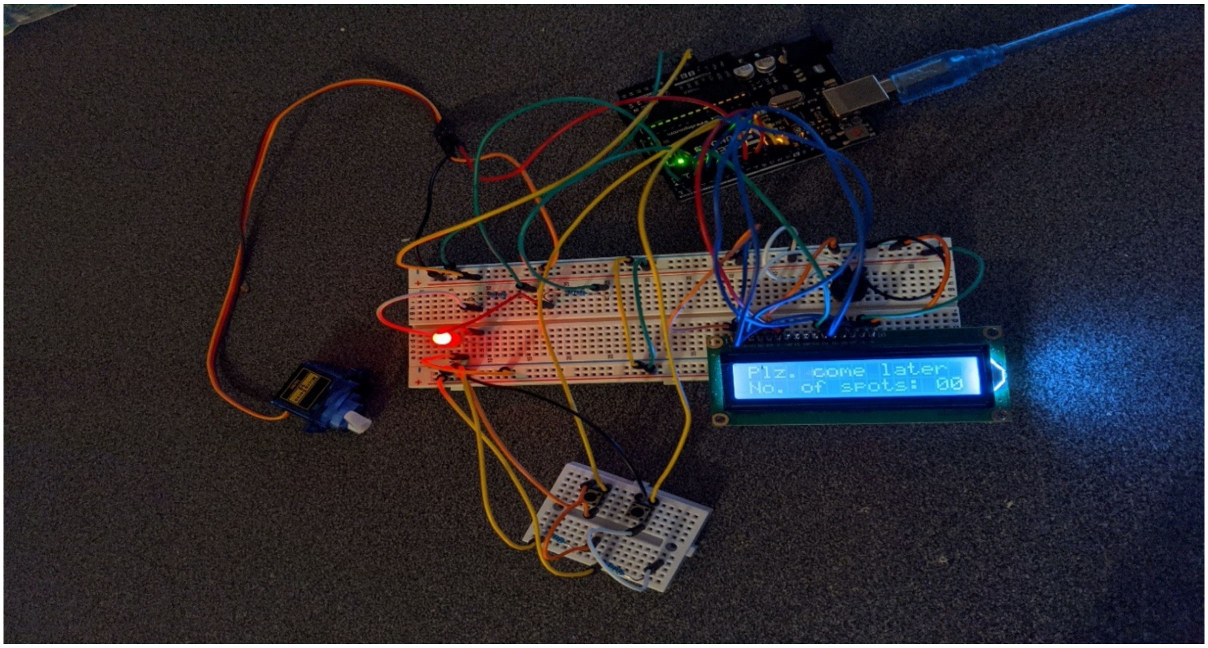


Figure 19: Output after test case 3

(D) 1 car enters the parking lot.

