Project Report

Smart Parking Management System

Instructor: Prof. Yasser Alginahi

Taarun Dev Karthikeyan

Student number: **110089870**

Group No.: **33** | Submission: 23rd November 2022

GENG8030: Computational Methods and Modeling for Engineering Applications

Master of Engineering – Electrical and Computer Engineering

DECLARATION OF ORIGINALITY

We therefore declare that this Smart Parking Management System project is entirely my own unique effort. Furthermore, I confirm that this work was written with the assistance of a professor and a graduate teaching assistant, and that all sources referenced in the work were clearly referenced in line with departmental regulations, in both the text and the bibliography or references. The work's data and findings have not been fabricated or inflated in any way.
I understand that making false claims about this work will result in disciplinary action under university or departmental regulations. I confirm that I understand that my work may be electronically tested for plagiarism using plagiarism detection software and saved on the server of a third party for future comparison.

ACKNOWLEDGEMENT

I would like to use this occasion to offer our profound gratitude to my mentor, **Prof. Yasser Alginahi**, for his direction, assistance, suggestions, and support in clearing my uncertainties during the project and inspiring me to put up our best efforts.
We shall be failing in our responsibility if we do not express our gratitude to the Graduate Teaching Assistant of the Department of Electrical and Computer Engineering who assisted me from time to time in understanding the issue and the different associated areas.
Finally, acknowledgement would not be complete without thanking all our friends and colleagues who assisted us directly or indirectly throughout our project preparation.

Table of Contents

1. Introduction 4
1.1. Overview 4
1.2. Design Objective 4
1.3. Product outcome 4
1.4. Advantages and Limitations of the Proposed Solution 4
1.4.1. Advantages of Smart Parking System 4
1.4.2. Limitations of Smart Parking System 4
2. Technical Components 5
2.1. Hardware Components 5
2.2. Software Description 6
3. Methodology and Detailed Explanation 7
3.1. Design Description 7
3.2. Workflow Description 7
4. Pseudocode 8
4.1. Entering Vehicle 8
4.2. Exiting Vehicle 9
5. Project Timeline 9
6. Circuit Diagram 10
7. Code 10
8. Testing Scenarios 13
9. Simulation 14
10. Lesson Learned 14
10.1. What went right? 14
10.2. What went wrong? 14
10.3. Recommendations 15
11. Conclusion 15
References 16

Table of Figures

Figure 1 - Arduino UNO 5
Figure 2 - Servo Motor 5
Figure 3 - LCD 5
Figure 4 - RGB LED module 6
Figure 5 - Button 6
Figure 6 - Basic Block Diagram 7
Figure 7 - Workflow Diagram for Entering Vehicle 7
Figure 8 - Workflow Diagram for Exiting Vehicle 8
Figure 9 - Gantt-chart | Project Timeline 10
Figure 10 – Circuit Diagram of Smart Parking Management System 10
Figure 11 – Proteus Design of Smart Parking Management System 14

List of Abbreviations

LCD	Liquid Crystal Display
RGB	Ged Green Blue
LED	Light Emitting Diode
USB	Universal Serial Bus
IDE	Integrated Development Environment
Gnd	Ground
Vcc	Supply Voltage

1. Introduction

Abstract- A smart technology that helps and manages car traffic while parked. In the actual world, if we need to park, we must first seek a parking place by driving around the parking lot. Because the system is aware of all available slots, it directs the user to the designated parking place. If no slots are available, the user will be notified.

1.1. Overview

A system designed to aid in the resolution of the parking problem in busy areas. We will monitor and control the vehicle flow using technology so that you may park your vehicles with convenience and without hassle. We offer a variety of smart parking options for vehicles, including automatic sensors and a smart system. The system greets the user with a welcoming message and directs them to an available slot. When it is full, it informs the user to return later. The parking situation is completely under the control of the system.

1.2. Design Objective

To help automate the parking experience with a smart system that regulates and monitors the vehicle and parking slots. To develop the system, that is aware of the available parking slots and directs the arriving vehicle accordingly based on the available slots. This may be accomplished and illustrated with an Arduino UNO microcontroller, servo motor, and basic electronic components.

