University of Windsor

Faculty of Engineering

GENG8030-5-R-2022S: Comp. Methods/Modeling for Eng.

Department of Electrical and Computer Engineering

Primary Project Report on

Smart Parking Management System

Group no: 13

Mohini Sarkar (110086149)

Umang Shrestha (110071431)

Instructor

Dr. Yasser M. Alginahi

TABLE OF CONTENTS

ABSTRACT

As we all know, the traditional parking systems are completely manual and needs human intervention at every step starting from entering data into the system, processing payments post every entry or exit as the system is established, updating the status of the slots available, and monitoring the overall operation of the system [1]. Traditional parking systems are extremely frustrating and slow. People in the queue are required to wait in line for longer period and the same can lead to traffic congestion and air as well as noise pollution around the neighbourhood. Removing the huge manual dependency from the entire parking process of parking is the whole purpose of this project [1].

The smart parking system will give an opportunity to automate the end-to-end process of parking and any individual can easily identify and grab an empty parking place as per the choice of their convenience. Smart parking technique when implemented successfully will ultimately help in the optimal usage of the available parking space [2]. Not only that, the overall efficiency of various public parking related activities, collection and storage of data for future predictions, and the unwanted traffic due to manual activities involved will improve [2].

The project's purpose is to create a Smart Parking Management System that continuously monitors and updates the live data about the available parking slots in each facility where the smart system is installed. The data is continuously updated and displayed near the entry and exit gates using MATLAB software and the Arduino Uno kit, thus enabling any new car visiting the parking area to take decision on whether to wait or start looking for a different place to park the car. Similar concept can be implemented on a wider scope by using multiple sensors and new adjustments as per the requirements.

LIST OF FIGURES

1.0 Introduction

In the recent era, finding a place to park the car is one of the most challenging tasks at hand. The number of vehicles on road is continuously increasing. Owning vehicle has become the ask of the time and is no more considered to be a luxury [1]. In urban cities, congestion has become a predominant factor due to the rapid growth of the automobile industry. The continuously increasing heavy traffic in the urban areas leads to air and noise pollution. Not only that, the amount of time and labour invested in the traditional approach of parking, calls for some automatic system that can smartly handle the hurdles of manual involvement. It is projected that by 2050, the urban population will account for 68 percent of the global population [1]. Every household owns at least one car if not more. This results in almost a quarter of all cars in cities searching for parking spots all the time [2].

In order to overcome the failings of the traditional parking system, smart parking is proposed as a solution. This project aims to create a Smart Parking Management System that monitors available parking spaces and assists drivers with parking. It utilizes MATLAB and Arduino technologies. The use of smart parking technology will solve this problem, enabling drivers to park more joyfully and save time. It will also reduce traffic flow and pollution resulting from automotive emissions. The below figure shows the smart parking system.

Figure 1: Smart Parking System [3]

2.0 Hardware Components

Below is the list of hardware components used in the Arduino implementation.

Name	Quantity	Component
UArduino Board	1	Arduino Uno R3
UDisplay	1	LCD 16 x 2
R1 R3 R4 R5 R6	5	220 Ω Resistor
SERVOGate	1	Positional Micro Servo
DEnter Led DExit Led	2	LED RGB
SExit Button SEnter Button	2	Pushbutton
R7 R2	2	1 kΩ Resistor

2.0.1 Arduino Uno

Microchip's ATmega328P microcontroller is the core of the Arduino Uno open-source microcontroller board. There are 14 digital inputs and outputs on the board The board has 14 pins; six of these pins are analog I/O pins and are programmable via type B USB cable with Arduino IDE. A picture of the Arduino Uno board used in the smart parking management system project [4] can be seen below.

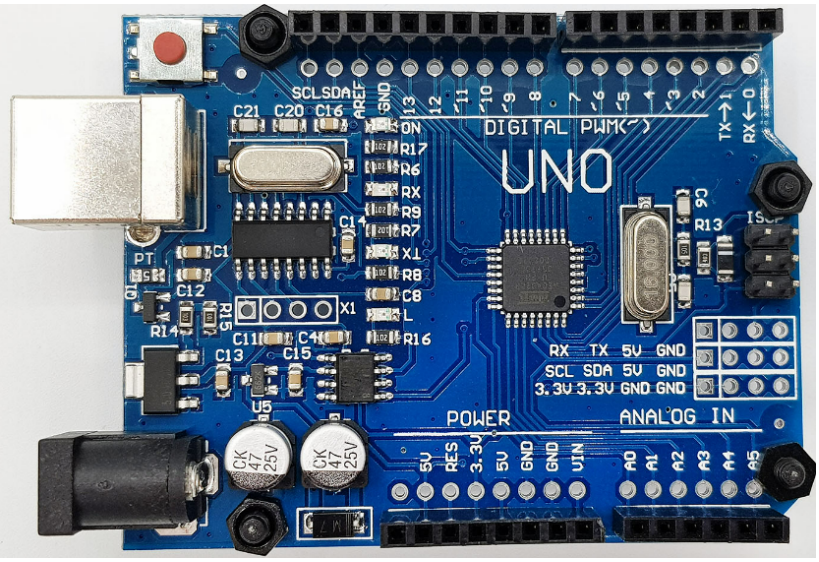

Figure 2: Arduino UNO [4]

2.0.2 Breadboard

Breadboards are boards for prototyping or building circuits. Circuit experiments can be conducted without soldering electronic components on it, so it is an ideal tool for developing temporary electronic circuits [5] .

