WEB-BASED CAR PARKING SLOT MONITORING SYSTEM

CH’NG SHU FEN

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering (Honours) Electronic Engineering

Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

April 2019

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ___________________________

Name : CH'NG SHU FEN

ID No. : 14AGB01950

Date : 16 APRIL 2019

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"WEB-BASED CAR PARKING SLOT MONITORING SYSTEM"** was prepared by **CH'NG SHU FEN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________________

Supervisor : DR. TOH PEK LAN

Date : 16 APRIL 2019

Signature : ____________________________

Co-Supervisor : DR. YEW TIONG KEAT

Date : 16 APRIL 2019

Specially dedicated to

my beloved father and mother, for their never-ending supports and love.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr Toh Pek Lan for her invaluable advice, guidance and her enormous patience throughout the development of research. Without her supervision, I would not complete this project successfully in the time frame given. In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement. I would also express gratitude to Microsoft Azure for providing a free platform, so that I can complete my project. Finally, I would like to express my gratitude towards my friend for providing me technical support and guidance during the progress of the project.

WEB-BASED CAR PARKING SLOT MONITORING SYSTEM

ABSTRACT

As town modernisation progress, the number of vehicles increases rapidly. Parking is a major problem especially in densely populated areas. Normally, places such as function hall, university, multiplexes and large industries, we need to search for the line that is empty to park our vehicle. Most of the car parking system face problems like wastage of fuel. This also caused us to consume a lot of time finding one car parking slot. To overcome this difficulty, there is need of an advanced car parking slot monitoring system. Thus, a more advanced car parking system is needed to monitor the empty slot of car parking. This carpark slot monitoring system can reduce the time taken used to find the available space by using microcontroller and Infrared sensors (IR) installed at each of the parking slot. The microcontroller serves as a programming tool to run the whole operation, and to make it a system that can meet market needs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

API	Application Programming Interface
CCTV	closed-circuit television
HTML	Hypertext Markup Language
IoT	internet of things
IPS	intelligent parking system
IR	infrared
JSON	JavaScript Object Notation
LCD	liquid-crystal display
PWM	pulse width modulation
QR	quick response
SCL	clock signal
SDA	data signal
SQL	structured query language
SPS	smart parking system
°	degree

CHAPTER 1

INTRODUCTION

1.1 Background

Wireless communication was developed over the years and is introduced in the 19th century. Wireless communication is an important medium in transmitting the information from one device to other devices. Wireless communication is crucial to convey the information rapidly to consumers. This also aids in complete the jobs anyplace on time and can improves the production, as user do not need to carry cables or wires whenever they go. Moreover, the urgent situation can be notified with the support of wireless communications immediately (Mary, 2018). Innovations in technology had enabled the efficiency of wireless driven the concept of Internet of Things. The core impression of this thought is the omnipresent around us with variety of objects or things that are connected to the Internet such as laptops, mobile phones, and daily used products, such as smart sensors. It is predicted that between the year of 2015 to 2020, the mobile data traffic will be experience 8-fold growth. Moreover, it is also estimated that the data emanating from mobile devices will reached the amount of 11.6 billion by the year of 2020 (Cisco.com, 2016). This is mainly caused by the aggressive growth and wide acceptance of the user around the world. These are originated by the integrated of technologies and communications solutions, such as artificial intelligence for smart objects, wired and wireless sensor and actuator networks, identification technologies and next generation communication protocols. The internet shortened the world by providing effective communication among devices around the world. Therefore, exchange of valuable info across the world, with the help of human involvement was possible. Internet of Things (IoT) is an advanced technology and it can be simply integrated into almost all the electrical or electronic devices in order to transform them into smarter devices. Many industries prefer IoT as it offers a variety of applications for the user, such as cloud data storage, analysis of collected data, real-time analytics, remote notifications and trigger an action based on a value. It also can be applied in industries that prefer automation, energy efficient systems, smart devices. The IoT system works on complex procedures and modules, thus it is not easy to learn. The exact value of the IoT is that it allows connected devices to collect users' data. For example, a hospital with data with connected devices is very important, as the data composed from those devices' yields information on the status of patients. Thus, the devices can run analytics on the various monitoring machine, serving the hospital to run as effectively as possible. The collection of statistics from devices will let clients, industries, and even whole connected cities to run more proficiently. Hence, the data collection is a key factor in the understanding of the Internet of Things and is also the end goal of IoT. Smart devices will develop in a more complicated way into our lives in coming (Medium, 2017).

1.2 Problem Statements

In today's busy working environment, drivers face difficulty in looking for vacancy slot in car park especially in peak hour. In many places, especially around universities, city centres, shopping complexes and other busy working environments, finding empty parking has become one of the stressful stuffs for the individuals who drive. The traditional method of finding parking, which is by using our naked eyes has several frustrating situations. In our daily lives, we can observe that where a driver is in the car or is walking towards a car, the other drivers who are finding parking frequently show some sign just trying to ask the other whether they are exiting the car park. Although this action would help at most of the times, but this will lead to inconvenience to other drivers (MWEBAZE, 2009). A smart car parking mobile application would help drivers to find a vacant spot by just monitoring on the system. By having a car parking system, it will reduce traffic congestion and air pollution. The environment will also be less polluted. Users will be more convenience to find car parking and this save their time. The most important is that the driver can find the nearest car parking through the mobile application without wasting time.

1.3 Aims and Objectives

The aim of this project is to provide a platform for user to find empty carpark slot easier and efficiently using web-based carpark slot monitoring system. The objectives of the study are shown as following:

i) To develop an intelligent and user-friendly carpark system, which decreases the manpower and traffic congestion.

ii) To overcome the problem of finding a parking space that consumes unnecessary time.

iii) To apply a programming language into the system.

iv) To display the number of available carpark slot on LCD screen and webpage.

v) To evaluate the carpark slot monitoring system for performance and accuracy.

CHAPTER 2

LITERATURE REVIEW

2.1 Background of Study

Nowadays, many people are facing problems, such as wastage of time and fuel to find available parking slot when we want to park our car in shopping complex. In most of the parking system, it is tough and time consuming to find out the empty space for large parking. Sometimes, it will cause late in important works or meeting. To solve these problems, we need to introduce a carpark slot monitoring system, so that users can find carpark slot easier.

2.1.1 Smart Parking System

Internet of Things (IoT) plays a vital role in connecting the surrounding environment devices, such as mobile application and web application to the network and can access the internet in anywhere. In an urban area, people are generally facing difficulties in finding empty slots. A Smart Parking System (SPS) is developed to enable users to find the nearest parking slot in that perspective parking area (SR, 2015; Parasher *et al*, 2017). The main idea is to reduce the time taken to find the availability of parking slot and avoid unnecessary travelling through the parking lots. Hence, this system will reduce fuel consumption and reduces carbon gases released from vehicles in the atmosphere. A system is developed to allow the users to have a look on the status of the parking spaces available before their trip. This is a challenge to use the existing resources to decrease the time taken for finding empty parking slot in a traffic congestion city within optimum level. Embedded system such as Raspberry Pi and Arduino modules can be used to develop IoT applications. In addition, parking system with sensors can be used to gather the info accurately, but the sensors are quite expensive. Hence, a system with better performance will be developed (Smart Car Parking System using Arduino UNO, 2017).

2.2 Literature Studies

2.2.1 A Street Parking System using Wireless Sensor Networks Presented by Zhang *et al.* (2013)

The key objective of this work is to develop a smart parking system (SPS) by using wireless sensor networks. In this paper, the proposed SPS includes routers, base station, routers, remote server and sensor nodes, as shown in Figure 2.1. The sensor nodes are

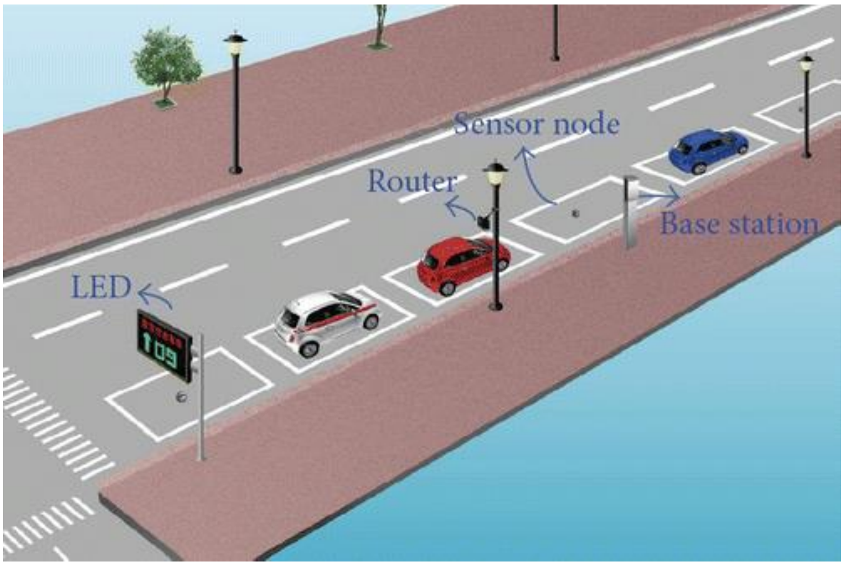
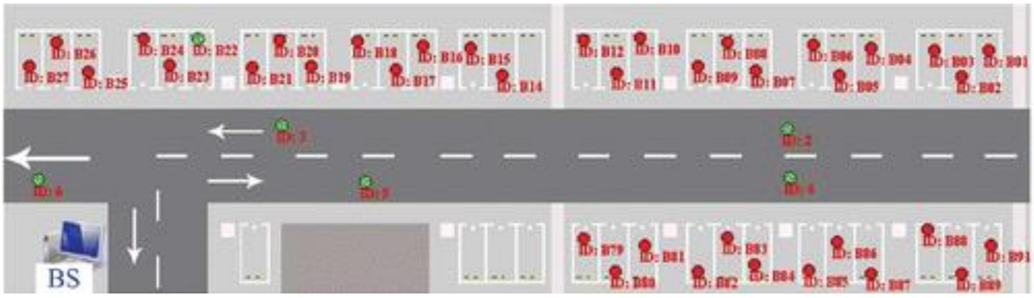


Figure 2.1: The Smart Parking System using Sensor (Zhang *et al.*, 2013).

positioned alongside the street and each node is placed on the centre floor of each car parking space. When the node sensed there is a car entering or leaving the parking space, the router receives a message and forward it to the base station. In addition, the data collected from different nodes will be combined, and the parking direction evidence will be displayed on LED board and remote server. The sensor node consists of HMC5883L magnetic sensor. The devices with magnetic sensors are fixed at the centre of the parking spaces. The routers furnished with a solar panel is used to forward the parking message, which the router is placed on the street light. Java language and MySQL database were developed as server system in this paper. By using the smart parking system using sensor, users can recognise the parking space that is empty or occupied. The occupied period of each parking space will also be displayed using the graphical client interface. The nodes which ID signed as integer are router nodes while nodes ID is started with “B,” such as B20, are shown in Figure 2.2(a). The sensor node with green dot shows the parking space is empty, whereas, a red dot indicates the parking space is unavailable. The blue lines as shown in Figure 2.2(b) with the arrow define the wireless network topology.



(a)

(b)

Figure 2.2: A Management System: (a) Sensor Nodes deployed on the Parking Spaces, and (b) Topology of the Network (Zhang *et al.*, 2013).