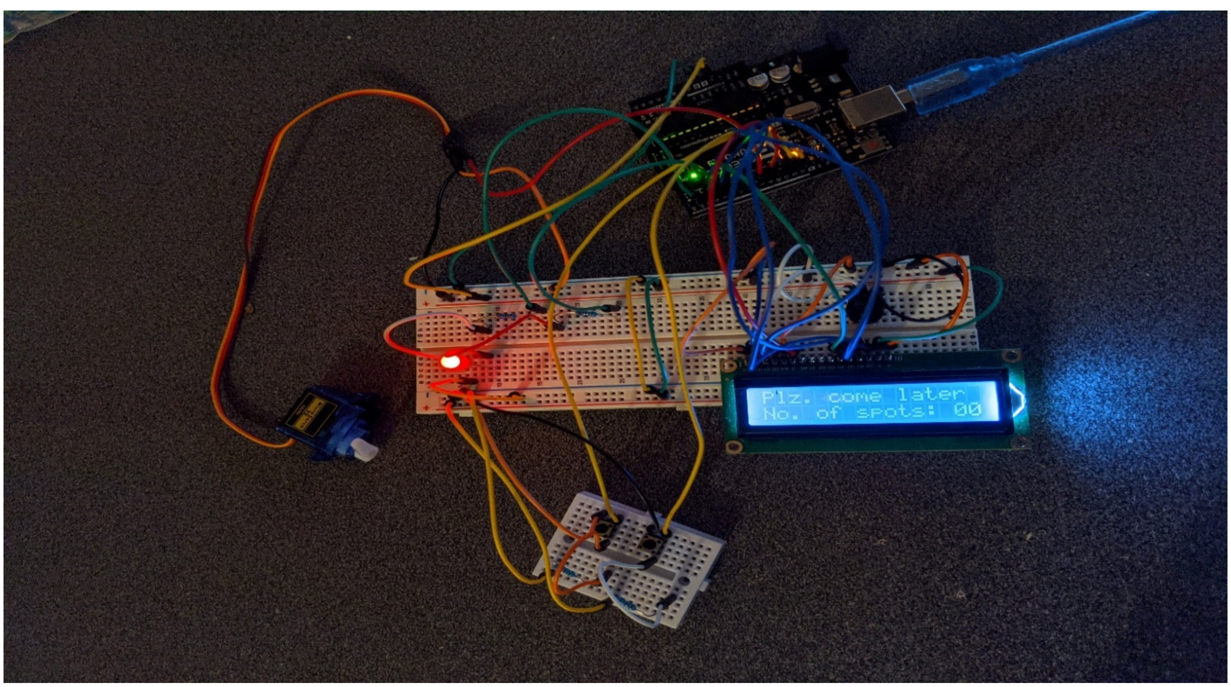


Figure 20: Output after test case 4

(E) 1 car leaves the parking lot.