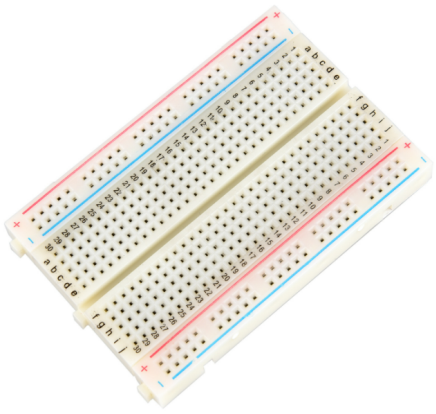

Figure 3: Breadboard [5]

2.0.3 Servo Motor

In servomotors, angle, velocity, and acceleration are precisely controlled with rotary actuators or linear actuators. Motors are coupled to sensors for feedback. VCC (red), GND (black/brown), and Signal pins (yellow/orange) make up the servo motor's three pins. The following image shows the Arduino Uno board connected to the servo motor [6].

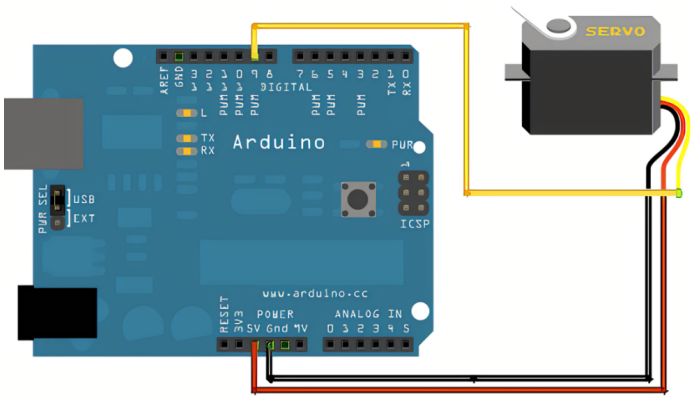

Figure 4: Servo Motor [6]

2.0.4 RGB LED

LEDs (Light Emitting Diodes) are semiconductor devices that produce infrared light when current is passed through them. LEDs are mainly used as an indicator [7]. Availability of slots is indicated by LEDs in Smart Parking System. In an electronic circuit, a resistor is a two-terminal component that is composed of a resistive material, which can limit the flow of current in a circuit. Figure 5 shows the led and resistor in an Arduino circuit [7].

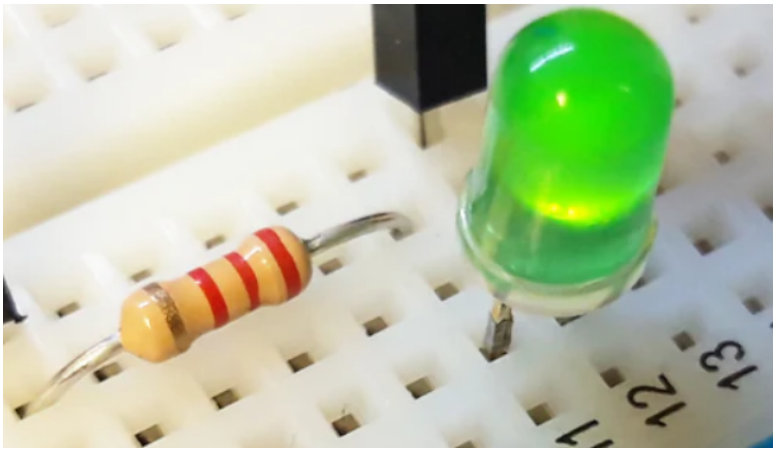

Figure 5: LED and Resistors [7]

2.0.5 LCD

Liquid Crystal Displays (LCDs) are electronic display modules that are used in a variety of applications. A major advantage of LCDs is their easy programmability and low cost [8]. The below image shows the LCD interfaced with Arduino [8].

