



University
of Windsor

**Computational Methods and Modelling for Engineering Applications
(GENG8030)**

Final Project Report

Smart Car Parking System using Matlab

By

Section: 4

Group Number:13

Team Members:

Sindhu Leela Sishitha Pilli (110060987)

Patel Vidhiben Rajendrakumar (110060676)

Cherub Singh Randhawa (110060793)

Fall 2021

Table of Contents

	List of Figures.....	3
1.	Introduction	4
2.	Diffrence between current and smart parking system	4
3.	Objective.....	5
4.	Project Desccription	6
	4.1 Flowchart of the project workflow	6
	4.2 Process Flow	7
5.	List of components	8
	5.1 Arduino Uno board.....	9
	5.2 Servo motor.....	10
	5.3 Pushbutton Switch.....	10
	5.4 LEDs.....	11
	5.5 Jumper Wires.....	12
	5.6 Breadboard.....	12
	5.7 Resistors.....	13
6.	Code for smart car parking App	14
7.	Conclusion.....	29
8.	References	30

2. LIST OF FIGURES

Figure-1: Overview of Parking Detection Process.	4
Figure-2: Smart parking app.....	5
Figure-3: Flowchart of parking system when entry button is pressed.....	6
Figure-4: Flowchart when exit button is pressed.....	7
Figure-5: ELEGOO UNO Project Super Starter Kit with Tutorial and UNO R3 Compatible with Arduino IDE.	8
Figure-6:. Arduino Uno pin diagram	9
Figure-7: Servo Motor.	10
Figure-8: Pushbutton switch.	11
Figure-9: LEDs.	11
Figure-10: Jumper wires.....	12
Figure-11: Breadboard.....	13
Figure-12: Resistors.....	13

1. Introduction

Getting a parking space in urban areas has become a hectic task due to the increase in the number of vehicles. Whenever a vehicle enters a parking lot driver always has to look for a parking spot which leads to wastage of time and resources. There is research which shows that 40% of traffic is generated due to parking search which also further contributes to extra pollution [1].

In this project, a smart parking management app is developed using the computational skills of MATLAB. MATLAB is a carefully designed software for the analysis, programming, and computation of engineering systems. The system overview is shown in Fig.1.

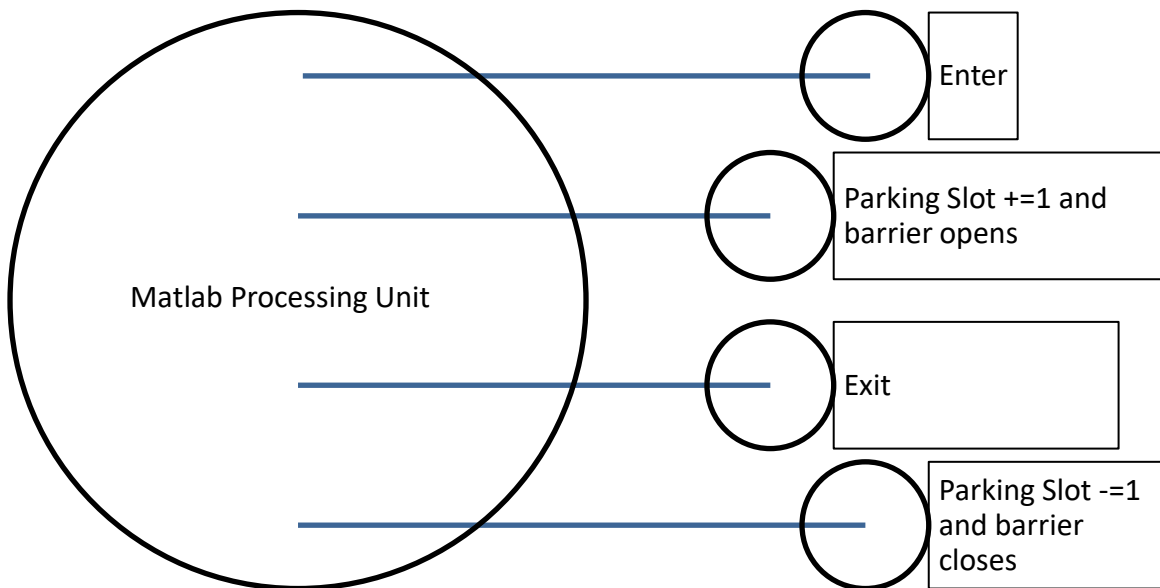


Figure-1: Overview of Parking Detection Process.

2. Differences between current and smart parking system

- The current parking system is less efficient as it relies on humans to do the job, whereas software-designed parking systems are easy to use.

- A smart parking system is comparatively faster than the other systems and it saves fuel consumption too.
- Smart car parking applications can improve the user experience with real-time parking availability information.
- Current parking systems are less secure compared to smart car parking systems.

3. Objective:

With progressing urbanization and increasing availability of the cars and ride hailing services, inefficient parking has become a huge burden for drivers [2]. On parking operator side there is an even bigger scope for transformation [2]. Hence, the main objective of designing smart car parking system is to improve the motorists' experience.