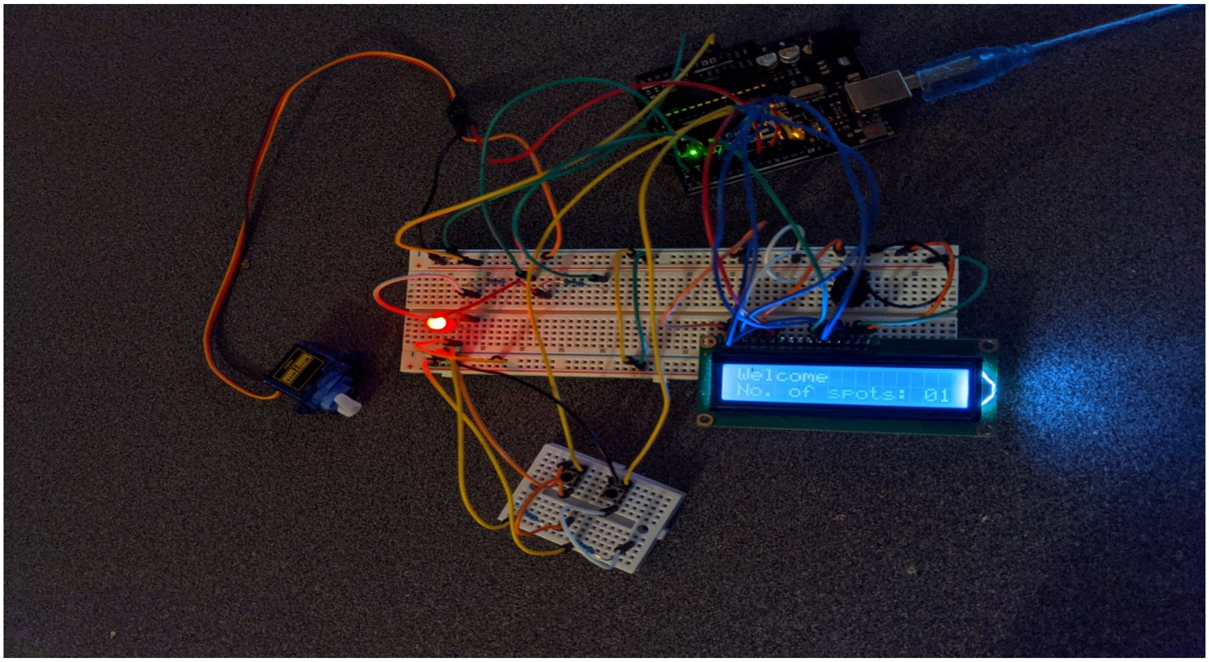


Figure 21: Output of test case 5

(F) 1 car enters the parking lot

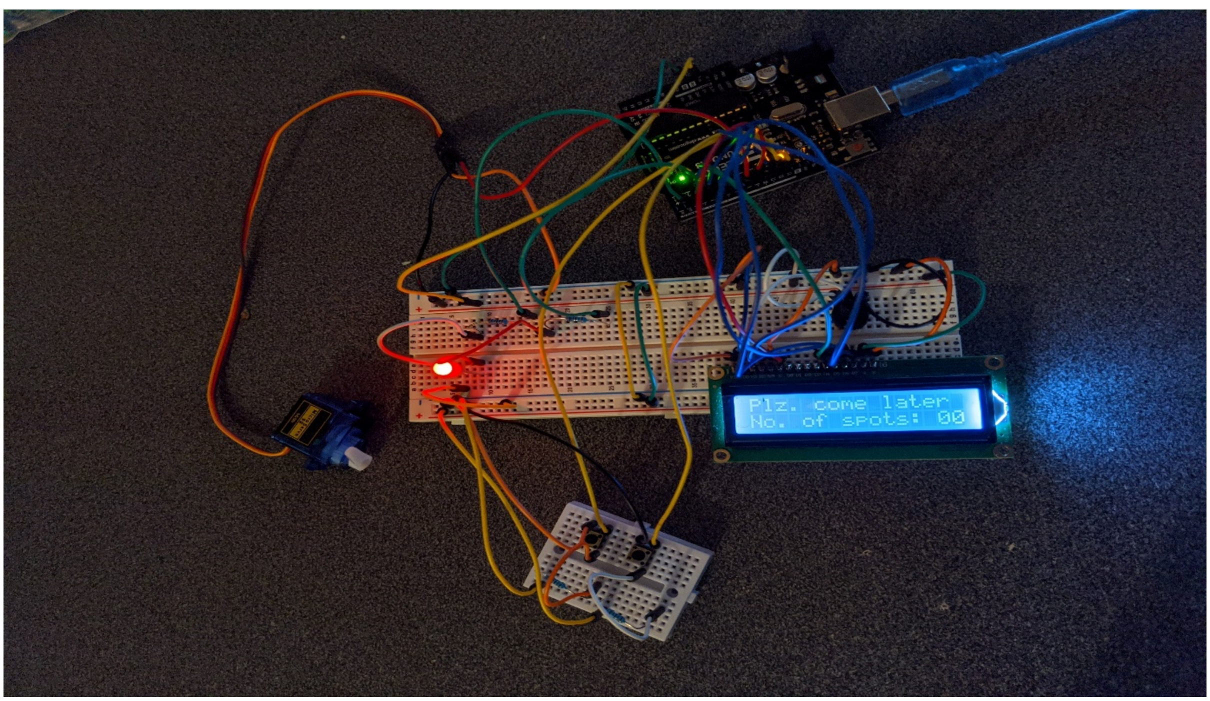


Figure 22: Output of test case 6

7. Challenges Faced During Development

- Since this project does not involve soldering, some of the connections tend to loosen over time, requiring constant attention and care.

- Attempting the counter module using Simulink was quite complicated and very cluttered, thus resulting in the use of a MATLAB function block to simulate a counter.

- Regulating input to the servo motor using a logic circuit as part of the gate subsystem was quite challenging.

