# SMART CAR PARKING SYSTEM USING GRAPH ALGORITHM

**NAN CHAW SU KYI**

**M.A.Sc.** **JUNE 2022**

# SMART CAR PARKING SYSTEM USING GRAPH ALGORITHM

**BY**

**NAN CHAW SU KYI**

**M.A.Sc.** **JUNE 2022**

# SMART CAR PARKING SYSTEM USING GRAPH ALGORITHM

**BY**

**NAN CHAW SU KYI**

**D.C.M.**

**A dissertation submitted in partial fulfillment of the requirements for the degree of**

**Master of Applied Science**

**(M.A.Sc.)**

**UNIVERSITY OF COMPUTER STUDIES, YANGON**

**JUNE 2022**

# ACKNOWLWDGEMENTS

I would like to take this opportunity to express my sincere thanks to those who helped me with various aspects of conducting research and writing this thesis. To complete this thesis, many things are needed like my hard work as well as the supporting of many people.

First and foremost, I would like to express my deepest gratitude and my sincere thanks to **Prof. Dr. Mie Mie Khin**, Rector, University of Computer Studies, Yangon, for her kind permission to submit this thesis.

My heartfelt thanks and respect go to Prof. **Dr. Khin Marlar Tun**, Principal and Pro-rector of Computer University (Hpa-an), for her invaluable and administrative support.

My sincere thanks and regards go to **Dr. Htar Htar Lwin**, Pro-rector, Head of Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her kind management throughout the completion of this thesis.

I would like to express my deeply thanks to **Dr. Amy Tun**, Professor, Course coordinator of Master's (CT), University of Computer Studies, Yangon, for her painstaking suggestion and encouragement throughout the development of the thesis.

My sincere thanks and regards go to my supervisor, **Dr. Khant Kyawt Kyawt Theint,** Lecturer, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her support, guidance, supervision, patience and encouragement during the period of study towards completion of this thesis.

I also wish to express my deepest gratitude to **Daw Hnin Yee Aung**, Lecturer, Department of English, University of Computer Studies, Yangon, for her editing this thesis from the language point of view.

Moreover, I would like to extend my thanks to all my teachers who taught me throughout the master's degree course and my friends for their cooperation.

I especially thank to my parents, all of my colleagues, and friends for their encouragement and help during my thesis.

# ABSTRACT

An effective car parking of a condo is conducted in this system with the aim of reducing the time for finding the free parking lots. The condo has an two-way road for each entrance and exit. Smart car parking enables the user to find the nearest parking areas. Push Buttons are used for validation when the user arrives the entrance. If the user presses to Push Button, the system will check for the nearest empty slot and show the user in LCD and the gate is opened by using servo motor. Graph Theory is applied to find the shortest path. As soon as a car has left from a slot, this system redefines the state of the slot as free. This system is implemented by using Arduino mega 2560 microcontroller, Push Button, LCD display, Ultrasonic sensor and Servo Motor. This system is implemented using C++ programming language with Arduino software IDE (Integrated Development Environment).

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER 1

# INTRODUCTION

Nowadays, electronic control technology is advancing rapidly around the world. New electronic technologies and manufacturing systems are also rapidly changing. In recent years, many control modes used in automobiles were developed by microcontrollers (minicomputers).

In densely populated cities, parking systems are an important part of life. The problem of parking leads to air pollution, traffic congestion and driver frustration in public places like shopping malls, cinemas, hotels and condos. The problem can be solved by making a smart car parking system using open source hardware and programmable sensors.

Many processes controlled by human operators in industry may not be automated using conventional control technologies. One reason is that the linear controllers commonly used in conventional control are not suitable for non-linear plants. Another reason is that humans have amassed a variety of information and integrated control strategies that could not be integrated into a single analysis law. Arduino logic captures the continuous nature of human decision-making processes and is a precise development over methods based on binary logic (widely used in mechanical control devices).

In this system, the sensors interact with each other. The 24 sensors in the parking system work, with one sensor sending the trigger and the other sending the trigger. The sensors send triggers one every 1 second to monitor all parking. After the checking, the system will send the result back to the machine whether the car parking is full or not. If the parking lot is less than 5 cars, it does not mean that the entrance is less than 5 cars. Only 24 sensors will build an Arduino to determine which holes are missing.

It is the key in embedded systems.The embedded system is designed to perform a separate function. An electronic / electronic device for running firmware and hardware, it is safe and effective for parking system.

## 1.1 Objectives of the Thesis

The following factors are described as the objective of the thesis.

- To avoid the unnecessary travelling time to fill parking lots.
- To display on the monitor at the gate before the driver is allowed to enter the parking lot.
- To find out the parking number for the driver in a short time.
- To discover the car parking finds easily
- To reduce time to find the car parking
- To know the knowledge about Arduino microcontroller, Push Button, Ultrasonic sensor, LCD and Servo Motor.

## 1.2 Motivation of the System

The motivation for this article is the desire to work in both hardware and software. From Arduino's applications are come up with ideas for some really cool game consoles. The organizers eventually agreed that the project would include the use of Arduinos Mega, the world-new Arduinos Mega, and the use of Arduino Mega, which had never been used before by the club.

The basic premise of knowledge-based (expert) control is to capture and implement the experience and knowledge available from experts. One type of knowledge-based control is the Arduino rule control. Control functions related to specific conditions of the system are set out in the terms of the Arduino if-then rules. The Arduino set is used to define quality values of controller inputs and outputs.

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital inputs / outputs (15 of which can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware nodes), 16 MHz crystal oscillator, USB connection power outlet ICSP heading, and reset button. It contains everything which are needed to support the microcontroller. It is connected it to a computer with a USB cable or power it up with an AC-to-DC adapter or battery to get started.

Some of research can create such as: Arduino mega controller hardware design and implementation, learning behaviors implemented as Arduino logic controllers for autonomous agents, measurement theoretic justifications of connectives in graph theory, shortest path by using graph theory in Arduino Mega microcontroller. Microcontroller based control systems have been developed an advancement of modern civilization and technology. I am very interesting in this area and so I decided to do the Microcontroller based vehicle speed control system with Arduino Mega controller system for my thesis.

## 1.3 Overview of the System

The diagram shows the input and output components connected to the Arduino Mega 2560. The input component is comprised of an LCD, ultrasonic sensors, push buttons, and a servo motor. The output components comprise an exit gate, a servo and ultrasonic sensors. The system's operation starts when the car is placed in front of the gate. When the driver presses the button, the counter circuit in the system will calculate whether there is an available vacant parking lot in the system and the shortest path between the two exit gates. If the vacant parking lot is available, the LCD will display the information about the parking lot available and the parking placement to the driver, and the entrance gate will open to allow the car to enter the parking. All of these can be summarized as shown in Figure 1.1.

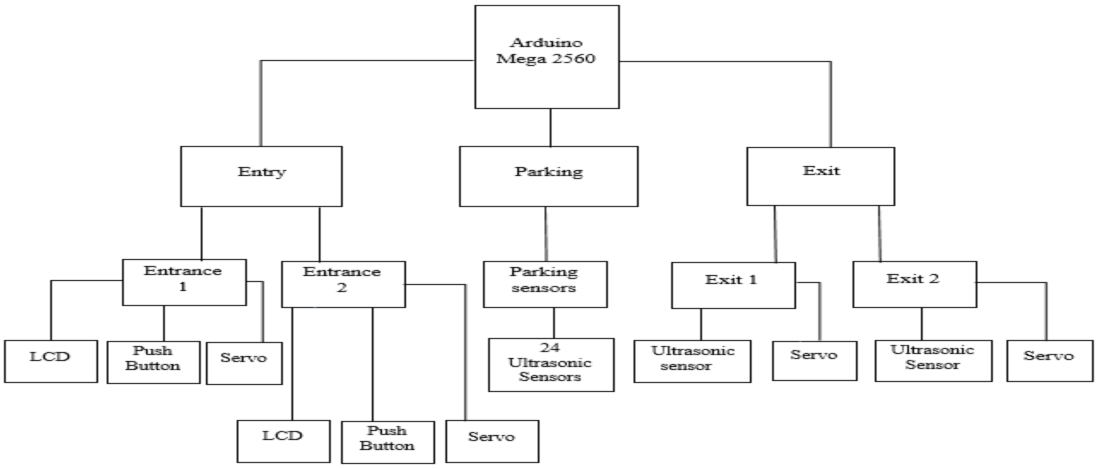


**Figure 1.1 Overview of the Smart Car Parking System**

## 1.4 Organization of the Thesis

There are five chapters in this thesis. Chapter 1 describes the introduction of the smart car parking system, interfacing with objectives of the Thesis, motivation of the system, overview of the system, and organization of the Thesis. Chapter 2 describes background theory of the system, the graph theory in control system.

