University of Windsor

Project Report

on

Smart Parking Management System

Computational Methods and Modelling for Engineering Applications

(GENG-8030-Summer-2020)

Submitted By: -

Project Group: - 17

Anmol Panchal	**110021741**
Nikhil Patel	**105148841**
Primit Patel	**110016936**

Submitted To:

Prof. Roozbeh Razavi-Far

Table of Content

Table of Figures

1. Introduction

In today's quickly growing world, there has been a tremendous increase in the number of automobiles. There are several reasons for this increase in the number of vehicles. The leading reason is the considerable distance between the workplace and residence. Most companies are located in the city centre and the most maximum of commuters reside far from that place. Because of this reason, companies need to reserve space for office staff in public parking which has led to creating traffic congestion in downtown. Along with this due to the rise in population nowadays it is common to have two or more vehicles even in a small-sized family. With such rapid growth and a fast-paced environment, it is getting harder and harder to cope up with this increasing demand, and to cope up with these demands we need a smart solution to work efficiently and effectively. Our goal is to propose a smart parking system that can minimize the congestion and improve driver's experience by efficiently letting the drivers let it in the parking and let them know in advance if it's full so they do not have to waste their time for searching a spot. This system can also keep track of available free spots and keep count of it along with that it will also great the driver when riding in the parking. Thus, the space for the parking will be efficiently used and most drivers would be benefitted from this service. Public parking, hospital, offices, shopping malls, etc are some of the obvious places where this technology can be consolidated and will be benefited by this project. The main aim of this project is to reduce the inconvenience faced by the people because of improper management of parking. Furthermore, this project would be cost-effective as well as it will eliminate the burden on the parking enforcement officials to regularly conduct patrols for illegal parking across the city.

2. Project Objective:

There are plenty of technical approaches available to solve this big problem. Many organisations conduct research to provide a real-time solution, but the solution should be based on a variety of factors such as budget, efficient automation and potential insights. The main aim of this project is to design and build a prototype to solve the parking problem in megacities. Developing this project would give one an idea of either how to execute a fairly complex project or give one enough hands-on experience to apply with full confidence for an entry-level job on Microcontrollers application. This solution is also economical as this reduces the manual workforce and reduce traffic flow. This system will also help in how to program Arduino and make it work on any system.

3. Project Scope:

The aim of this project is to find out if the car park has free slots, or if all of them are complete by showing multiple free slots at the entrance to the parking. The driver must press Enter button to open the gate barrier. The barrier arm was removed after parking and the traffic light green to red again, and the number shown decreased by one. To exit, the driver must press the Exit button. When the car leaves the number of parking spaces open to it will be reduced by one. If the parking lot is full, the next entry bar can be closed by pressing the Enter button. Show changes from welcome to 'please come later, full parking' post.

4. Project Description

In this project we will also use some components; Arduino UNO, servo motor, LCD module, RGB LED panel, button, 5 Volt relay and a few wires. Here, we consider 13 available parking spaces. LCD module used to display messages according to opened parking spots. Arduino UNO board can run the machine using MATLAB code. For various tasks the signal will be indicated by the RGB module. The LCD module shows "Welcome" message when one of 13 parking spaces is open otherwise it shows "Please come later'. The parking barrier arm has to be in the closed position with a RED traffic light on. If a car wants to enter the parking lot, the driver must press Enter to open the barrier gate. Once the gate is open, the RED light will change GREEN to allow the car to enter. The traffic light turns back into RED, and once the car is parked, the number of available parking spaces is reduced by one. If a driver tries to reach the car park by pressing the Enter button and no spaces are vacant, the barrier bar is not raised. To exit a car from the parking lot the driver must press the Exit button. The barrier bracket is opening, and the lights are turning from RED to GREEN. The barrier bar will close when the vehicle leaves the parking lot. The traffic light turns to RED after that, and the number of available parking spaces shown at the LCD increases by one. This logic will be implemented as a code for the tasks performed in a loop.