- All pin connections were verified to ensure proper output.

8. Benefits of Smart Parking System

❖ Save money
Obviously, automation which results in less manual activity reduces the cost of managing and labor needs.

❖ Optimized parking options
The model provides an integral and detailed approach with less human interference resulting in optimal parking.

❖ Scalability
Smart parking management system will be easier to handle, reducing manpower needs.

❖ Diminished emissions
The advantage of spending less kilometres when looking for a parking spot is that individual emission is that.

❖ Better customer experience
Uncertainty about availability of parking spots can be quite stressful. The smart system provides information to the driver automatically about the number of slots available and ensures enhanced user experience. Further, the barrier arm opens automatically only when parking spot is available. Thus, customers need not have to come out from a car to open the gate.

9. Conclusion

The conceptual model used MATLAB and Arduino technologies to introduce a new automated car parking management system. The benefits of Smart Parking System would go beyond minimizing the city blockage's excessive circumference. It is a solution to the current traffic clog that reduces the annoyance of the driver by providing exact details about the parking lot status. The interest in the smart car parking system will naturally continue to expand in the years ahead and beyond. Parking is, as we know, a challenging and costly job, but the implementation of the Smart Parking Program provides the best way to make use of money and techniques, which in effect increases revenue.

The methods and techniques used to develop this model improve the lifestyle of many people owning cars, saving a lot of time, reducing traffic congestion, saving money and delivering better public service, thus reducing vehicle emissions and pollution problems, and enhancing city visitors. The goal and criteria for designing the smart parking system that were provided were met successfully and would solve the problem efficiently. Any potential improvements could be added upon this existing system to make the parking system more efficient and simpler.

10. References

[1] Faheem,S.A.Mahmud,G.M.Khan,M.Rahman,H.Zafar, “A Survey of Intelligent Car Parking System ” [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1665642313715803. [Accessed: February 3, 2020].

[2] A. Basu, "Happiest Minds," [Online]. Available: https://www.happiestminds.com/whitepapers/smart- parking.pdf. [Accessed: February 3, 2020].

[3] Steve Symanovich, “The future of IoT: 10 predictions about the Internet of Things,” Norton by Symantec, n.d. [Online]. Available: https://us.norton.com/internetsecurity-iot5-predictions-for-the-future-of-iot.html. [Accessed: February 3, 2020].

[4] Sparkfun, “Arduino Board”. Available: https://www.sparkfun.com/products/11021. [Accessed: March 15, 2020].

[5] Electronic Geek, “Cable”. Available: https://www.electronicgeek.ca/collections/arduino-accessories-1/products/hi-speed-usb-a-male-to-b-male-printer-cable-arduino-uno-and-mega. [Accessed: March 15, 2020].

[6] Circuit Uncle, “Breadboard”. Available: https://www.circuituncle.com/product/breadboard-large-buy-online-india/. [Accessed: March 15, 2020].

[7] Last Minute Engineer, “LCD”. Available: https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/. [Accessed: March 15, 2020].

[8] Components 101, “Servo Motor”. Available: https://components101.com/servo-motor-basics-pinout-datasheet. [Accessed: March 15, 2020].

[9] Circuit Bread, “LED”. Available: https://www.circuitbread.com/tutorials/how-rgb-leds-work-and-how-to-control-color. [Accessed: March 15, 2020].

[10] Antratek, “Button”. Available: https://www.antratek.com/mini-push-button-switch. [Accessed: March 15, 2020].

[11] PICAXE Store, “Potentiometer”. Available: https://www.picaxestore.com/res021. [Accessed: March 15, 2020].

[12] GM Electronics, “Resistor”. Available: https://www.gme.cz/rm-1k2-0309-1w-1. [Accessed: March 15, 2020].

[13] Sparkfun, “Wires”. Available: https://www.sparkfun.com/products/11026. [Accessed: March 15, 2020].