1.3. Product outcome

The system will reduce the confusion associated with vehicle parking. This system notifies the incoming if there are no available spots; this saves a lot of time and resources that would otherwise be spent looking for vacant spots. This can be implemented in shopping centers, malls, and movie theatres, among other places.

1.4. Advantages and Limitations of the Proposed Solution

1.4.1. Advantages of Smart Parking System

- It eliminates the need for the driver to exit the vehicle to find a parking space and then return to the vehicle to move it.
- Provides a seamless, user-friendly experience
- Increased turnover implies that more people will be able to find parking and will not have to wait as long.
- Congestion in parking lots and on streets is reduced.
- Reduces CO2 emissions by reducing traffic and driving.

1.4.2. Limitations of Smart Parking System

- For a larger deployment, the infrastructure needs can be expensive.
- The sensors' power consumption is an issue.

- A back-end management system that controls all parking details such as booking slots, time, and space would be a great addition.

2. Technical Components

2.1. Hardware Components

- **Arduino UNO** - The Arduino Uno is a microcontroller board. It contains 14 digital input/output pins, 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. [1]

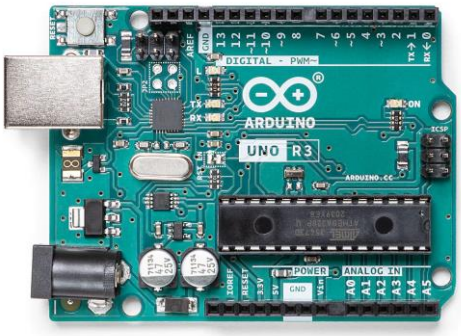

Figure 1 - Arduino UNO

- **Servo Motor** - A servo motor is a rotary actuator that enables accurate angular position control. It is comprised of a motor and a sensor for position feedback. To complete the system, a servo drive is also required. [2]

Figure 2 - Servo Motor

- **LCD (Liquid Crystal Display)** - It is a type of flat panel display that operates primarily with liquid crystals. [3]

Figure 3 - LCD

- **Common cathode RGB LED module** - RGB LED modules can produce a wide range of light colors. They are made by encasing three red, green, and blue LEDs in a transparent or semitransparent plastic shell with four pins. [4]

Figure 4 - RGB LED module

- **Button** - When you press a pushbutton or switch, it connects two points in a circuit. [5]

Figure 5 - Button

2.2. Software Description

- **Arduino C++** - The Arduino code is written in C++, with the addition of several specific methods and functions that will be used for certain functionality. C++ is a computer language that is easy to understand and use. When you produce a 'sketch' (Arduino code file), it is analyzed and compiled to machine language. [6]
- **Arduino IDE** - The major text editing application used for Arduino programming is the Arduino Integrated Development Environment (IDE). It's where you'll write your code before uploading it to the board, you're programming. Sketches are the name given to Arduino code. [7]
- **MATLAB** - You may use MATLAB to interface with an Arduino board using the MATLAB Support Package for Arduino® Hardware. Because MATLAB is an interpreted high-level language, you may view results from I/O commands without compiling. MATLAB has dozens of built-in math, engineering, and graphing tools that may be used to swiftly analyze and display data from your Arduino. [7]

3. Methodology and Detailed Explanation

3.1. Design Description

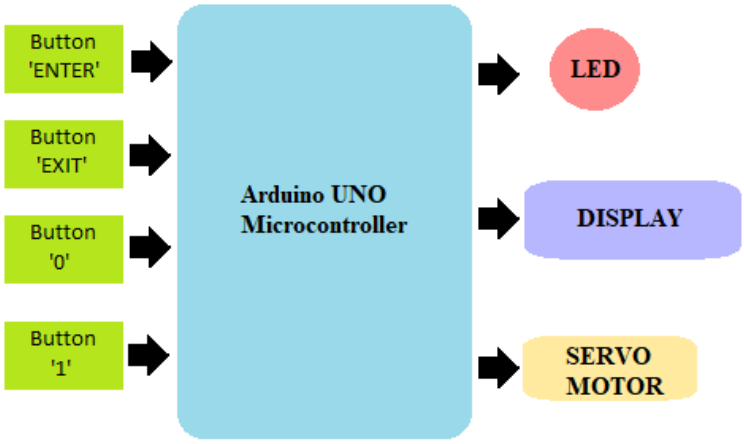

Figure 6 - Basic Block Diagram

The Arduino UNO Microcontroller will be used to build the Smart Parking system prototype. The system is operated via four buttons. The first two buttons are '*ENTER*' and '*EXIT*', while the last two are '*0*' and '*1*'. These are the sources of input for the Microcontroller.