- Provide real-time parking availability information.
- Signal using the light control indicators; accordingly, either red or green
- Easy to use application
- Automate the system of manual searching of parking spots
- Minimize the time taken for locating the parking space

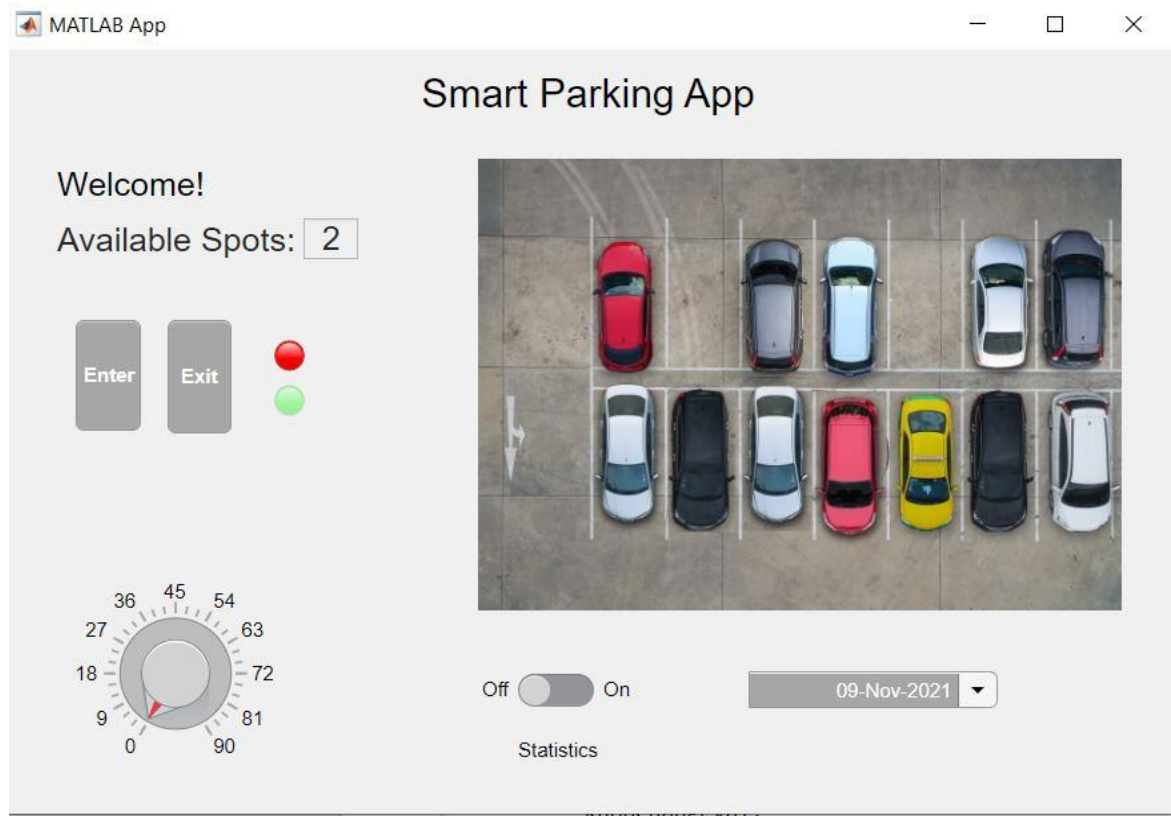


Figure-2: Smart parking app.

4. Project Description:

In this project, we are using MATLAB and its app designer tool to create an app for the smart car parking system. When the car approaches the gate, the app will display a “Welcome!” message, and the available slot is displayed on the screen. If the slot is available, the driver can press the entry button and the green light blinks simultaneously with the barrier getting opened.

The car is allowed to enter the parking lot and the available slots are decreased by 1 with each entry and is displayed in the app. Similarly, the car existing from the parking lot by pressing the exit button and available slots increased by 1.

Whereas, if there is no parking space available, the system will not allow an additional car to enter the parking area and show the number of slots available as zero. Even if the driver tries to press the entry button, the gate will not open.

