GENG 8030 - COMPUTATIONAL METHODS AND MODELING FOR ENGINEERING

(SECTION – 4)

University of Windsor

FINAL REPORT ON

SMART PARKING MANAGEMENT SYSTEM

SUBMITTED BY

PROJECT GROUP 24

STUDENT NAME	STUDENT ID
DALER SINGH RANJEETH SINGH	110040840
POOJA RANI SUNDARRAJAN	110060046
THARUNAVARSHINI MANIMARAN	110040248

TO

PROFESSOR: DR. RASHID RASHIDZADEH

ON

DATE: 11/25/2021

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

With an ever-increasing population and high number of vehicles on the road, we've begun to consider parking space. With everyone owning at least one car at home, it is difficult to find parking space in public places such as malls, supermarkets, and so on. The rise in the traffic congestion, as well as the uncertainty surrounding parking availability and payment, have emphasized the need for Smart Parking systems.

Smart parking technology will aid in the optimization of parking space usage, the efficiency of parking operations, and the smoother flow of traffic. Smart Parking solutions are being introduced to provide drivers with an all-in-one solution for their journey from start to finish, eliminating the need to search for parking, costs, travel time, and so on.

In this project, a model is developed using MATLAB and Arduino Uno to simulate the entire process of a parking space, with cars entering and exiting on time. The prototype has a great deal of potential to grow into a packed parking management system, and it can be implemented in any suitable location in the future with very few minor changes.

1. INTRODUCTION

Almost every country in the world is experiencing parking space issues because of its own advancement in automotive technology, which has increased demand for vehicles. This increase in vehicle demand has increased the need for more parking space in public places. The public has a tough time finding a parking spot and getting to their destination. This also turns out to be a fact, causing them to simply delay their desired travel schedule. The main objective of this project is to avoid all the above problems by developing a smart parking management system using MATLAB and Arduino.

Figure 1. Smart Parking System [1]

2. COMPONENT DESCRIPTION

This project's output is obtained using several components. A brief description about the component and its functionality is listed below,

2.1 ARDUINO UNO

The ATmega328P-based Arduino Uno is a microcontroller board (datasheet). It has 14 digital I/O pins (six of which can be used as PWM outputs), six analogue inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header, and a reset button. It comes with everything you need to support the microcontroller; simply connect it to a

computer via USB or power it via an AC-to-DC adapter or battery to get started. The Arduino UNO is a low-cost, versatile, and simple-to-use programmable open-source microcontroller board that can be integrated into a wide range of electronic projects. As an output, this board can control relays, LEDs, servos, and motors and can be interfaced with other Arduino boards, Arduino shields, and Raspberry Pi boards [2].