We have an RGB LED module that signals the car to stop or move. A display is also used to show messages to the user that the smart system wants to show them. The servo motor closes and opens the parking terminal gate.

3.2. Workflow Description

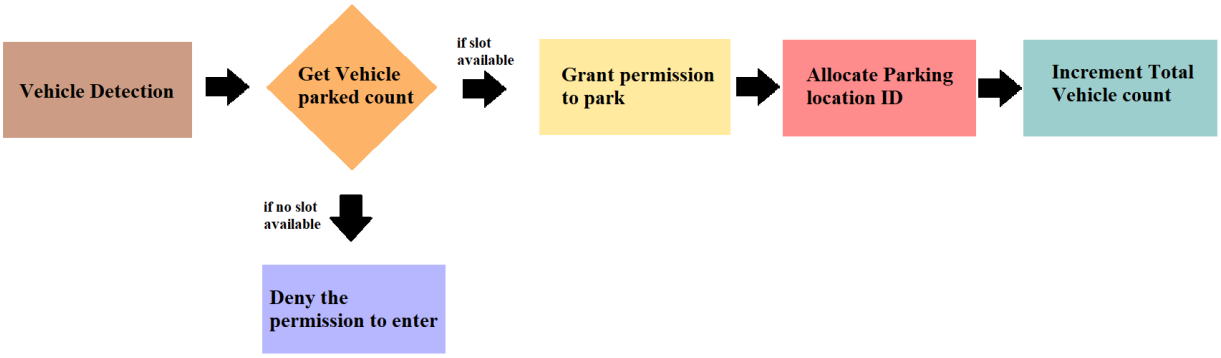

Figure 7 - Workflow Diagram for Entering Vehicle

When there are empty parking spaces, an LCD display must indicate the total number of open parking spaces as well as a "Welcome!!!" message. The parking barrier arm must be closed with a RED traffic signal turned on.

To enter the parking garage, the driver must push the 'Enter' button. The system then assigns the Vehicle a four-digit ID number and shows it on the LCD. The parking slot number

in which the automobile must be parked is subsequently shown by the system. The gate will then be opened, and the RED traffic signal will turn to GREEN, allowing the automobile to enter. The barrier arm shuts once the automobile enters, the traffic signal returns to RED, and the number of available parking spaces is decreased by one.

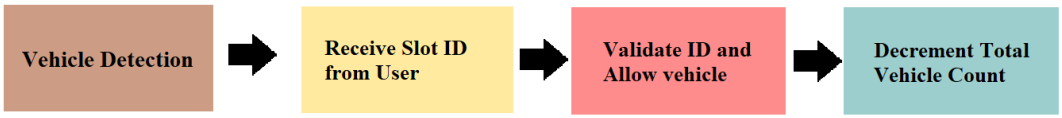

Figure 8 - Workflow Diagram for Exiting Vehicle

To escape the parking lot, the driver must press the Exit button. The system will then prompt you for your ID. The driver must provide his four-digit ID number. The ID will be checked. If the ID is not genuine, the system prompts the driver to input it again. If it is legitimate, the barrier arm will be raised, and the traffic signal will change from red to green. When the automobile exits the parking lot, the barrier bar shuts again, the traffic signal goes RED, and the number of available parking spaces on the LCD increases by one.

The parking lot is anticipated to have a capacity of 13 spaces. When there are no more parking spaces available, the message on the LCD must change from "Welcome!!!" to "Please come later." If a driver attempts to enter the parking lot by hitting the Enter button, the barrier bar should not open.