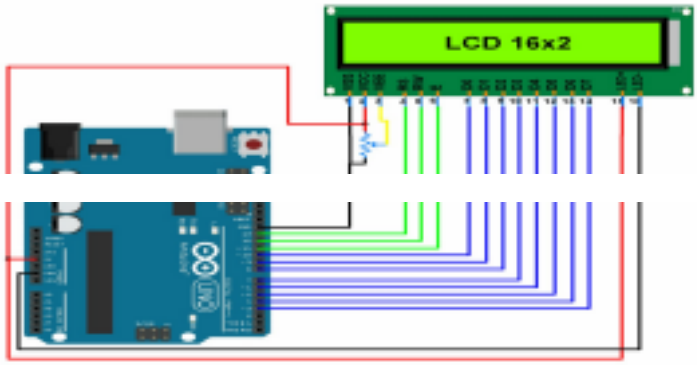

Figure 6: LCD [8]

2.0.6 Press Button Switch

Push buttons work by allowing the current to pass through them when pressed [9]. Pressing the button will default to the closed state, which is the default state of the switch. The below image shows the Arduino interface with the press button switch [9].

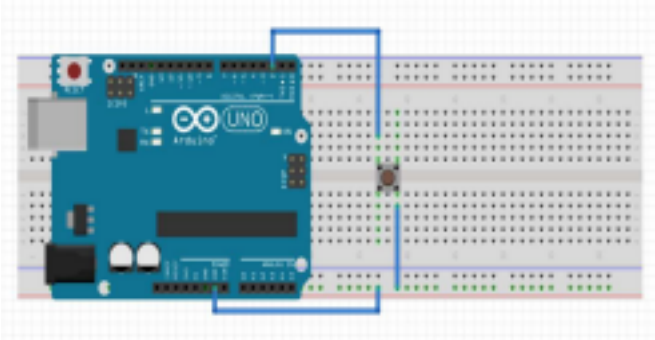

Figure 7: Press Button Switch [9]

2.0.7 Wires

A circuit is connected by a wire that acts as a medium for carrying current between the circuits [10]. Wires and jumper wires have been used to bridge the connections between the components in this project. The figure 8 shows the jumper wires.

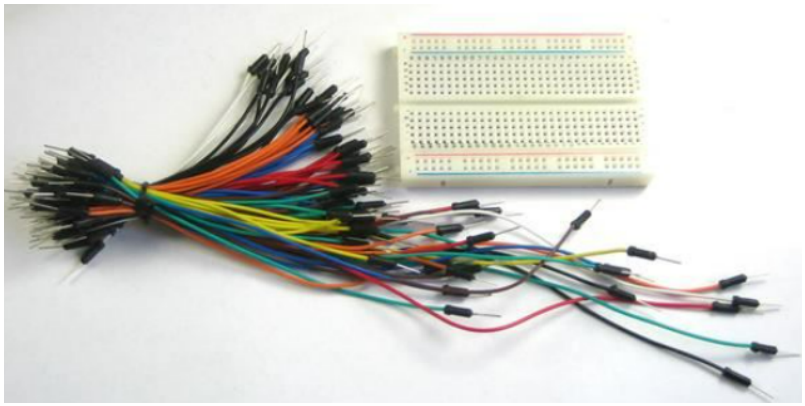

Figure 8: Jumper Wires [10]

2.1 Software Components

1. MATLAB
2. TinkerCad

3.0 METHODOLOGY

Thorough research was executed to understand the planned mechanism of the smart parking system. The basic pseudocode and flowchart were built, followed by the Arduino code to

simulate the same logic on TinkerCad (Arduino). When the Arduino code was successfully implemented and tested with the all the flows and scenarios identified in the flowchart, efforts were made to mimic the same in MATLAB. The MATLAB code was further integrated with Arduino hardware using the Arduino and LCD supported MATLAB add-ons. The Arduino pins were connected to a breadboard which were further connected to various components such as a LCD display, 5 resistors, 2 LEDs each for entry ad exit gates, 2 push button switches each for entry and exit, 6 resistors and a servomotor.

The base of the circuit was the Arduino hardware and the other identified hardware components were connected based on the circuit diagram. The servomotor is attached to the The Arduino's digital pins D5 and D6 are linked to the negative ports of the LED light near the entry gate and, D13 and D14 are linked to the negative ports of the LED lights near the exit gate. The positive port of the LED lights is connected to the ground passing through resistors. The Arduino's D3 and D4 digital pins are used to link the enter and exit switch buttons. Arduino's digital pin D7, which has one end connected to the ground and the other to the breadboard's power supply. The jumper wires from the Arduino's Data pins D8 to D11 are connected to the data pins of the LCD, D12 to the enable pin and D13 of the Arduino to the resistance pin to connect the LCD to the breadboard.

MATLAB code is executed by Arduino using the MATLAB software, which communicates with the software’s code. The Arduino runs the code and operates the servo motor and the LED lights using the jumper wires that connect to the Arduino’s digital pin.