5. Hardware Requirement

❖ **Arduino Uno R3 with the ATmega328 microcontroller**

Figure 1. Arduino UNO REV3

The Arduino Uno R3 is a microcontroller board based entirely on an ATmega328 AVR microcontroller with removable dual-inline package (DIP). It has twenty digital I / O pins

(6 can be used as PWM outputs and 6 can be used as analogue input). You can get programmes from the easy-to-use Arduino computer programme such as using MATLAB or C, C++ or Java programming in the Arduino. The Arduino has a widespread community of help, which makes working with embedded electronics a very handy way to get started. The Arduino Uno is microcontroller-based device. It has 20 digital I /O pins (6 of which can be used as PWM outputs and 6 can be used as an analogue input), 16 MHz resonator, USB connexion, power jack, ICSP header, and reset button. It contains everything you need to support the microcontroller; simply connect it to a computer with a USB cable or power it to get started with an AC-to - DC adapter or battery [1].

❖ **LCD Screen**

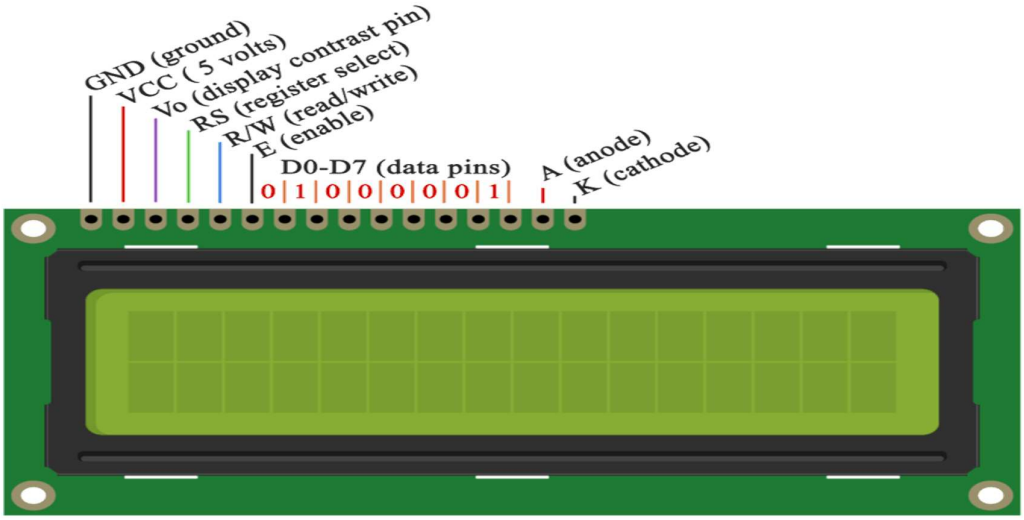

Figure 2. LCD 16*2 Module

Liquid Crystal Display is controlled by the Liquid Crystal library and consists of 16 pins with 8 data pins, 1 read / write pin, 1 display contrast pin, 2 power pins and LED backlight pins [2].

❖ **Servo Motor**

Figure 3. Servo Motor

The servo motor used in the Arduino project consists of three leads, out of which the colour red is 5V, the colour brown or black is ground, and the other colour is orange, usually connected to the digital pin 10. The length of a pulse determines the position of the servo motor and for every 20 milliseconds it is expected to receive a pulse. Servo

motors are generally used for accurate control in varying linear and angular position, velocity and acceleration [3].

- **Common Cathode RGB LED Module**

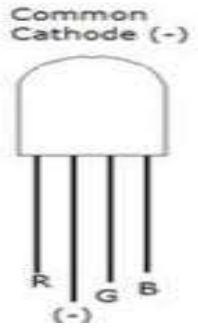

Figure 4. Common Cathode RGB LED

This consists of four pins, one of which is cathode(ground), the other three are RED, GREEN and BLUE colours. Three colour pins also share a common negative cathode pin connection [4].