4.1 Flowchart of the project workflow

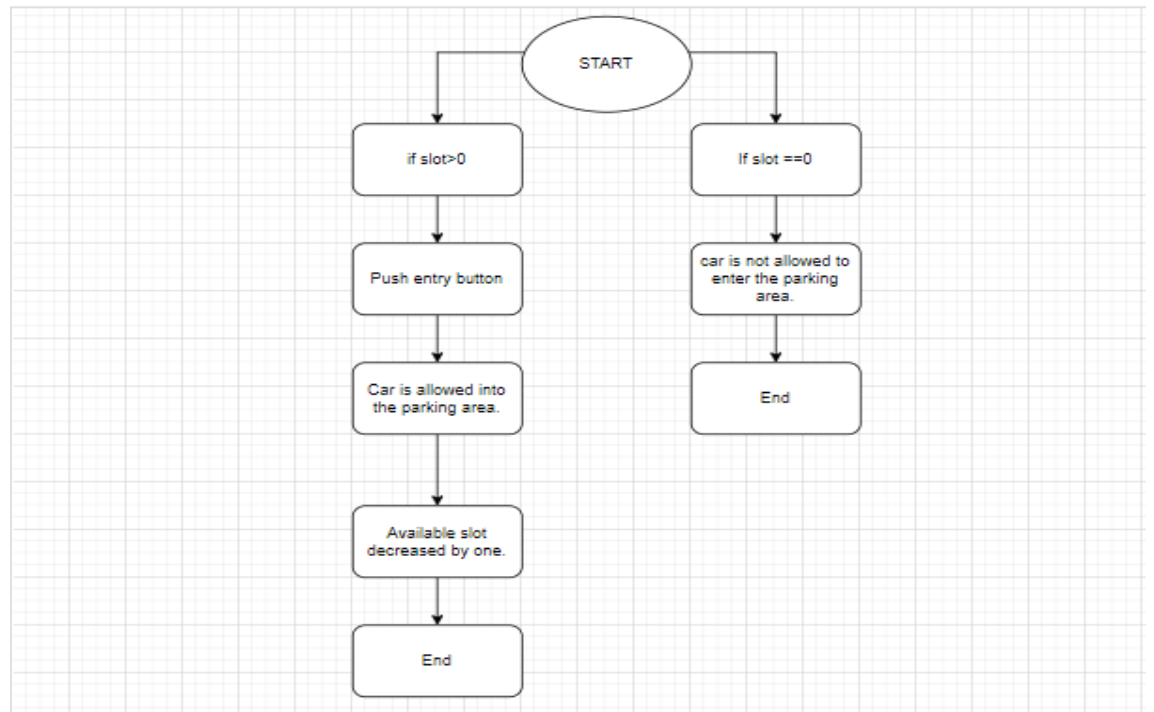


Figure-3: Flowchart of parking system when entry button is pressed

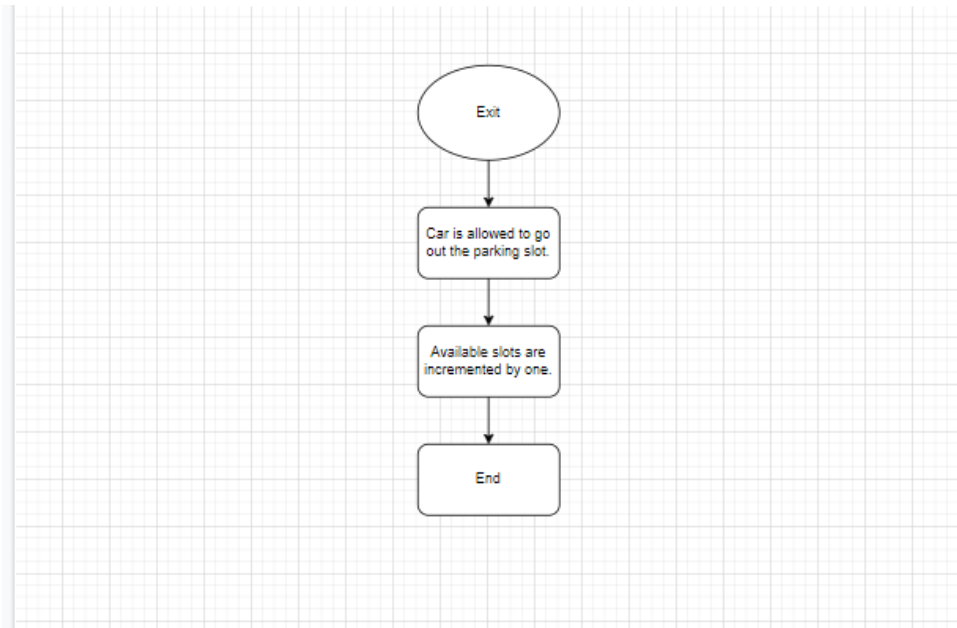


Figure-4: Flowchart when exit button is pressed.

4.2 Process Flow:

Initially, the Arduino and the necessary hardware components are integrated together. By installing the “MATLAB Support Package for Arduino”, the hardware set is connected through a USB Port to the monitoring device. Required code is written for individual components in the hardware setup and run accordingly. The ‘Home page’ of MATLAB contains an option to shift to the app designing tool. Once on the new page, components from library are used to design an aesthetic user interface.

The drag and drop of components create the required code in “Code view” and upon further review it is reconstructed.

The integration with IOT is done through Thing Speak. ThingSpeak is an analytic IoT platform service that allows one to aggregate, visualize and analyze live data streams in the cloud [3].