Chapter 3 describes hardware and design methodology, Arduino Mega microcontroller, Servo motor, Push Button, Liquid Crystal Display and Ultrasonic Sensor. Chapter 4 describes software implementation of the system, Hardware Implementation of the system, System Design and experiment results. Finally, the conclusion, advantages, limitation and further extension are presented in Chapter 5.

CHAPTER 2

# BACKGROUND THEORY

In the Embedded based Smart Car Parking Using Graph Theory, Microcontroller Arduino Mega 2560 is implemented with Graph Theory (Graph Algorithm) system.

## 2.1 Control System

A control system is a set of components that work together under the direction of some intelligence. In most cases, electronic circuits provide intelligence, and electronic components, such as motors and sensors, provide the physical world. A control system is used to control its output to some particular values. The common method of representing for a control system is using block diagram. As shown in Figure 2.1, it consists of a block representing each component in a control system connected by lines that represent the signal paths.

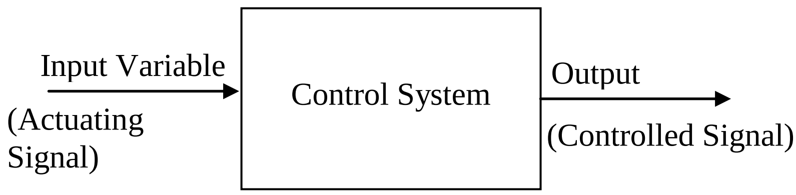


**Figure 2.1 Block diagram of Basic Control System**

Each component receives an input signal from some part of the system and provides an output signal for the other part of the system. The signals can be electric current, voltage, air pressure, liquid flow rate, temperature, speed or others. Signal paths are connected to power lines. It could be an atomic tube or something moving from one part of a signal to another. The component may use some sources of energy to increase the power of the output signal.

### 2.1.1 Open-loop Control System

The control systems are usually of two types: the function is the physical system output and the independent open control system and closed-loop control systems (also called response control systems), in it the control function depends on the physical system output. One of the characteristics of an open-loop controller is that it does not use feedback to determine whether the output achieves the desired goal of the input. This means that the system does not monitor the output of the processes it controls. As a result, a truly open-loop system can be involved in machine learning and fix any mistakes it can make. It does not care for any interruptions in the system. The general form of an open-loop control system is illustrated in Figure 2.2 [6].

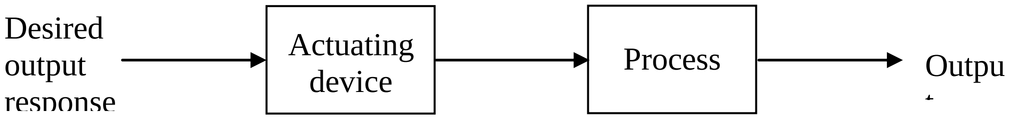


**Figure 2.2 Open-loop Control System (without feedback)**

## 2.1.2 Close-loop Control System

The system must first measure to control any physical changes. This system is called a sensor to measure a controlled signal. Under a controlled signal, the physical system is called a plant. In the bracket system, the system's forced signals (called inputs and outputs) are determined by the system's responses (called outputs). For the proximity control system, an additional system known as a control device, or regulator, needs to be connected to the box to obtain satisfactory responses and symptoms. The general form of a closed-loop control system is illustrated in Figure 2.3.

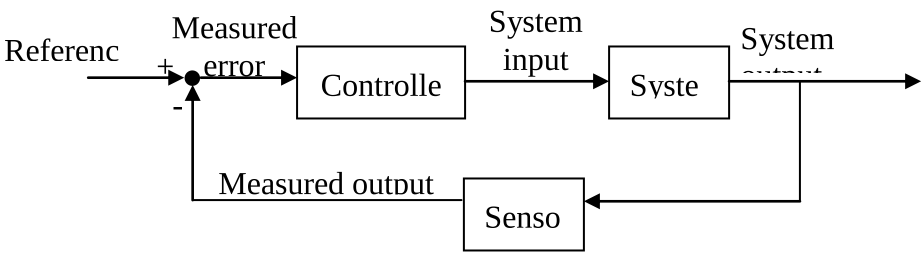


**Figure 2.3 Closed-loop Feedback Control System (with feedback)**

## 2.2 Embedded System

An embedded system is an electronic / electronic-mechanical system designed to perform a specific function, combining both hardware and software.

It can be part of a larger system (other than a computer, such as a keyboard, screen, etc.) and performs pre-programmed instructions for controlling electronic devices. Systems dedicated to data collection enable data acquisition from the outside world.

Embedded systems are often programmed and controlled by a program that incorporates a separate function within a larger mechanical or electrical system, often with real-time computational constraints. Most of these embedded systems are known as real-time systems, this means that real-time properties such as reaction time, worse case resolution means more important design issues. It is often included as part of a complete machine, including hardware and mechanical components. Embedded systems control many of the most commonly used devices today.

Most installed systems are based on the processor / controller. The processor can be either a microprocessor or a microcontroller or a digital signal processor, depending on the domain and application. Most systems integrated with industrial control and monitoring applications use available microprocessors or microcontrollers. Modern integrated systems are often based on microcontrollers (i.e., microprocessors with integrated memory and backup interfaces), but common microprocessors (using external chips for memory and backup circuits) are also used, especially in more complex systems. In any case, the processor (s) used range from general purpose to specialized types in some computational classes, or it could be custom categories for the app at hand.

Embedded systems are designed to perform a specific task rather than a general purpose computer for many tasks. There are some performance limitations over time for reasons such as safety and usability. Embedded hardware / software systems basically send control signals to devices connected to actuators or O / P ports of the system to control a physical change or control the condition of certain devices. In order to control the condition of some devices, sensors connected to inputs.

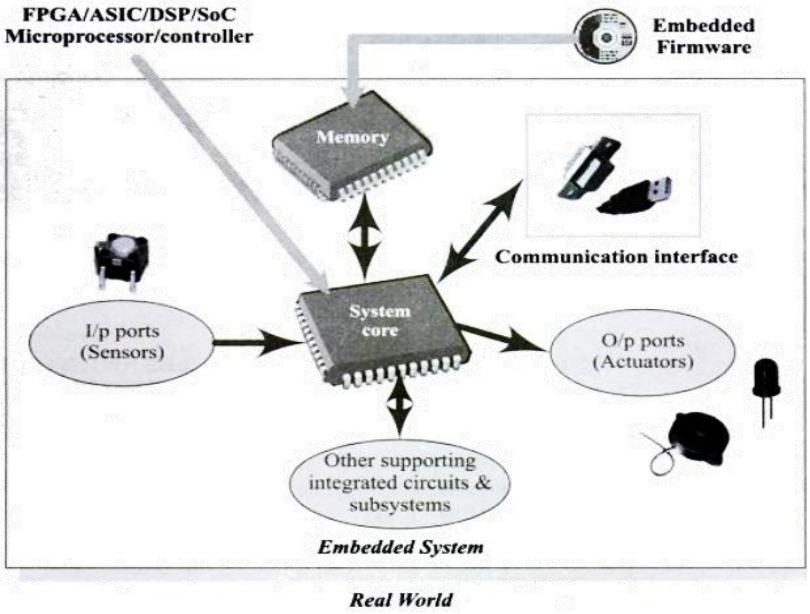


**Figure 2.4 Basic Design for Embedded System**

## 2.3 Introduction to Microcontroller

Microcontrollers are widely used in all fields of science and engineering. Hundreds of electronic products are based on a variety of microcontrollers that simplify our daily tasks.

Microcontroller has become an excellent tool for learning about electronic, digital interfacing and programming. They also provide the capabilities to fairly easily and inexpensively create sophisticated electronic applications application that control real-world devices.

A microcontroller is a tiny complete computer contained in a single chip and it has built-in memory circuit for the program and input/output ports to communicate with the outside-world. The microcontroller's ability is to store and run unique program makes it as the heart of a limitless variety of different electronic applications.

One of the most popular and easy to use microcontroller is "Arduino Mega 2560" microcontroller from Microchip. The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital inputs / outputs (15 of which can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware nodes), 16 MHz crystal oscillator, USB connection power outlet ICSP heading and reset button.

Arduino MEGA 2560 is available for I / O lines, designed for projects that require more RAM, more RAM and more RAM, and for larger projects for the sketch, this is the recommended board for 3D printers and robotic projects. Arduino Mega microcontrollers are built using CMOS technology, which significantly reduces the size and power requirements of current chips.

All the chips in Arduino series are RISC (Reduced Instruction Set Computer) processor which make them relatively easily to write the software. Microcontroller has resets, memory timers, interrupts, input/output (I/O) ports and other capabilities.