2.2.2 Designing a Smart Car Parking System (PoC) Prototype Utilizing CCTV Nodes: A Vision of an IoT Parking System via UCD Process studied by Fraifer and Fernström (2017)

The objective of this study is to develop an IoT parking system prototype based on computer vision, as shown in Figure 2.3. This study is carried out in parallel with technical investigations, such as integration of system components and incremental development. They had discovered a smart parking system based on CCTVs/Web-cams (nodes), which can allocate drivers with appropriate and straightforward parking service information. First of all, drivers need to have the support of realtime navigation support, such as using Google Maps. Next, a parking information service for the driver and the parking owner need to be provided. Thirdly, CCTV cameras (nodes) are used for safety purposes also serve as budgetary option. The core purpose of the system is to accomplish and gather information for the real-time parking status and assist drivers to be directed straight to empty parking spaces. In this study, the author uses a LAMP server as the open-source platform, MySQL as the relational database management system, Linux as the operating system and Apache as the Web server. The core technique of the system is to monitor the system through IoT cams. At the same time,

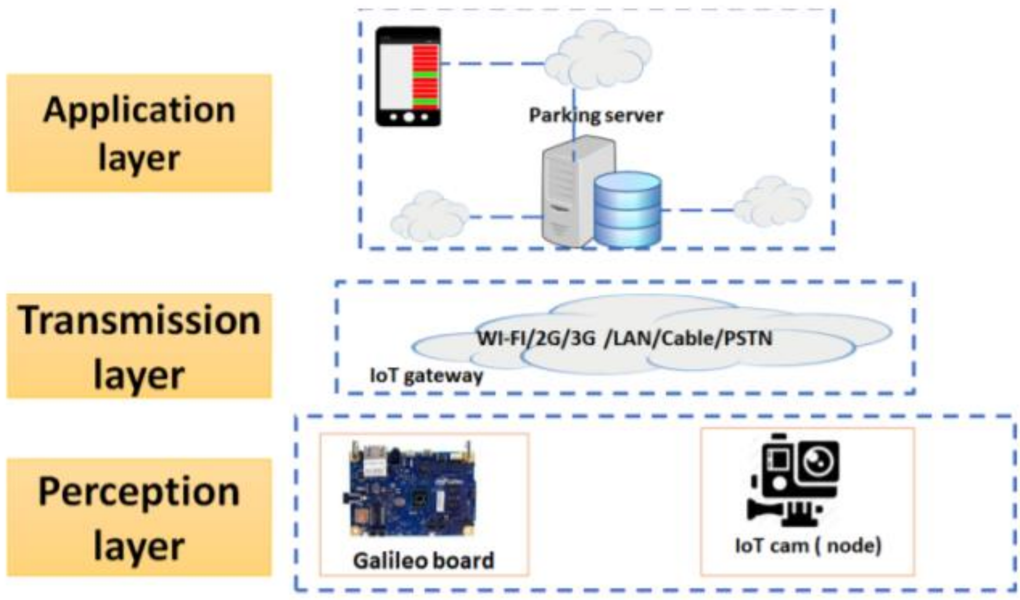


Figure 2.3: Smart Parking IoT Application (Fraifer and Fernström, 2017).

the video is processed by examining the live streaming of statistics and the changes are saved in a MySQL database. The latest car parking data and services can be sent to the user in an easy and friendly method. This smart parking system needs the collaboration of several users finding for parking slots within the same time. However, there is no barrier to the parking spots. Moreover, there are some circumstances and limitation in this system. The system can be operating well in day time but is not intended for use at night. For instance, if night-time operation is required, night vision cameras or artificial illumination would have to be practiced.

2.2.3 Intelligent Parking System using Android Application by Anitha *et al* (2017)

The objective of this paper is to offer intelligent, universal and user-friendly automated parking system application. The aim of this project is to reduce the user's time, escape from traffic congestion in city areas and can find available parking slots within restricted area. In this paper, the "Intelligent Parking System (IPS)" is applied using the Android operating system. First, the user may request the Parking Control Unit to get the current situation of existing parking spaces. Once user appeal for the parking, all the free slots in the carpark will be shown to the user. Next, the users may book the parking slot and continue to payment if they had confirmed the parking spaces. After the payment, the users will be directed to the parking spaces by following the path. Once the user successfully parks at the parking spaces given, the slots will be updated immediately in the administrator's database. Therefore, the period to discover for a vacant parking slot is diminished. The core duty of the Intelligent Parking System (IPS) is to aid the user to find the parking spaces, where there are still have free slots. Consequently, the user's energy and time of searching a parking slot will be reduced. There are three main modules in this Intelligent Parking System. The modules are known as user module, administrator module and booking module, as shown in Figure 2.4. By using this module, users can register, log in, booking and make the payment whenever they want. If there are new user who wants to register for this system, they just need to register the application form by filling the profile information. Once the registration is done, user may login as usual. At this time, they may browse the parking spaces and proceed to booking the parking spaces by making online payment. Furthermore, Administrator Module works as the backend for supervision the database and accomplishes various activities on it. The administrator is responsible for keeping all the user's details in the database safely. If there is any amendment on booking or payment, users need to adjust

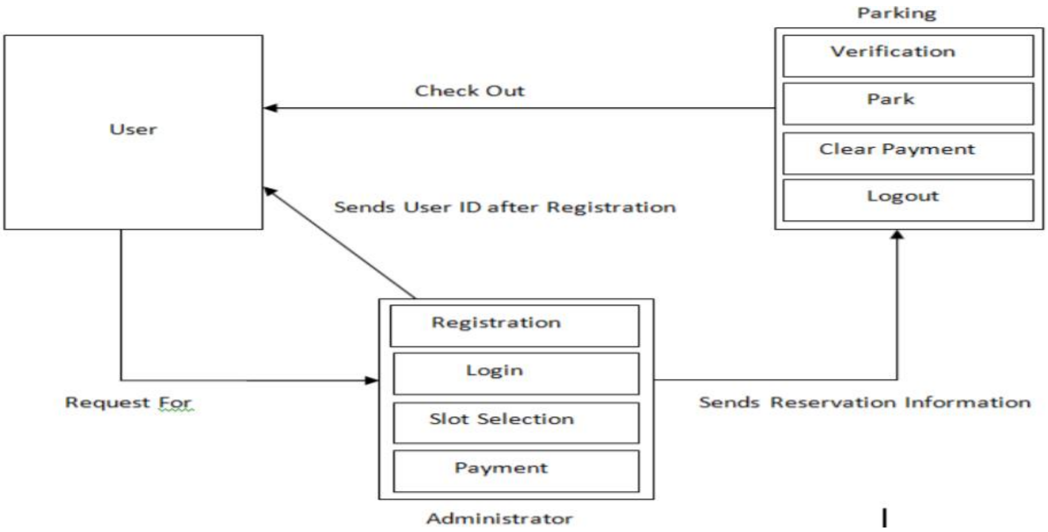


Figure 2.4: Architecture of Intelligent Parking System (IPS) User Module (Anitha *et al.*, 2017).

it with administrator. Next, the Booking Modules responsible for parking spaces reservation. When users are ready for reservation, the available slot, cost to reserve the spaces and required processing are done by this booking module.

CHAPTER 3

METHODOLOGY

3.1 Design Architecture in Detailed

Microcontroller used in this project are NodeMCU and Arduino UNO. The number of carpark slot is set as 12 in my project. In this carpark slot monitoring system, the NodeMCU V3 board and Arduino UNO are used to control the infrared (IR) sensor modules, Liquid-crystal display (LCD) and servo motor. NodeMCU has a built-in ESP8266 inside, which is an electronic device that allows us to connect to the internet. NodeMCU V3 board and smartphone are required to connect to the hotspot network in this project in order to receive data. First, a powerbank is used to power up the NodeMCU V3 board via micro USB cable. The NodeMCU V3 board supplies 3.0V to infrared sensor. There are two servo motors placed at the entrance and exit of carpark respectively. A sensor is also placed at the entrance and exit of carpark for counting the number of parking slot. When the NodeMCU V3 board receives the signal from IR sensor at the entrance and exit, it controls the movement of servo motor. At the same time, the parking slot available will also display on webpage. When the number of cars is less than 12, the cars will be allowed to enter until it reaches the maximum value, which is 12 cars. When the number of cars in carpark is maximum, no car will be allowed to enter the car park. The number of carpark slot available will be shown on the LCD display. User may also scan QR code before entering the carpark, so that they will know the exact slot that is empty and find the empty slot in the shortest time frame. The web interface in the browser can act as a monitoring system. Users can access the webpage via smartphone, personal computer or any device with browser. Figure 3.1 shows the basic block diagram of a car parking system.

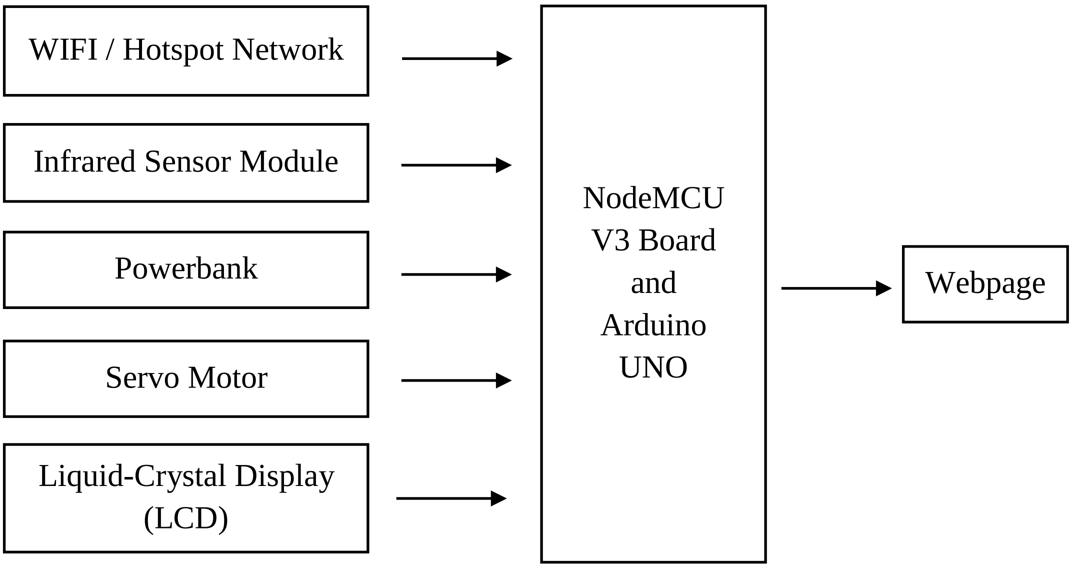


Figure 3.1: Block Diagram of a Car Parking System.

3.2 Project Management

The project schedules are shown in Tables 3.1 and 3.2, respectively.

Table 3.1: Gantt Chart for FYP 1.

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Discussion with supervisor														
Project title selection														
Research														
Build Prototype														
FYP 1														
Introduction														
Problem statement														
Objective														
Literature review														
Methodology														
Presentation														
FYP 1 presentation														

Table 3.2: Gantt Chart for FYP 2.

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Build Prototype														
FYP 2														
Results														
Conclusion														
Presentation														
Poster Presentation														
FYP 2 Presentation														

3.3 Hardware used in a Car Parking System

3.3.1 NodeMCU V3 based ESP8266 Module Board

The main reason of choosing NodeMCU V3 based ESP8266 board in this project is due to its open source, Arduino-like hardware, lower cost, lighter weight, small size, 3.3V operated, can be powered by USB, ESP8266 built-in WiFi and contains GPIO pins. Arduino module is a great choice to the project, but these modules do not feature a built-in WiFi capability. However, NodeMCU V3 based ESP8266 module board, as show in Figure 3.2, incorporates a built-in WiFi support, giving an easy pathway to design IoT applications. The features as stated above make the NodeMCU enormously powerful tool for Wifi networking. NodeMCU can be used to host a web server, used as access point, or link to internet to upload or fetch data. It is an open-source firmware that is very important in designing IoT product using a few Lua scripts. The components such as infrared sensor modules can be connected externally on the GPIO pins of NodeMCU board (Aqeel and G+, 2019). The pin layout of NodeMVU V3 board is shown in Figure3.3.

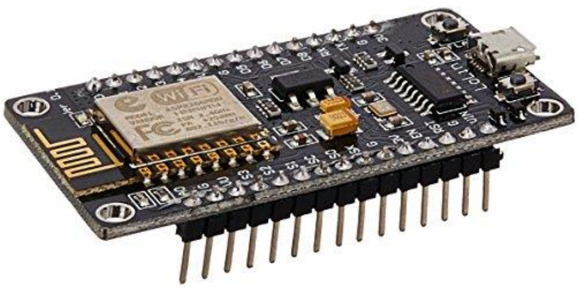


Figure 3.2: NodeMCU V3 based ESP8266 Module Board (Future Electronics Egypt, 2019).

