Project Report on

Smart Parking Management System

By

Course Number: GENG8030-4-R-2021F
Section: 04

Group no. 17

Name	Student Id
Divya Shrikant Puram	110059604
Vishnudev Krishnadas	110059296
Shivam Prajapati	110070074

Abstract:

Smart parking management system provides solution to ever-increasing problem of vehicle parking which is certainly an important matter of concern in this automated world. With the increase in number of vehicles on the roadways, parking an automobile has turned out to be a major concern for everyone. Finding a parking spot in crowded areas is a daunting task. The Smart Parking Management System systematically regulates traffic in parking areas thus reducing chaos and saving time.

The Smart Parking Management System is developed in MATLAB and is linked with cloud via ThingSpeak where the inflow and outflow of traffic is recorded and stored. The project is implemented using an Arduino UNO microcontroller, coupled to a servo motor and a series of LEDs used as indicators in the parking zone. The entire hardware assembly is interfaced to MATLAB UI, an application which includes a graphical representation of the entire system. The system can be operated directly from this application. When the system is activated, available number of parking spots is displayed on the screen. If spots are available, the driver presses the enter button, the signal changes from red to green color and the servo motor barrier is lifted to allow the passage of the vehicle. Once the vehicle is parked, the barrier closes and again the red signal is turned on displaying the current number of available spots in the specified parking zone.

Consequently, when a car exits, the driver presses the exit switch, the servo motor barrier is lifted, red signal turns green and the car is allowed to exit. After departure, the count resets, barrier is closed, green signal changes to red and the current available spots are displayed in the application.

A parking zone equipped with a Smart Parking System, facilitates automatic control and synchronization of incoming and outgoing traffic. Moreover, the data from the cloud enables one to record and track the behavior of traffic over a specific period. This provides a hassle-free and swift parking experience to the drivers.

Table of Contents

List of Figures

1. Introduction:

The increase in the rate of private and public vehicles especially in urban areas is now a major concern of the transport and traffic management departments all over the world. This is creating barrier to the normal flow of the traffic [1]. Even in the congested area such as shopping malls, stadiums, offices and cities managing the car parking is too difficult. Illegal parking is resulting in traffic jams and blocking of roadways thus disrupting other emergency services such as ambulance, fire trucks and police vehicles.

The Smart Parking Management System helps to mitigate this concern by assisting the driver to look for an open parking spot resulting in regulated flow of traffic. The IoT based interface allows the system to connect to the cloud and the data to be centralized. This provides real-time status about parking availability also enabling the traffic authorities to monitor the behavior of vehicle congestions, calculate the availability of on-street parking spaces or public and private parking facilities and make amendments in infrastructure wherever necessary [2].

Fig.1 : Smart Parking System [3].

2. Objective:

The objective is to build a Smart Parking Management System application using MATLAB to display the availability parking spots allowing smooth parking of vehicles. Once the incoming driver presses the enter button, the system will automatically check the availability of parking spots and will permit the passage of vehicle once the condition is satisfied. If the parking is full the entry of vehicles will be denied. In this project, the number of available spots is considered as 14. The code is generated in MATLAB and tested on Arduino kit where the entire assembly is connected to cloud by ThingSpeak.

3. Hardware:

The project comprises of the following hardware components:

- Arduino Uno R3 Controller Board.
- Servo Motor
- Resistors
- Breadboard
- LED (Red, Yellow , Green)

3.1Arduino UNO R3 Controller Board:

Arduino Uno R3 controller board is an electronic device which comprises of dual-inline package ATmega328 AVR microcontroller with 14 digital input/output pins(of which 6 can be used as PWM outputs), 6 analog inputs, 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Various sensors and input/output devices can be connected to these pins to read/write values. It is an open-source electronics platform based in easy-to-use hardware and software. Arduino controller boards are able to read inputs such as light detection by a sensor, a finger on a button, or a text message, and can activate output devices such as a motor or an LED [4].

The Arduino software is installed on windows which is used in MATLAB working environment. The board is connected to the computer through a USB and the program can be loaded and checked.