- **Jumper wires with connector pins**

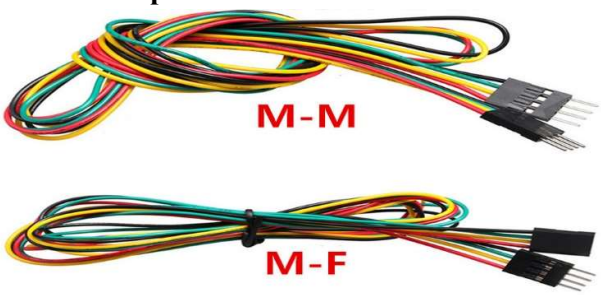

Figure 5. Jumper Wires (M-F, M-M)

Jumper wires are nothing more than simple wires with connector ends on both sides that are useful for connecting Arduino to breadboard. These are available in different colours and it means nothing but distinguishing between Arduino breadboard connexions. Three types of jumper wires are also available, such as male-to - male, female-to-female and male-to - female wires, and from these, male to male wires are commonly used. We can also use steppers and wired insulated wires to make jumper wires and it's most useful for beginners.

- **Breadboard**

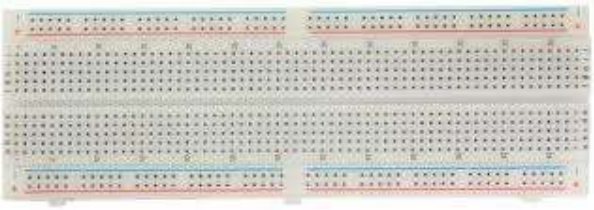

Figure 6. Bread Board

This component is called solderless breadboard, as the connexion of wires to the Arduino does not require soldering and is mostly used for temporary project experimentation. It is also used for quick testing before any design of the circuits is finalised. Two rows top and bottom are connected horizontally, and the remaining holes are linked vertically. Jumper wires are used to connect to Arduino in breadboards, and the top and bottom rows of holes are used to supply the power.

- **Resister**

Figure 7. Resister

It is an electrical component which, by limiting the current, resists current flow and is a passive component which has two terminals at both ends. It is also useful for purposes such as adjusting signal levels and dividing voltages. Resistance department is ohm, and resistor is linked between LED and ground cathode.

- **Potentiometer**

Figure 8. Potentiometer

A potentiometer is a variable resistor and is called a voltage regulator, because it helps to regulate circuit voltage and has three terminals. One pin goes out of 3 terminals to the ground, another to the VEE and the last to the V0 LCD (display contrast pin) [5].

- **Push Button**

Figure 9. Push Button

It is an electric component that connects two points in the electrical circuit when the push button is pressed. The two wires go into two vertical rows in the power supply board and the third one goes into the Arduino board's digital pin. [6].

6. Software Requirement

The Arduino Support package makes it easy to write a simple code and upload it to the Arduino board.