5. List of Components:

1. Arduino Uno board
2. Servo motor
3. Pushbutton Switch
4. LEDs
5. Jumper Wires
6. Breadboard
7. Resistors

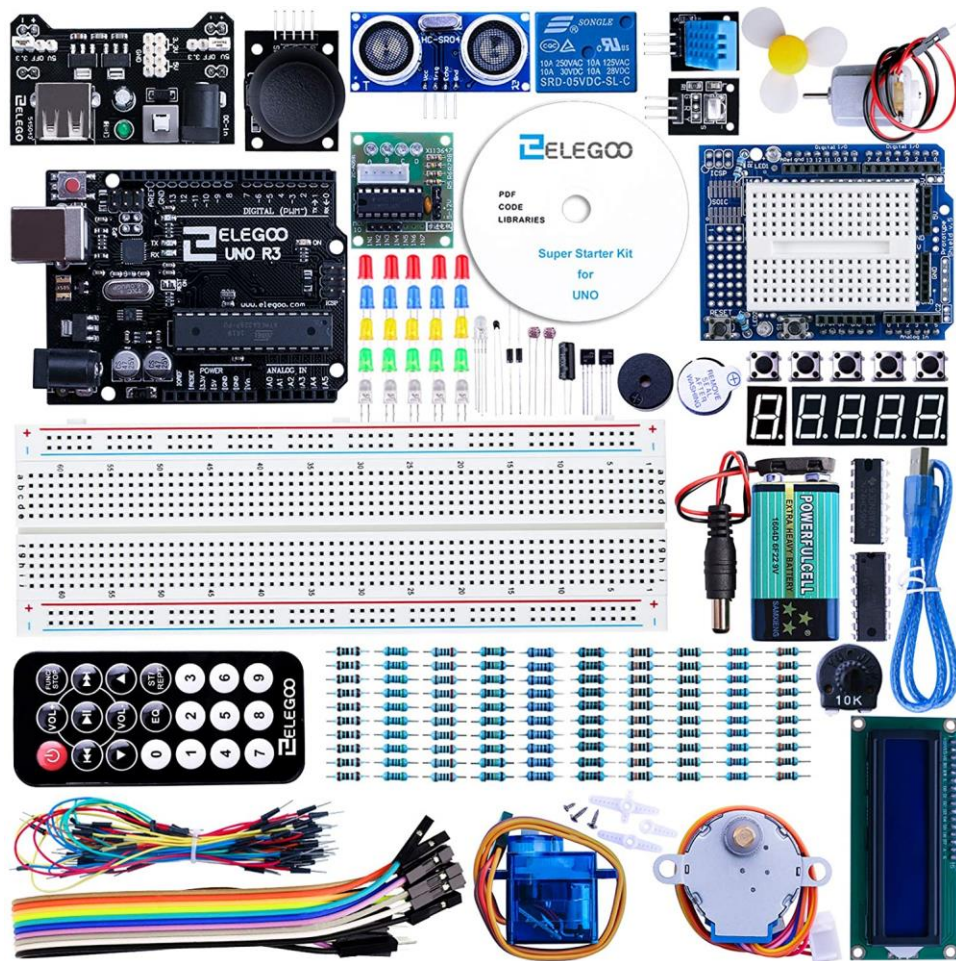


Figure-5: ELEGOO UNO Project Super Starter Kit with Tutorial and UNO R3 Compatible with Arduino IDE [4]

5.1 Arduino Uno

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online [5]. The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well [5]. The Arduino UNO is the best board to get started with electronics and coding [6]. Arduino Uno is a microcontroller board based on the ATmega328P (datasheet) [6]. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button [6]. The UNO is the most used and documented board of the whole Arduino family [6].

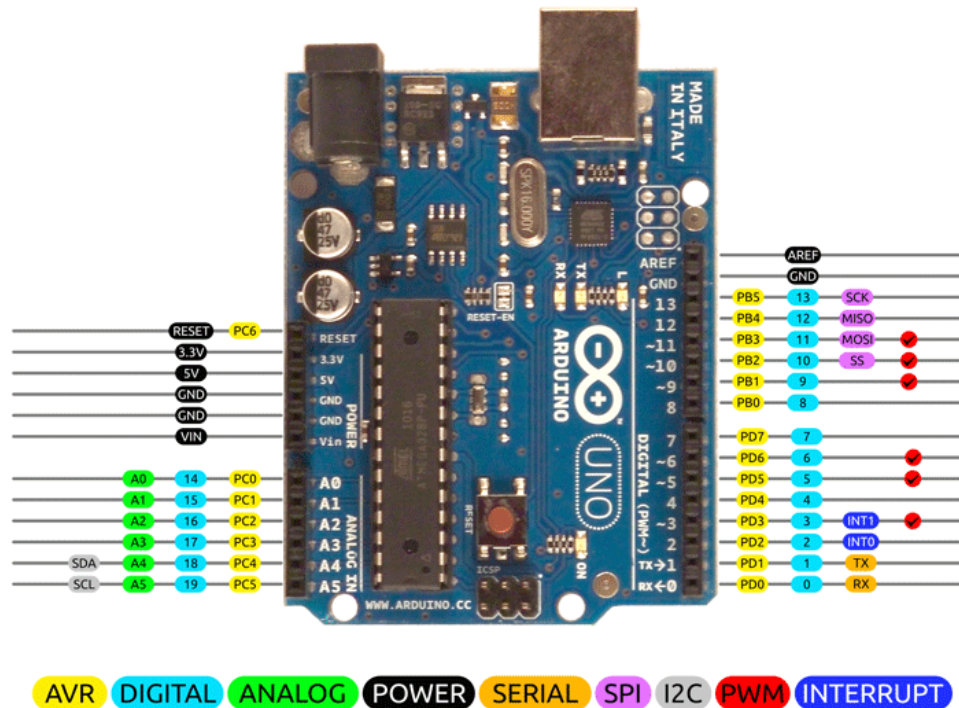


Figure-6: Arduino Uno pin diagram [7].

5.2 Servo Motor

A Servo Motor is a small device that has an output shaft [9]. This shaft can be positioned to specific angular positions by sending the servo a coded signal [9]. The servo will maintain the angular position of the shaft as long as the coded signal exists on the input line. If the coded signal changes, the angular position of the shaft will also change. [9]. Servo motors have three wires: power, ground, and signal [10]. The power wire is typically red and should be connected to the 5V pin on the Arduino board [10]. The ground wire is typically black or brown and should be connected to a ground pin on the board [10]. The signal pin is typically yellow, orange or white and should be connected to pin 9 on the board [10].