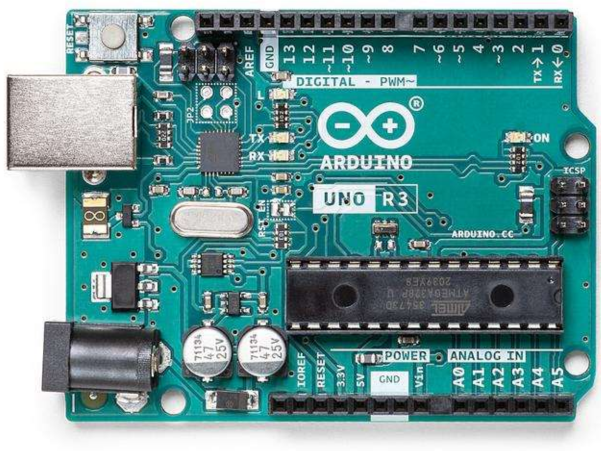

Fig.2 : Arduino UNO R3 Controller Board [4].

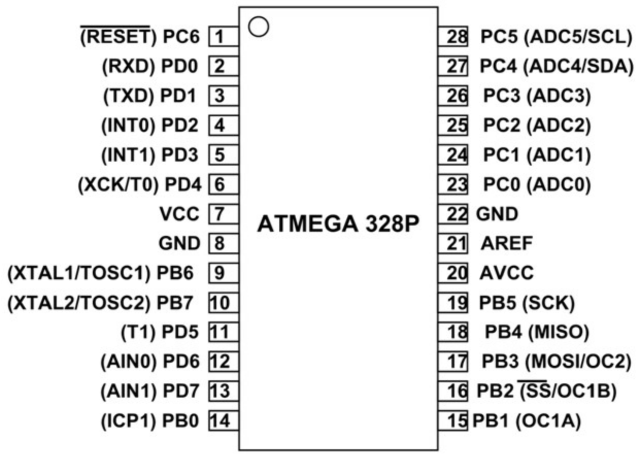

Fig.3 : ATMega 328P Pin Configuration [5].

3.2. Servo Motor:

A servo motor is an electronic motor that can rotate with excellent accuracy. The control circuit of this motor provides a feedback signal of the current position of the shaft allowing the motor to rotate with greater accuracy. By using servo motor, an object can be rotated to some specific angle or distance. This is simple motor works on a servo mechanism [6]. This servo motor is used in the project for opening and closing the barrier to permit the incoming and outgoing of vehicles.

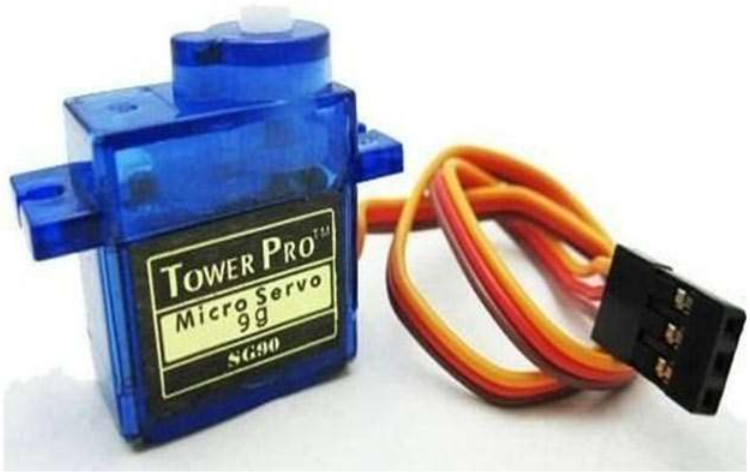

Fig.4 : Servo Motor [6].

3.3. Resistors (220 ohm):

Resistors are electronic components that are used to limit the amount of current in the circuit consisting of LEDs and integrated circuits [7]. Resistors have different value that is indicated by their different color-code. The resistors used in the project have a value of 220Ω . The 220Ω resistor has a color pattern of red, red, and brown stripes in that order. The last stripe represents the tolerance. Gold means ±5 [8]. These resistors are used to limit the current flowing through the LEDs.