## 2.4 Variable Pulse Width Control Servo Position

Pulse Width Modulation (PWM) is a fancy term used to describe a type of digital signal. Pulse width modulation is used in a variety of applications, including state-of-the-art control circuitry. RGB LEDs are often used here to control dimming or to control the orientation of a server. Because users can achieve multiple results in both applications, pulse width modulation can vary the duration of the signal in analog form. The signal is always high (usually 5V) or low (ground). The signal can change to a higher time ratio compared to a fixed time period [14].

Servo motors last a long time and are used in many applications. They are small in size but large in packaging and extremely energy efficient. These features include remote control toys or radio controls. It can operate robots and aircraft. Servo motors can be used in industrial equipment and in robots. It is also used in medicine and food services. The servo circuitry is built inside the motor unit and usually has a position shaft mounted on the gear (as shown below). The motor is controlled by an electric signal that determines the amount of movement of the shaft.

Servos are controlled by sending a pulse width modulation (PWM) that can be changed via a control wire. Minimum pressure; there is a maximum pulse rate and a recurrence rate. The servo motor can usually rotate 90 ° and move a total of 180 °. The neutral position of the motor is defined as the position where the servo has the same amount of rotation in both clockwise and counterclockwise directions.

The PWM sent to the motor determines the position of the shaft and is based on the duration of the pulse transmitted through the control wire. The helicopter will rotate to the desired position. The servo motor detects the pulse rate every 20 milliseconds (ms) and the pulse rate determines the rotation distance of the motor. For example, The 1.5ms pulse turns the motor 90 °. If it is shorter than 1.5ms, it is moved clockwise back to 0 °, and if it is longer than 1.5ms, the server is rotated clockwise 180 ° [15].

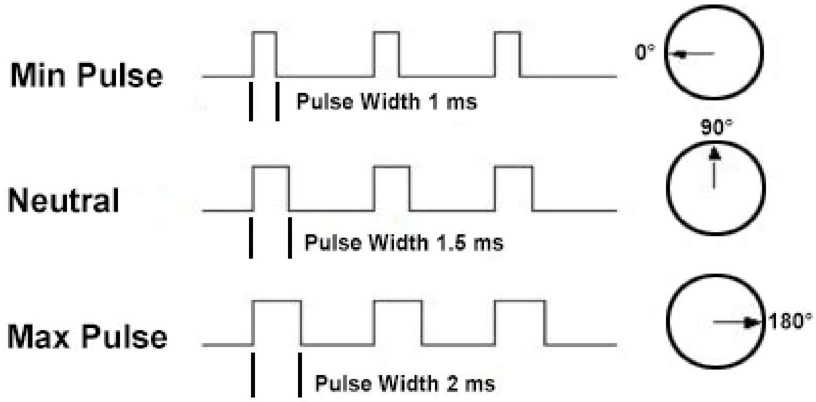


**Figure 2.5 Variable Pulse Width Control Servo Position**

These servers are ordered to be moved, they will move on and hold on to that position. If an external force pushes the servo will resist movement from that position. The maximum amount thrust of a servo is called the torque level of the servo. Servos will not hold their position forever. The position pulse must be repeated to instruct the servo to stay in position.

## 2.5 Graph Theory

Graph are data structures used to represent "connections" between elements. These elements are called nodes. They are objects in real life, represents people or objects. Connections between nodes are called edges.

Graph theory is a mathematical theory of the properties and applications of graphs (networks). Graph theory is the ultimate study of relationships. In mathematics, graph theory is the study of graphs, which are mathematical shapes used to model the relationships between objects. A graph in this content is made up of bounded edges (also called links or lines) (also called nodes or points).

A graph G = (V, E) is a set of vertical V and E vertices connected between u, v ∈ V at each end. Graphing theory helps to calculate many moving parts of a number and dynamics system, as it provides a set of notes and connections that allow you to view everything from urban settings to computer data.

In graph theory, a path is a program of different verticals and edges connecting two nodes. They can be multiple paths from one source node to the destination node. The number of paths can be multiplied by the definition of input, but only a fraction of those paths reduce the sum of the margins, and they are called the shortest paths.

Graph Algorithms: Shortest Path

**G = (V, E)**, weighing function W : E ⟶ R to shape the edges with real value.

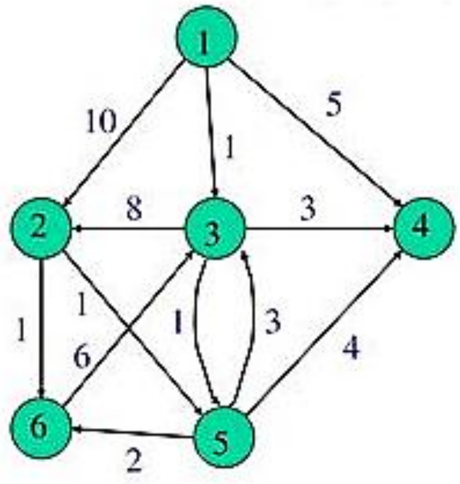


**Figure 2.6 Graph Theory**

Graph theory has many multiplication algorithms. There are several algorithms, including Dijkstras' shortest path algorithm, Breadth First Search algorithm, and Depth First Search algorithm.

Depth First Search Algorithm: The Depth First Search (DFS) is the most fundamental search algorithm used to explore nodes and edges of a graph. It runs with a time complexity of 0(V+E) and is often used as a building block in other algorithm. By itself the DFS isn't all that useful, but when augmented to perform other tasks such as count connected components, determined connectivity, or find articulation points them DFS really shine.

Breadth First Search Algorithm: The BFF is another fundamental search algorithm used to explore nodes and edges of a graph. It runs with a time complexity of 0(V+E) and is often used as a building block in other algorithms. The BFF algorithm is particularly useful for one thing to find the shortest path on unweighted graphs. A BFF starts at some arbitrary node of a graph and explores the neighbours nodes first before moving to the next level neighbours.

Dijkstras' Shortest Path Algorithm: Dijkstras' algorithm is a single source shortest path (SSSP) algorithm for graphs with non-negative edges weights.

Depending on how the algorithm is implemented and what data structures are used the time complexity is typically 0(E*log(V)) which is competitive against other shortest path algorithm.

In this system, Graph theory uses direct graph and calculates using it procedures. The edges of the three floors are calculated with different weights. For edges on the first floor, weight values 3, 2 and 1 are used. For edges on the second floor, weight value 7, 2 and 1. Weight values 11, 2 and 1 are used for edges on the third floor. The path from floor to floor is set to weight 4 and weight 4 is added to each floor. The weights for these edges are set to find the nearest parking lot to the exit. There are two exits and the user must choose OUT1 or OUT2. In this selection, the system calculated the weights specified for the edges and direct graph theory to find the vacant parking lot nearest to the exit.

### 2.5.1 Directed Graph and Weight Graph

It does not indicate the shortest route problem, can be specified for graphs whether directed or mixed. The definition for directional graphs is that the direction of the path needs to be connected in series vertices with the appropriate indication points. The relationships are based on the direction of the edges, it can be one way relationship or two way relationship, but it must explicitly started.

The indicated graph, or graph, is a directional graph at the edges. For example, the edge (u, v) is the edge from note u to note v. The indicated graph (or figure) is a set of nodes connected by polarities, and at the extremes there is a direction associated with them. For example, an arc (x, y) is connected to point from x to y, and arc (y, x) is an inverse link. y is the successor to x and x is the successor to y.

Thus, the graphs are represented by arrows as shown by the vertical pairs (as opposed to the non-linear vertical pairs in an unstructured graph) with the indicated edges of a symbol. The verticals of a number are called the degrees of output. It can be able to now count the number of pointed edges flowing away from them and the number of borders flowing towards them called degrees.

This system uses a directed graph because it has inputs and outputs. Weight graphs: Many graphs are cost-effective, distance the amount and there are weighted edges defined to represent unjust values. The application for graphics is very different. It can be used to analyze electrical circuits, project scheduling, finding the shortest routes, it can be used to build models for the analysis and solution of social problems and many other problems.

- ❖ Step1: Find the free space, which is the vertex, and calculate the distance from the vertex. Set this vertex to current.
- ❖ Step2: From current vertex, calculate the distance to get to OUT1 by adding the specified weights.
- ❖ Step3: Again, find all the vertex from current vertex to OUT2. Calculate the distance up to OUT2 by adding the weights assigned to each edge for each vertex.
- ❖ Step4: Mark the OUT1 (or) OUT2 with the smallest distance from current vertex and repeat from step1.

## 2.6 LCD Display

Lately numerous systems use intelligent clear screen (LCD) modules. Not just figures but letters, the capability to display a variety of words and symbols makes them more protean than familiar 7- member light emitting diode (LED) displays. Although character- grounded modules are limited, a wide range of shapes and sizes are also available. 8, 16, 20, 24 Line lengths of 32 and 40 characters are standard, two and four line performance.