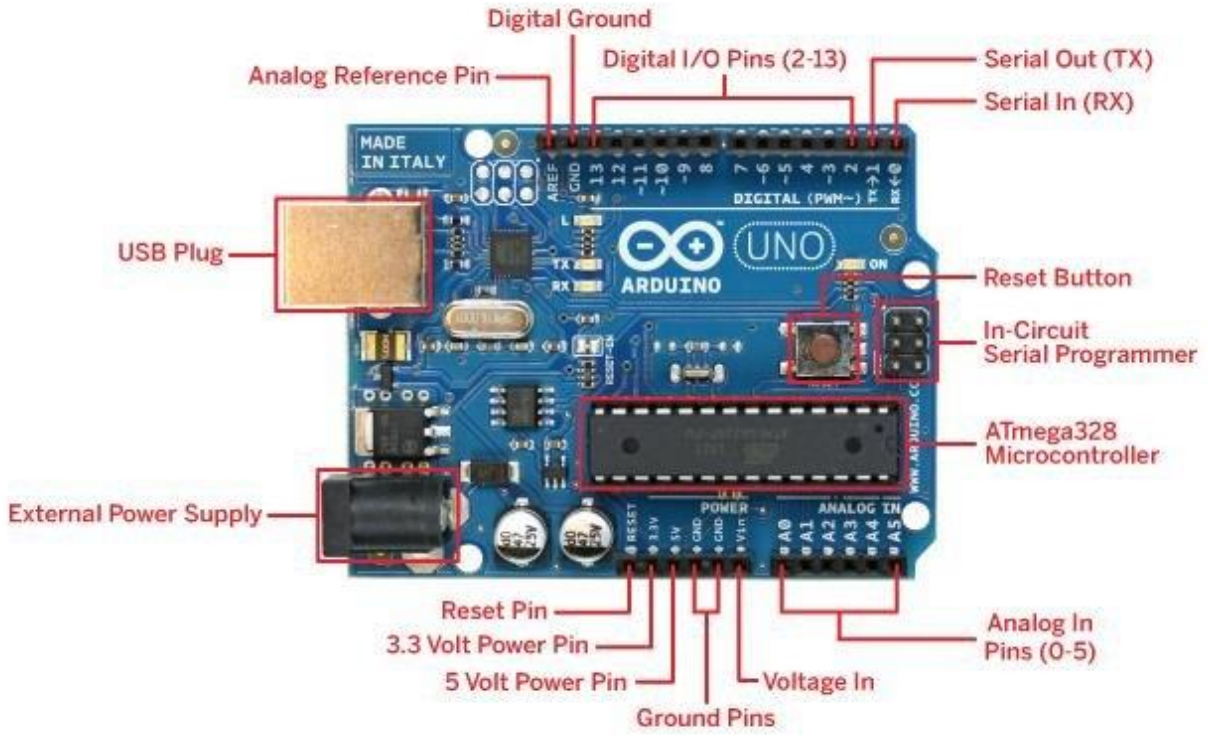

Figure 2. Arduino Uno [3]

2.2 BREADBOARD

An electronics breadboard refers to a solderless breadboard. These are excellent for making temporary circuits and prototyping because they do not require any soldering. When working with a development board, such as an Arduino, you can simply draw power from the female headers of the Arduino. The Arduino has several power and ground pins that can be connected to power rails or other rows on a breadboard [4].

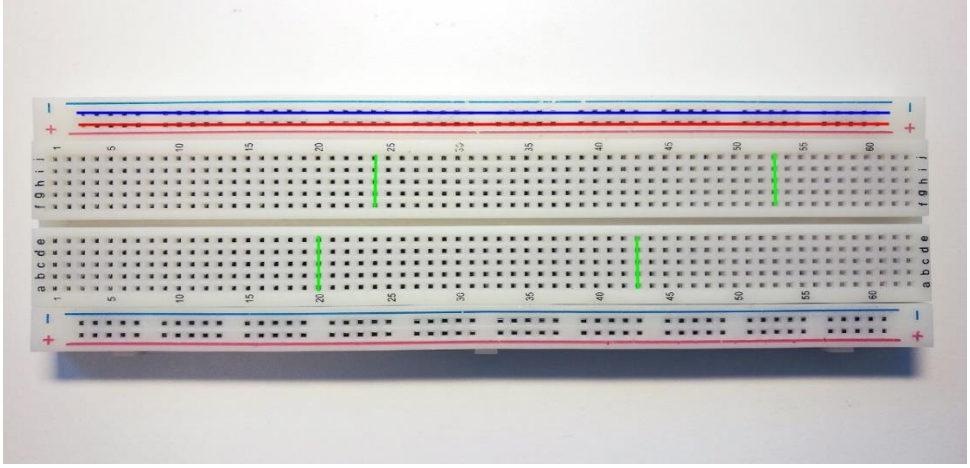

Figure 3. Breadboard [4]

2.3 JUMPER WIRE

Jumper wires are simply wires with connector pins at each end that can be used to connect two points without soldering. Jumper wires are commonly used with breadboards and other prototyping tools to allow for easy circuit changes as needed. Although jumper wires are available in a variety of colors, the colors have no meaning. However, you can use the colors to your advantage by distinguishing between different types of connections, such as ground or power.

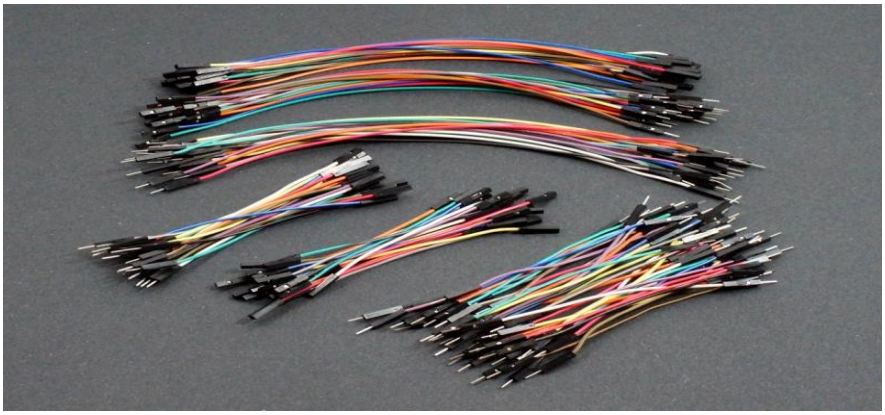

Figure 4. Jumper Wire [5]

2.4 SERVO MOTOR

A servo motor is a rotary actuator that allows for precise angular position control. It is made up of a motor and a sensor for position feedback. To complete the system, a servo drive is required. The feedback sensor is used by the drive to precisely control the rotary position of the motor [6].