3.0.1 Pseudo Code

SET used parking spots, num as 0
SET Enter Led, L2 as Red
SET Exit Led, L2 as Green
SET Servo Position, pos as 0
DISPLAY Student Name and Student Id

```
LOOP FOREVER
        DISPLAY "Welcome!!!" in first row of LCD
        DISPLAY "Available:", num in second row of LCD
        IF Enter Button is pushed
                IF num < 13
                        INCREMENT num by 1
                        SET L1 as Green
                        SET pos as 90
                        SLEEP for 1 second
                        SET L1 as Red
                        SET pos as 0
                ELSE
                        DISPLAY "Please come later!!!" in first row of LCD
                END IF
        ELSE IF Exit Button is pushed
                IF num > 0
                        DECREMENT num by 1
                        SET L2 as Green
                        SET pos as 90
                        SLEEP for 1 second
                        SET L2 as Red
                        SET pos as 0
                ELSE
                        DISPLAY "No Car Inside!!!" in first row of LCD
                END IF
        END IF
END LOOP
```

3.0.2 Flow Chart

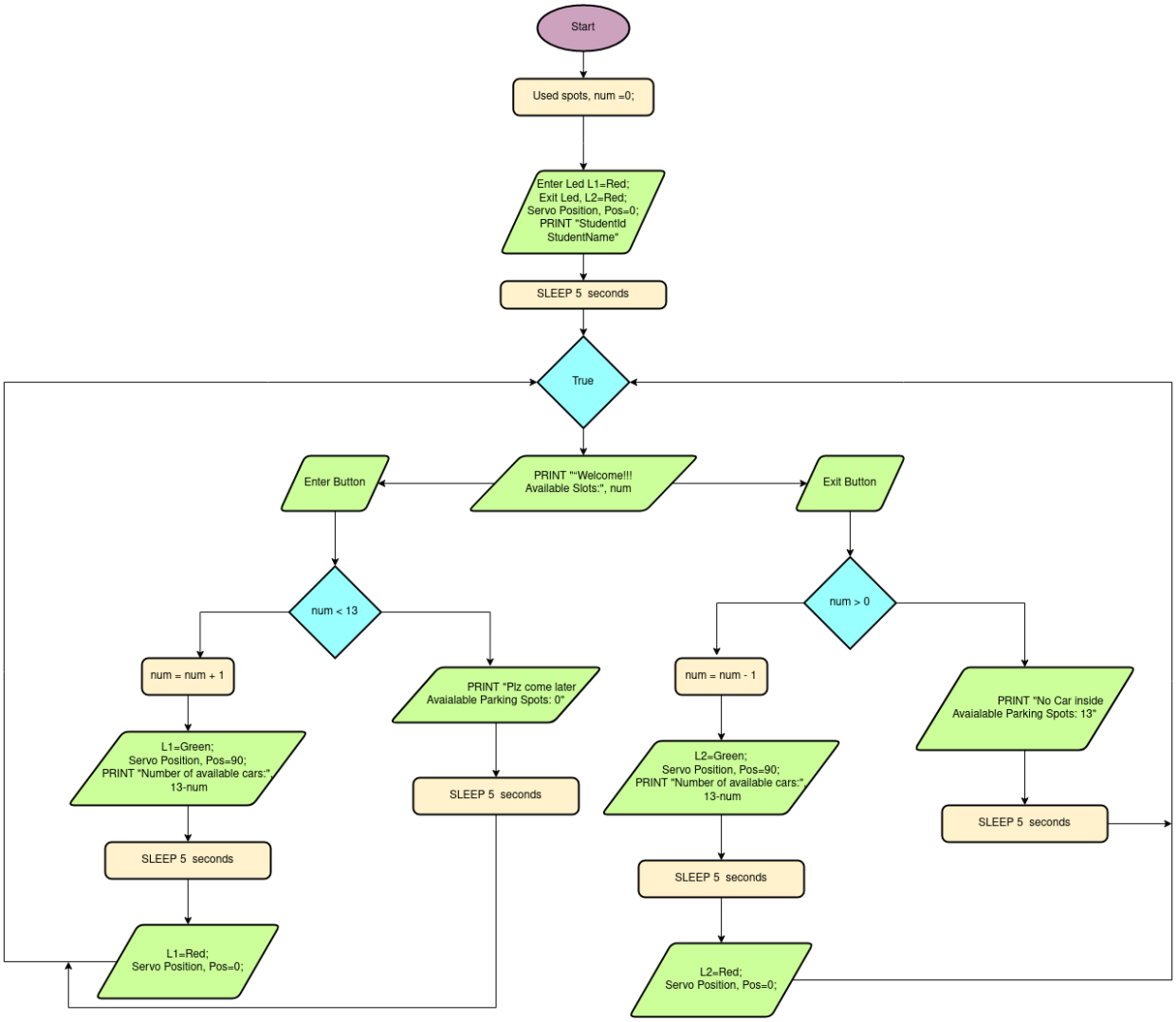

Figure 9: Flow Chart

3.0.3 Schematic Diagram

The underlying schematic diagram for the simulation is also given below:

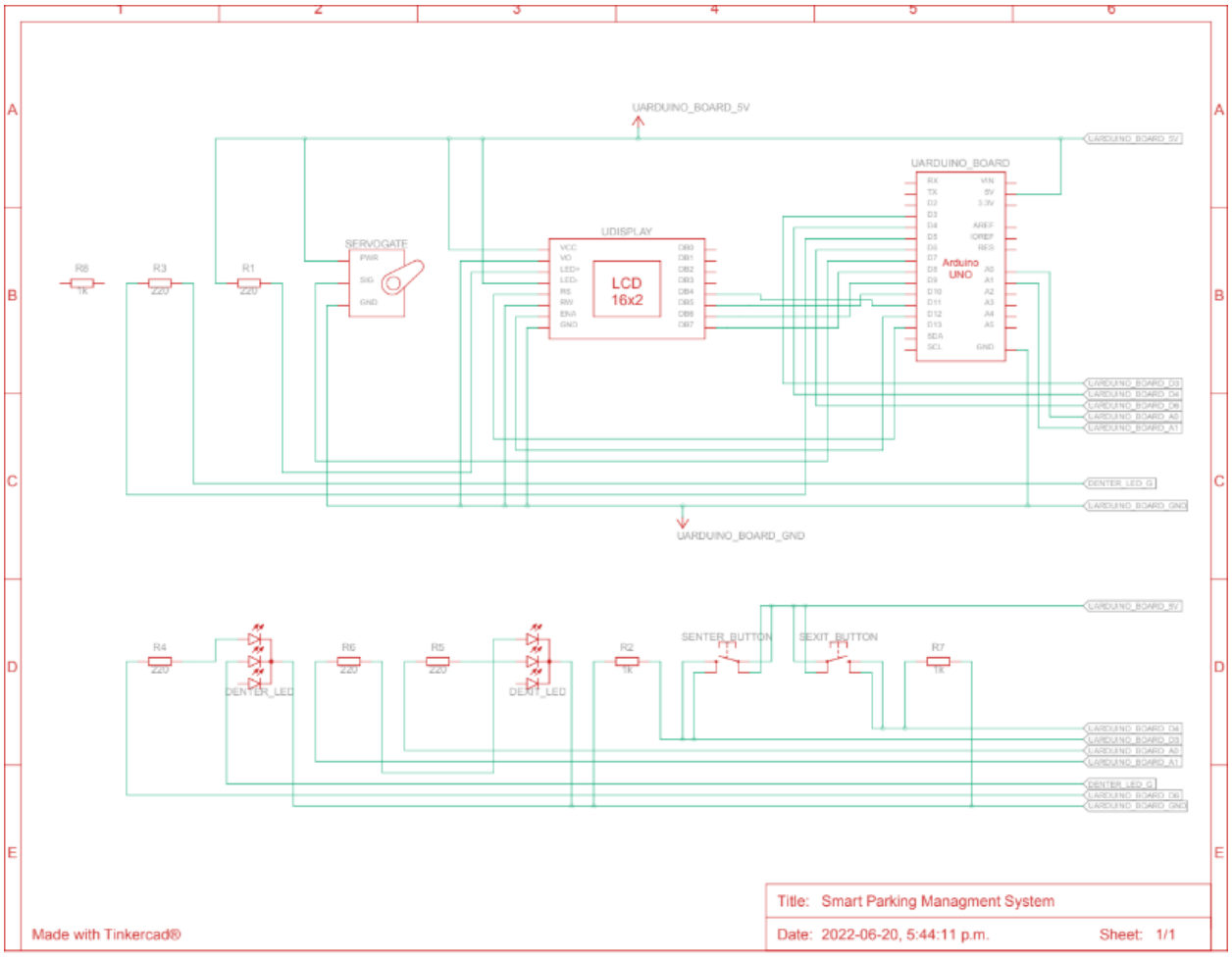

Figure 10: Schematic Diagram

3.0.4 Arduino Code

```
#include <LiquidCrystal.h>
#include <Servo.h>

/* LCD */
#define RS 13
#define EN 12
```

```
#define D4 11
#define D5 10
#define D6 9
#define D7 8
/* creating LCD object */
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);

/* ServoMoter */
#define SERVO_PIN 7
#define setGate_OPEN 90
#define setGate_CLOSE 0
/* Creating Servo Object */
Servo servo;

// ENTER_LED
#define RED_LED_ENTER 6
#define GREEN_LED_ENTER 5

//  BUTTON
#define BUTTON_ENTER 3
#define BUTTON_EXIT 4

// EXIT_LED
#define RED_LED_EXIT 14
#define GREEN_LED_EXIT 15

// ENUM
#define ON 255
#define OFF 0

//color
#define RED 1
#define GREEN 0
#define LED_DELAY 50
```

```
// state
#define MAX_ALLOCATED_SPACE 13
int usedSlots = 0;

void exitLED(int isRED) {
 if (isRED) {
  analogWrite(GREEN_LED_EXIT, OFF);
  delay(LED_DELAY);
  analogWrite(RED_LED_EXIT, ON);
 } else {
  analogWrite(RED_LED_EXIT, OFF);
  delay(LED_DELAY);
  analogWrite(GREEN_LED_EXIT, ON);
 }
}

void enterLED(int isRED) {
 if (isRED) {
  analogWrite(GREEN_LED_ENTER, OFF);
  delay(LED_DELAY);
  analogWrite(RED_LED_ENTER, ON);
 } else {
  analogWrite(RED_LED_ENTER, OFF);
  delay(LED_DELAY);
  analogWrite(GREEN_LED_ENTER, ON);
 }
}

int isPushed(int button) {
 return digitalRead(button);
}

void initialPrint() {
```

```
 lcd.print("Smart Parking");
 lcd.setCursor(6, 1);
 lcd.print("By Team 13");

 delay(1000);

 lcd.setCursor(0, 0);
 lcd.print("Mohini Sarkar          ");
 lcd.setCursor(0, 1);
 lcd.print("110086149              ");

 delay(1000);
 lcd.setCursor(0, 0);
 lcd.print("Umang Shrestha         ");
 lcd.setCursor(0, 1);
 lcd.print("110071431              ");
 delay(1000);
}

void setGate(int pos) {
 Serial.begin(9600);
 servo.write(pos);
 delay(10);
 servo.write(pos);
}

void intermediatePrint() {
 lcd.setCursor(0, 1);