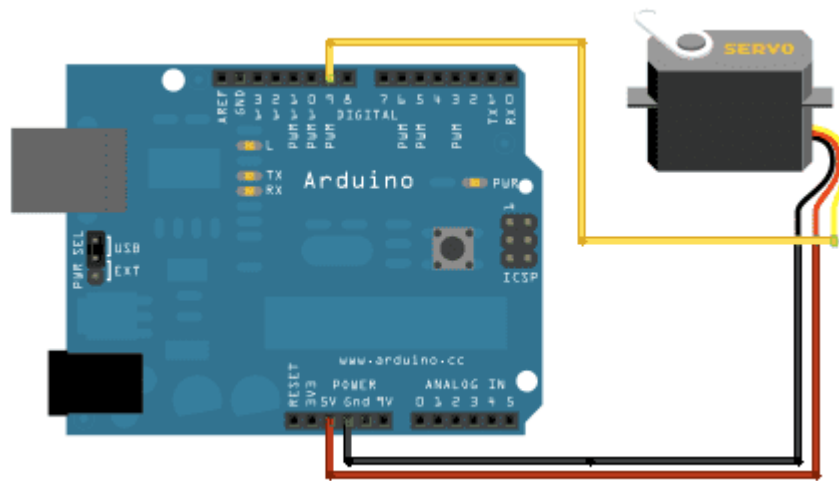


Figure-7: Servo motor and its pin connection [10].

5.3 Pushbutton

The pushbutton switch has thousands of familiar uses, from elevators to car stereos [11]. The push-button switch is a small, sealed device that completes an electric circuit when pressed [11]. Electricity flows through it when two wires contact a small metal spring inside. When it's off the spring retracts, contact is interrupted, and the current will not flow [11]. The body of the switch is made of non-conducting plastic [11].



Figure-8: Pushbutton switch [12]

5.4 LEDs

LEDs come in a variety of shapes, sizes, and color; this gives designers the ability to tailor them to their product [13]. Their low power and small sizes make them an excellent choice for many distinct products as they can be worked into the design more seamlessly to make it an overall better device [13]. It is used as an indicator in most electronic circuits. LEDs can easily be powered from a small battery source and last a long time [13]. It has two legs; longer leg is anode (+) and shorter is cathode (-).



Figure-9: LEDs [13]

5.5 Jumper Wires

An electrical wire (also known as a jumper wire) is an electrical cable that has a connector or pin at each end, which is typically used to connect the components of a

breadboard or other prototyping or test circuit, either internally or with other equipment or components, without soldering [14]. Jump wires are connected by inserting the end connectors into the slots on a breadboard or the header connector on a circuit board [14].

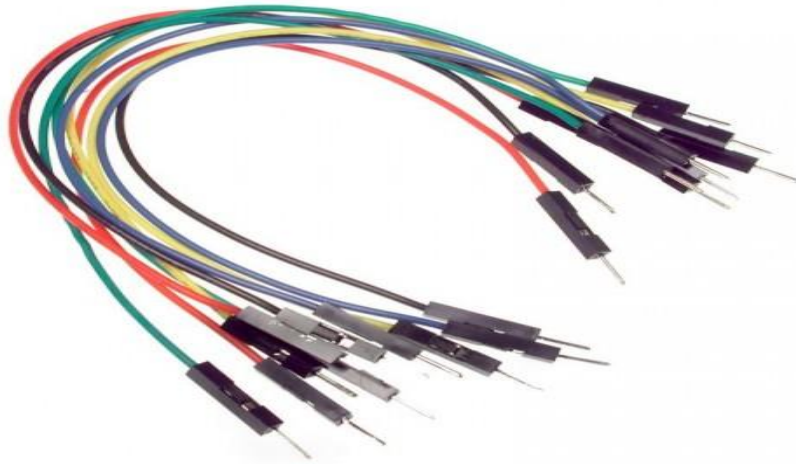


Figure-10: Jumper wires [15]

5.6 Breadboard

A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted [16]. The breadboard has strips of metal that run underneath the board and connect the holes on the top of the board [16]. The long top and bottom row of holes are usually used for power supply connections [16]. The rest of the circuit is built by placing components and connecting them with jumper wires [16]. Breadboard is specifically used for extension of positive and ground terminals.

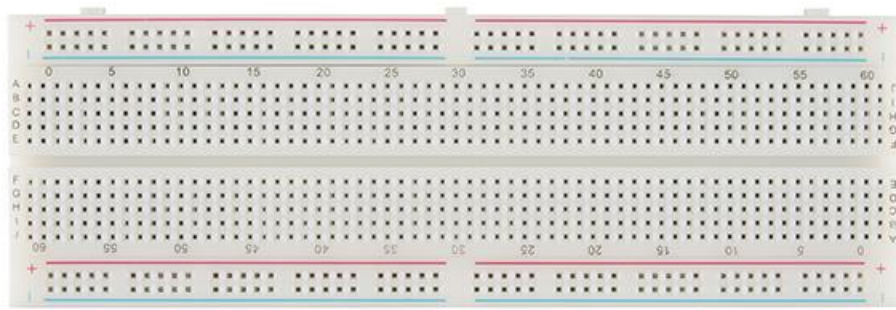


Figure-11: Breadboard [17]

5.7 Resistors

Resistors come in varieties of shapes and sizes. The resistor resistance limits the flow of electrons through a circuit [18]. They are passive components, meaning they only consume power and cannot generate it [18]. Resistors are usually added to circuits, where they complement active components like microcontrollers, and other integrated circuits. Commonly, resistors are used to limit current, divide voltages, and pull up I/O lines [18].