- ❖ Arduino Support for MATLAB
- ❖ MATLAB Software

7. Circuit Diagram

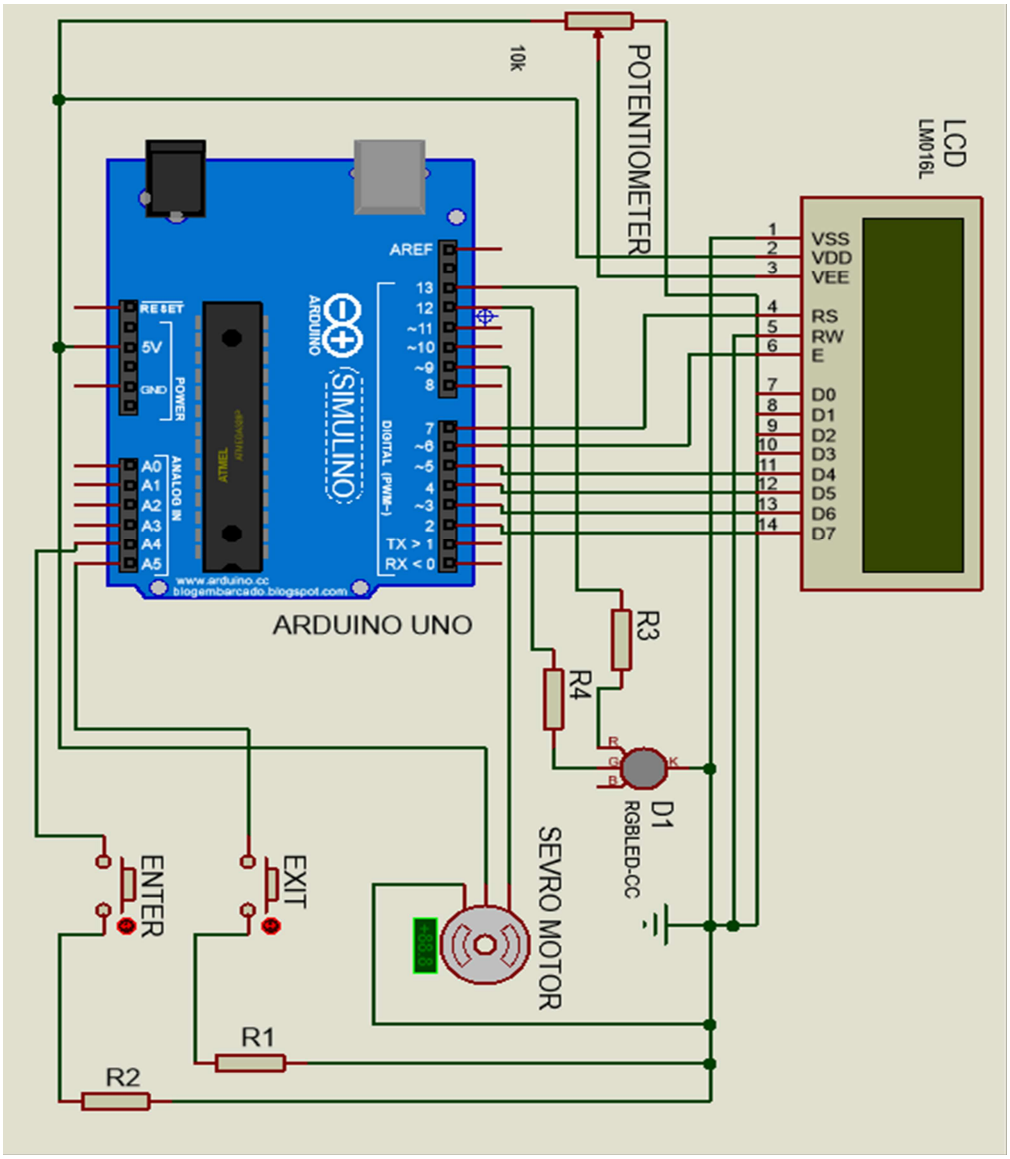

Figure 10. Circuit Diagram

8. Flow Chart

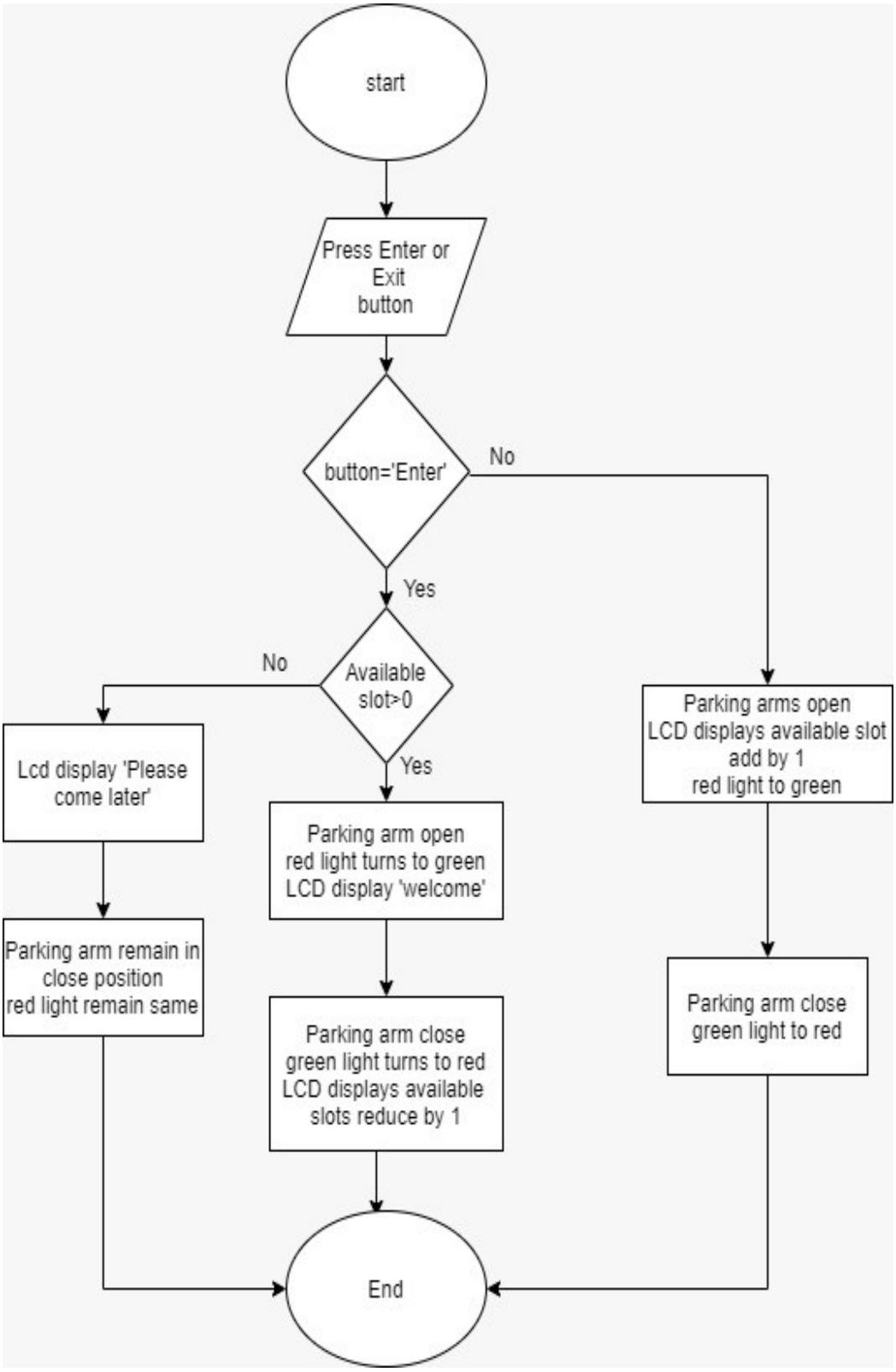

Figure 11. Flowchart of Project

9. Development and Procedure

Overall, circuit connection of smart parking system follows in these steps. The servo motor red pin connects to a positive 5V supply in the overall connected circuit, black is connected ground, orange pin connected to D9 in Arduino board. Next, one wire from each is grounded out with resistance of two push buttons, and the other wire from each connects to A4 and A5 (entry button and exit button). Then, RED colour pin connects to D13 in the RGB cathode circuit, GREEN pin connects to D12 in Arduino and cathode pin is grounded. Potentiometer one pin is connected to positive after that, another pin is grounded, and the last pin is connected to LCD display V0. Finally, K and VSS are connected to positives and negatives in the LCD display circuit, A and VDD are connected to positive and RW is connected to negative. The remaining LCD pins, such as D7, D6, D5, D4, E, RS are connected to Arduino's D2, D3, D4, D5, D6, D7.