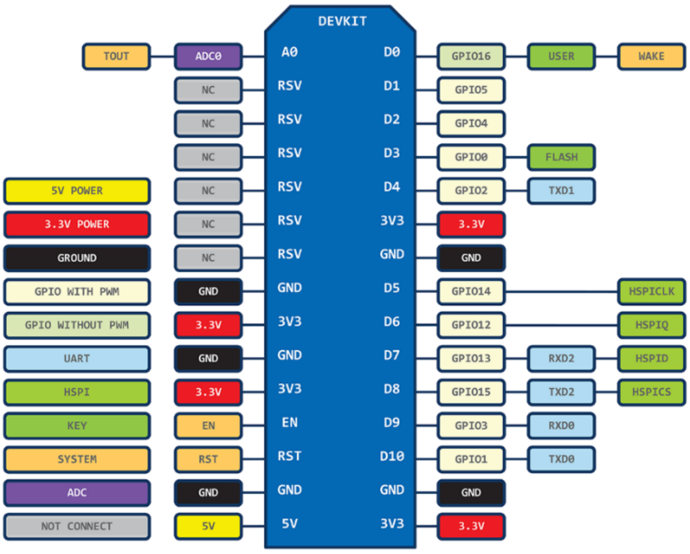


Figure 3.3: Pin Layout of NodeMCU V3 based ESP8266 Module Board (My2cents, 2019).

3.3.2 Infrared Sensor

The main reason to choose infrared IR sensor module in this project is that it has an excessive ability of the ambient light and having a pair of infrared transmitter and receiver. Sensors are electronic devices, which are used to sense the changes occur in the surrounding environment. Figure 3.4 shows a pair of infrared transmitter and receiver. Infrared transmitter is a light emitting diode (LED), which emits infrared (IR) radiations. Moreover, infrared receivers sense the radiation from an IR transmitter. Next, IR receivers come in the form of photodiodes and phototransistors. It is different from normal LED as infrared photodiodes only detect infrared radiation. The working principle of IR sensor, which is Object Detection Sensor is described by Figure 3.5. The transmitter is used to emit radiation of required wavelength. When the radiation from IR sensor reaches the car, it reflects back and is detected by the IR receiver. In initial, the power LED will be turned on when there is power supply. When there is no car parked at the slot, there is no obstacle and hence, the obstacle LED will not be turned on. When there is a car parked at the slot, there is obstacle detected, thus the obstacle LED is illuminated. For the detection distance adjust, the distance is set as 5 cm in this project. For different light intensity at the surrounding, different distance need to be adjusted, as the light intensity will affect the obstacle distance. The distance can be increased by turning the distance adjust in clockwise direction, while anticlockwise direction is adjusted in order to decrease the detection distance.

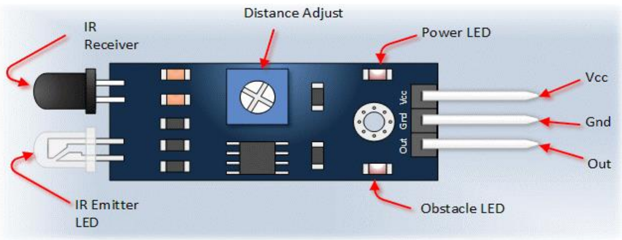


Figure 3.4: Diagram of Infrared Sensor (Henry's Bench, 2019).

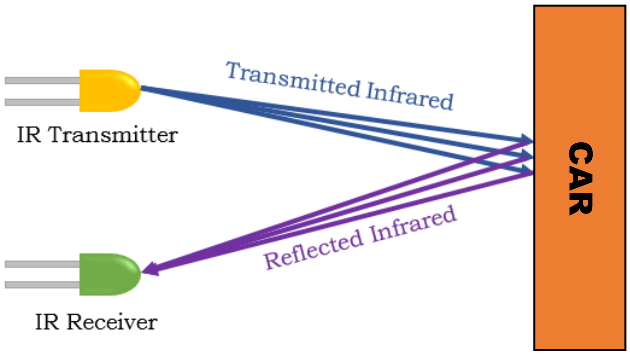


Figure 3.5: Infrared Sensor Working Principle (Electricalfundablog.com, 2019).

3.3.3 SG90 Servo Motor

The main reason to choose SG90 Servo Motor is that it is lightweight and tiny with high output power. Servo motor as shown in Figure 3.6 consists of a DC motor, potentiometer and control panel. The resistance on the potentiometer changes as the motor rotates. Therefore, the control panel can regulate the rotation, speed, acceleration and direction of rotation precisely. It is controlled by sending Pulse Width Modulation (PWM) signals via the control wire, which is known as the output wire. In this project, two SG90 Servo Motors are used. One of the servo motors is used for entrance barrier, while another servo motor is used as exit barrier. The maximum angle that can be rotate by Servo Motor SG90 is 180°. It has sufficient angle for this project, due to the opening of the carpark entrance and exit of carpark entrance requires a movement of 90°. Input servo motor action is depending on the IR sensor placed at the carpark entrance and carpark exit. Servo motor will move 90° to raise the barrier if there is car detected at the carpark entrance. If there is a car detected at the entrance, but the number of available parking slot is empty, the servo motor will not take any action. In the case of output servo, the action will depend on the IR sensor placed at the exit. If the IR sensor detects the car at the exit, servo motor will move 90° to raise the barrier.

Figure 3.6: SG90 Servo Motor (electricalfundablog.com, 2019).

3.3.4 16×2 I2C LCD Display

16×2 I2C LCD display is a combination of two modules, which are an I2C module and 16×2 LCD module. This Liquid Crystal Liquid (LCD) is used as it consists of 2 lines and 16 characters for each line, which can display a total of 32 characters on LCD display screen, as shown in Figure 3.7. The I2C module is built-in with a 16×2 LCD module which helps to save the usage of GPIO pins from Arduino UNO. There is a total of four input signals, which are input, ground, clock signal (SCL) and data signal (SDA). The contrast of the backlight can be adjusted by gently twisting the mini trim potentiometer. It is used to display all the messages including the basic information of the number of slots available in carpark.

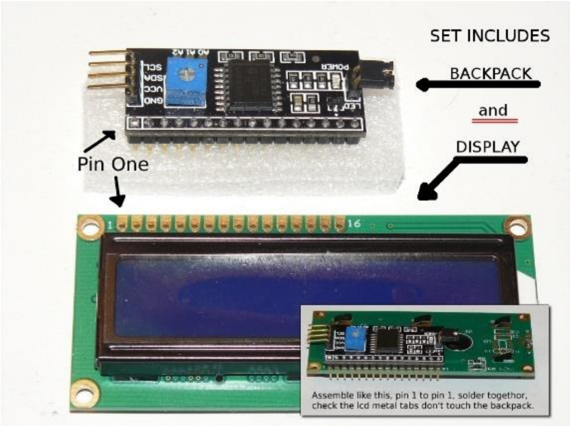


Figure 3.7: 16 × 2 I2C LCD Display (Sparks.gogo.co.nz, 2019).

3.4 Arduino IDE (Integrated Development Environment)

3.4.1 Defining NodeMCU Pins

There are two NodeMCUs used in this project. Table 3.3 shows the electronic devices and its attached pin. D0, D1, D2, D3, D4, and D5 are attached to slots A to slot H of first NodeMCU, while D0, D1, D2, D3, D4, and D5 are attached to slots H to slot L of second NodeMCU respectively. This is because NodeMCU has only 9 pins that can be used in Arduino IDE, which is lesser output pin compared to Arduino UNO. However, it has a WiFi connection, which enable it to update the latest information on webpage.

Table 3.3: Electronic Devices and Pin attached to NodeMCU.

Electronic devices	Pin Attached
Slot A	D0 (NodeMCU 1)
Slot B	D1 (NodeMCU 1)
Slot C	D2 (NodeMCU 1)
Slot D	D3 (NodeMCU 1)
Slot E	D4 (NodeMCU 1)
Slot F	D5 (NodeMCU 1)

Slot G	D0 (NodeMCU 2)
Slot H	D1 (NodeMCU 2)
Slot I	D2 (NodeMCU 2)
Slot J	D3 (NodeMCU 2)
Slot K	D4 (NodeMCU 2)
Slot L	D5 (NodeMCU 2)

3.4.2 WiFi Setup Function

The WiFi setup function of the NodeMCU will be executed once it is powered on. The Wifi.begin() initializes the WiFi's library network settings and provide the current status. In the first line of sketch, *library <ESP8266WiFi.h>* is included as it provides ESP8266 specific WiFi routines calling to network. Actual connection to WiFi is initialized by calling *WiFi.begin(WIFI_SSID, WIFI_PASSWORD)*. The process of connection may take a few seconds and we can check for this complete in the following loop by the *while()* loop as shown in Figure 3.8. In this case, the while loop will continue looping while *WiFi.status()* is other than *WL_CONNECTED*. The status will change to *WL_connected* when it exits the loop. It is crucial to connect the network as to display the realtime status of carpark on webpage.

```
#include <ESP8266WiFi.h>                    //library for NodeMCU
void setup_wifi() {
   delay(100);
  // We start by connecting to a WiFi network
    Serial.print("Connecting to ");
    Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("----------------------");

}
```

Figure 3.8: Nodemcu Void Setup Wifi Code.

3.4.3 Defining Arduino UNO pins

Table 3.4 lists electronic devices and its attached pin to Arduino UNO. Electronic devices connected to components, such as servo motor, LCD display and IR sensor. IR sensor at the entrance and exit are attached to pin 2 and pin 4 of Arduino UNO respectively. Servo motor at the entrance and exit are attached to pin 3 and pin 5 of Arduino UNO respectively. SDA of LCD display is attached to A4 pin of Arduino UNO, while SCL of LCD display is attached to A5 pin of the Arduino UNO.

Table 3.4: Electronic Devices and its Pin attached to Arduino Uno.

Electronic devices	Pin Attached
IR Sensor (Entrance)	2
Servo Motor (Entrance)	3
IR Sensor (Exit)	4
Servo Motor (Exit)	5
LCD Display SDA	A4
LCD Display SCL	A5

3.4.4 Setting up Servo Motor

The servo motor will be rotated to 90° clockwise about the origin when the car enters or exit the carpark. The *delay()* function is very important for code writing. Without the *delay()* function, most of the electronic modules will not work as expected. This is because the Arduino code is being executed millions of times in 1 second, the electronic module, such as a servo motor will not be able to respond that fast in real life. Therefore, a delay is needed for an electronic module to perform its assigned task. When the carpark has any available spaces, cars can enter the carpark. The counter will deduct by one once a car enters the carpark. When the carpark is full, no car can be entered to the carpark, hence servo motor barrier will not lift even there are cars want to enter. The part of motor servo coding is presented in Figure 3.9.

```
void INentrance()
{
    if (digitalRead(INsensor) == LOW && counter >= 1 && counter <= 12)
    //When detected (which is LOW), up the barrier
    {
    Serial.println("Entrance IN detected!");
    INservo.write(INbarrierup);
    int CIN;                                      //Declare people entering the classroom
    CIN = counter--;                              //Introduce increase counter
    car_counter();                            //Call subroutine of people_counter()
    delay(3000);
    }

    if (digitalRead(INsensor) == HIGH)
    //When detected (which is HIGH), the barrier back/remain the position
    {
    delay(3000);
    INservo.write(INbarrierori);
    }
}
```

Figure 3.9: The Coding of Servo Motor.

3.4.5 Setting up IR Sensor

The IR sensor attached to the NodeMCU is defined by the obstacle LED. *Low* is meant by parked and *High* means empty. If there is car detected by obstacle LED of IR sensor, the slot will show "parked" on webpage, as the result will connected to firebase realtime database. If there is no car detected by detection LED of IR sensor, the slot will show "empty" on webpage. Thus, a strong WiFi or network need to be connected to the NodeMCU as to update the latest status of the carpark slot. The coding of IR sensor is shown in Figure 3.10.

```
void loop()
{

  if (digitalRead(slotA) == LOW)
  {
    Serial.println("SLOT A PARKED");
    obj["CA_SLOT_A"]=String("PARKED");
    Firebase.set("/CARPARK_A1", obj);
  }
  else {
    Serial.println("SLOT A EMPTY");
    obj["CA_SLOT_A"]=String("EMPTY");
    Firebase.set("/CARPARK_A1", obj);
  }
```

Figure 3.10: Setting up IR Sensor

3.5 Webpage Interface

3.5.1 Microsoft Azure

There are many ways to create a webpage. One of the examples is using Microsoft Azure. Microsoft Azure is a cloud computing service formed by Microsoft for testing, deploying, building and managing application or services for data management. It supports many different programming languages, tools and frameworks.