Figure 5. Servo Motor [6]

2.5 BUTTONS

When you press a push button or switch, it connects two points in a circuit. When the pushbutton is open (not pressed), there is no connection between the two legs, so the pin is connected to ground (via the pull-down resistor) and we read a LOW. When the button is pressed, a connection is formed between its two legs, connecting the pin to 5 volts, resulting in a HIGH reading [7].

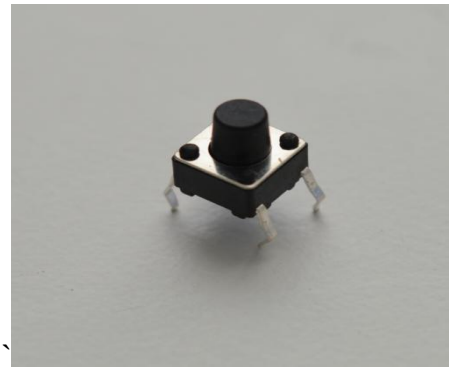

Figure 6. Buttons [7]

2.6 LED MODULE

We use Green and Red LEDs in the Smart Parking Management System. When a car tries to access the parking space, the light turns green, and the barrier opens. After the car passes, barrier is closed, and the light turns red.

Figure 7. LED Modules (Red and Green)

3. THINGSPEAK INTERFACE

Thingspeak is an IOT analytics platform that supports MATLAB to aggregate and visualize data. In this project, the MATLAB application communicates with Thingspeak platform to store the number of available parking spots on the cloud and generates a graph to visually analyze its data.