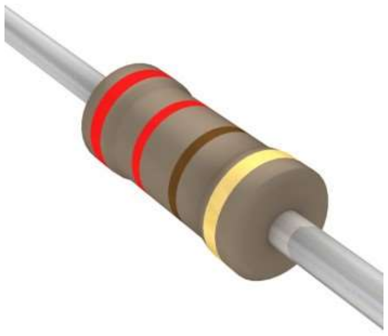

Fig.5 : Resistors (220 ohm) [8].

3.4 LEDs:

LEDs (Light Emitting Diode) are semiconductor devices that convert electrical energy directly into light, delivering efficient light generation with little-wasted electricity [9]. Based on the semiconductor material used, LEDs are available in different colors. Here LEDs are used to indicate the availability of parking spots. Red light indicates that parking space is not available, yellow light is used to look out before entry and exit and green light indicates the availability of parking spots.

Fig.6 : LED [10].

3.5 Bread Board:

A breadboard is a rectangular plastic board having a tiny hole. It is generally used for the circuit design testing. By installing electronic components, it's easy to build prototype of an electronic circuits. And the connections are not permanent, so it is easy to remove and make new connection in it [11].

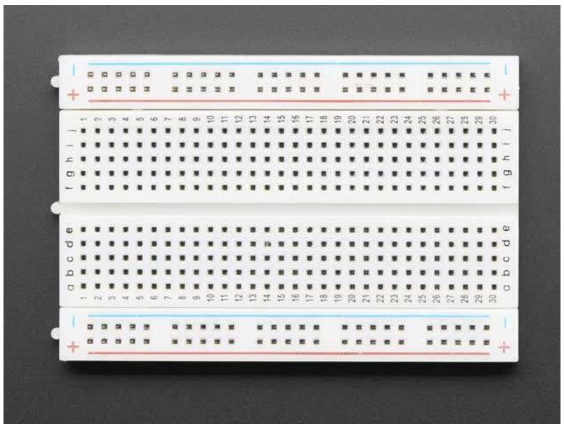

Fig.7 : Bread Board [11].

4. Software:

The following are the software packages included in the project.

- MATLAB
- Arduino Hardware support package by MATLAB.

4.1.MATLAB:

MATLAB is a programming platform used by engineers to create models, develop applications and algorithms and analyze data for numeric computing. It uses a matrix and array based programming language for a user friendly demonstration of computer mathematics [12].

4.2.Arduino Hardware support package by MATLAB:

This support package allows communication with an Arduino board using MATLAB . One can read and write data to/from a sensor through the Arduino controller board and see the results instantly in MATLAB [12].