3.5.2 Firebase Realtime Database

Firebase is a web application and mobile development platform. Firebase offers a realtime database and provide application developers an Application Programming Interface (API). The application data is stored in Firebase's cloud, as it can be synchronized across clients. The Firebase Realtime Database is known as a cloud-hosted database, which the data is stored in the form of JavaScript Object Notation (JSON). All client will share one Realtime Database and receive the latest update automatically as it is synchronized in realtime to every connected client. This concept is applied on the carpark slot monitoring system, which the latest status of the carpark slot can be updated automatically on webpage. The realtime database is shown in Figure 3.11.

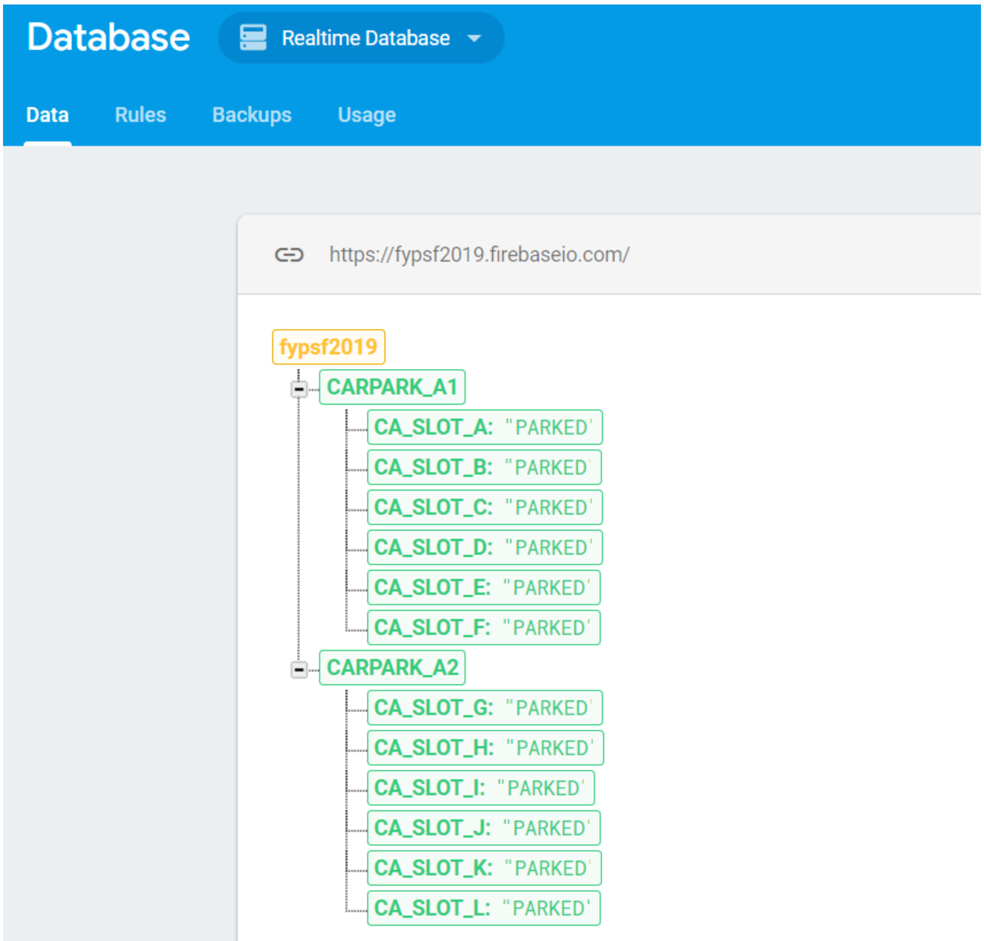


Figure 3.11: Realtime Database of Carpark Slot Monitoring System

3.5.3 Microsoft Visual Studio

In Microsoft Visual Studio, HTML programming language is used. Visual Studio is a combination of the powerful developer tooling with a source code editor. Hypertext Markup Language (HTML) is the standard markup language for generating web application and web pages. Figure 3.12 shows the HTML code on Microsoft Visual Studio. After that, the webpage needs to be published (as demonstrated in Figure 3.13) before users can use it.

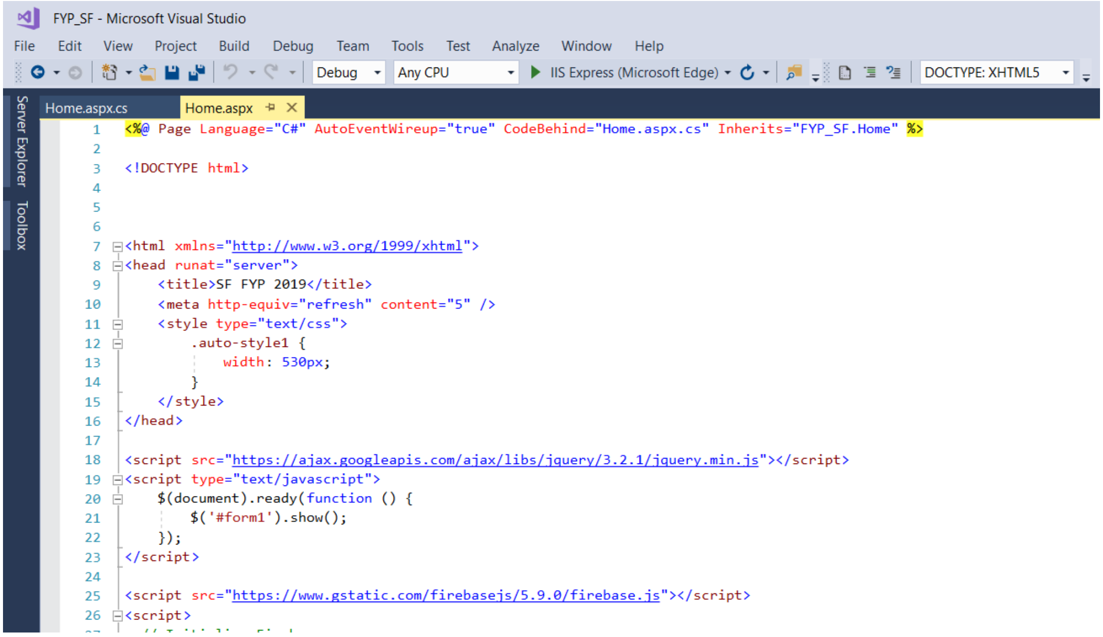


Figure 3.12: HTML Code on Microsoft Visual Studio.

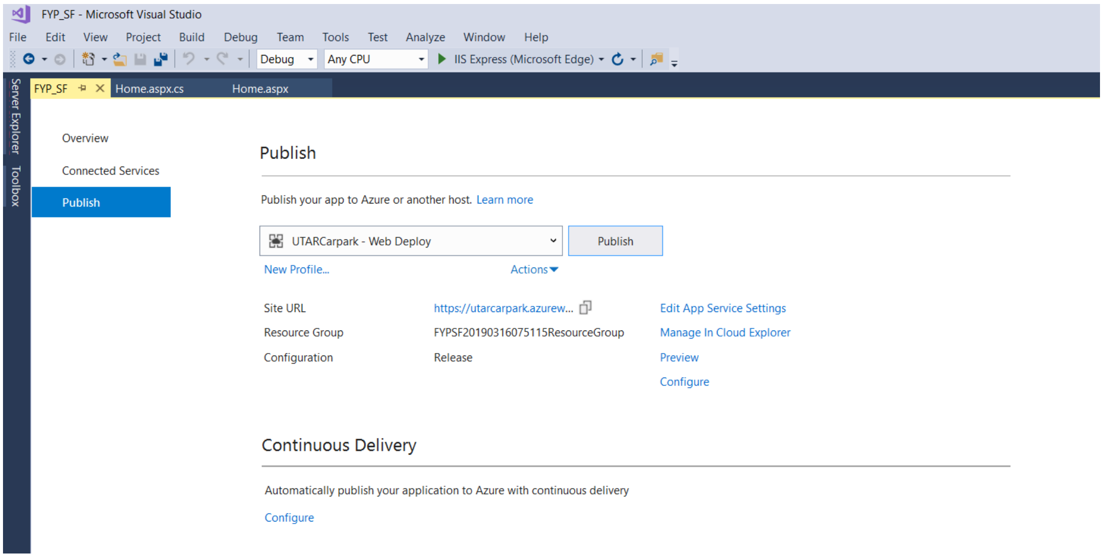


Figure 3.13: Publish the Webpage.

3.10 Equipment and Cost Analysis

Table 3.5 shows the components list with price. The total cost for this final year project is RM160.60.

Table 3.5: Components List with Price.

No.	Components	Unit Price (RM)	Unit	Total Price (RM)	Remarks
1.	NodeMCU V3 board	19.70	2	39.40	From Shopee, Robotedu
2.	IR Sensor Module	4.60	12	64.40	From Shopee, Robotedu
3.	Jumper Wire	5.00	2	10.00	From Shopee, Robotedu
4.	Servo Motor	8.00	2	16.00	From Shopee, Robotedu
5.	Arduino UNO	18.60	1	18.60	From Shopee, Robotedu
6.	I2C LCD Display	12.20	1	12.20	From Shopee, Robotedu
			Total	**160.60**	-

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Schematic Diagram of a Carpark Slot Monitoring System

The carpark slot monitoring system consists of two main microcontrollers, which are Arduino UNO and NodeMCU. A schematic diagram of a carpark slot monitoring system is generated using the Eagle software as shown in Figures 4.1 and 4.2, respectively.

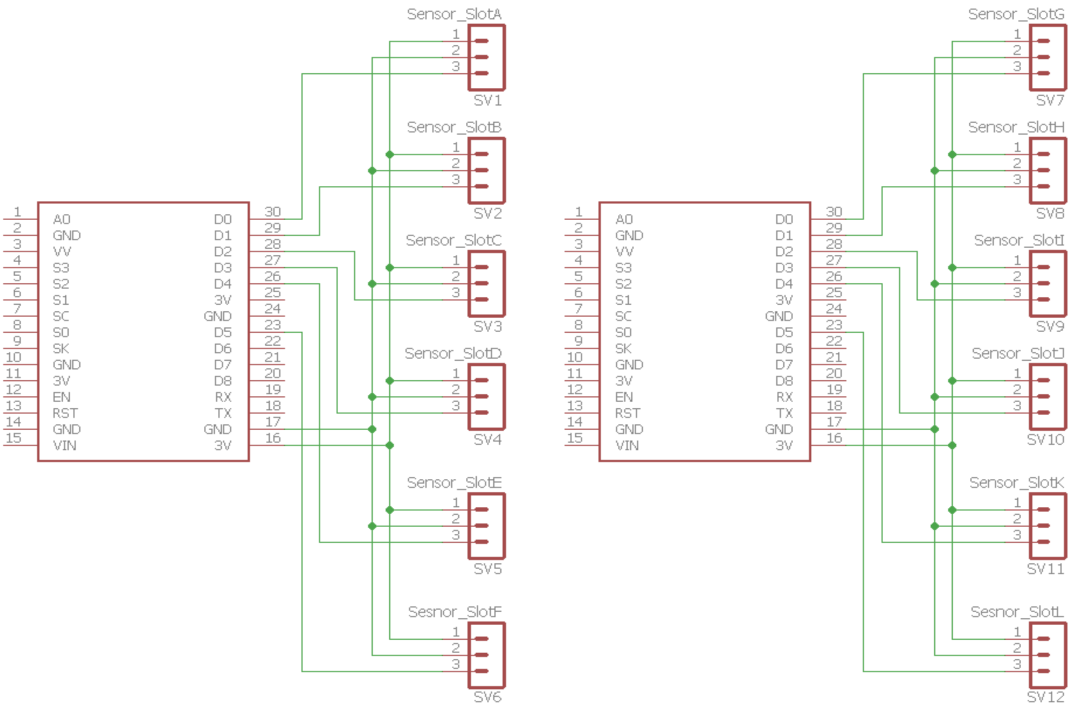


Figure 4.1: Schematic Diagram of Carpark Slot Monitoring System for NodeMCU.