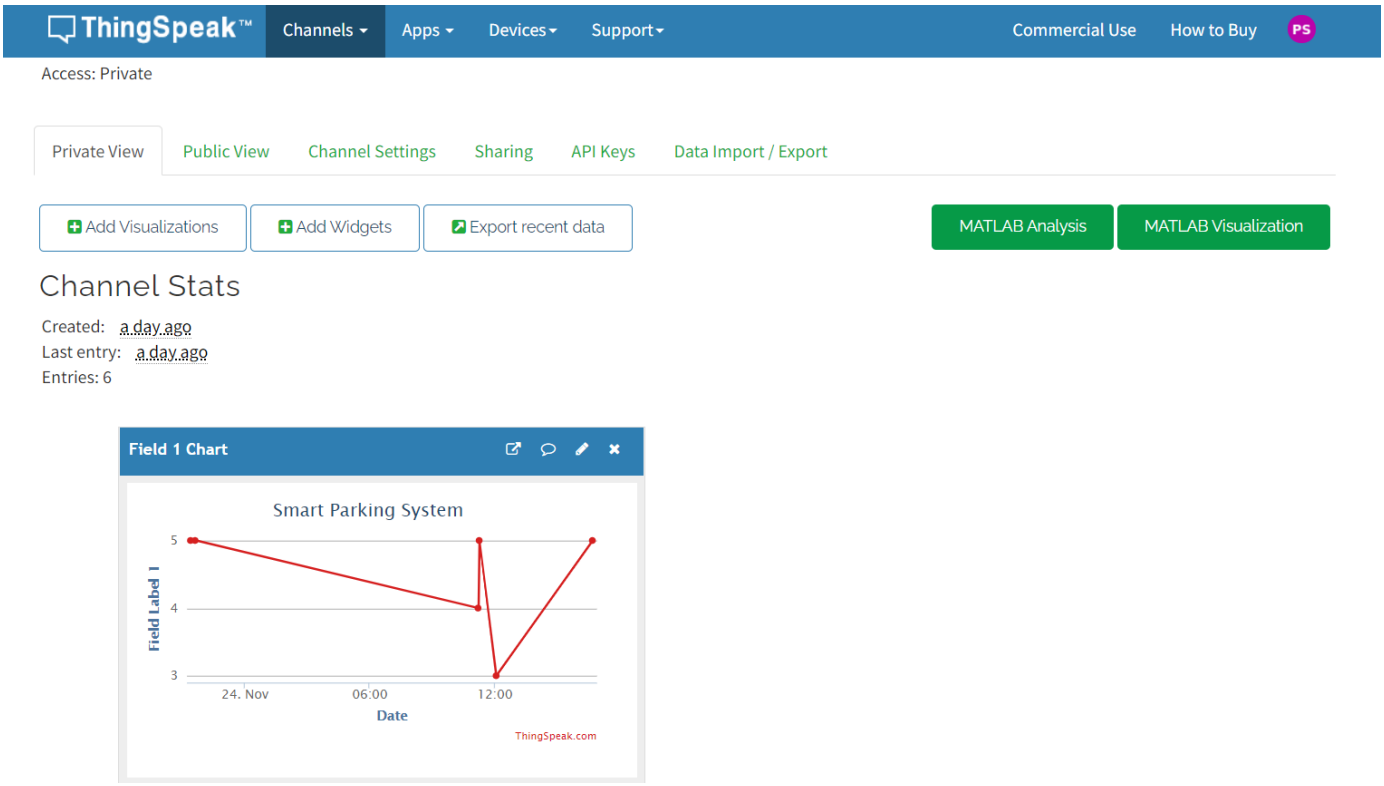

Figure 8. Cloud Connectivity with ThingSpeak Interface

4. WORKING MODEL

The flowchart below clearly details the working process of the Smart Parking Management System.

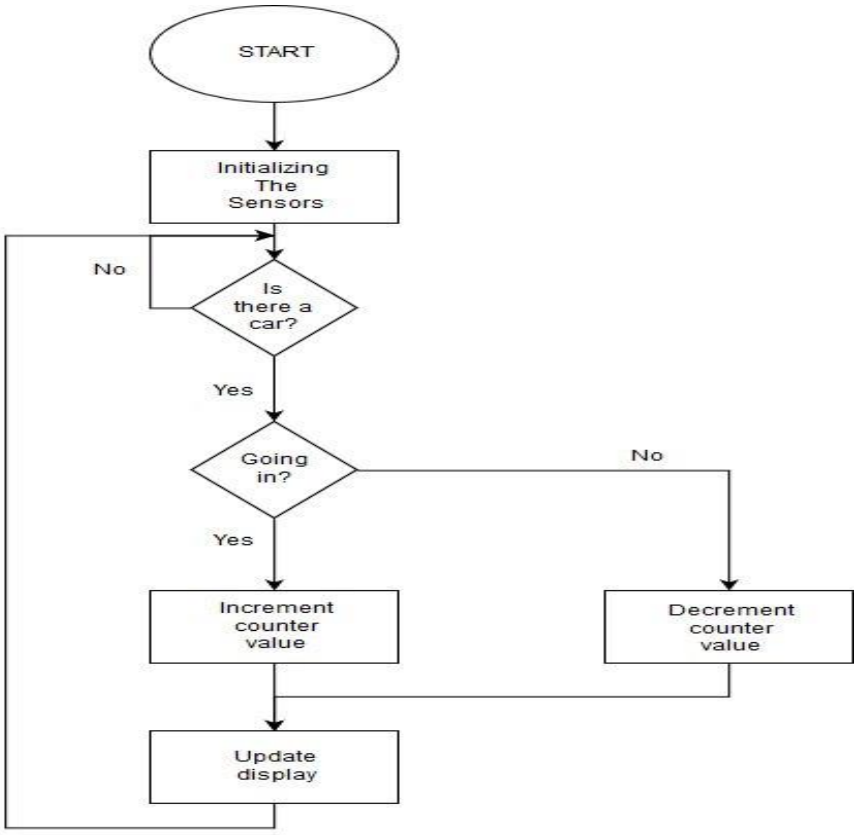

Figure 9. Flowchart of Smart Parking Management System

After connecting all components to the breadboard, the hardware system looks as follows,

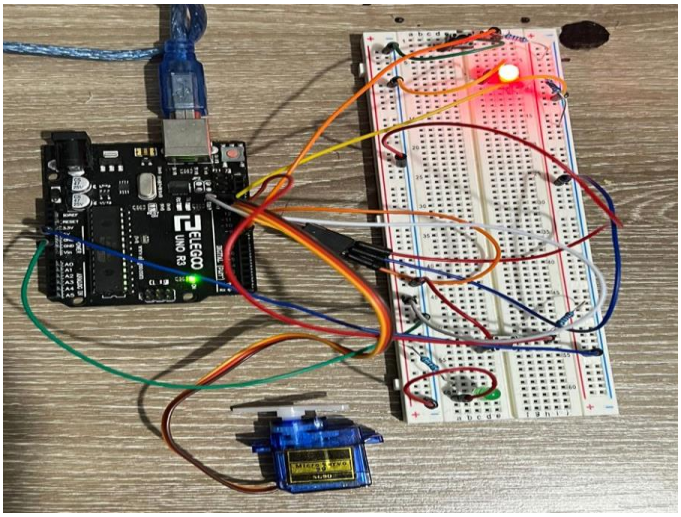

Figure 10. Smart Parking Management System Hardware System

After connecting Arduino UNO with the MATLAB application and designing the User interface, an interactive parking system is established.