5. Flowchart

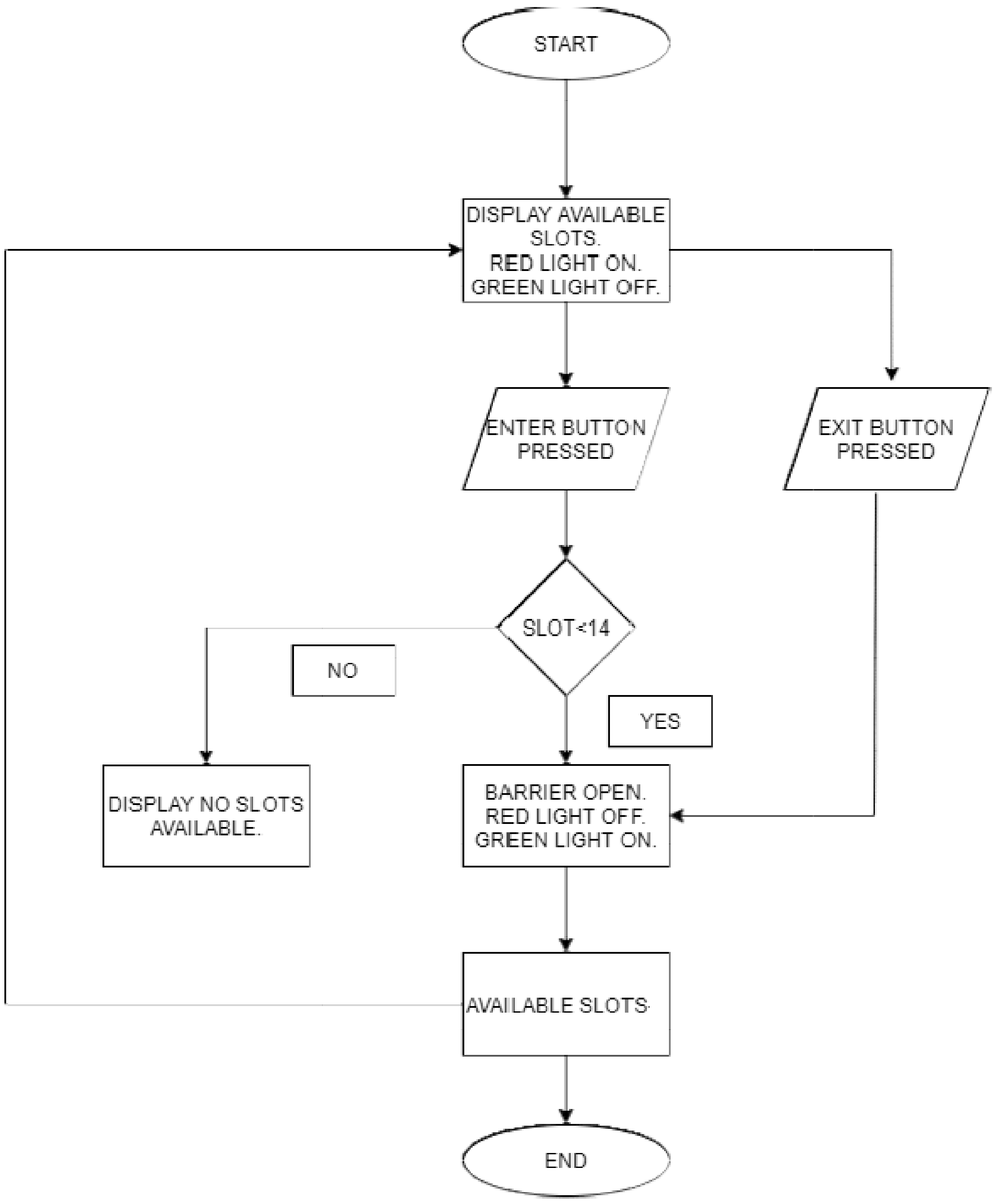
START
DISPLAY AVAILABLE SLOTS. RED LIGHT ON. GREEN LIGHT OFF.
ENTER BUTTON PRESSED
EXIT BUTTON PRESSED
SLOT<14
NO
YES
DISPLAY NO SLOTS AVAILABLE.
BARRIER OPEN. RED LIGHT OFF. GREEN LIGHT ON.
AVAILABLE SLOTS
END

Fig.8 : Flowchart of Smart Parking Management System

6. Working:

The Smart parking management system is a capable to determine the number of available parking spots in a parking zone. The collected data from the parking area will be transferred to the cloud database by using an IoT platform called ThingSpeak. The total number of empty spots is displayed on the screen. If the number of spots is less than 14 but greater than zero the driver can enter. In order to enter the parking space, the driver needs to press the enter button. Once the enter button is pressed, the red signal light changes to green and the barrier opens to 90 degrees angle with the help of a servo motor to allow the vehicle to pass. A delay is provided for the time the vehicle passes and then the barrier is back to closed position and the red light turns on. The number of available spots is now reduced by 1 and the difference is displayed on the screen.

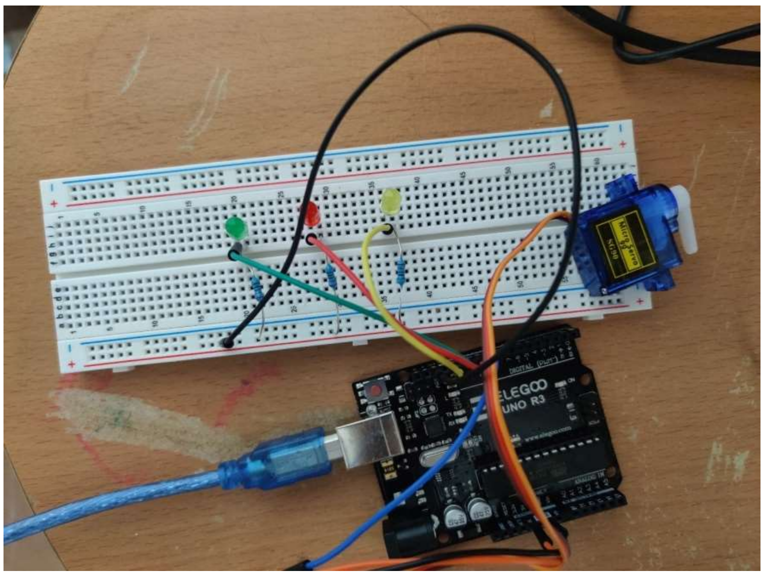
Fig.9 : Smart Parking Management System hardware assembly with Arduino board, 3 LED lights and a servo motor