We have 13 identification cards and 13 parking spaces. The IDs are ['0001',0010',0011', '0100', '0101', '0110','0111', '1000', '1001','1010','1011','1100','1101'] and the Parking Slots are [Slot1, Slot 2, Slot 3, Slot 4, Slot 5, Slot 6, Slot 7, Slot 8, Slot 10, Slot 11, Slot 12, Slot 13]. These IDs and Slots are assigned on a first-come, first-served basis. The slots must be filled in the following order: Slot 1 to Slot 13. That implies the first car should park in Slot 1 and the last vehicle should park in Slot 13. Driver ID 0001 is allocated to Slot 1, Driver ID 0010 is assigned to Slot 2, and Driver ID 1101 is assigned to Slot 13.

4. Pseudocode

4.1. Entering Vehicle

START

while true

Display available slot.

Set LED color to ‘RED’.

User presses the ‘ENTER’ button.

Display “Welcome!” greeting to the user.

Get total vehicle count.

If slot available:

Allocate a 4-digit ID to the vehicle.

Change LED color to ‘GREEN’.

Trigger the servo motor to lift the bar.

Increment total vehicle count.

Change back the LED color to ‘RED’.

Else:

Display “Please come later”.

Exit.

Exit

END.

4.2. Exiting Vehicle

START

while true

Display available slot.

Set LED color to ‘RED’.

Listen to the user for slot id.

Verify slot id occupancy.

Change LED color to ‘GREEN’.

Decrement total vehicle count.

Trigger the servo motor to lift the bar.

Change back the LED color to ‘RED’.

exit

END.

5. Project Timeline

The Work Breakdown Structures with respect to their timeline is shown in the below Gantt-chart.

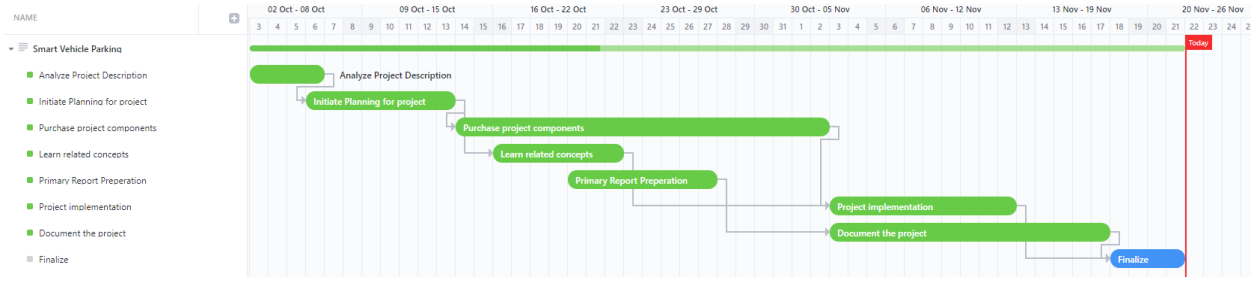

Figure 9 - Gantt-chart | Project Timeline

Work Breakdown Structures and their respective progresses are listed in below image.

6. Circuit Diagram

The project's circuit diagram is shown below.