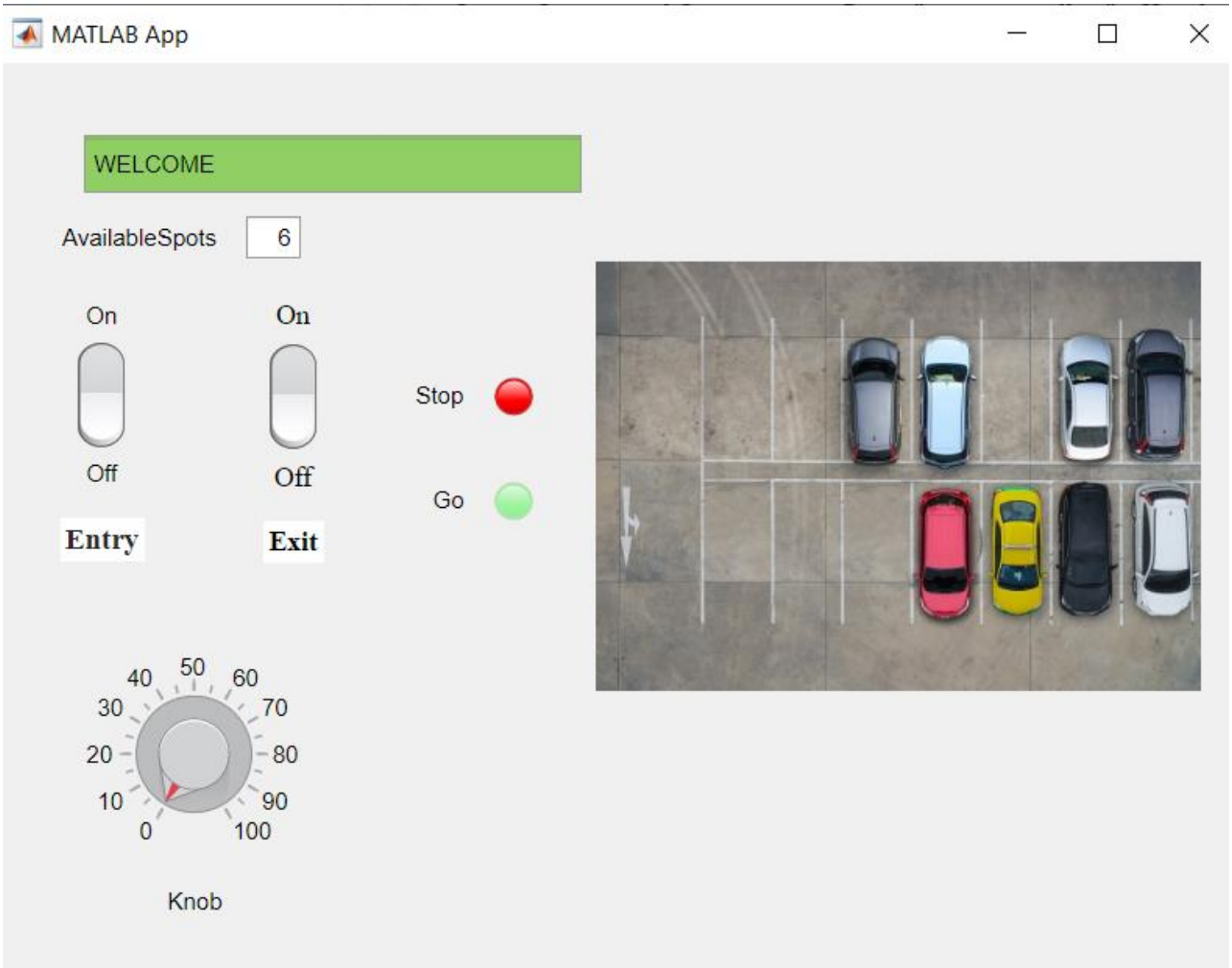

Figure 11. Smart Parking Management System Interface in MATLAB Application

5. MATLAB SCRIPT

Classdef app1 < matlab.apps.AppBase

```
    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        Knob                    matlab.ui.control.Knob
        KnobLabel               matlab.ui.control.Label
        GoLamp                  matlab.ui.control.Lamp
        GoLampLabel             matlab.ui.control.Label
        StopLamp                matlab.ui.control.Lamp
```

```
    StopLampLabel               matlab.ui.control.Label
    EditField2                  matlab.ui.control.EditField
    Image                       matlab.ui.control.Image
    AvailableSpotsEditField     matlab.ui.control.NumericEditField
    AvailableSpotsEditFieldLabel  matlab.ui.control.Label
    ExitSwitch                  matlab.ui.control.RockerSwitch
    ExitSwitchLabel             matlab.ui.control.Label
    EntrySwitch                 matlab.ui.control.RockerSwitch
    EntrySwitchLabel            matlab.ui.control.Label
end


properties (Access = private)
    Counter=6;
    a;
    Servo;
    response;
end


% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        app.a=arduino();
        app.Servo=servo(app.a,'D9');
        app.EditField2.Value='WELCOME'
        app.AvailableSpotsEditField.Value=6;
        app.Image.ImageSource='p8.jpg'
        writeDigitalPin(app.a,'D12',1);
        writeDigitalPin(app.a,'D13',0);
        app.StopLamp.Color='red';
        app.StopLamp.Enable='on';
        app.GoLamp.Enable='off';
    end

    % Close request function: UIFigure
    function UIFigureCloseRequest(app, event)
       delete(app)
    end

    % Callback function
    function SwitchValueChanged(app, event)
    end
```

```
% Callback function
function KnobValueChanged(app, event)
end

% Value changed function: EntrySwitch
function EntrySwitchValueChanged2(app, event)
    value = app.EntrySwitch.Value;
    app.Counter=app.Counter-1;
    if app.Counter==5
        app.StopLamp.Enable='off';
        pause(1);
        app.GoLamp.Enable='on';
        pause(3);
        app.StopLamp.Enable='on';
        app.GoLamp.Enable='off';
    app.Image.ImageSource='p9.jpg'
    elseif app.Counter==4
        app.StopLamp.Enable='off';
        pause(1);
        app.GoLamp.Enable='on';
        pause(3);
        app.StopLamp.Enable='on';
        app.GoLamp.Enable='off';
        app.Image.ImageSource='p10.jpg'
    elseif app.Counter==3
        app.StopLamp.Enable='off';
        pause(1);
        app.GoLamp.Enable='on';
        pause(3);
        app.StopLamp.Enable='on';
        app.GoLamp.Enable='off';
        app.Image.ImageSource='p11.jpg'
    elseif app.Counter==2
        app.StopLamp.Enable='off';
        pause(1);
        app.GoLamp.Enable='on';
        pause(3);
        app.StopLamp.Enable='on';
        app.GoLamp.Enable='off';
        app.Image.ImageSource='p12.jpg'
    elseif app.Counter==1
        app.StopLamp.Enable='off';
        pause(1);
        app.GoLamp.Enable='on';
        pause(3);
        app.StopLamp.Enable='on';
```

```
            app.GoLamp.Enable='off';
            app.Image.ImageSource='p13.jpg'
        elseif app.Counter==0
            app.StopLamp.Enable='on';
            app.Image.ImageSource='p14.jpg'
            app.EditField2.Value='PLEASE COME BACK LATER'
        end
        app.AvailableSpotsEditField.Value=app.Counter;
        writeDigitalPin(app.a,'D12',0);
        writeDigitalPin(app.a,'D13',1);
        writePosition(app.Servo,1);
        pause(3);
        writePosition(app.Servo,0);