Figure-12: Resistors [17]

6. Code for smart car parking App

```
classdef SmartParkingApp < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
    UIFigure          matlab.ui.Figure
```

```
    EditFiled_AvailableSpots    matlab.ui.control.NumericEditField
```

```
    AvailableSpotsEditFieldLabel    matlab.ui.control.Label
```

```
    Button_Exit          matlab.ui.control.Button
```

```
    Button_Enter          matlab.ui.control.Button
```

```
    DatePicker            matlab.ui.control.DatePicker
```

```
    Label_Welcome          matlab.ui.control.Label
```

```
    Knob                   matlab.ui.control.Knob
```

```
    Lamp_Green             matlab.ui.control.Lamp
```

```
    Lamp_Red               matlab.ui.control.Lamp
```

```
    Switch_Statistics      matlab.ui.control.Switch
```

```
    StatisticsSwitchLabel  matlab.ui.control.Label
```

```
    Image                   matlab.ui.control.Image
```

```
    LabelTitle             matlab.ui.control.Label
```

```
end
```

```

        properties (Access = private)
            ardUino;
        serVo;
        Capacity = 14;
        AvailableSpots;
        timeToHold=15;
        onoffstats;
        APIKEY = "; %API Key TBD
        channelid = 123; %Channel ID TBD
        knobChangeVal;
    end

    methods (Access = private)
        function checkStatus(app)
            if(app.AvailableSpots == app.Capacity)
                app.Button_Exit.Enable = 'off';
            elseif(app.AvailableSpots==0)
                app.Button_Enter.Enable='off';
                app.Label_Welcome.Text='PARKING FULL..';
            else
                app.Label_Welcome.Text='Welcome!';
            end
        end
    end

```


end

```
function changeimage(app, carpos)
switch app.Capacity-carpos
case 0
app.Image.ImageSource = 'p0.jpg';
case 1
app.Image.ImageSource = 'p1.jpg';
case 2
app.Image.ImageSource = 'p2.jpg';
case 3
app.Image.ImageSource = 'p3.jpg';
case 4
app.Image.ImageSource = 'p4.jpg';
case 5
app.Image.ImageSource = 'p5.jpg';
case 6
app.Image.ImageSource = 'p6.jpg';
case 7
app.Image.ImageSource = 'p7.jpg';
case 8
app.Image.ImageSource = 'p8.jpg';
```

case 9

app.Image.ImageSource = 'p9.jpg';

case 10

app.Image.ImageSource = 'p10.jpg';

case 11

app.Image.ImageSource = 'p11.jpg';

case 12

app.Image.ImageSource = 'p12.jpg';

case 13

app.Image.ImageSource = 'p13.jpg';

case 14

app.Image.ImageSource = 'p14.jpg';

end

end

function logdatatothingspeak(app,status)

if (strcmp(app.onoffstats,'On'))

tstamps = datetime('now');

thingSpeakWrite(app.channelid,[app.AvailableSpots,status],"Timestamp",tstamps,'writekey',app.APIKEY);

end

end

end

```

% Callbacks that handle component events

methods (Access = private)


% Code that executes after component creation

function startupFcn(app)

    clear app.arduino app.serveo;

app.arduino = arduino('COM9','Uno','Libraries','Servo');
app.serveo = servo(app.arduino,'D10');
writePosition(app.serveo,0);
writeDigitalPin(app.arduino,'D3',1);
app.Lamp_Red.Color = 'r';
app.Lamp_Green.Color = 'g';

    app.Lamp_Green.Enable = 'off';

app.AvailableSpots=0;
app.Label_Welcome.Text='Welcome!';
app.onoffstats='Off';

app.EditFiled_AvailableSpots.Value = app.AvailableSpots;
app.EditFiled_AvailableSpots.Editable = 'off';
app.Switch_Statistics.Value = app.onoffstats;

```

```

        changeimage(app,app.AvailableSpots);
        app.checkStatus();
    end

% Button pushed function: Button_Enter
function Button_EnterPressed(app, event)

    disp("Enter btn pressed");

    if(app.AvailableSpots>0)
app.Lamp_Red.Enable = 'off';
app.Button_Enter.Enable='off';
app.Button_Exit.Enable='off';
writePosition(app.serVo,0.5);
writeDigitalPin(app.ardUino,'D3',0);
time=0;
while time < app.timeToHold
app.Lamp_Green.Enable = 'on';

                                writeDigitalPin(app.ardUino,'D4',0);

changeimage(app,app.AvailableSpots)
pause(0.5)
app.Lamp_Green.Enable = 'off';

                                writeDigitalPin(app.ardUino,'D4',1);

changeimage(app,app.AvailableSpots-1)