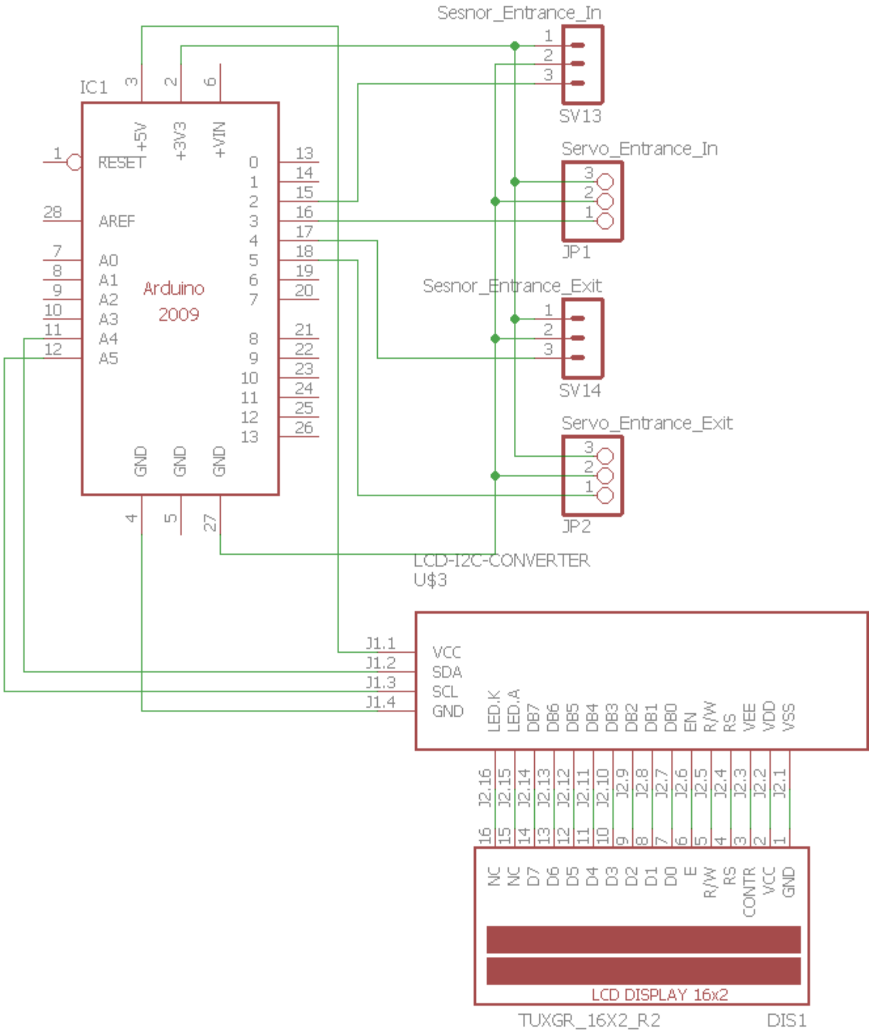


Figure 4.2: Schematic Diagram of Carpark Slot Monitoring System for Arduino UNO.

4.2 Overview of the Framework of Carpark Slot Monitoring System

The carpark slot monitoring system is built using electronic modules, such as LCD display, IR sensors, servo motors, NodeMCU and Arduino UNO. The top view of carpark slot monitoring system is shown in Figure 4.3 and the front view of the carpark slot monitoring system is shown in Figure 4.4. IR sensors are powered up by a powerbank with 3.0V, whereas LCD display and servo motor are powered up by powerbank with 5.0V. All the sensors and devices are connected to a common ground to avoid unnecessary problems.

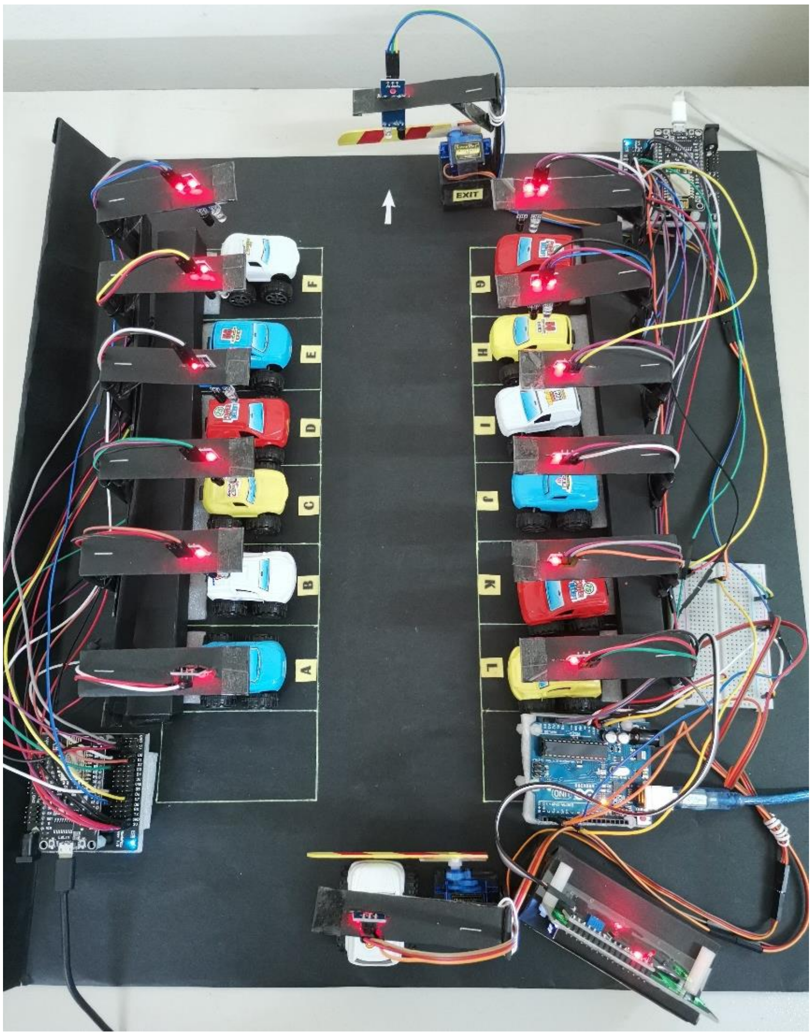


Figure 4.3: Top view of Carpark Slot Monitoring System.

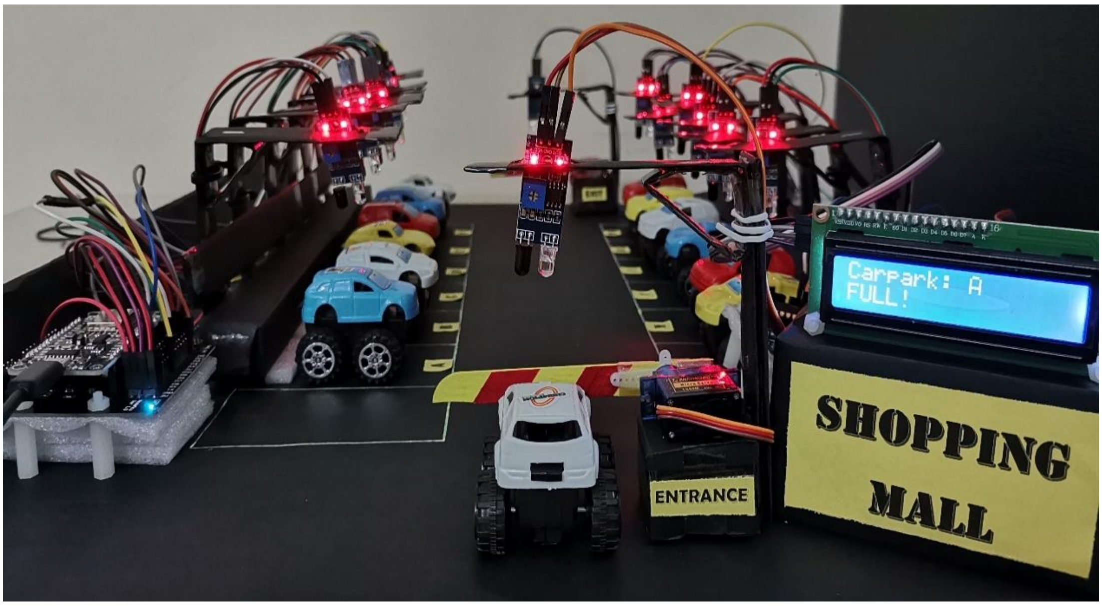


Figure 4.4: Front View of Carpark Slot Monitoring System.

4.3 Results from Servo Motor

There are two servo motors used in this project. One servo motor is placed at the entrance, and another servo motor is placed at the exit of carpark. The barrier of servo motor will not lift when the carpark is full, as shown in Figure 4.5. The barrier of servo motor will lift, when there is car at the entrance of carpark and there is still have available parking slot inside carpark. Figure 4.6 shows the barrier of servo motor does not lift, as there is no car at the carpark exit. It will only lift when there is car at the exit of carpark.

Figure 4.5: Servo Motor placed at the Entrance of Carpark.

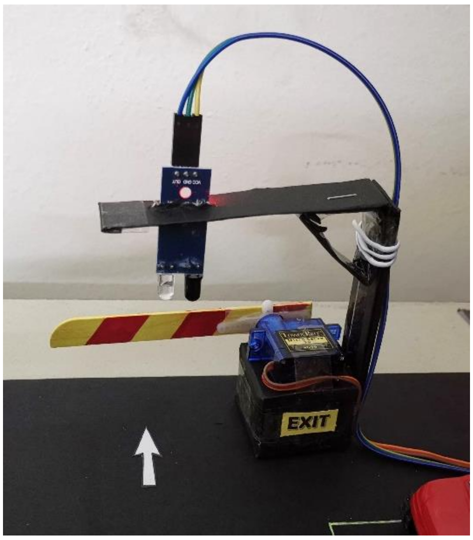


Figure 4.6: Servo Motor placed at the Exit of Carpark.

4.4 Results from 16×2 LCD Module

In this section, a 16×2 LCD module displays the basic information included the carpark section and the number of available free slots in the carpark. The 16×2 LCD module is named as Carpark A. There are 12 slots in the carpark. When the carpark slot is empty, the LCD will show the number of available carpark slot, which is 12. After one car enters, the number of carpark slot available is reduced from 12 to 11, and this result will be displayed on LCD display. When two cars enter the carpark, the number of carpark slot available will reduce to 10. The cars can enter the carpark until it is full, which is 12 cars. When the carpark is full, even if there are cars want to enter the carpark, no car will be able to enter, as the servo motor will not lift if there is no any available parking slot. If there is one car exit carpark, the carpark will have one more available slot. Therefore, the LCD display will show one slot is available for parking. At this time, one more car can be entered to the carpark. All the results from 16×2 LCD Module are presented in Figure 4.7 to Figure 4.11.