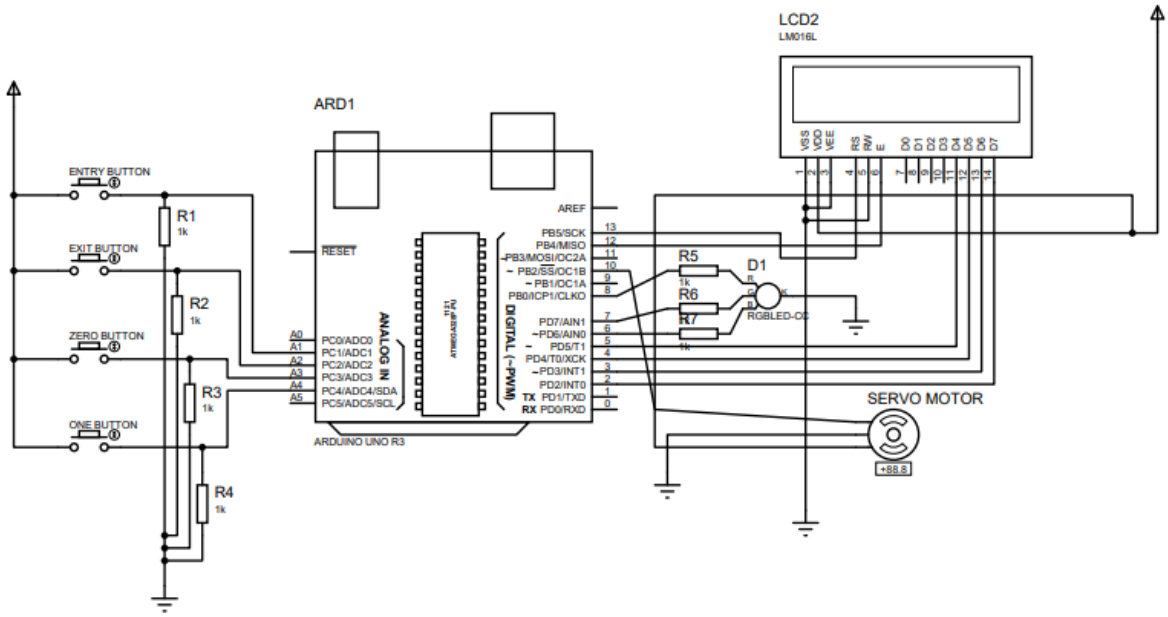

Figure 10 - Circuit diagram of Smart Parking Management System

7. Code

```
% ============================================================
% Project: Smart Parking Management System
% Author: Taarun Dev Karthikeyan
%
% Date: 22-11-2022
% Time: 2pm
% Desc.: A smart technology that helps and manages car
%   traffic while parked. In the actual world, if we need
%   to park, we must first seek a parking place by driving
%   around the parking lot. Because the system is aware of
```

```
%   all available slots, it directs the user to the
%   designated parking place. If no slots are available,
%   the user will be notified.
% ============================================================

a = arduino('COM3','Uno','Libraries',{'LCD/LCDAddon','Servo'}); % Arduino Connection
+ Library import for Arduino

% LCD object
lcd                                                                                 =
addon(a,'LCD/LCDAddon','RegisterSelectPin','D13','EnablePin','D12','DataPins',{'D5','
D4','D3','D2'});
initializeLCD(lcd);

% Constants
RED = 'D8';
GREEN = 'D7';
ON = 1;
OFF = 0;
availablity                                                                         =
["0001","0010","0011","0100","0101","0110","0111","1000","1001","1010","1011","1100",
"1101"] ;
availableSlots = 13;
parked = string;
pin = [];
isExitPressed = false;
welcomeMSG = true;

% Servo Object
gate = servo(a,"D10");
writePosition(gate,0);

while true

    %init buttons
    enter = readVoltage(a,'A1');
    exit = readVoltage(a,'A2');
    zero = readVoltage(a,'A3');
    one = readVoltage(a,'A4');

    %init singal color RED
    writeDigitalPin(a,RED,ON);

    parked(cellfun('isempty',parked)) = [];

    % Welcome Message
    if ~isExitPressed && welcomeMSG
        clearLCD(lcd);
        printLCD(lcd,'Welcome!');
        printLCD(lcd, strcat('Slot avail.: ', num2str(availableSlots)));
        welcomeMSG = false;
    end

    % On press of ENTER button
    if enter>=4
```

```
    clearLCD(lcd);
    if ~isempty(availablity)
        availableSlots = availableSlots - 1;
        writeDigitalPin(a,RED,OFF);
        writeDigitalPin(a,GREEN,ON);
        writePosition(gate,0.5);

        parked(end+1) = string(availablity(1));
        clearLCD(lcd);
        printLCD(lcd, strcat('Parking ID: ', num2str(availablity(1))));
        printLCD(lcd, strcat('Slot avail.: ', num2str(availableSlots)));
        availablity = availablity(2:end);
        pause(3);
        writePosition(gate,0);
        writeDigitalPin(a,GREEN,OFF);
        writeDigitalPin(a,RED,ON);
    else
        clearLCD(lcd);
        printLCD(lcd, 'Parking Full!');
        printLCD(lcd, 'Please ComeLater');
        pause(3);
    end
   welcomeMSG = true;
end
% On press of EXIT button
if exit>=4
    isExitPressed = true;
    pin = [];
    clearLCD(lcd);
    printLCD(lcd,'Enter Pin');
    printLCD(lcd, num2str(pin));
    welcomeMSG = true;
end
% Zero button functionality
if zero>=4 && isExitPressed
   pin = [pin, OFF];
   clearLCD(lcd);
   printLCD(lcd, num2str(pin));
end
% One button functionality
if one>=4 && isExitPressed
   pin = [pin, ON];
   clearLCD(lcd);
   printLCD(lcd, num2str(pin));
end
% Exit Logic
if length(pin) == 4 && isExitPressed
    if length(parked) > 0
        oldSize = length(parked);
        parked = erase(parked, strrep(join(string(pin)),' ',''));
        parked(cellfun('isempty',parked)) = [];
        if length(parked) < oldSize
            availableSlots = availableSlots + 1;
            availablity(end+1) = strrep(join(string(pin)),' ','');
            clearLCD(lcd);
```

```
                printLCD(lcd, 'Thank You!!');
                writeDigitalPin(a,RED,OFF);
                writeDigitalPin(a,GREEN,ON);
                writePosition(gate,0.5);
                pause(3);
                writePosition(gate,0);
                writeDigitalPin(a,GREEN,OFF);
                writeDigitalPin(a,RED,ON);
            else
                clearLCD(lcd);
                printLCD(lcd, 'Incorrect ID');
                pause(3);
            end
        else
            clearLCD(lcd);
            printLCD(lcd, 'NoVehicle in lot');
            pause(3);
        end
        isExitPressed = false;
        welcomeMSG = true;
    end
end
```