```

```

pause(0.5)
time = time+1;
end
app.Lamp_Red.Enable = 'on';
app.Button_Enter.Enable='on';
app.Button_Exit.Enable='on';
app.AvailableSpots = app.AvailableSpots-1;
app.EditFiled_AvailableSpots.Value=app.AvailableSpots;
%
% writePosition(app.serVo,0);
%
% writeDigitalPin(app.ardUino,'D3',1);
%
% writeDigitalPin(app.ardUino,'D4',0);
checkStatus(app);
logdatatothingspeak(app,1);
end
end

% Button pushed function: Button_Exit
function Button_ExitPressed(app, event)
    disp("Exit btn pressed");
    if(app.AvailableSpots<app.Capacity)
app.Lamp_Red.Enable = 'off';
app.Button_Enter.Enable='off';

```

```

app.Button_Exit.Enable='off';
writePosition(app.serVo,0.5);
writeDigitalPin(app.ardUino,'D3',0);
time=0;
while time< app.timeToHold
app.Lamp_Green.Enable = 'on';
writeDigitalPin(app.ardUino,'D4',0);
changeimage(app,app.AvailableSpots)
pause(0.5)
changeimage(app,app.AvailableSpots+1)
app.Lamp_Green.Enable = 'off';
writeDigitalPin(app.ardUino,'D4',1);
pause(0.5)
time = time+1;
end
app.Lamp_Red.Enable = 'on';
app.Button_Enter.Enable='on';
app.Button_Exit.Enable='on';
app.AvailableSpots = app.AvailableSpots+1;
app.EditFiled_AvailableSpots.Value=app.AvailableSpots;
writePosition(app.serVo,0);
writeDigitalPin(app.ardUino,'D3',1);
writeDigitalPin(app.ardUino,'D4',0);

```

```

checkStatus(app);
logdatatothingspeak(app,-1);
end

end

% Value changed function: Knob
function KnobValueChanged(app, event)
    value = app.Knob.Value;
app.knobChangeVal=value/180;
%             writePosition(app.serVo,app.knobChangeVal);
    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created

```

```
app.UIFigure = uifigure('Visible', 'off');  
app.UIFigure.Color = [0.9412 0.9412 0.9412];  
app.UIFigure.Position = [100 100 702 463];  
app.UIFigure.Name = 'MATLAB App';
```

```
% Create LabelTitle
```

```
app.LabelTitle = uilabel(app.UIFigure);  
app.LabelTitle.HorizontalAlignment = 'center';  
app.LabelTitle.FontSize = 24;  
app.LabelTitle.Position = [1 419 702 39];  
app.LabelTitle.Text = 'Smart Parking App';
```

```
% Create Image
```

```
app.Image = uiimage(app.UIFigure);  
app.Image.Position = [264 125 430 274];
```

```
% Create StatisticsSwitchLabel
```

```
app.StatisticsSwitchLabel = uilabel(app.UIFigure);  
app.StatisticsSwitchLabel.HorizontalAlignment = 'center';  
app.StatisticsSwitchLabel.Position = [307 30 54 22];
```



```

app.StatisticsSwitchLabel.Text = 'Statistics';

% Create Switch_Statistics
app.Switch_Statistics = uiswitch(app.UIFigure, 'slider');
app.Switch_Statistics.Position = [310 67 45 20];

% Create Lamp_Red
app.Lamp_Red = uilamp(app.UIFigure);
app.Lamp_Red.Position = [163 270 19 19];

% Create Lamp_Green
app.Lamp_Green = uilamp(app.UIFigure);
app.Lamp_Green.Position = [163 243 19 19];

% Create Knob
app.Knob = uiknob(app.UIFigure, 'continuous');
app.Knob.Limits = [0 90];
app.Knob.ValueChangedFcn = createCallbackFcn(app,
@KnobValueChanged, true);
app.Knob.Position = [70 53 68 68];

```

```

% Create Label_Welcome

app.Label_Welcome = uilabel(app.UIFigure);
app.Label_Welcome.FontSize = 20;
app.Label_Welcome.Position = [33 368 211 31];
app.Label_Welcome.Text = 'Welcome!';


% Create DatePicker

app.DatePicker = uidepicker(app.UIFigure);
app.DatePicker.FontColor = [1 1 1];
app.DatePicker.BackgroundColor = [0.651 0.651 0.651];
app.DatePicker.Position = [448 66 150 22];
app.DatePicker.Value = datetime([2021 11 9]);


% Create Button_Enter

app.Button_Enter = uibutton(app.UIFigure, 'push');

app.Button_Enter.ButtonPushedFcn = createCallbackFcn(app,
@Button_EnterPressed, true);

app.Button_Enter.BackgroundColor = [0.651 0.651 0.651];
app.Button_Enter.FontWeight = 'bold';
app.Button_Enter.FontColor = [1 1 1];

```

```

app.Button_Enter.Position = [44 234 39 67];
app.Button_Enter.Text = 'Enter';

% Create Button_Exit

app.Button_Exit = uibutton(app.UIFigure, 'push');

app.Button_Exit.ButtonPushedFcn = createCallbackFcn(app,
@Button_ExitPressed, true);

app.Button_Exit.BackgroundColor = [0.651 0.651 0.651];
app.Button_Exit.FontWeight = 'bold';
app.Button_Exit.FontColor = [1 1 1];
app.Button_Exit.Position = [99 232 39 69];
app.Button_Exit.Text = 'Exit';

% Create AvailableSpotsEditFieldLabel

app.AvailableSpotsEditFieldLabel = uilabel(app.UIFigure);

app.AvailableSpotsEditFieldLabel.BackgroundColor = [0.9412
0.9412 0.9412];

app.AvailableSpotsEditFieldLabel.FontSize = 20;
app.AvailableSpotsEditFieldLabel.FontColor = [0.149 0.149 0.149];
app.AvailableSpotsEditFieldLabel.Position = [33 338 154 25];
app.AvailableSpotsEditFieldLabel.Text = 'Available Spots: ';