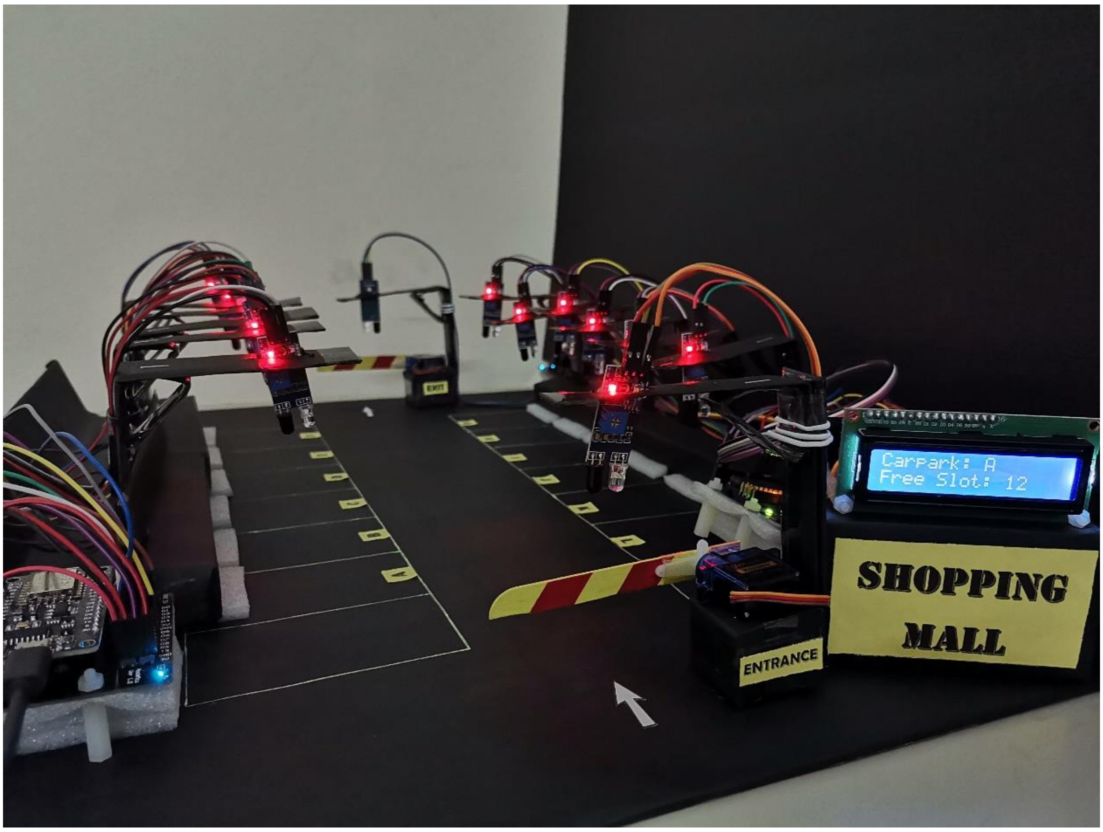


Figure 4.7: Entrance of Carpark when it is Empty.

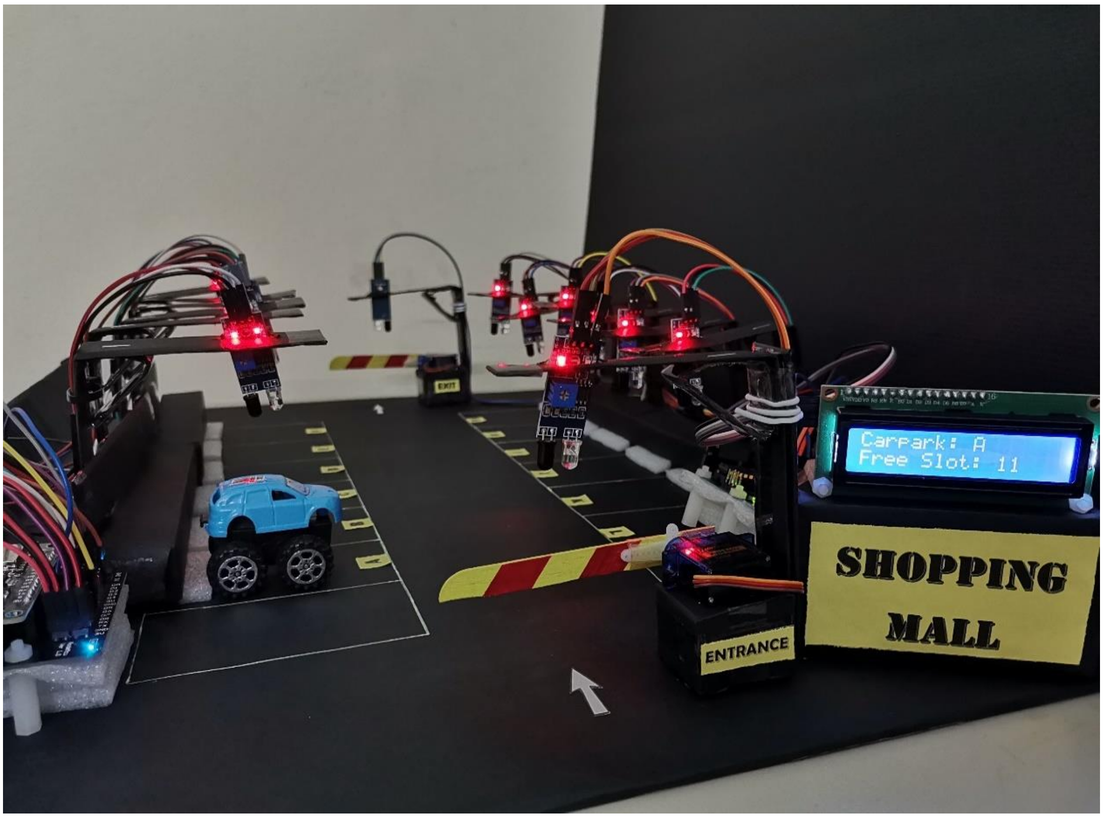


Figure 4.8: Entrance of Carpark when One Car is Parked.

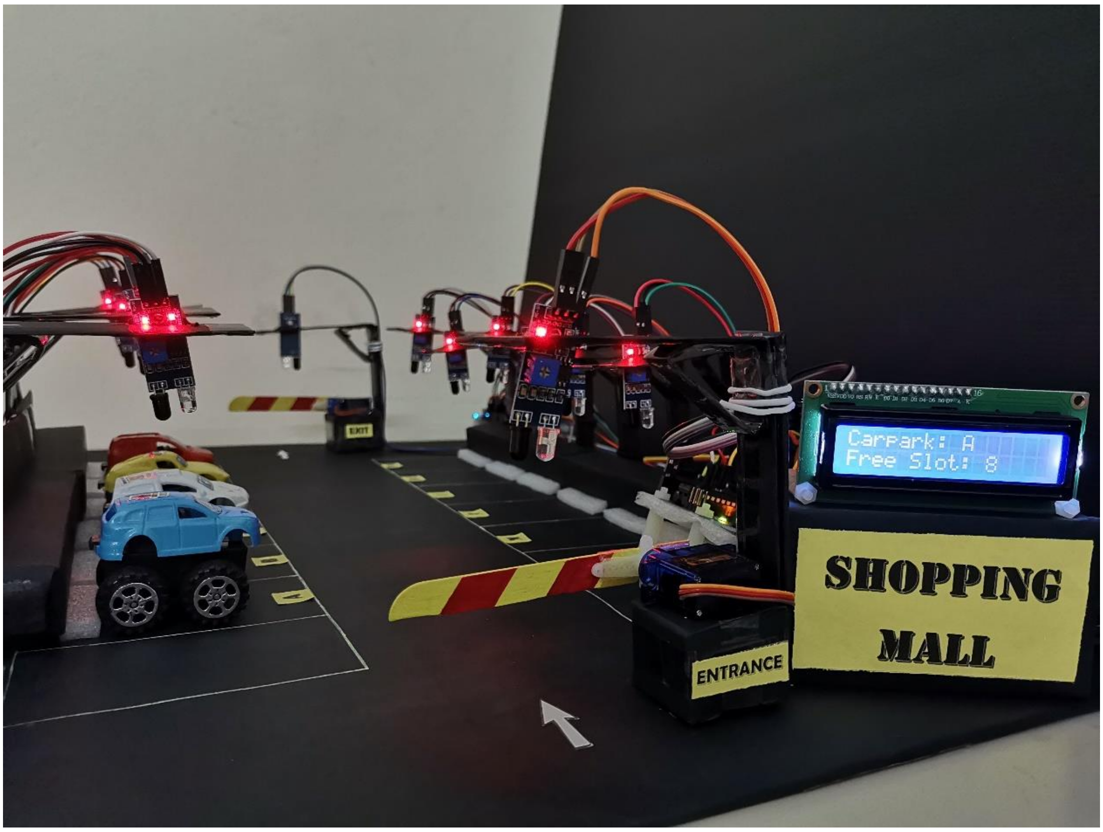


Figure 4.9: Entrance of Carpark when Four Cars are Parked.

Figure 4.10: Entrance of Carpark when Eight Cars are Parked.

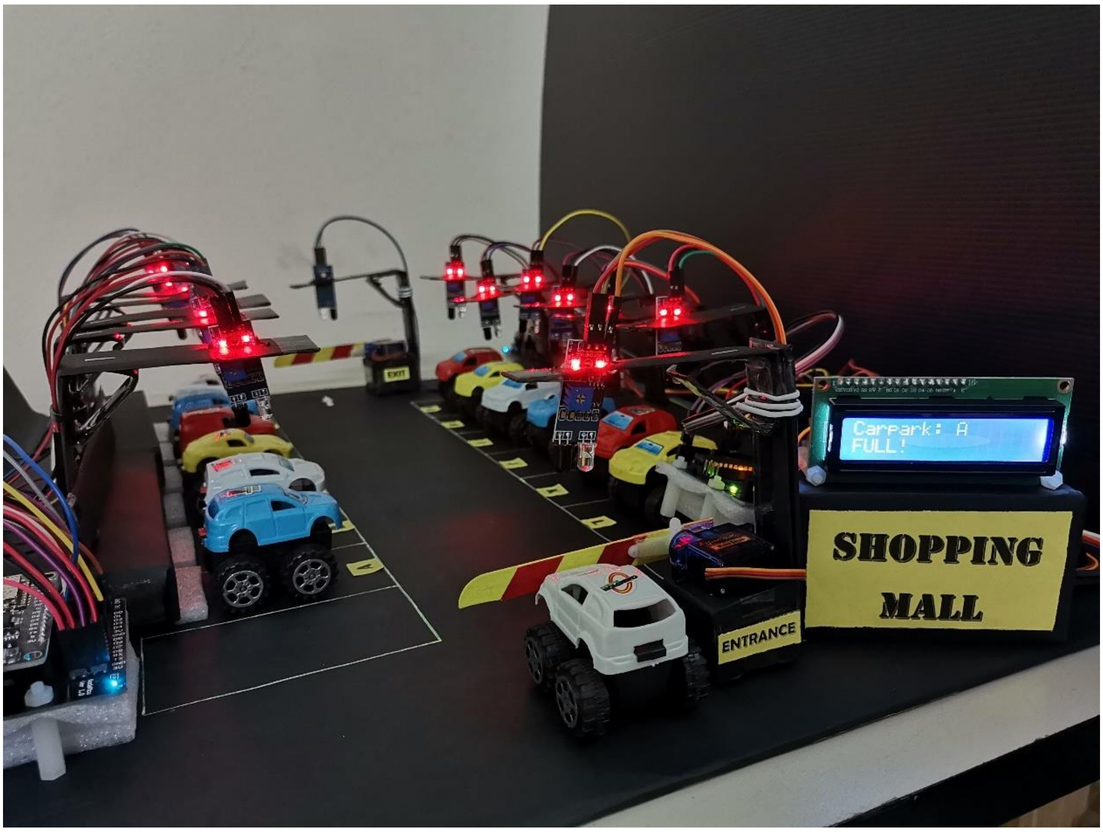


Figure 4.11: Entrance of Carpark when it is Full.

4.5 Results from Web-based Carpark Slot Monitoring System

In this section, the exact location of the carpark slot, which are slots A until slot L are shown on the published webpage. When the Carpark A is empty, the results displayed on webpage is shown in Figure 4.12. When slot A is parked, and other carpark slot is empty, the results displayed is shown in Figure 4.13. When slot A, slot B, slot C and slot D are parked, whereas other carpark slots are empty, the result is demonstrated as shown in Figure 4.14. When all slots are parked, except slot G, slot H, slot I and slot J is empty, the result is demonstrated in Figure 4.15. When all slots are parked, the result is demonstrated in Figure 4.16.