 lcd.print("Available:             ");
 lcd.setCursor(14, 1);
 lcd.print(MAX_ALLOCATED_SPACE - usedSlots);
}

void setup() {
```

```
 // ENTER LED
 pinMode(RED_LED_ENTER, OUTPUT);
 pinMode(GREEN_LED_ENTER, OUTPUT);
 //  BUTTON
 pinMode(BUTTON_ENTER, INPUT);
 pinMode(BUTTON_EXIT, INPUT);
 // EXIT LED
 pinMode(RED_LED_EXIT, OUTPUT);
 pinMode(GREEN_LED_EXIT, OUTPUT);
 // setting default color RED
 enterLED(RED);
 exitLED(RED);
 // LCD
 lcd.begin(16, 2);
 // SERVO
 servo.attach(SERVO_PIN);
 setGate(setGate_CLOSE);
 initialPrint();
}

void loop() {
 lcd.setCursor(0, 0);
 lcd.print("Welcome!!!               ");
 intermediatePrint();

 if (isPushed(BUTTON_ENTER)) {
   if (usedSlots < MAX_ALLOCATED_SPACE) {
            usedSlots++;
            setGate(setGate_OPEN);
            enterLED(GREEN);
            lcd.setCursor(0, 0);
            lcd.print("PLEASE ENTER!!!    ");
            intermediatePrint();
            delay(1000);
```

```
            setGate(setGate_CLOSE);
            enterLED(RED);
  } else {
            lcd.setCursor(0, 0);
            lcd.print("PLZ Come Later:       ");
            intermediatePrint();
  }
  delay(1000);

 } else if (isPushed(BUTTON_EXIT)) {
  if (usedSlots > 0) {
            usedSlots--;
            setGate(setGate_OPEN);
            exitLED(GREEN);
            lcd.setCursor(0, 0);
            lcd.print("PLEASE EXIT!!!       ");
            intermediatePrint();
            delay(1000);
            setGate(setGate_CLOSE);
            exitLED(RED);
  } else {
            lcd.setCursor(0, 0);
            lcd.print("No Car Inside!!!         ");
            intermediatePrint();
  }
  delay(1000);
 }
 delay(100);
}
```

4.0 Timeline

The project Timeline is divided into 3 major milestones to achieve. All the details about the milestones achieved so far and the planned checkpoints along with its estimated completion time is indicated in the below Gantt Chart.