When a car needs to exit, the driver presses the exit button. The barrier opens and the red light changes from red to green. The driver is permitted to pass. Once the car is exited, the number of available spots is updated by a +1 and the sum is now displayed as available spots on the screen. If there are no parking spots available and driver presses the enter button intending to park the car. The screen will display a message as ‘NO AVAILABLE SPOTS’ with a red signal.

The entire system is connected to the cloud and the log can be recorded. The data of incoming and outgoing cars can be obtained anytime by logging into ThingSpeak.

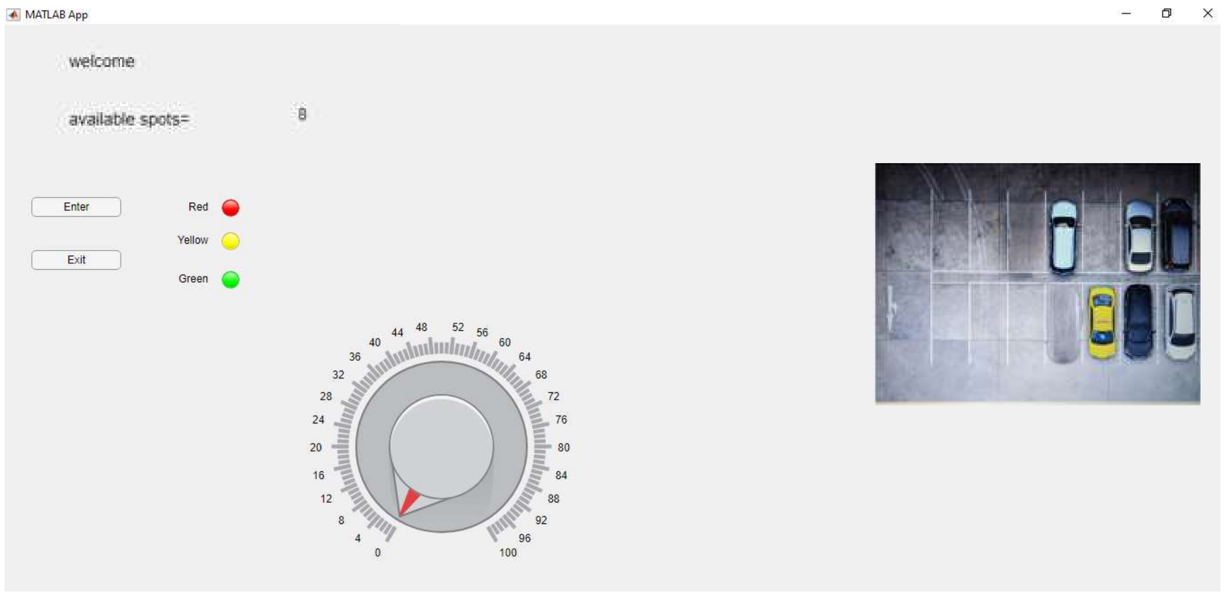

Fig.10 : Smart Parking Management System Application Screen in MATLAB

7. MATLAB Program for Smart Parking Management System

```
classdef app1trial < matlab.apps.AppBase


    % Properties that correspond to app components
    properties (Access = public)

        UIFigure          matlab.ui.Figure

        Label_3           matlab.ui.control.Label

        Label_4           matlab.ui.control.Label

        Label_5           matlab.ui.control.Label

        ExitButton        matlab.ui.control.Button

        EnterButton       matlab.ui.control.Button

        Image             matlab.ui.control.Image

        GreenLamp         matlab.ui.control.Lamp
        GreenLampLabel   matlab.ui.control.Label

        YellowLamp       matlab.ui.control.Lamp

        YellowLampLabel  matlab.ui.control.Label

        RedLamp          matlab.ui.control.Lamp

        RedLampLabel     matlab.ui.control.Label

        Knob             matlab.ui.control.Knob

        KnobLabel        matlab.ui.control.Label

    end


available_spots=10;
    end



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


        % Code that executes after component creation
        function startupFcn(app)
app.ArduinoUno=arduino('com7','UNO','Libraries','Servo');
app.Servo=servo(app.ArduinoUno,'D9');
writePosition(app.Servo,0);
writeDigitalPin(app.ArduinoUno,'D11',1);


filename=strcat("p",int2str(14-str2num(app.Label_3.Text)),".jpg");
             app.Image = uiimage(app.UIFigure);
            app.Image.Position = [297,59,322,258];
            app.Image.ImageSource = filename;




        end


        % Button pushed function: EnterButton
        function EnterButtonPushed(app, event)


          if app.available_spots>0
app.Label_5.Text='welcome';
app.Label_4.Text='available spots=';
app.available_spots=app.available_spots-1;
app.Label_3.Text=sprintf('%d',app.available_spots);