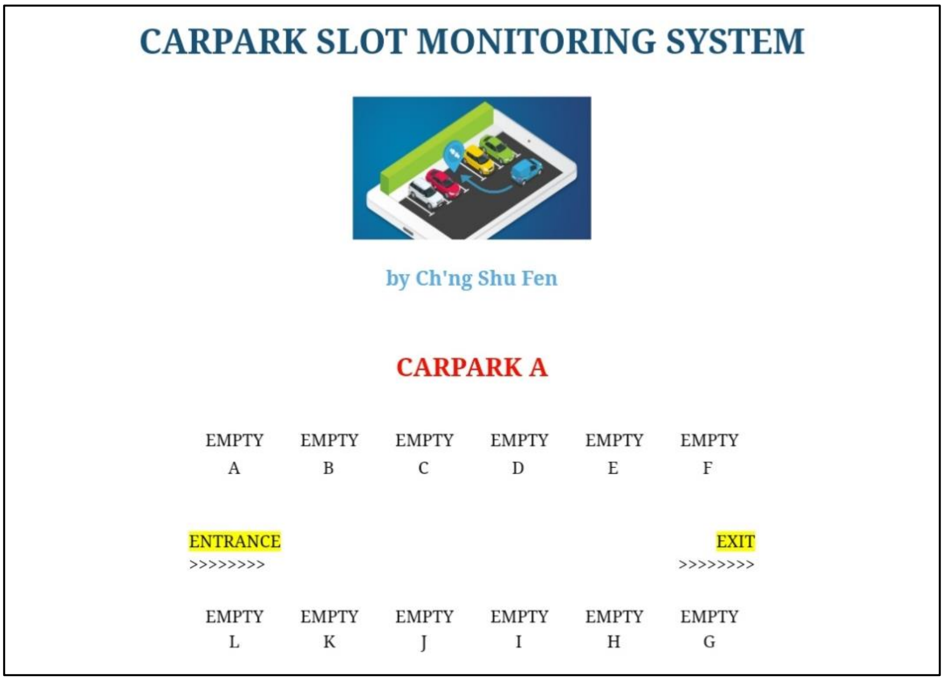


Figure 4.12: Web-based Carpark Slot Monitoring System when all Carpark Slot are Empty.

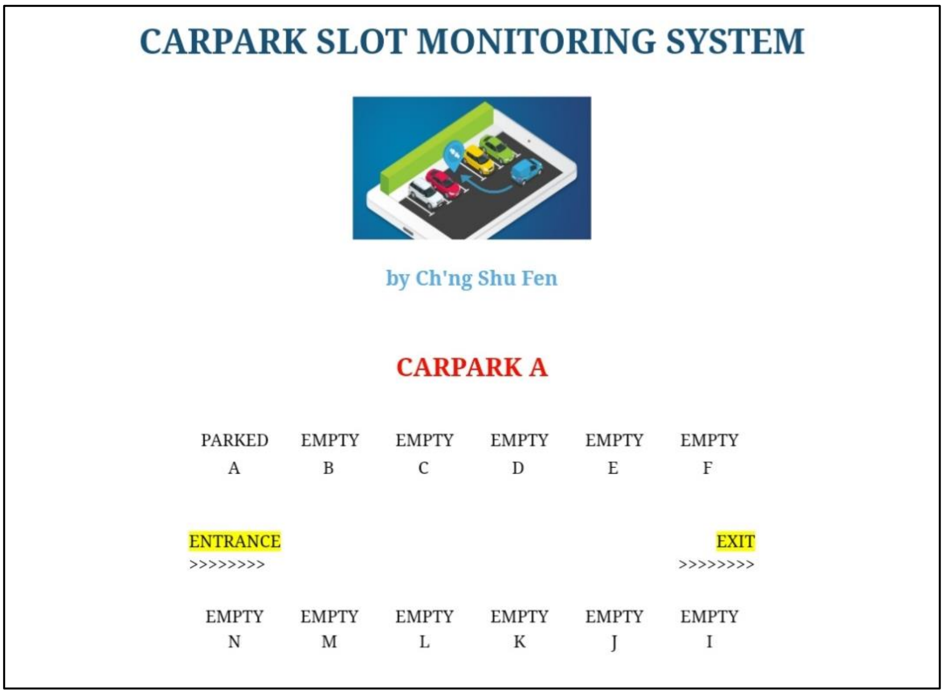


Figure 4.13: Web-based Carpark Slot Monitoring System when Slot A is Parked.

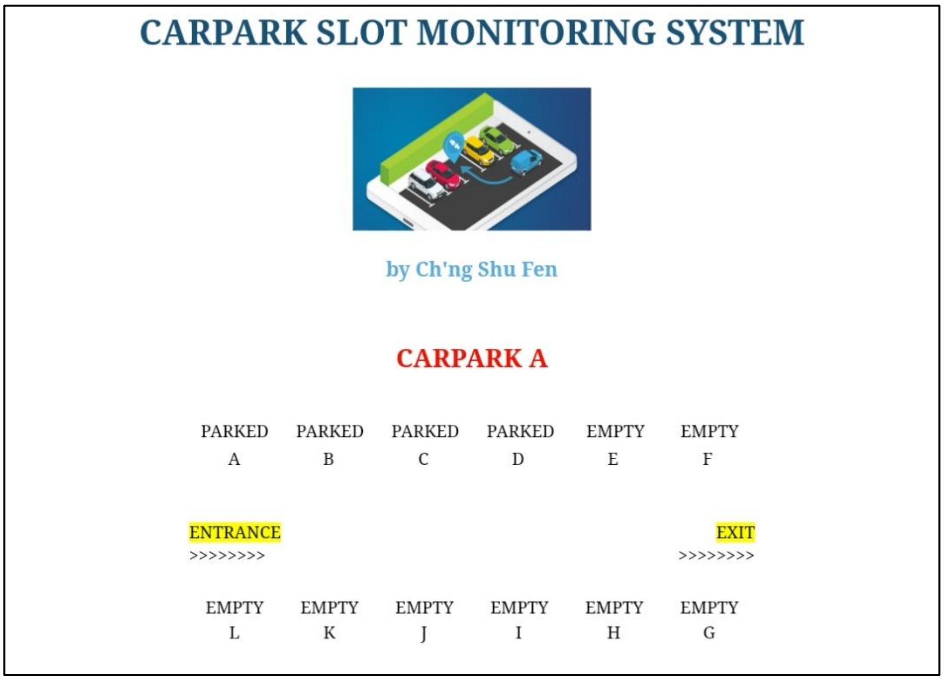


Figure 4.14: Web-based Carpark Slot Monitoring System when Slot A, Slot B, Slot C and Slot D are parked.

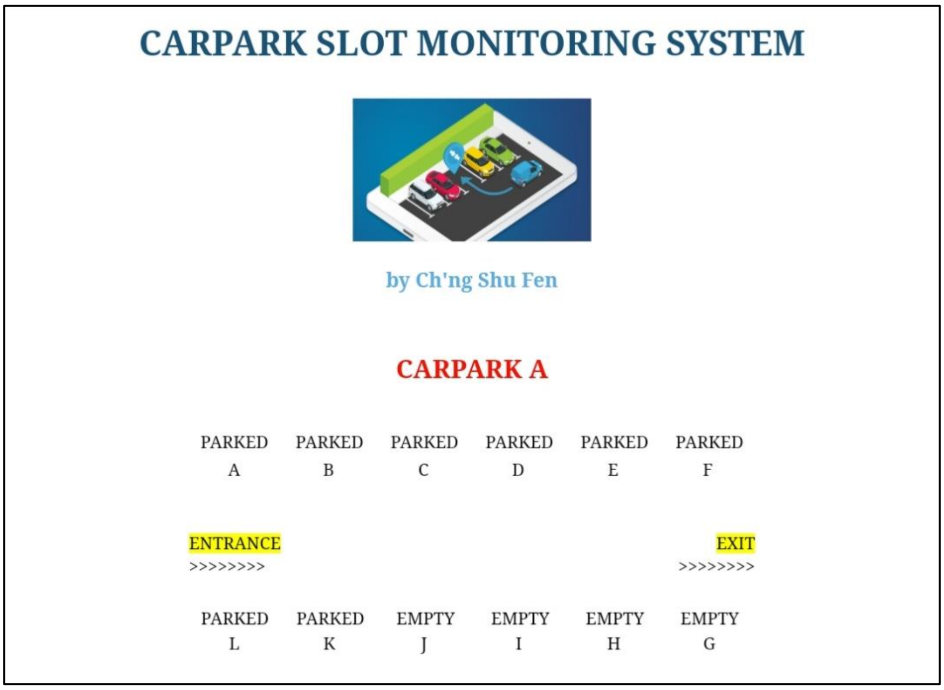


Figure 4.15: Web-based Carpark Slot Monitoring System when Slot G, Slot H, Slot I and Slot J are Empty.

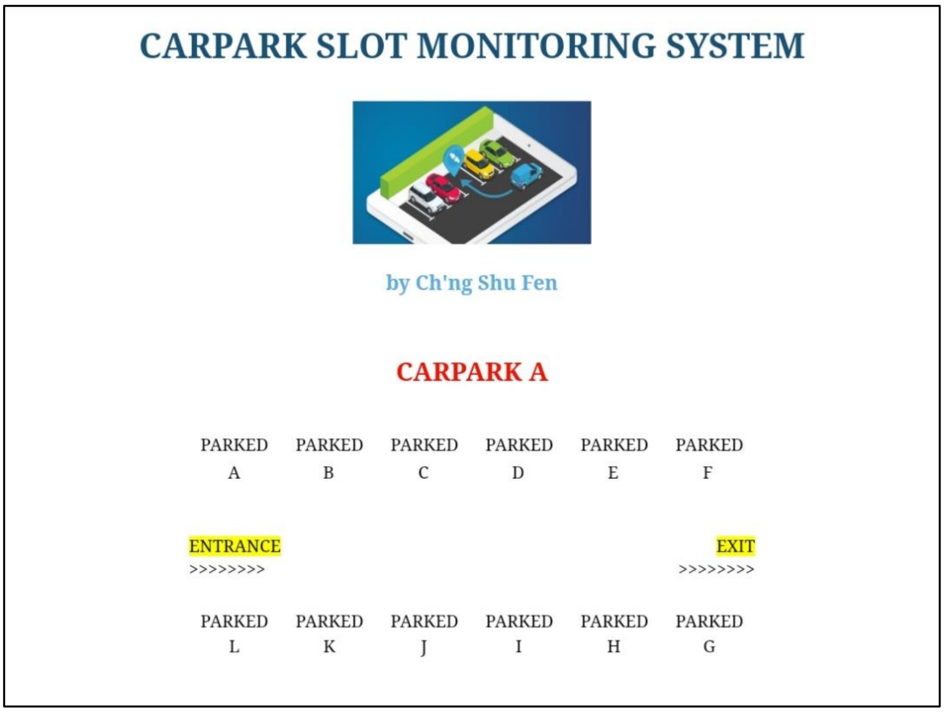


Figure 4.16: Web-based Carpark Slot Monitoring System when all Carpark Slots are Parked.

4.6 Webpage Interface QR Code

QR code generator is used to create a QR code as shown in Figure 4.17. This generator can be found easily online as it can be found online easily. The created QR code is used to link the webpage interface.

Figure 4.17: QR Code for Web-based Carpark Slot Monitoring System.

4.7 Percentage of Accuracy for Carpark A

In this IoT project, the new electronic devices and platforms used are very important. To compare and evaluate the new electronic devices, some characterises of IoT were carried out, i.e. data storage, network connectivity, and power supply management. For example, the circumstances of entering and exiting the carpark are observed and recorded. The percentage of accuracy of Carpark A is presented in Table 4.1. This will provide more reliable car parking system.

Table 4.1: Detection Result for Carpark A.

Case	Situation	The detected number of cars	Percentage of Accuracy (%)
1	Entering the carpark	12	100
2	Exiting the carpark	12	100

4.8 Discussion on Actions to be Avoided

At first, the design of prototype looks as published in Figure 4.18, which is the IR sensor is facing opposite to each other. When the IR sensors are powered up, it cannot function well, and the obstacle LED of IR sensors keep light up. This is because when the IR transmitter releases radiation, it reaches the IR receiver of the opposite side. Therefore, the output of the sensor is defined based on the intensity of the reception by the IR receiver. This may cause the obstacle LED keep showing the detection LED as high, means there are obstacle, and the result is affected. Moreover, one more limitation is that the light intensity of the surrounding will also affect the overall result. Thus, the carpark slot monitoring system should put indoor and not under strong light intensity.