        writeDigitalPin(app.a,'D12',1);
        writeDigitalPin(app.a,'D13',0);

app.response=thingSpeakWrite(1582534,app.Counter,'Fields',1,'WriteKey','BETR9GUSITBGC
GB6');
    end

    % Value changed function: ExitSwitch
    function ExitSwitchValueChanged(app, event)
        value = app.ExitSwitch.Value;
        app.Counter=app.Counter+1;
        if strcmp(value,'On')
            app.StopLamp.Color = 'red' ;
        else
            app.StopLamp.Color = 'green';
        end
        if app.Counter==6
            app.StopLamp.Enable='off';
            pause(1);
            app.GoLamp.Enable='on';
            pause(3);
            app.StopLamp.Enable='on';
            app.GoLamp.Enable='off';
            app.Image.ImageSource='p8.jpg';
        elseif app.Counter==5
            app.StopLamp.Enable='off';
            pause(1);
            app.GoLamp.Enable='on';
            pause(3);
            app.StopLamp.Enable='on';
            app.GoLamp.Enable='off';
            app.Image.ImageSource='p9.jpg';
            app.EditField2.Value='WELCOME';
```

```
elseif app.Counter==4
    app.StopLamp.Enable='off';
    pause(1);
    app.GoLamp.Enable='on';
    pause(3);
    app.StopLamp.Enable='on';
    app.GoLamp.Enable='off';
    app.Image.ImageSource='p10.jpg';
    app.EditField2.Value='WELCOME';
elseif app.Counter==3
    app.StopLamp.Enable='off';
    pause(1);
    app.GoLamp.Enable='on';
    pause(3);
    app.StopLamp.Enable='on';
    app.GoLamp.Enable='off';
    app.Image.ImageSource='p11.jpg';
    app.EditField2.Value='WELCOME';
elseif app.Counter==2
    app.StopLamp.Enable='off';
    pause(1);
    app.GoLamp.Enable='on';
    pause(3);
    app.StopLamp.Enable='on';
    app.GoLamp.Enable='off';
    app.Image.ImageSource='p12.jpg';
    app.EditField2.Value='WELCOME';
elseif app.Counter==1
    app.StopLamp.Enable='off';
    pause(1);
    app.GoLamp.Enable='on';
    pause(3);
    app.StopLamp.Enable='on';
    app.GoLamp.Enable='off';
    app.Image.ImageSource='p13.jpg';
    app.EditField2.Value='WELCOME';
elseif app.Counter==0
    app.Image.ImageSource='p14.jpg';
    app.EditField2.Value='PLease Come Back Later';
end
app.AvailableSpotsEditField.Value=app.Counter;
writeDigitalPin(app.a,'D12',0);
writeDigitalPin(app.a,'D13',1);
writePosition(app.Servo,1);
pause(3);
writePosition(app.Servo,0);
```

```
            writeDigitalPin(app.a,'D12',1);
            writeDigitalPin(app.a,'D13',0);

%app.response=thingSpeakWrite(1582534,app.Counter,'Fields',1,'WriteKey','BETR9GUSITBG
CGB6');
        end

        % Button down function: UIFigure
        function UIFigureButtonDown(app, event)
        end

        % Value changing function: Knob
        function KnobValueChanging(app, event)
            changingValue = event.Value;
            value = app.Knob.Value;
            writePosition(app.Servo,value/100);
        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.Position = [100 100 640 480];
            app.UIFigure.Name = 'MATLAB App';
            app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest,
true);
            app.UIFigure.ButtonDownFcn = createCallbackFcn(app, @UIFigureButtonDown, true);

            % Create EntrySwitchLabel
            app.EntrySwitchLabel = uilabel(app.UIFigure);
            app.EntrySwitchLabel.BackgroundColor = [1 1 1];
            app.EntrySwitchLabel.HorizontalAlignment = 'center';
            app.EntrySwitchLabel.FontName = 'Times New Roman';
            app.EntrySwitchLabel.FontSize = 15;
            app.EntrySwitchLabel.FontWeight = 'bold';
            app.EntrySwitchLabel.FontColor = [0.149 0.149 0.149];