app.RedLamp.Color=[1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno,'D11',0);
writeDigitalPin(app.ArduinoUno,'D13',1);
app.YellowLamp.Color='y';
pause(0.5);
writeDigitalPin(app.ArduinoUno,'D13',0);
app.YellowLamp.Color=[1.00,1.00,1.00];
writePosition(app.Servo,0.5);
app.Knob.Value=100;
for i=0:6
filename=strcat("p",int2str(13-str2num(app.Label_3.Text)),".jpg");
app.Image = uiimage(app.UIFigure);
app.Image.Position = [297,59,322,258];
app.Image.ImageSource = filename;
app.GreenLamp.Color= 'g';
writeDigitalPin(app.ArduinoUno, 'D12', 1);
pause(0.2);
filename=strcat("p",int2str(14-str2num(app.Label_3.Text)),".jpg");
app.Image = uiimage(app.UIFigure);
app.Image.Position = [297,59,322,258];
app.Image.ImageSource = filename;
app.GreenLamp.Color= [1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno, 'D12', 0);
pause(0.2);
end
writePosition(app.Servo,0);
app.Knob.Value=0;
app.RedLamp.Color='r';
writeDigitalPin(app.ArduinoUno,'D11',1);
app.YellowLamp.Color=[1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno,'D12',0);
app.GreenLamp.Color=[1.00,1.00,1.00];
else
app.Label_4.Text='No available spots';
end
%drawnow
```

```
thingSpeakWrite(1579183,app.available_spots,'WriteKey','O3G5F0J8PUQH2GPO');
end


% Button pushed function: ExitButton
function ExitButtonPushed(app, event)
if app.available_spots<14
app.Label_4.Text='available spots=';
app.available_spots=app.available_spots+1;
app.Label_3.Text=sprintf('%d',app.available_spots);
app.RedLamp.Color=[1.00,1.00,1.00 ];
writeDigitalPin(app.ArduinoUno,'D11',0);
app.YellowLamp.Color='y';
writeDigitalPin(app.ArduinoUno,'D13',1);
pause(0.5);
writeDigitalPin(app.ArduinoUno,'D13',0);
app.YellowLamp.Color=[1.00,1.00,1.00];
writePosition(app.Servo,0.5);
app.Knob.Value=100;
for i=0:6
filename=strcat("p",int2str(15-str2num(app.Label_3.Text)),".jpg");
app.Image = uiimage(app.UIFigure);
app.Image.Position = [297,59,322,258];
app.Image.ImageSource = filename;
app.GreenLamp.Color= 'g';
writeDigitalPin(app.ArduinoUno, 'D12', 1);
pause(0.2);
filename=strcat("p",int2str(14-str2num(app.Label_3.Text)),".jpg");
app.Image = uiimage(app.UIFigure);
app.Image.Position = [297,59,322,258];
app.Image.ImageSource = filename;
app.GreenLamp.Color= [1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno,'D12',0);
pause(0.2);
end
writePosition(app.Servo,0);
app.Knob.Value=0;
app.GreenLamp.Color=[1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno,'D12',0);
app.YellowLamp.Color=[1.00,1.00,1.00];
writeDigitalPin(app.ArduinoUno,'D13',0);
writeDigitalPin(app.ArduinoUno,'D11',1);
app.RedLamp.Color='r';
end
```

```
thingSpeakWrite(1579183,app.available_spots,'WriteKey','O3G5F0J8PUQH2GPO');