There are numerous different liquid demitasse technologies. For illustration" Supertwist" types offer a better discrepancy and perspective than the aged" twisted nematic" types. Some modules are available with back- lighting so they can be viewed in dim conditions. The backlight may be an" electric light" that requires a high voltage inverter circuit or a simpler LED light.

Most LCD modules meet the standard interface specifications. Eight data lines, a 14- pin access point with three control lines and three power lines (14 holes for welding or IDC connectors) is handed. Connections are housed in two rows of 7 pin figures, one of two common structures, or a single row of 14-pins.

Indeed on utmost displays, the figures are numbered on the LCD print circuit board, but else it's relatively easy to find pin 1. Because this pin number is connected to the ground, it has a thicker PCB path connected to it, and it's generally connected to the essence assiduity at some point. The functionality of each connection is shown in

Table2.1. Pin 1 and 2 have power force lines; Vss and Vdd. The Vdd pin should connect the positive force and the Vss to the 0V force or ground.

LCD module data wastes give 5V dc support (only in a many millimeters), but both 6V and4.5 V support work well, and indeed 3V for some modules is sufficient. As a result, these modules are more battery-effective and effective. It can be done economically.

Pin 3 is a control switch connected to a variable voltage force (this is Vee). A predefined potentiometer is connected between the power force lines that the wiper connected to its discrepancy pin is suitable in utmost cases, but some modules may bear negative conduction. In some cases 17V is lower. For simplicity, connecting this pin to 0V is frequently sufficient.

Pin 4 is the Register Select (RS) line, the first of three command control inputs. When these rulings are short, database transfers to the screen are considered commands, and the read databases on the screen indicate its status. By setting the RS line high, character data can be transferred to the module.

Pin 5 is the Read/ Write (R/ W) line. To write commands or character data in a module, draw the line at least to draw this line or to read character data or status information from its registers.

Pin 6 is the Enable (E) line. This input is used to actually transfer commands or character data between the module and the data lines. When writing to the screen data is only transferred from the height of this signal to the smallest. Still, when reading from the screen Data will be available until the signal is reduced.

Pins 7 to 14 are the eight data machine lines (D0 to D7). Data can be transferred to the screen as a single 8- bit byte or as two 4- bit" nibbles". In the rearmost case, only the below four data lines (D4 to D7) are used. This 4- bit mode is useful when using a microcontroller as it requires smaller input/ output lines.

**Table 2.1 Pinout Functions for all the LCD Type**

| Pin No. | Name | Function |
|---|---|---|
| 1 | Vss | Ground |
| 2 | Vdd | +ve supply |
| 3 | Vee | Contrast |
| 4 | RS | Register Select |
| 5 | R/W | Read/Write |
| 6 | E | Enable |
| 7 | D0 | Data bit 0 |
| 8 | D1 | Data bit 1 |
| 9 | D2 | Data bit 2 |
| 10 | D3 | Data bit 3 |
| 11 | D4 | Data bit 4 |
| 12 | D5 | Data bit 5 |
| 13 | D6 | Data bit 6 |
| 14 | D7 | Data bit 8 |

CHAPTER 3

# HARDWARE AND DESIGN METHODOLOGY

In smart car parking lot management system hardware design consists of Arduino Mega 2560, Ultrasonic Sensor, Push Button, Servo motor and Liquid Crystal Display (LCD).

## 3.1 Arduino Mega 2560 Microcontroller

Arduino has better educational applications. Compatible with both hardware and software. An open source electronic duplication platform based on hardware and software.

The Arduino senses the environment by receiving input from a variety of sensors, including lights, flashlights and it can harm its environment by controlling motors and other actuators. Arduino projects can interact with software running on a computer. The Arduino Mega 2560 is compatible with most other Arduino boards designed for protection. The Arduino Mega 2560 is the ideal choice for projects that require more memory space to use multiple on-board number pins.

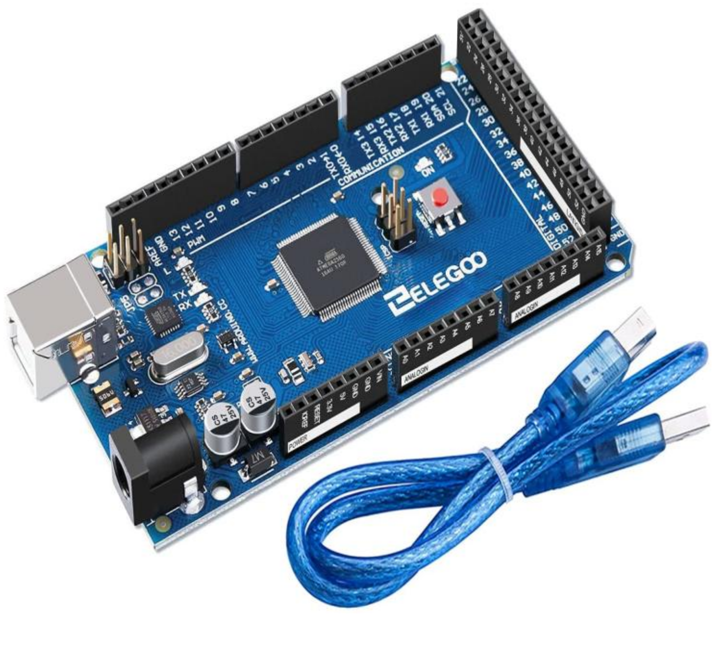


**Figure 3.1 Arduino Mega 2560**

## 3.2 Hardware Overview of Arduino Mega 2560

In addition to 54 digital I / O pins and 16 analog inputs, it has four UART (hardware serial numbers), 16 MHz crystal oscillator, USB connection, power outlet, includes ICSP header and reset button.

It provides everything needed to support a microcontroller. It is simply connect to a computer with a USB cable or power it up with an AC-to-DC adapter or battery to get started.

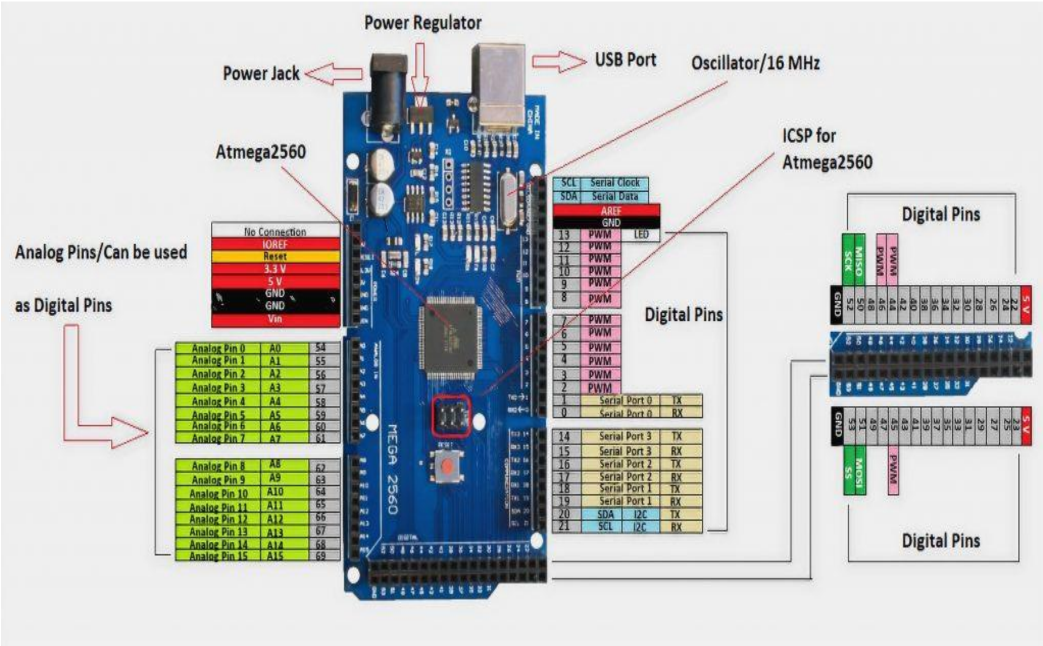


**Figure 3.2 Pin Diagram of Arduino Mega 2560**

The 86 pins, 72 pins available on the mega board are related to the input and output. These 54 pins (D0 to D53) are digital IO pins that can be used to configure applications using pinMode (), digitalWrite () and digitalRead () functions.

All of these digital IO pins can source or submerge current 20mA (maximum 40mA is allowed). Another feature of digital IO pins is the internal pull-up resistor (not connected to the original). The value of the internal resistance will be in the range of 20KΩ to 50KΩ.