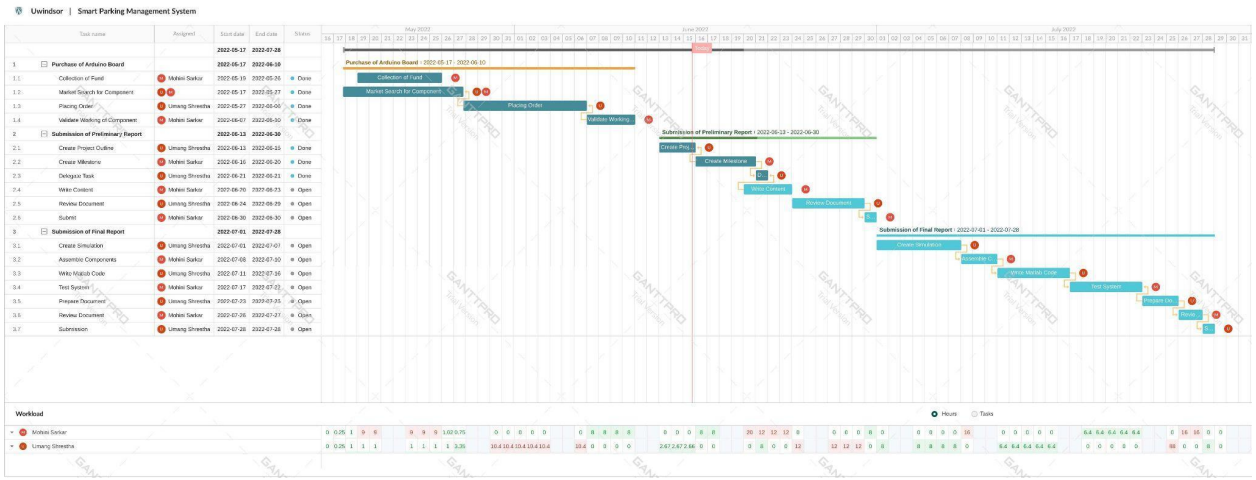

Figure 11: Gantt Chart

4.0.1 Limitation/Risks

1) Electrical components are fragile and are easily damaged by using incorrect value of resistance.
2) Components are not easily available in the market. In case of any damage to any of the components, significant investment of time and money is required.

5.0 CONCLUSION

We successfully designed, developed and tested all the project requirements pertaining to Smart Parking Management system. Software simulation followed by hardware implementation on Arduino kit was executed within the planned timeline. This project will prove to be helpful in mitigating all the drawbacks of the traditional manual parking system.

Reference

[1] A. Kurniawan, Ed., “Arduino Programming using MATLAB,” 2015. https://www.researchgate.net/profile/Abdullah-Mamun-50/post/how_to_implement_a_matlab_programme_in_arduino/attachment/5ab65a794cde266d5893836b/AS%3A607707442724864%401521900153103/download/Arduino-Programming-Using-MATLAB-Agus-Kurniawan.pdf (accessed Jun. 10, 2022).

[2] “MATLAB Support Package for Arduino Hardware User’s Guide,” *dokumen.tips*. https://dokumen.tips/documents/matlab-support-package-for-arduino-hardware-users-guide.html (accessed Jun. 11, 2022).

[3] *smart parking system*. 2021. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=SBPpYi1n&id=67EC76595387740BFAF37C64040ED1278EB0D39B&thid=OIP.SBPpYi1ntG287ureZYAw-wHaFj&mediaurl=https%3A%2F%2Fsc01.alicdn.com%2Fkf%2FHTB1dQ20IFXXXXbqaXXXq6xXFXXX9%2F200167884%2FHTB1dQ20IFXXXXbqaXXXq6xXFXXX9.jpg&cdnurl=https%3A%2F%2Fth.bing.com%2Fth%2Fid%2FR.4813e9622d67b46dbceeeade658030fb%3Frik%3Dm9OwjifRDgRkfA%26pid%3DImgRaw%26r%3D0&exph=640&expw=853&q=smart+parking+system&simid=608022710690195692&form=IRPRST&ck=FF16C272DBD7ED31353E6454D5C1C00D&selectedindex=45&ajaxhist=0&ajaxserp=0&vt=3&sim=1&cal=0.21222322&cab=0.81547266&cat=0.57978725&car=0.5189644&hotspot=1

[4] *Arduino Uno*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=tV4VNNuB&id=86BA51BF4FBED1B8CA62A36E58B623F4B0CAF5A6&thid=OIP.tV4VNNuBfmarY2kfEuDjBAHaFT&mediaurl=https%3a%2f%2fwolles-elektronikkiste.de%2fwp-content%2fuploads%2f2019%2f07%2fUNO_2.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.b55e1534db817e66ab63691f12e0e304%3frik%3dpvXKsPQjtlhuow%26pid%3dImgRaw%26r%3d0&exph=1073&expw=1500&q=arduino+uno&simid=608005749860949732&FORM=IRPRST&ck=25202905DBAFDD4F92952B9DA3540928&selectedIndex=8&ajaxhist=0&ajaxserp=0

[5] *Breadboard*. Accessed: Jun. 21, 2022. [Online]. Available: [1]Arduino Uno. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=tV4VNNuB&id=86BA51BF4FBED1B8CA62A36E58B623F4B0CAF5A6&thid=OIP.tV4VNNuBfmarY2kfEuDjBAHaFT&mediaurl=https%3a%2f%2fwolles-elektronikkiste.de%2fwp-content%2fuploads%2f2019%2f07%2fUNO_2.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.b55e1534db817e66ab63691f12e0e304%3frik%3dpvXKsPQjtlhuow%26pid%3dImgRaw%26r%3d0&exph=1073&expw=1500&q=arduino+uno&simid=608005749860949732&FORM=IRPRST&ck=25202905DBAFDD4F92952B9DA3540928&selectedIndex=8&ajaxhist=0&ajaxserp=0