8. Testing Scenarios

The project will be thoroughly tested to ensure that the project objectives are met. The test technique is shown in the following steps.

Case 1: When the parking lot is empty, six cars enter sequentially.

Observation: Parked Vehicles – 6, available slots – 7.

Case 2: When the parking lot is empty, one car enters, and the same car exits.

Observation: Parked Vehicles – 0, available slots – 13.

Case 3: When the parking lot is empty, one car enters, and the same car tries to exit with a different pin.

Observation: Parked Vehicles – 1, available slots – 12, Message Displayed – 'Incorrect PIN'.

Case 4: When the parking lot is empty, 14 cars enter sequentially.

Observation: Parked Vehicles – 13, available slots – 0, Message Displayed – 'Parking Full! Please Come Later'.8

Case 5: When the parking lot is empty, try to exit with a pin.

Observation: Parked Vehicles – 0, available slots – 13, Message Displayed – 'No Vehicle in lot'

9. Simulation

Proteus Design Suite software is used to realistically simulate the same project. Unfortunately, I was unable to connect to proteus and run the MATLAB code. As a result, similar Arduino C++ code is written.

The Proteus Design Suite is a closed-source software tool suite primarily used for electrical design automation. Electronic design experts and technicians mostly utilize the program to develop schematics and electronic prints for the fabrication of printed circuit boards. [9]

Before ordering a physical prototype, Proteus can be used to design, test, and debug fully embedded systems inside schematic capture. VSM introduces AGILE development to embedded workflows.

The following is the Design of the same Circuit in Proteus for Simulation purpose.