There are also 16 analog input pins (A0 to A15). All analog input pins provide a 10-bit resolution ADC feature that can be read using the analogRead () function. An important point about analog input pins is that can be able to configure them as

Digital IO pins if needed. Digital IO pins 2 - 13 and 44 - 46 can output 8-bit PWM signals. It can use the analogWrite () function for this [8].

## 3.3 Simplification and Security

The ATmega + 2560 on the Elegoo Mega comes preprogrammed with a bootloader that allows the system. The Mega R3 comes with a resettable fuse to protect their computer's USB ports from shorts and over-current. Most computers protect their interior, but the fuse provides an extra layer of protection.

If it uses a USB port with more than 500 mA, **t**he fuse will automatically disconnect until the short or overload is removed. Memory Store: ATMega + 2560 of ELEGOO Mega R3 Controller Board contains 256 KB of flash memory for coding (8 KB of which is used as bootloader), 8 KB of Spram and EEPROM 4 KB (read and write with library).

## 3.4 Push Button

Thrust is a simple switch that controls some part of a machine or process. The buttons are usually made of plastic or metal. Momentary, Latching, Electric and Pneumatic are types of buttons used for "Push to make" and "Push to break”.

**Figure 3.3 Push Button**

A button or simple button is a simple button that controls some part of a machine or process. The surface is usually flat or shaped and can be adjusted to fit a human finger or hand **[**9].

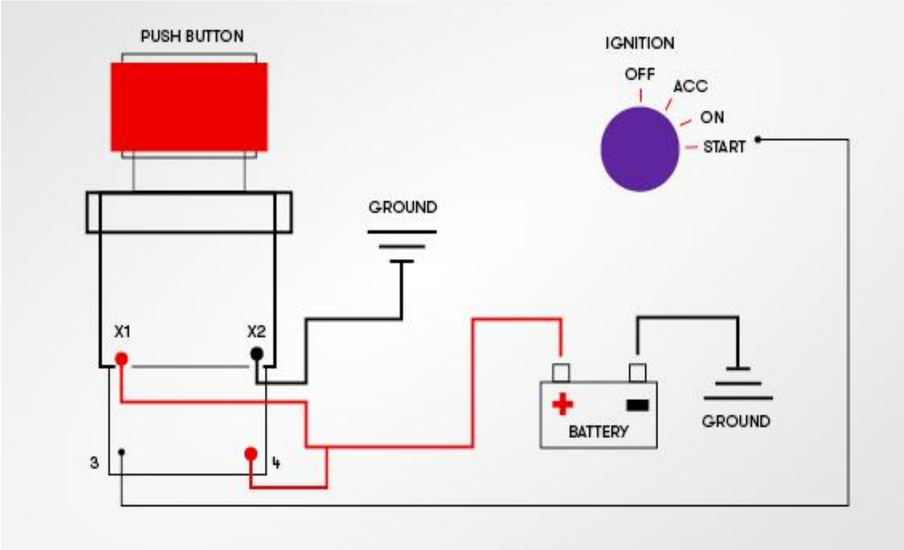


**Figure 3.4 Pin Diagram of Push Button**

Specifications of Push Button:

- Item: Push Button Switches and Non-lighted models
- Mechanical: Momentary action
- Electrical: 500,000 operations min.
- Ambient operating Temperature *1:  -25· to 70 C

## 3.5Ultrasonic Sensor

A sensor is a transducer device that converts energy from one form of energy to another for measurement or control purposes. Changes in the system environment are detected by sensors connected to the input of the system embedded in the variables.

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves are faster than the speed of sound that can be heard (that is, the sound that humans can hear).

Ultrasonic sensors have two main components: the transmitter (which emits sound using pizzo electron crystals) and the receiver (after contact with the target). In this system, Ultrasonic sensors use 28 parking spaces and two entrances.


**Figure 3.5 Ultrasonic Sensor**

The distance to the target is determined using an ultrasonic sensor. The ultrasonic sensor sends the pulse through the trigger pin to the object, which receives it back from the echo pin and sends it to the Arduino board.

Use IO trigger for signal level at least 10µs higher. The module automatically sends eight 40KHz signals to determine if the pulse signal is returning. The ultrasonic sensor (HC-SR04) supports 2cm-400cm or 1 inch to 13 ft measuring distance, with accuracy up to 3mm [10].

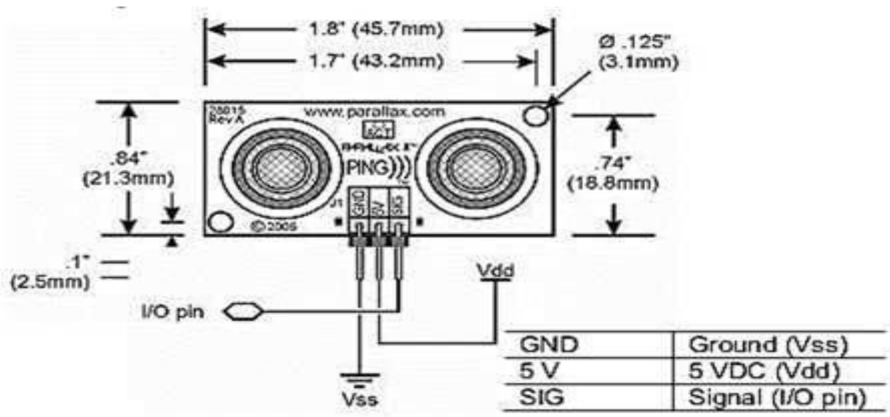


**Figure 3.6 Pin Diagram of Ultrasonic Sensor**

Specifications of the Ultrasonic Sensor

- Working Voltage:  DC 5v
- Working Current: 15mA
- Working Frequency: 40KHz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angel: 15 degree
- Trigger Input Signal: 10µs TTL pulse

- Echo Output Signal: Input TTL lever signal and the range in proportion

## 3.6 Servo Motor

The servo motor has an angular position that is not present in a normal motor. A rotary actuator or motor that allows precise control of acceleration and velocity.

It uses a standard motor and attaches to a sensor for location feedback. The controller is the most sophisticated part of a servo motor. In this system, the servo motor is used to turn on and off.


**Figure 3.7 Servo Motor**

A servo motor is a rotary actuator or linear actuator with an angular or linear position. A rotary actuator or linear actuator that precisely controls the speed and velocity. A servo motor is an electrical device that produces torque and velocity depending on the supply current and voltage.The servo motor operates as part of a closed-loop system that supplies torque and velocity as ordered by the server controller, using a circuit breaker [11].

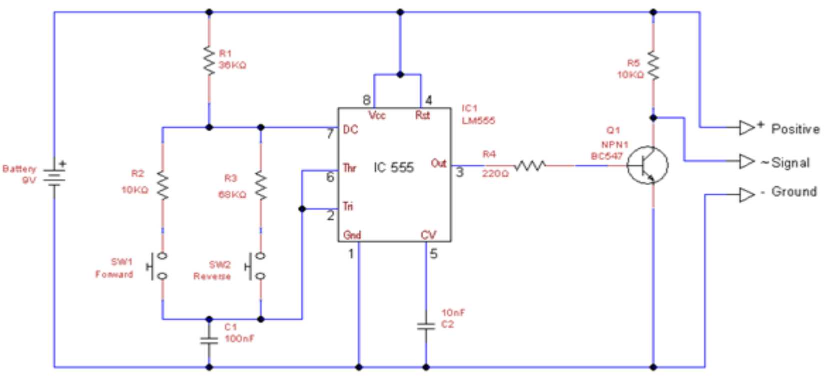


**Figure 3.8 Pin Diagram Servo Motor**

Specifications of the Servo Motor

- Size: 23 12.2 29mm
- Weight: 9g
- Torque: 1.6KG/CM
- Reaction speed: 0.12-0.13 seconds/60 degrees
- Operating temperature: -30 ~ +60 degrees
- Dead Setting: 5 microseconds
- Rotation angel: 180 degrees
- Maximum Working voltage: 3.5-6v

## 3.7 Liquid Crystal Display (I2C)

Liquid Crystal Display is a digital device that uses crystal cells to change the reflection in the applied electric field. Portable computer displays, it can be use calculators and digital clocks. But the most common is the 16 character * 2 lines display.

Alphanumeric dot matrix liquid crystal display (LCDs) Symbols used to display alphanumeric characters and icons. These modules are used to refresh the surface, control tools for easy implementation of logic for multiple inserts and updates, includes circuits related to driver character generator RAM / ROM [15].


**Figure 3.9 Liquid Crystal Display**

Alphanumeric LCD screens have become so popular for microcontroller applications that they can be integrated into a project in many different ways.