10. Attained Results

- Start point when code is uploaded

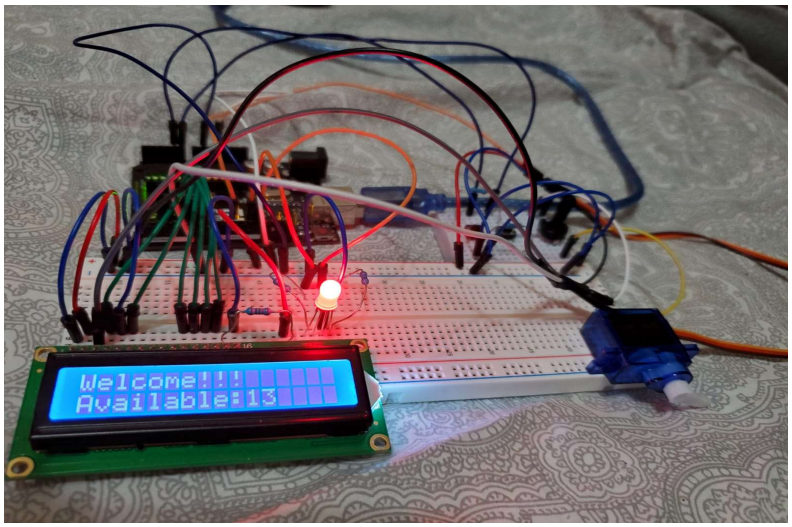

Figure 12. Initially Arm is Closed

- Arm is open when car enter

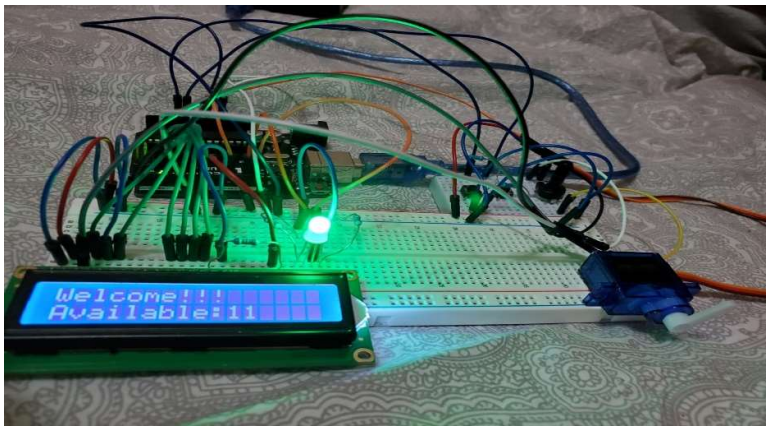

Figure 13. When Car is Entering in Slot

❖ All slots are full

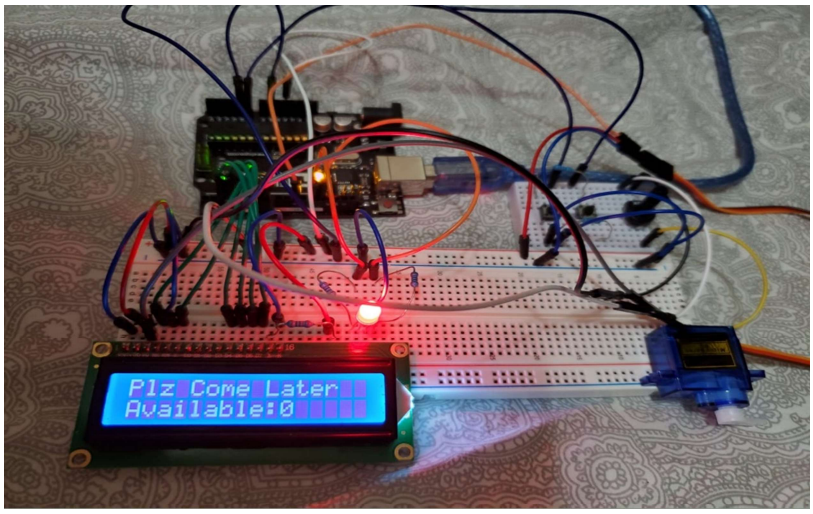

Figure 14. 13 Cars Parked

11. MATLAB Code for Smart Parking System

```
% smart parking system for 13 slot
% entry and exit button in main logic here (A4 and A5 pin of arduino)
% if A4 or A5 is high then red and green led logic apply with arm open and close position
%lcd display the message bbased on logic
arduino_uno = arduino('com3','UNO','Libraries',{'ExampleLCD/LCDAddon','Servo'});
% arduino libraries
Parking_Spot = 13; % as per requirement parking spot
Entry = 'A4'; % first logic for required code
Exit = 'A5';
Red_LED = 'D13'; % led sign for entry and exit
Green_LED = 'D12';
Control_Logic = 'D9'; % sevro motor control logic
Angle_Close = 0; % close arm
Angle_Open = 0.5; % open arm
delay = 2;
Servo_Position = servo (arduino_uno, Control_Logic); % get close for waiting of entry or
exit
writePosition (Servo_Position, Angle_Close);
lcd =
addon(arduino_uno,'ExampleLCD/LCDAddon','RegisterSelectPin','D7','EnablePin','D6','
DataPins',{'D5','D4','D3','D2'}); %lcd declaration
initializeLCD (lcd, 'Rows', 2, 'Columns', 16); % use both rows for print
clearLCD(lcd);
configurePin (arduino_uno, Entry, 'Pullup'); % pins required
configurePin (arduino_uno, Exit, 'Pullup');
configurePin (arduino_uno, Red_LED, 'DigitalOutput');
```

```
configurePin (arduino_uno, Green_LED, 'DigitalOutput');
while 1 % create loop
 Car_Entry = readDigitalPin (arduino_uno, Entry); % entry logic point
 Car_Exit = readDigitalPin (arduino_uno, Exit); % exit logic point

if (Car_Entry == 0 && Parking_Spot > 0)
        writeDigitalPin (arduino_uno, Red_LED,0);
        writeDigitalPin (arduino_uno, Green_LED,1); % green led on
        writePosition (Servo_Position, Angle_Open); % arm open
        pause(delay);
        writePosition (Servo_Position, Angle_Close); % arm close
        Parking_Spot = Parking_Spot - 1; % parking slot decrease
        writeDigitalPin (arduino_uno, Green_LED,0);
        writeDigitalPin (arduino_uno, Red_LED,1); % red led on
    elseif ((Car_Exit == 0) && (Parking_Spot < 13))
        writeDigitalPin(arduino_uno, Red_LED,0);
        writeDigitalPin(arduino_uno, Green_LED,1); % green led
        writePosition(Servo_Position, Angle_Open); % arm open
        pause(delay);