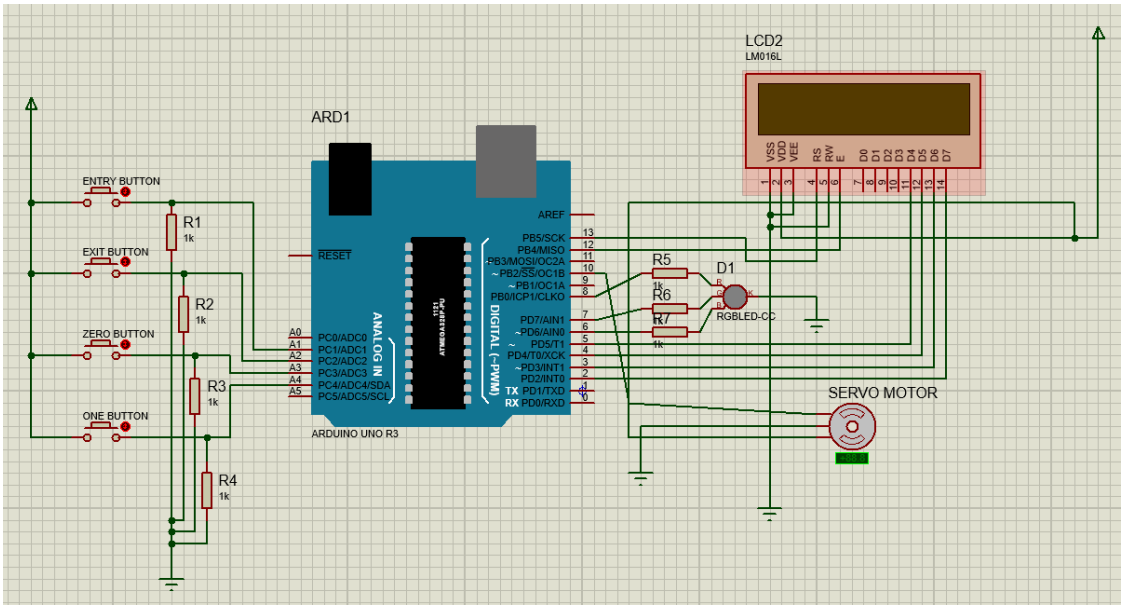

Table 11 - Proteus Design of Smart Parking Management System

10. Lesson Learnt

10.1. What went well?

- Capable of acquiring all component knowledge and implementing it according to the planned timetable.
- Able to design and code the project on the respective timeline.
- Using MATLAB, learned how to create an add-on file for the Arduino interface.

10.2. What went wrong?

- Unable to connect virtually and use MATLAB code for simulation purposes

- Because only one participant was involved, the project's cost and effort appeared to be higher.

10.3. Recommendations

For simulation purposes only, Arduino C++ code is used instead of MATLAB code.

11. Conclusion

As a result, the technology presented in this project may be utilized to lessen the uncertainty around car parking. If there are no open places, this system alerts the arriving party; this saves a lot of time and resources that would otherwise be required to search for open spots. This may be done, among other locations, in malls, movie theatres, and shopping complexes.

References

[1] *Arduino Uno REV3*. Arduino Official Store. (n.d.). Retrieved October 26, 2022, from https://store.arduino.cc/products/arduino-uno-rev3

[2] *Servo Motors*. Kollmorgen. (2021, November 18). Retrieved October 26, 2022, from https://www.kollmorgen.com/en-us/products/motors/servo/servo-motors/

[3] Wikimedia Foundation. (2022, October 22). *Liquid-crystal display*. Wikipedia. Retrieved October 26, 2022, from https://en.wikipedia.org/wiki/Liquid-crystal_display

[4] *RGB led module*. Wiki. (n.d.). Retrieved October 26, 2022, from http://wiki.sunfounder.cc/index.php?title=RGB_LED_Module

[5] Team, T. A. (n.d.). *Button*. Arduino. Retrieved October 26, 2022, from https://www.arduino.cc/en/Tutorial/BuiltInExamples/Button

[6] *Everything you need to know about arduino code*. circuito.io blog. (2019, January 1). Retrieved October 26, 2022, from https://www.circuito.io/blog/arduino-code

[7] Team, T. A. (n.d.). *Software*. Arduino. Retrieved October 26, 2022, from https://www.arduino.cc/en/software

[8] *Arduino support from Matlab*. Hardware Support - MATLAB & Simulink. (n.d.). Retrieved October 26, 2022, from https://www.mathworks.com/hardware-support/arduino-matlab.html

[9] *PCB Design & Simulation Made Easy*. PCB Design and Circuit Simulator Software - Proteus. (n.d.). Retrieved November 22, 2022, from https://www.labcenter.com/