[6] *Servo Motor*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=Dycntwa2&id=4F06924D64991FDD72D681AFF07DBAC6194692DF&thid=OIP.Dycntwa2ygfjxEEs24eMEQAAAA&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.0f2727b706b6ca07e3c4412cdb878c11%3frik%3d35JGGca6ffCvgQ%26riu%3dhttp%253a%252f%252fwww.ozeki.hu%252fattachments%252f3034%252fservo-motor-wiring-diagram.png%26ehk%3ds5IBqbJXAscXbwX3t8CJhPGpWRZrdGgxa8pL1cI1tS4%253d%26risl%3d%26pid%3dImgRaw%26r%3d0%26sres%3d1%26sresct%3d1%26srh%3d742%26srw%3d1300&exph=270&expw=473&q=servo+motor+arduino+code&simid=608007940295032999&FORM=IRPRST&ck=0B4049D2D8328AA18E501B1EB6DACBA7&selectedIndex=50&ajaxhist=0&ajaxserp=0

[7] *LED*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=E1WoXhq6&id=AC6FCCF3DA508AC936A030D90D35926BCC3CDD12&thid=OIP.E1WoXhq6fwZ3JwxZW-q0JQHaD_&mediaurl=https%3a%2f%2fcdn.shopify.com%2fs%2ffiles%2f1%2f0254%2f1191%2f1743%2farticles%2fLEDResistor_800x.jpg%3fv%3d1600352226&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.1355a85e1aba7f0677270c595beab425%3frik%3dEt08zGuSNQ3ZMA%26pid%3dImgRaw%26r%3d0%26sres%3d1%26sresct%3d1%26srh%3d702%26srw%3d1300&exph=432&expw=800&q=led+and+resistors&simid=608022285486679250&FORM=IRPRST&ck=C210EA85D791DA15AA672563001BEACB&selectedIndex=21&ajaxhist=0&ajaxserp=0

[8] *LCD Display Arduino Code*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=nB6YT8d8&id=25BBA09E490A9E362B24172FC13802B02EC85864&thid=OIP.nB6YT8d8EWnlLXO1v_6PfgHaC-&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.9c1e984fc77c1169e52d73b5bffe8f7e%3frik%3dZFjILrACOMEvFw%26riu%3dhttp%253a%252f%252fwww.meccanismocomplesso.org%252fwp-content%252fuploads%252f2017%252f01%252fFritzing-sketch-for-LCD1602-and-Arduino.jpg%26ehk%3dzLeMgBB%252fDRLV7p8leD2cn66XQxylt2LnbR2Nad0dqjw%253d%26risl%3d%26pid%3dImgRaw%26r%3d0&exph=527&expw=1308&q=lcd+display+arduino+code&simid=608050550664876916&FORM=IRPRST&ck=15AC6B4B2D1A5D9CAAF2D72BEA2F7D00&selectedIndex=4&ajaxhist=0&ajaxserp=0

[9] *Press Button Switch*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=nzdWeGS%2f&id=5048644E5E6F17BEC92294D964D30E2921E15D58&thid=OIP.nzdWeGS_6ziuMnhZScYeeQHaHa&mediaurl=https%3a%2f%2fwww.sparkpcb.com%2fimages%2fwatermarked%2f1%2fdetailed%2f3%2f12mm_NO_Red_Momentary_Push_Button_Switch1.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.9f37567864bfeb38ae32785949c61e79%3frik%3dWF3hISkO02TZlA%26pid%3dImgRaw%26r%3d0&exph=700&expw=700&q=press+button+switch&simid=608032498920066323&FORM=IRPRST&ck=6FA5C208D7C41BABEAA6E19FE27B95B4&selectedIndex=30&ajaxhist=0&ajaxserp=0

[10] *Wires on Breadboard*. Accessed: Jun. 21, 2022. [Online]. Available: https://www.bing.com/images/search?view=detailV2&ccid=zu2lWZMj&id=F6020483A458B3837E3E81BB55C97D5EF63ED4E2&thid=OIP.zu2lWZMjeGwoz1-3vq85ZQHaHa&mediaurl=https%3A%2F%2Fmikroelectron.com%2FProImg%2FX4%2Fced759cc-44c8-4197-8fd9-b017ba24ae63.jpg&cdnurl=https%3A%2F%2Fth.bing.com%2Fth%2Fid%2FR.ceeda5599323786c28cf5fb7beaf3965%3Frik%3D4tQ%252b9l59yVW7gQ%26pid%3DImgRaw%26r%3D0&exph=600&expw=600&q=wires+on+bradboard&simid=608019961910732446&form=IRPRST&ck=F8EE81752221AF06393B11FB7CD0D18E&selectedindex=1&ajaxhist=0&ajaxserp=0&pivotparams=insightsToken%3Dccid_j9FB7F%252FP*cp_83FCBFD6977A87BB30DB1D65A8794AE0*mid_94A0FC72C708BC26C8CAE25514E366DD5F1489CE*simid_608056078289346993*thid_OIP.j9FB7F!_PlvaPVZ3Q12OX6AHaHa&vt=0&sim=11&iss=VSI&ajaxhist=0&ajaxserp=0