I2C_LCD is an easy display module to use and it makes the display easier. Using the reduction of hassle creativity, it is able to focus on the core of the work. Users can get complex graphics and text display features with just a few lines of code in the Arduino Library for I2C_LCD. In this system LCD is displayed the range of Servo Motor and it is connecting to Arduino Mega 2560. The pin diagram of the LCD is shown in Figure 3.10.

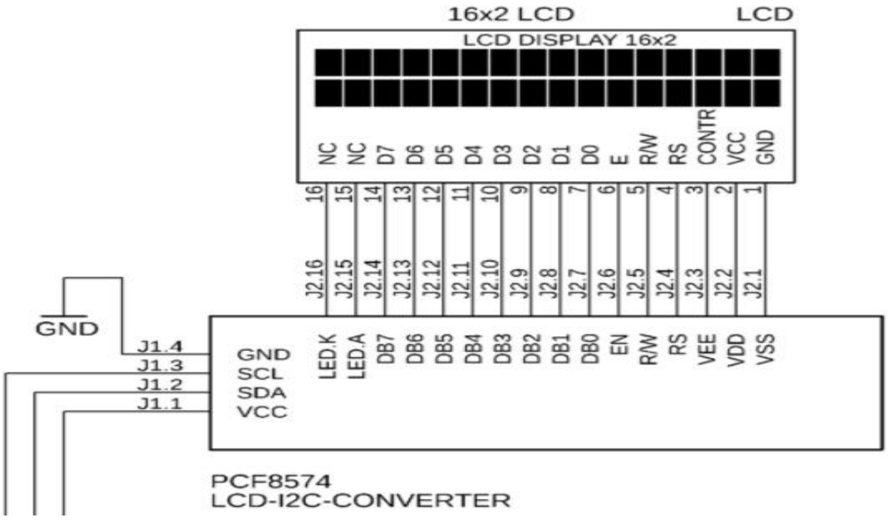


**Figure 3.10 Pin Diagram of LCD 12C**

Specifications of the LCD

- Backlight Color: Blue Backlight Wide viewing angle and high contrast
- Chip: HD44780
- LCM Type: Characters
- Line and Characters: 2-Lines X 16 Characters
- DC Input: Operate with 5V DC
- Module Dimension: Approx.80mm x 35mm x 9mm
- Viewing Area Size: Approx. 64.5mm x 16mm

## 3.8 Key Features of Arduino Mega 2560

Arduino Mega 2560 can be operated with oscillator up to 16MHz clock input. The explanations of ATmega 2560 key features are shown in Table 3.1.

**Table 3.1 Key Features of ATmega 2560**

| **Microcontroller** | **ATmega2560** |
|---|---|
| Operation Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 15 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40mA |
| DC Current for 3.3V Pin | 50mA |
| Flash Memory | 256 KB of which 8KB used by bootloader |
| SRAM | 8KB |
| EEPROM Data Memory | 4KB |
| Clock Speed | 16Mhz |

CHAPTER 4

# IMPLEMENTATION AND EXPERIMENTAL RESULTS

Software implementation and hardware implementation is required to develop the smart car parking lot system. Matlab programming software is used to control the route of the parking system using graph theory. Arduino Mega 2560 and sensors are used to carry out the hardware implementation.

## 4.1 Software Implementation of the System

The flowchart for the application program is given in Figure 4.1. In this thesis, graph theory is used to control the devices.

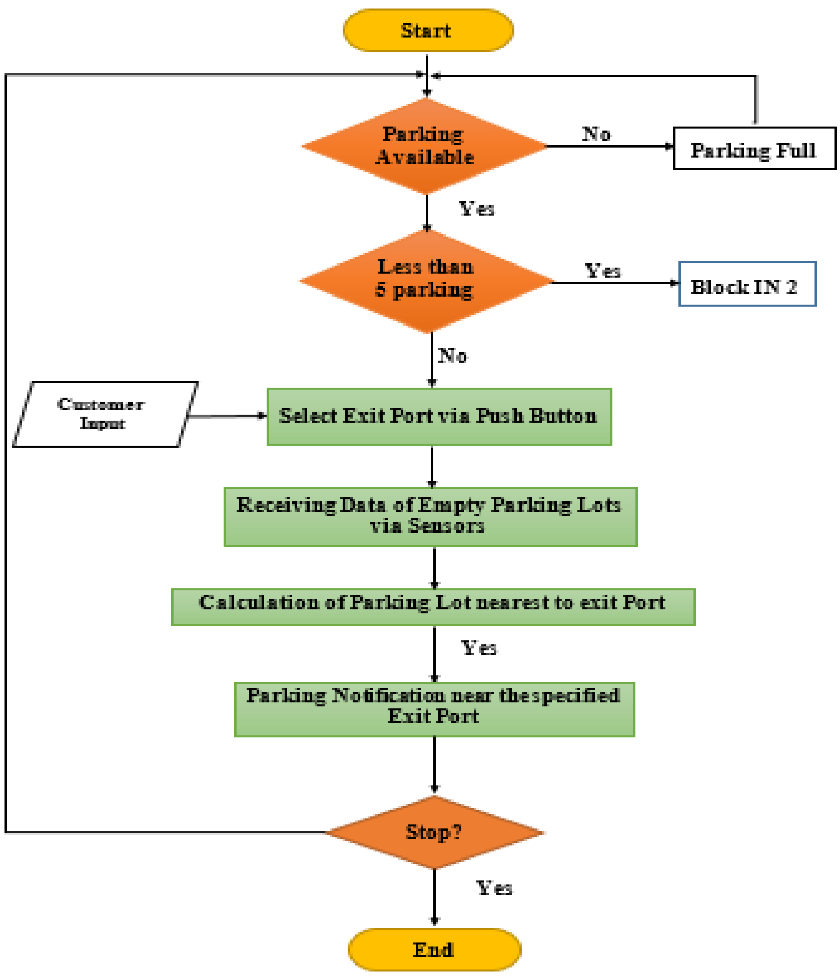


**Figure 4.1 Flow Chart of Smart Car Parking System**

As soon as a car enters, the initialization process begins. The gate will not be opened if the parking is not free. With only five parking spaces left, entrance IN 2 will be closed. When a car enters, the user selects exit port via push button. Receiving data of empty parking lots via sensor, it calculates the closet exit to display on LCD. If the car comes to the exit without parking, the gate will open. If the car does not enter the parking lot, this system will be started. After leaving the car, the system will be given a record of leaving the car park.

### 4.1.1 Result for First Floor

Then, coding is run in Matlab Programming.

- ❖ First, it is chosen to enter from Entrance 1(IN1), Exit 3(OUT1) and Floor 1, the result display 28 and next choose for Entrance 1(IN1), Exit 4(OUT2) and Floor1, the result display 24.

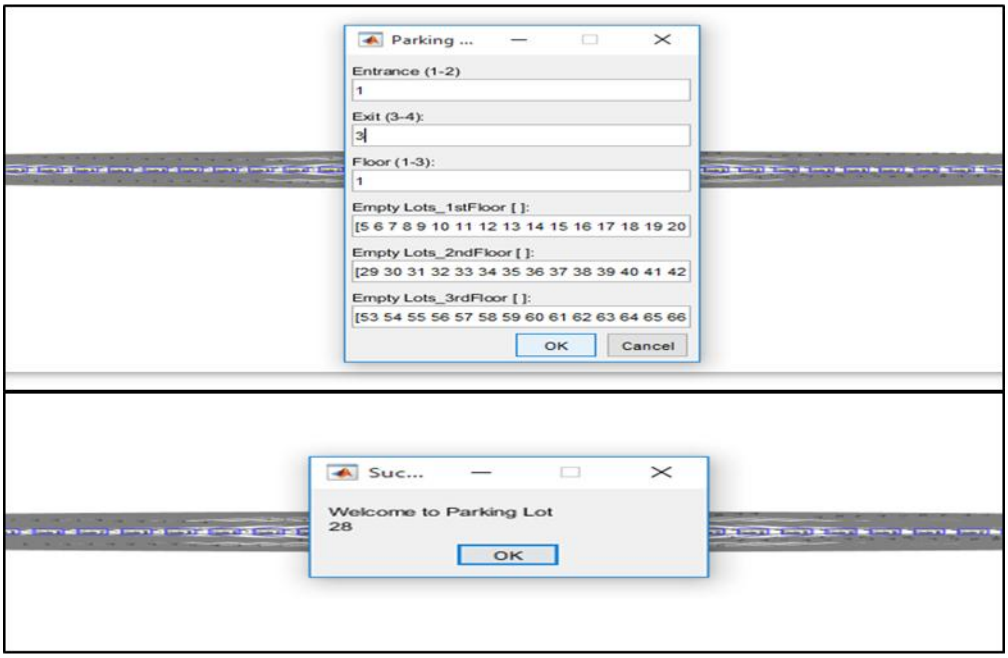


**Figure 4.2 Result for Entrance 1(IN1), Exit 3(OUT1) and Floor 1**

- ❖ Second, it is chosen to enter from Entrance 2(IN2), Exit 3(OUT1) and Floor 1, the result display 28 and next choose for Entrance 2(IN2), Exit 4(OUT2) and Floor1, the result display 24.