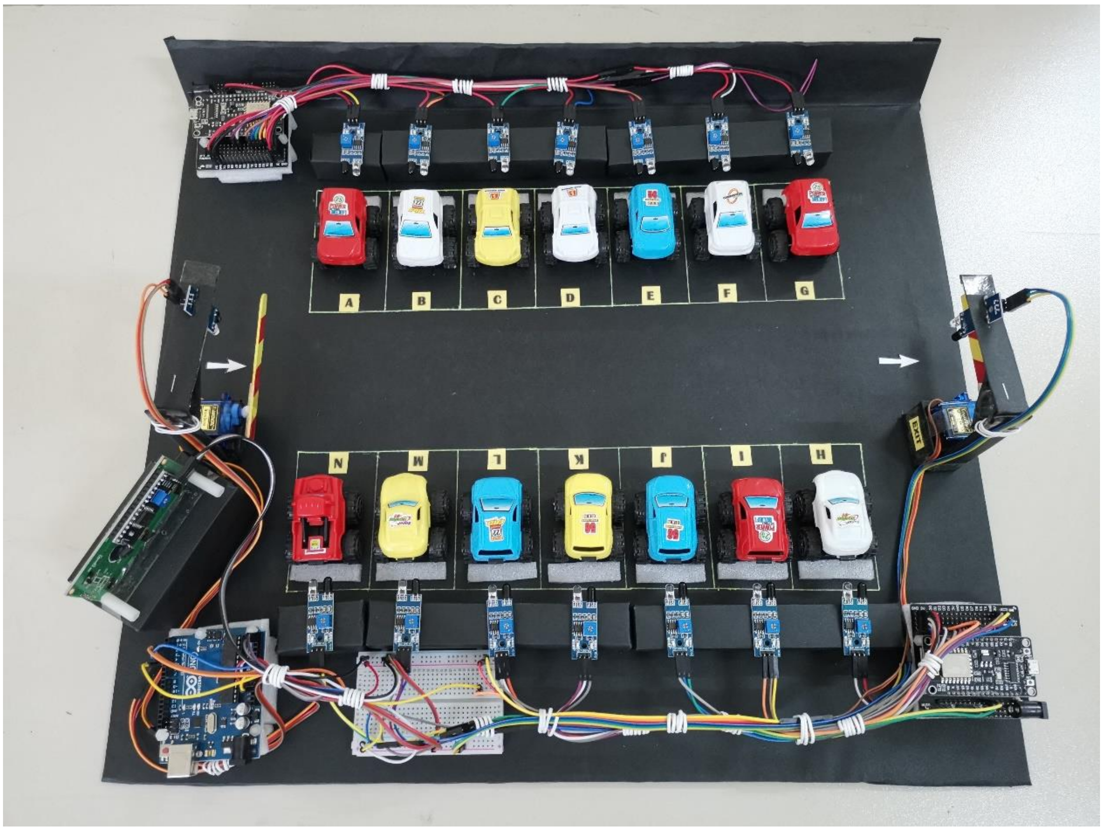


Figure 4.18: The First Design of Carpark Slot Monitoring System.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

A carpark slot monitoring system is developed successfully and working fine. The LCD display can display the number of parking slot available accurately. The web-based carpark slot monitoring system also display the correct slots parked. The servo motor allows the car to enter carpark, while there is any available carpark slot. There is no car allowed to enter carpark when the carpark is fully parked. The limitations of the carpark slot are obvious. When the internet is not smooth enough, the data cannot be transmitted from the IR sensor to the web-based carpark slot monitoring system. Thus, the result will not be updated without network connection. Users had to be depended on the IR sensor to find the available parking slot if there is no internet connection. Arduino C language and HTML are used to implement this system. It is a simple idea of a carpark slot monitoring system, which opens to future users to make modifications and improvements to perform more functions and specifications.

5.2 Recommendations

In the future, the features which can allow users to be directed to the empty carpark slot via mobile application can be developed. The webpage also can be designed in a more interesting view to capture the empty carpark slots easily.This will save much more time to find an empty carpark and save fuel at the same time. Furthermore, an LCD screen could replace with a larger size of the LCD screen which can display more than two rows of the texts.

REFERENCES

Anitha, J., Thoyajakshi, Y., Ramya, A., Sravani, V. and Kumar, P. (2017). Intelligent Parking System Using Android Application. *International Journal of Pure and Applied Mathematics*, [online] 114(7), pp.165-174. Available at: https://acadpubl.eu/jsi/2017-114-7-ICPCIT-2017/articles/7/16.pdf [Accessed 4 Aug. 2018].

Aqeel, A. and G+ (2019). *Introduction to NodeMCU V3 - The Engineering Projects*. [online] The Engineering Projects. Available at: https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html [Accessed 17 Mar. 2019].

Cisco.com. (2016). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020*. [online] Available at: https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf [Accessed 30 Jul. 2018].

electricalfundablog.com. (2019). *Infrared Sensor - How it Works, Types, Applications, Advantage & Disadvantage*. [online] Available at: https://electricalfundablog.com/infrared-sensor/ [Accessed 2 Apr. 2019].

Fraifer, M. and Fernström, M. (2017). Designing a Smart Car Parking System (PoC) Prototype Utilizing CCTV Nodes: A vision of an IoT parking system via UCD process. *dvances in Science, Technology and Engineering Systems Journal*, [online] 2(3), pp.755-746. Available at: https://www.astesj.com/publications/ASTESJ_020396.pdf [Accessed 4 Aug. 2018].

Future Electronics Egypt (Arduino Egypt). (2019). *NodeMCU (ESP8266 WiFi Programming & Development Kit)*. [online] Available at: https://store.fut-electronics.com/products/nodemcu-esp8266-programming-and-development-kit [Accessed 16 Mar. 2019].

Henry's Bench. (2019). *Arduino IR Obstacle Sensor: Tutorial and Manual*. [online] Available at: http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-ir-obstacle-sensor-tutorial-and-manual/ [Accessed 2 Apr. 2019].

Mary, R. (2018). What is wireless communication technology and its types | Infrared, Satellite Communication & Broadcast Radio. Retrieved from https://www.engineersgarage.com/articles/wireless_communication

Medium. (2017). *Importance of Internet of Things (IoT) in Our Live – Teksun – Medium*. [online] Available at: https://medium.com/@TeksunGroup/importance-of-internet-of-things-iot-in-our-live-b71e53d50a44 [Accessed 30 Jul. 2018].

MWEBAZE, A. (2009). *THE DESIGN OF AN INTELLIGENT PARKING SYSTEM USING WIRELESS SENSOR NETWORKS AND MULTI-PROTOCOL LABEL SWITCHING*. [online] Pubs.cs.uct.ac.za. Available at: http://pubs.cs.uct.ac.za/archive/00000679/01/tony_Thesis_changes_graduation_june_2010Version2.pdf [Accessed 30 Jul. 2018].

my2cents. (2019). *Comparison of ESP8266 NodeMCU development boards • my2cents*. [online] Available at: https://frightanic.com/iot/comparison-of-esp8266-nodemcu-development-boards/ [Accessed 16 Mar. 2019].

Parasher, A., Pandey, B. and Raj Sharma, S. (2017). Intelligent Parking System Using Internet Of Things (IoT). *INTERNATIONAL JOURNAL FOR RESEARCH & DEVELOPMENT IN TECHNOLOGY*, [online] 7(3), p.390. Available at: http://www.ijrdt.org/upload/8421975-Intelligent%20Parking%20System%20Using%20Internet%20Of%20Things%20(IoT).pdf [Accessed 14 Aug. 2018].

Smart Car Parking System using Arduino UNO. (2018). *International Journal of Computer Applications*, [online] 169(1), p.13. Available at: https://www.ijcaonline.org/archives/volume169/number1/nandyal-2017-ijca-914425.pdf [Accessed 31 Jul. 2018].

Sparks.gogo.co.nz. (2019). *Buy this 16x2 LCD with I2C Backpack For Arduino etc.. at our Dev Boards Online Shop*. [online] Available at: http://sparks.gogo.co.nz/catalog/Microcontrollers-255/Dev-Boards-112/16x2-LCD-with-I2C-Backpack-For-Arduino-etc.-391.html [Accessed 2 Apr. 2019].

SR, B. (2015). Automatic Smart Parking System using Internet of Things (IOT). *International Journal of Scientific and Research Publications*, [online] 5(1), p.629. Available at: http://www.ijsrp.org/research-paper-1215/ijsrp-p4898.pdf [Accessed 14 Aug. 2018].

Zhang, Z., Li, X. and Yuan, H. (2013). A Street Parking System Using Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, [online] 9(6), p.12. Available at: http://journals.sagepub.com/doi/full/10.1155/2013/107975 [Accessed 4 Aug. 2018].

APPENDICES

```
#include <Wire.h>                          //library for serial connection
#include <LiquidCrystal_I2C.h>             //library for I2C 20x4 LCD
#include <Servo.h>
#include <stdio.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
// 0x27 means the address of this 1602 I2C LCD display
// Different lcd may have the different address
// A4 pin - SDA;  A5 pin - SCL

int counter = 12;              //Initial counter value is 0

Servo INservo, OUTservo;

// For Servo IN
int INsensor = 2;               //pin 2 on-board UNO
int INservopin = 3;             //pin 3 on-board UNO
int INbarrierup = 85;           //Up position of the barrier
int INbarrierori = 15;          //Origin position of the barrier

// For Servo OUT
int OUTsensor = 4;              //pin 4 on-board UNO
int OUTservopin = 5;            //pin 5 on-board UNO
int OUTbarrierup = 110;         //Up position of the barrier
int OUTbarrierori = 15;         //Origin position of the barrier

void setup()
{
  Serial.begin(9600);
  pinMode(INsensor, INPUT);
  pinMode(OUTsensor, INPUT);
  pinMode(INservopin, OUTPUT);
  pinMode(OUTservopin, OUTPUT);

  INservo.attach(INservopin);
  OUTservo.attach(OUTservopin);

  // LCD
  lcd.begin();                  //Initialize the LCD
  lcd.backlight();              //Enable or Turn On the backlight for LCD
  lcd.clear();                  //Clear the LCD
  lcd.setCursor(0,0);
  lcd.print("Carpark: A");
  lcd.setCursor(0,1);            //Set cursor of LCD at 1st col & 1st row
  lcd.print("Free Slot: ");      //LCD displays “People Count: ”
  lcd.setCursor(11,1);           //Set cursor of LCD at 1st col & 2nd row
  lcd.print(counter);
}

void car_counter()
//Subroutine for counter included range of student in classroom
```

```
{
  lcd.clear();                      //Clear the LCD
  lcd.setCursor(0,0);
  lcd.print("Carpark: A");
  lcd.setCursor(0,1);               //Set cursor of LCD at 1st col & 1st row
  lcd.print("Free Slot: ");         //LCD displays “People Count: ”
  lcd.setCursor(11,1);              //Set cursor of LCD at 1st col & 2nd row
  lcd.print(counter);               //LCD displays the number of counter
  Serial.println("Carpark: A");
  Serial.println(counter);


  if(counter == 0)
  {
  lcd.clear();                       //Clear the LCD
  lcd.setCursor(0,0);
  lcd.print("Carpark: A");
  lcd.setCursor(0,1);                //Set cursor of LCD at 1st col & 1st row
  lcd.print("FULL!");                //LCD displays “People Count: ”
  }
}

void INentrance()
{
    if (digitalRead(INsensor) == LOW && counter >= 1 && counter <= 12)
    //When detected (which is LOW), up the barrier
    {
    Serial.println("Entrance IN detected!");
    INservo.write(INbarrierup);
    int CIN;
    //Declare people entering the classroom
    CIN = counter--;        //Introduce increase counter
    car_counter();          //Call subroutine of people_counter()
    delay(3000);
    }

    if (digitalRead(INsensor) == HIGH)
    //When detected (which is HIGH), the barrier back/remain the position
    {
    delay(3000);
    INservo.write(INbarrierori);
    }
}

void OUTentrance()
{
    if (digitalRead(OUTsensor) == LOW && counter >=0 && counter <= 11)
//When detected (which is LOW), up the barrier
      {
      Serial.println("Entrance OUT detected!");
      OUTservo.write(OUTbarrierup);
      int COUT;                       //Declare people entering the classroom
      COUT = counter++;               //Introduce increase counter
      car_counter();                  //Call subroutine of people_counter()
      delay(3000);
      }

      if (digitalRead(OUTsensor) == HIGH)
  //When detected (which is HIGH), the barrier back/remain the position
      {
      delay(3000);
      OUTservo.write(OUTbarrierori);
      }
  }

  void loop()
```

```
{
    INentrance();
    OUTentrance();
}
```