```

```

% Create EditFiled_AvailableSpots
app.EditFiled_AvailableSpots = uieditfield(app.UIFigure, 'numeric');
app.EditFiled_AvailableSpots.Limits = [0 14];
app.EditFiled_AvailableSpots.ValueDisplayFormat = '%.0f';
app.EditFiled_AvailableSpots.HorizontalAlignment = 'center';
app.EditFiled_AvailableSpots.FontSize = 20;
app.EditFiled_AvailableSpots.FontColor = [0.149 0.149 0.149];
app.EditFiled_AvailableSpots.BackgroundColor = [0.9412 0.9412
0.9412];
app.EditFiled_AvailableSpots.Position = [181 338 33 25];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app

```

```

function app = SmartParkingApp

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)

if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

```

```
        % Delete UIFigure when app is deleted  
        delete(app.UIFigure)  
    end  
end  
end
```

7. Conclusion.

If this project replaces the current parking system, many current problems will be minimized. It cuts the waiting time to find empty parking spaces and reduces pollution. It can overcome the disadvantages of the current parking system and help make the most of the parking space.

8. References

- [1] D. Shoup, "Is 30 Percent of Traffic Cruising for Parking?," *parkingtoday*, Feb. 2019, [Online]. Available: <https://www.parkingtoday.com/articledetails.php?id=2624&t=is-30-percent-of-traffic-cruising-for-parking> [Accessed: Nov. 21, 2021]
- [2] "Advantages and Disadvantages of Smart Parking Sensors," *Nwave*, Dec. 10, 2019. [Online]. Available: <https://www.nwave.io/pros-and-cons-of-smart-parking-systems/#:~:text=With%20progressing%20urbanisation%20and%20increasing,causes%20much%20frustration%20and%20stress> [Accessed: Nov. 21, 2021]
- [3] "Learn More - ThingSpeak IoT.," *Thingspeak*, n.d. [Online]. Available: https://thingspeak.com/pages/learn_more [Accessed: Nov. 21, 2021]
- [4] "ELEGOO UNO Project Super Starter Kit with Tutorial and UNO R3 Compatible with Arduino IDE," *amazon*, n.d. Available: <https://www.amazon.com/ELEGOO-Project-Tutorial-Controller-Projects/dp/B01D8KOZF4> [Accessed: Nov. 20, 2021]
- [5] "What is Arduino?," *Arduino*, 2018.. Available: <https://www.arduino.cc/en/Guide/Introduction> [Accessed: Nov. 20, 2021]
- [6] "Overview," *Arduino*, n.d. . Available: <https://store.arduino.cc/products/arduino-uno-rev3> [Accessed: Nov. 20, 2021]

- [7]“Arduino Uno,” *components101*, 2021. Available:
<https://components101.com/microcontrollers/arduino-uno> [Accessed: Nov. 20,2021]
- [8]“Arduino - Servo Motor,” *tutorialspoint*, 2018. Available:
https://www.tutorialspoint.com/arduino/arduino_servo_motor.htm [Accessed: Nov. 21,2021]
- [9] “Sweep,” *Arduino*, n.d. Available:
<https://www.arduino.cc/en/Tutorial/LibraryExamples/Sweep> [Accessed: Nov. 21,2021]
- [10]]“How Do Push Button Switches Work in an Electrical Circuit?,” *sciencing*, n.d. Available: <https://sciencing.com/push-switches-work-electrical-circuit-5030234.html> [Accessed: Nov. 22,2021]
- [11]“Momentary Push Button Switch,”*moderndevise*, n.d. Available:
<https://moderndevise.com/product/momentary-push-button-switch/> [Accessed: Nov. 22,2021]
- [12] S.Taylor, “How Does a 5mm LED Work?,” *ledsupply*, 2019. Available: <
<https://www.ledsupply.com/blog/how-does-a-5mm-led-work/> [Accessed: Nov. 23,2021]
- [13]“Jump Wire,” *arduino*, 2021. Available:
<https://www.arduino.cc/en/Tutorial/LibraryExamples/Sweep> [Accessed: Nov. 23,2021]
- [14] “MALE TO MALE JUMPER WIRE,” *elementzonline* , n.d. Available:
<https://www.elementzonline.com/male-to-male-jumper-wire-132> [Accessed: Nov. 23,2021]
- [15]“What Is a Breadboard?,” *StudyLib*, n.d. Available:
<https://studylib.net/doc/18066529/what-is-a-breadboard%3F> [Accessed: Nov. 24,2021]
- [16]“How to Use a Breadboard, ” *learn.sparkfun*, n.d. Available:
<https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all> [Accessed: Nov. 24,2021]

[17]“Resistors,” *learn.sparkfun*, n.d. Available:
<https://learn.sparkfun.com/tutorials/resistors/all> [Accessed 24 Nov. 2021]