end

% Value changed function: Knob
function KnobValueChanged(app, event)
value = app.Knob.Value;
writePosition(app.Servo,value/200);
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';

% Create KnobLabel
app.KnobLabel = uilabel(app.UIFigure);
app.KnobLabel.HorizontalAlignment = 'center';
app.KnobLabel.Position = [105 38 34 22];
app.KnobLabel.Text = 'Knob';

% Create Knob
app.Knob = uiknob(app.UIFigure, 'continuous');
app.Knob.ValueChangedFcn = createCallbackFcn(app, @KnobValueChanged, true);
app.Knob.Position = [91 94 60 60];

% Create RedLampLabel
app.RedLampLabel = uilabel(app.UIFigure);
app.RedLampLabel.HorizontalAlignment = 'right';
app.RedLampLabel.Position = [201 267 27 22];
app.RedLampLabel.Text = 'Red';
```

```
% Create RedLamp
app.RedLamp = uilamp(app.UIFigure);
app.RedLamp.Position = [243 267 20 20];
app.RedLamp.Color = [1 0 0];

% Create YellowLampLabel
app.YellowLampLabel = uilabel(app.UIFigure);
app.YellowLampLabel.HorizontalAlignment = 'right';
app.YellowLampLabel.Position = [189 230 39 22];
app.YellowLampLabel.Text = 'Yellow';

% Create YellowLamp
app.YellowLamp = uilamp(app.UIFigure);
app.YellowLamp.Position = [243 230 20 20];
app.YellowLamp.Color = [1 1 0];

% Create GreenLampLabel
app.GreenLampLabel = uilabel(app.UIFigure);
app.GreenLampLabel.HorizontalAlignment = 'right';
app.GreenLampLabel.Position = [189 187 39 22];
app.GreenLampLabel.Text = 'Green';

% Create GreenLamp
app.GreenLamp = uilamp(app.UIFigure);
app.GreenLamp.Position = [243 187 20 20];

% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.Position = [297 59 322 258];
app.Image.ImageSource = 'p4.jpg';

% Create EnterButton
app.EnterButton = uibutton(app.UIFigure, 'push');
app.EnterButton.ButtonPushedFcn = createCallbackFcn(app, @EnterButtonPushed,
true);
app.EnterButton.Position = [31 267 100 22];
app.EnterButton.Text = 'Enter';
```

```
% Create ExitButton
app.ExitButton = uibutton(app.UIFigure, 'push');
app.ExitButton.ButtonPushedFcn = createCallbackFcn(app, @ExitButtonPushed,
true);
app.ExitButton.Position = [31 208 100 22];
app.ExitButton.Text = 'Exit';


% Create Label_5
app.Label_5 = uilabel(app.UIFigure);
app.Label_5.Position = [47 430 143 25];
app.Label_5.Text = '';


% Create Label_4
app.Label_4 = uilabel(app.UIFigure);
app.Label_4.Position = [47 387 128 28];
app.Label_4.Text = '';


% Create Label_3
app.Label_3 = uilabel(app.UIFigure);
app.Label_3.Position = [211 393 399 22];
app.Label_3.Text = '';


% 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 = app1trial


% 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
```