            app.EntrySwitchLabel.Position = [33 226 43 22];
            app.EntrySwitchLabel.Text = 'Entry';

            % Create EntrySwitch
            app.EntrySwitch = uiswitch(app.UIFigure, 'rocker');
```

```
        app.EntrySwitch.ValueChangedFcn = createCallbackFcn(app,
@EntrySwitchValueChanged2, true);
        app.EntrySwitch.Position = [42 284 24 55];

        % Create ExitSwitchLabel
        app.ExitSwitchLabel = uilabel(app.UIFigure);
        app.ExitSwitchLabel.BackgroundColor = [1 1 1];
        app.ExitSwitchLabel.HorizontalAlignment = 'center';
        app.ExitSwitchLabel.FontName = 'Times New Roman';
        app.ExitSwitchLabel.FontSize = 14;
        app.ExitSwitchLabel.FontWeight = 'bold';
        app.ExitSwitchLabel.Position = [137 225 31 22];
        app.ExitSwitchLabel.Text = 'Exit';

        % Create ExitSwitch
        app.ExitSwitch = uiswitch(app.UIFigure, 'rocker');
        app.ExitSwitch.ValueChangedFcn = createCallbackFcn(app,
@ExitSwitchValueChanged, true);
        app.ExitSwitch.FontName = 'Times New Roman';
        app.ExitSwitch.FontSize = 14;
        app.ExitSwitch.Position = [140 283 24 54];

        % Create AvailableSpotsEditFieldLabel
        app.AvailableSpotsEditFieldLabel = uilabel(app.UIFigure);
        app.AvailableSpotsEditFieldLabel.HorizontalAlignment = 'right';
        app.AvailableSpotsEditFieldLabel.Position = [29 381 84 22];
        app.AvailableSpotsEditFieldLabel.Text = 'AvailableSpots';

        % Create AvailableSpotsEditField
        app.AvailableSpotsEditField = uieditfield(app.UIFigure, 'numeric');
        app.AvailableSpotsEditField.Position = [128 381 28 22];
        app.AvailableSpotsEditField.Value = 6;

        % Create Image
        app.Image = uiimage(app.UIFigure);
        app.Image.Position = [307 123 311 293];

        % Create EditField2
        app.EditField2 = uieditfield(app.UIFigure, 'text');
        app.EditField2.BackgroundColor = [0.5647 0.8118 0.3882];
        app.EditField2.Position = [45 415 255 29];
        app.EditField2.Value = 'WELCOME';

        % Create StopLampLabel
        app.StopLampLabel = uilabel(app.UIFigure);
        app.StopLampLabel.HorizontalAlignment = 'right';
```

```
        app.StopLampLabel.Position = [210 300 30 22];
        app.StopLampLabel.Text = 'Stop';

        % Create StopLamp
        app.StopLamp = uilamp(app.UIFigure);
        app.StopLamp.Position = [255 300 20 20];
        app.StopLamp.Color = [1 0 0];

        % Create GoLampLabel
        app.GoLampLabel = uilabel(app.UIFigure);
        app.GoLampLabel.HorizontalAlignment = 'right';
        app.GoLampLabel.Position = [215 247 25 22];
        app.GoLampLabel.Text = 'Go';

        % Create GoLamp
        app.GoLamp = uilamp(app.UIFigure);
        app.GoLamp.Position = [255 247 20 20];

        % Create KnobLabel
        app.KnobLabel = uilabel(app.UIFigure);
        app.KnobLabel.HorizontalAlignment = 'center';
        app.KnobLabel.Position = [85 41 34 22];
        app.KnobLabel.Text = 'Knob';

        % Create Knob
        app.Knob = uiknob(app.UIFigure, 'continuous');
        app.Knob.ValueChangingFcn = createCallbackFcn(app, @KnobValueChanging, true);
        app.Knob.Position = [71 97 60 60];

        % 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 = app1

        % Create UIFigure and components
        createComponents(app)

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

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

            if nargout == 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
```