        writePosition(Servo_Position, Angle_Close); % arm close
        Parking_Spot = Parking_Spot + 1; % parking slot increase
        writeDigitalPin(arduino_uno, Green_LED,0);
        writeDigitalPin(arduino_uno, Red_LED,1); % red led on
    end

    if(Parking_Spot > 0) % any slot is free
        printLCD(lcd,'Welcome!!!');
        printLCD(lcd,['Available:',num2str(Parking_Spot)]); % how many slot free
        writeDigitalPin (arduino_uno, Red_LED,1);
    elseif(Parking_Spot == 0) % slot full
        printLCD(lcd,'Plz Come Later');
        printLCD(lcd,['Available:',num2str(Parking_Spot)]); % how many free (all)
        writeDigitalPin(arduino_uno, Red_LED,1);
    end
end
```

12. Advantages and Disadvantages

❖ **Advantages**

1. This smart parking system greatly reduces the time taken to find parking spots by driver, and the gas consumption in vehicles is reduced.
2. It is an automated system, so in this scenario, it does not require any manpower and is an added advantage.
3. Traffic caused by vehicles finding vehicles to park is reduced and the pollution is reduced as a result.

❖ **Disadvantages**

1. In this automated system, if any breakdown occurs, then the entire system will stop working.
2. The power consumption is high for this smart parking management system to operate.
3. Maintenance and operating costs to run this system are high.

13. Conclusion

Thus, we have successfully developed an automated model of the parking system with the help of Arduino and MATLAB software to overcome parking related issues in metropolitan cities around the world. In this project we designed this model if the full capacity is 13 parking spots and we tested for different conditions, and it worked perfectly. Through this project we have been able to overcome the demerits that were there in the conventional system.

14. References

[1] "Arduino UNO rev3", [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3

[2] "Specification of LCD module", 2008. [online]. Available: https://components101.com/16x2-lcd-pinout-datasheet

[3] Simon Monk, "Arduino Lesson 14. Servo Motors", 2018. [Online]. Available: https://learn.adafruit.com/adafruit-arduino-lesson-14-servo-motors

[4] "5mm RGB LED common cathode", [Online]. Available: https://www.arabsmakers.com/wp-content/uploads/2017/05/upload-5mm_RGB_led_common_cathode.pdf

[5] "Potentiometer 10k", [Online]. Available: https://components101.com/potentiometer

[6] "Push Button Switch", [Online]. Available: https://components101.com/switches/push-button