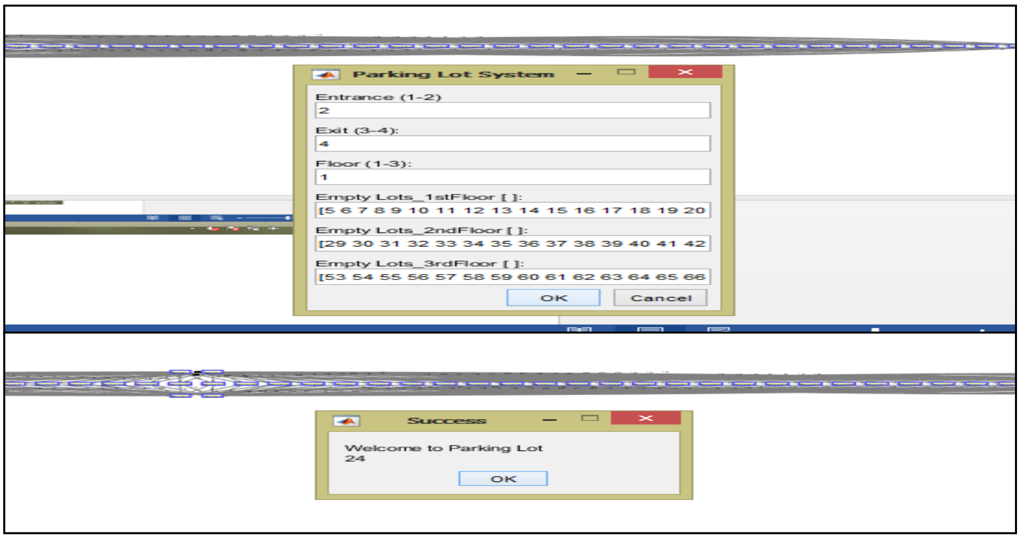


**Figure 4.3 Result for Entrance 2(IN2), Exit 4(OUT1) and Floor 1**

### 4.1.2 Result for Second Floor

❖ Step1, , it is selected to enter from Entrance 1, Exit 3 and Floor 2, the result display 52 and next choose for Entrance 1, Exit 4 and Floor2, the result display 48.

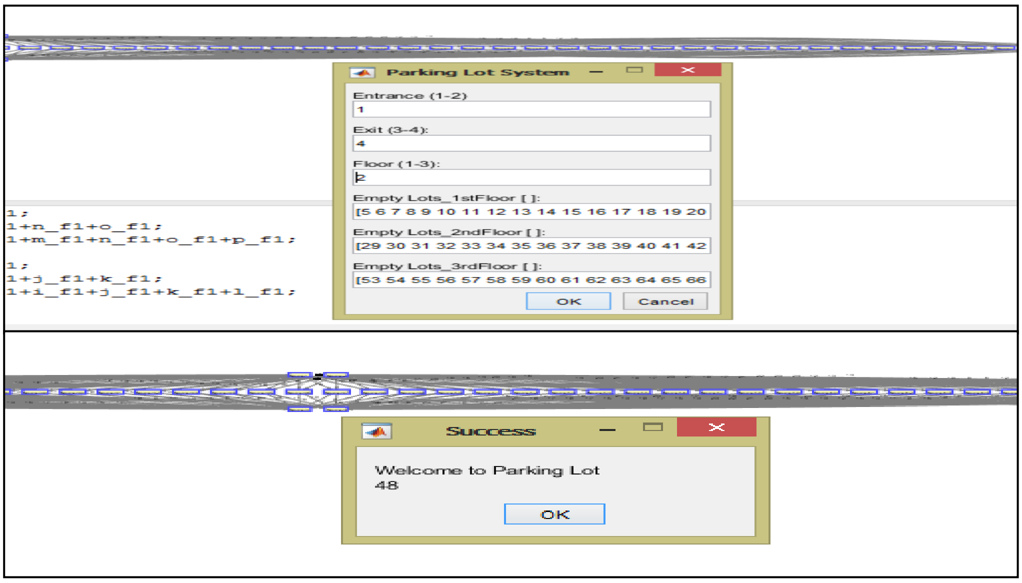


**Figure 4.4 Result for Entrance 1(IN1), Exit 4(OUT1) and Floor 2**

❖ Step2, it is selected to enter from Entrance 2(IN2), Exit 3(OUT1) and Floor 2, the result display 52 and next choose for Entrance 2(IN2), Exit 4(OUT2) and Floor2, the result display 48.

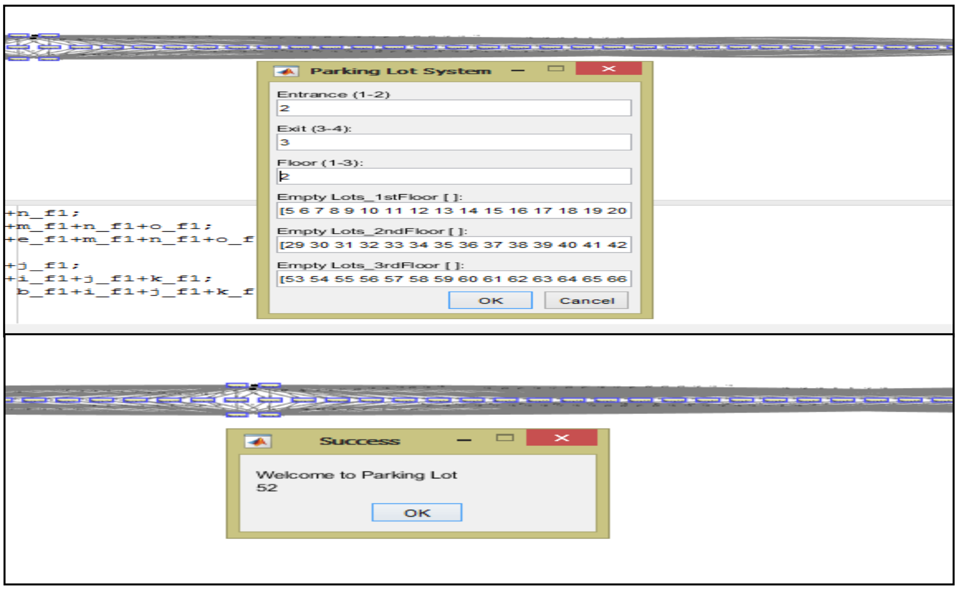


**Figure 4.5 Result for Entrance 2(IN2), Exit 3(OUT1) and Floor 2**

### 4.1.3 Result for Third Floor

❖ First, it is selected to enter from Entrance 1(IN1), Exit 3(OUT1) and Floor 3, the result display 76 and next choose for Entrance 1(IN1), Exit 4(OUT2) and Floor3, the result display 72.

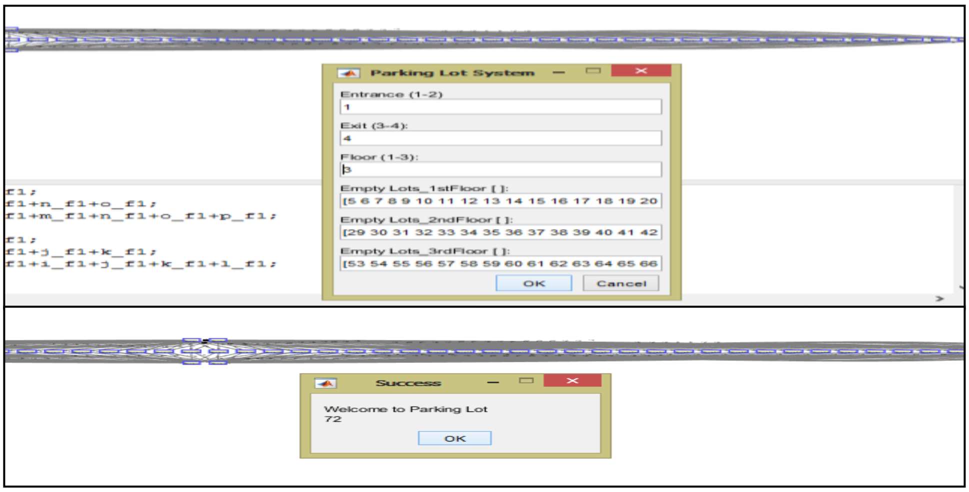


**Figure 4.6 Result for Entrance 1(IN1), Exit 4(OUT1) and Floor 3**

❖ Second, it is selected to enter from Entrance 2(IN2), Exit 3(OUT1) and Floor 3, the result display 76 and next choose for Entrance 2(IN2), Exit 4(OUT2) and Floor3, the result display 72.