8. ThingSpeak:

ThingSpeak is an IoT analytics cloud platform that allows to visualize, aggregate and analyze live data logs and streams in the cloud. One can send or receive data from ThingSpeak from any device and create visuals [13].

ThingSpeak is used in this project to generate a graph of Available spots vs Time. The following graph shows the data log.

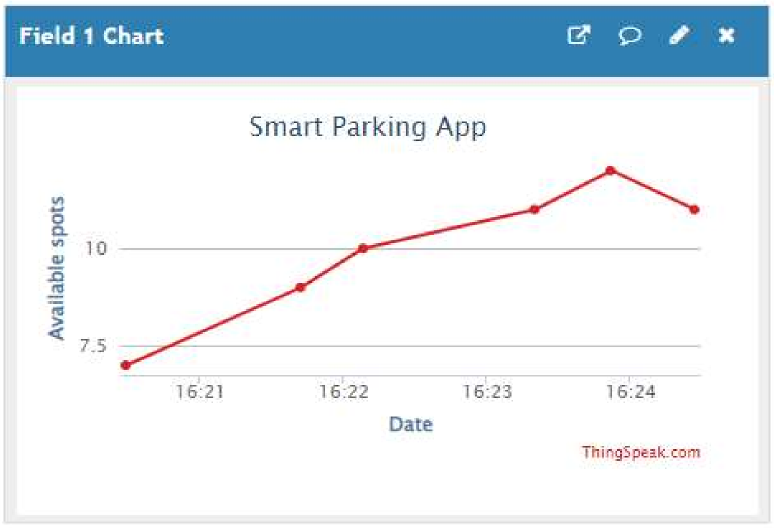

Fig.11 : Graph of Available Spots vs Time

9. Conclusion:

Traffic regulation in the parking space is achieved using the Arduino kit and MATLAB platform.Moreover, a record of incoming and outgoing traffic data is also maintained using ThingSpeak which is accessible anytime. The objective of this project to develop a Smart Parking Management System with live status of available parking spots is thus achieved. This application can be implemented in many areas especially shopping malls, business hubs, economically busy urban areas to manage the flow of traffic.

10.Refrences:

[1] J.Parmar, P.Das and S. M. Dave, “Study on demand and characteristics of parking system in urban areas: A review”, Journal *of Traffic and Transportation Engineering, Volume 7, Issues 1*, Feb2020.[Online].
Available: https://www.sciencedirect.com/science/article/pii/S2095756418305786
[Accessed : November 18, 2021]

[2] Lucia burbano,’What is a Smart Parking System? Functionalities and Benefits”,*Tomorrow City*, Aug 17, 2021. [Online]. Available: https://tomorrow.city/a/smart-parking . [Accessed: November 21, 2021]

[3] IPVS, “Address your parking woes with a Smart Parking Management Solution”, *Matrix Security Solutions,* Jan 27, 2020.[Online]. Available : https://www.matrixvideosurveillance.com/blog/address-your-parking-woes-with-a-smart-parking-management-solution/ [Accessed:November 18, 2021].

[4] Arduino.cc,“Arduino Uno R3” Store. [Online].
Available:https://store.arduino.cc/products/arduino-uno-rev3 [Accessed: November 15, 2021]

[5] Components 101, “ATMega328P”, April 4, 2018.[Online].
Available : https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet
[Accessed:November 20, 2021].

[6] Apoorve, “Understanding the basics of servomotor working” *CircuitDigest* , Aug 01, 2015.
[Online]. Available: https://circuitdigest.com/article/servo-motor-working-and-basics
[Accessed: November 17, 2021.]

[7] **“**Resistor,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Resistor
[Accessed: November 18, 2021]

[8] “220 ohm resistor color code – Overview and tips” SM TECH [Online]. Available:
https://somanytech.com/220-ohm-resistor-color-code/ [Accessed: November 18, 2021]

[9] “What is an LEDs?,” *LEDs Magazine*, Sept 01 2004. [Online]. Available:
https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led
[Accessed: November 19, 2021]

[10] “Light Emitting Diode Basics”, *Electronics Hub*, May 20, 2021. [Online]. Available:
https://www.electronicshub.org/light-emitting-diode-basics/ . [Accessed: 19th November 2021]

[11] “How to use a breadboard,” Science buddies [Online]. Available:[https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard] [Accessed: November 20, 2021]

[12] “MATLAB Support Package for Arduino Hardware”, MathWorks. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/47522-matlab-support-package-for-arduino-hardware?s_tid=srchtitle_arduino%2520hardware_7. [Accessed : November 22, 2021].

[13] “ThingSpeak for IoT Projects”, ThingSpeak. [Online]. Available: https://thingspeak.com/ . [Accessed: November 22, 2021]