6. RESULT

The Smart Parking Management System has been designed and implemented successfully in MATLAB application using Arduino Uno. Cloud connectivity has also been established with ThingSpeak. The project has also been tested successfully under all the testing conditions.

7. CONCLUSION

Today's technology, booming every year, makes its presence felt in almost all sectors with its wide range of applications. One such is automation, which reduces manpower and saves time. Implementing it in the parking systems as shown in our Smart Parking Management System prototype will greatly alleviate the parking problem and allow people to park their automobiles fast by just looking at a screen for an open parking place. The initial testing of the prototype model has demonstrated that the project can be readily adapted to a suitable parking location, after which further enhancements can be made based on the feedback.

8. REFERENCES

1. Akarat Phasura 2017, *Smart Parking lot Guidance System with Overhead Indicators, Intelligent sensors assist control/monitor*, Shutterstock, viewed 10 November 2021, <https://www.shutterstock.com/image-illustration/smart-parking-guidance-system-overhead-indicators-1253068696 >

2. Emma Ashely 2021, *What is Arduino UNO? A Getting Started Guide*, DesignSpark, viewed 10 November 2021, <https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide >

3. Samarjith Biswas 2017, *Assessment of Surface Roughness Using LVDT: A Convenient and Inexpensive Way of Measuring Surface Irregularities*, ResearchGate, viewed 10 November 2021, <https://www.researchgate.net/figure/Arduino-UNO-The-Arduino-program-contains-two-main-parts-setup-and-loop-The-name_fig3_325115625>

4. Joel Eb 2019, *How to use a Breadboard*, Sparkfun, viewed 11 November 2021, <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all>

5. ProtoSupplies 2020, *Jumper Wires*, ProtoSupplies, viewed 12 November 2021, <https://protosupplies.com/product/jumper-wires-dupont-style-assortment-140-pcs/>

6. RealPars 2020, *What is a Servo Motor And How It Works?*, RealPars, viewed 12 November 2021, <https://realpars.com/servo-motor/>

7. Codebender 2018, *How to Use a Push Button - Arduino Tutorial*, Instructables Circuits, viewed 13 November 2021, <https://docs.onion.io/omega2-arduino-dock-starter-kit/arduino-kit-reading-a-push-button.html>