**Figure 4.7 Result for Entrance 2(IN2), Exit 3(OUT2) and Floor 3**

❖ Another is that when the parking spaces on each floor are full, the result is that there is no parking space.

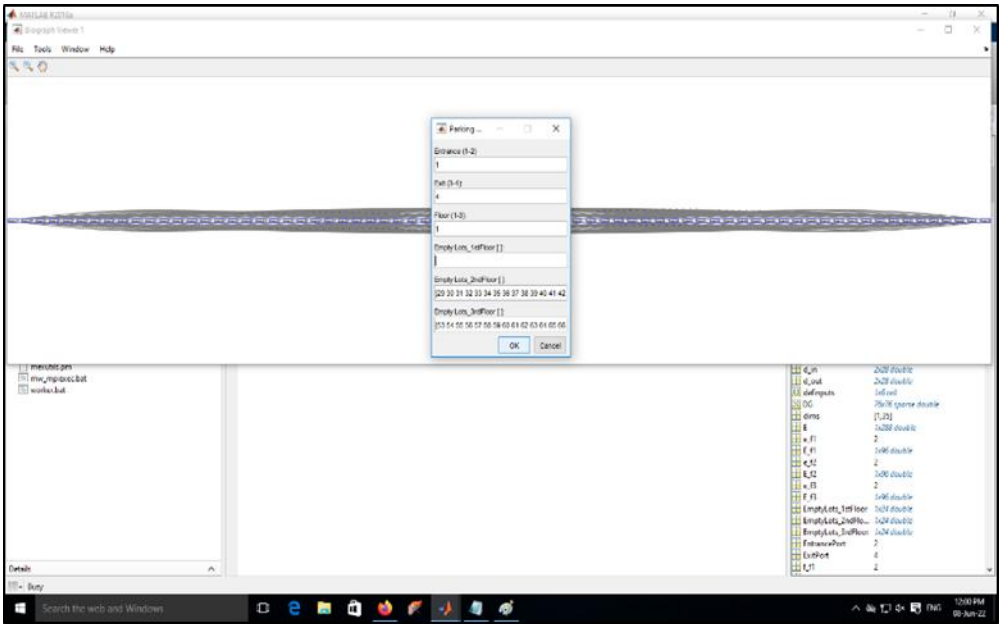

**Figure 4.8 Result for No Space on the First Floor**


**Figure 4.9 Result for No Parking Space**

## 4.2 Hardware Implementation of Smart Car Parking System

The smart car parking system can be operated with the following parts Push Button, Ultrasonic Sensor, Servo Motor, LCD (Liquid Crystal Display) and Arduino Mega 2560.

### 4.2.1 Circuit Design

The diagram shows the input and output components connected to the Arduino Mega 2560. The input component is comprised of ultrasonic sensors, push buttons, an LCD, and a servo motor. The parking system consists of four main circuits are a sensor circuit, a counter circuit, a gate circuit and LCD display circuit show in figure 4.10.

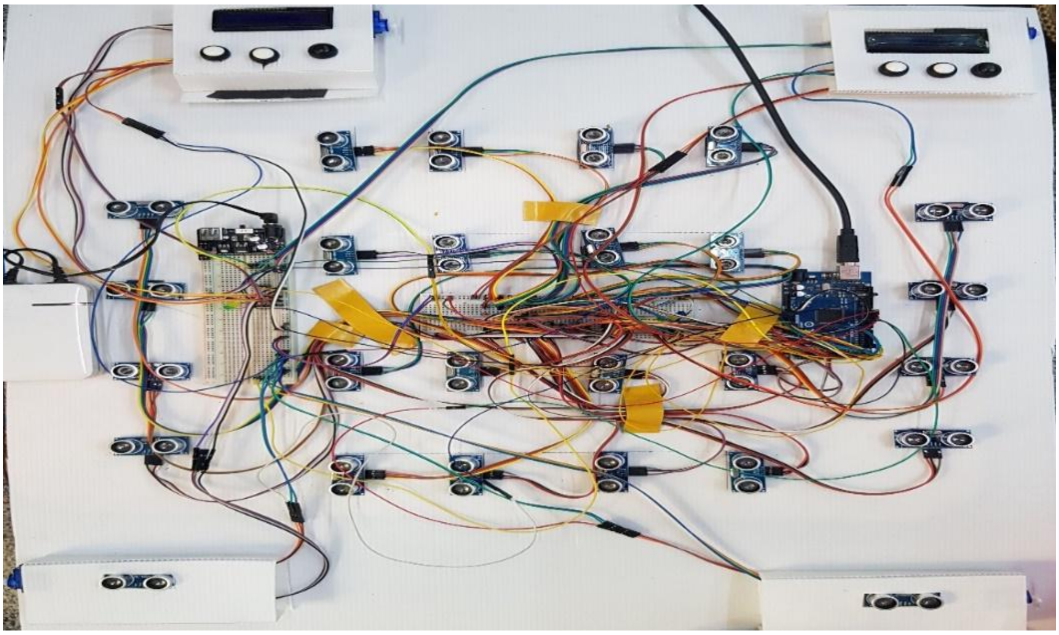

**Figure 4.10 Prototype for Smart Car Parking System**

Ultrasonic sensors are used as both inputs to the main circuit, but not for the gate. Push button is used as input. Then, an analog digital converter is used to convert an analog signal from an ultrasonic sensor to a digital signal. All this circuit are controlled by Arduino Mega 2560. It controls the circuit based on the software installed on the Arduino.

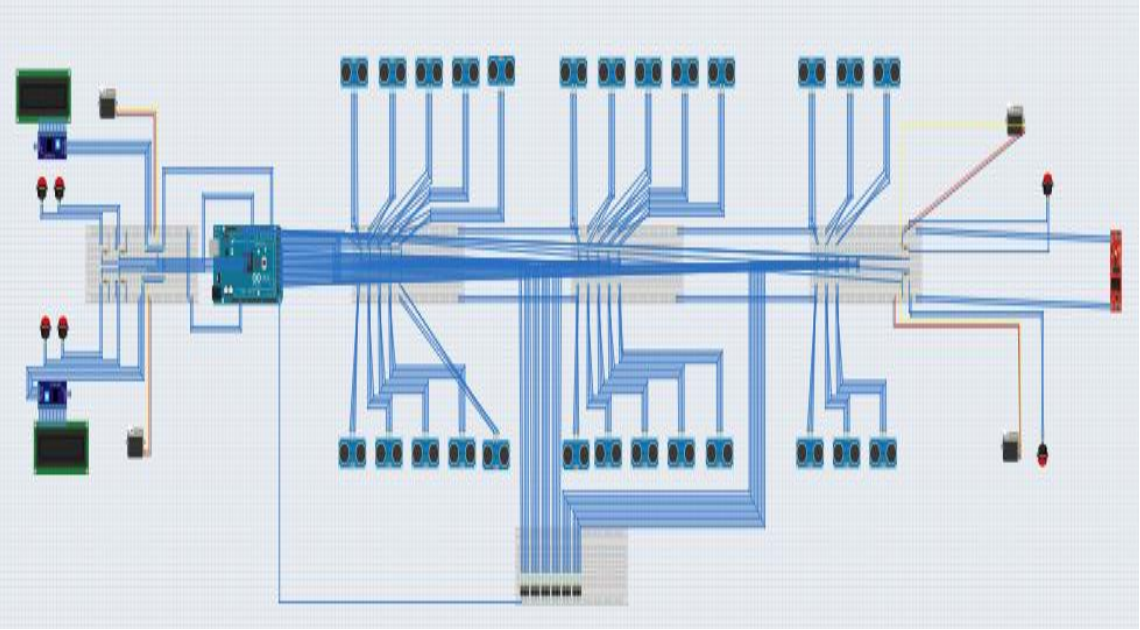

**Figure 4.11 Printed Diagram of Smart Car Parking System**

The programming is executed in the Arduino. The output is displayed on a Liquid Crystal Display (LCD). After parking an ultrasonic sensor in the parking lot scans the car's current and sends a captured signal to the counter circuit. The counter will then be recalculated to update the availability status.

In term of sensors, the sensors interact with each other. The 24 sensors in the parking system work, with one sensor sending the trigger and the other sending the trigger. The sensors send triggers one every 1 second to monitor all parking. After checking the system will send the result back to the machine whether the car parking is full or not. If the parking lot is less than 5 cars, it does not mean that the entrance is less than 5 cars. Only 24 sensors will build an arduino to determine which holes are missing.

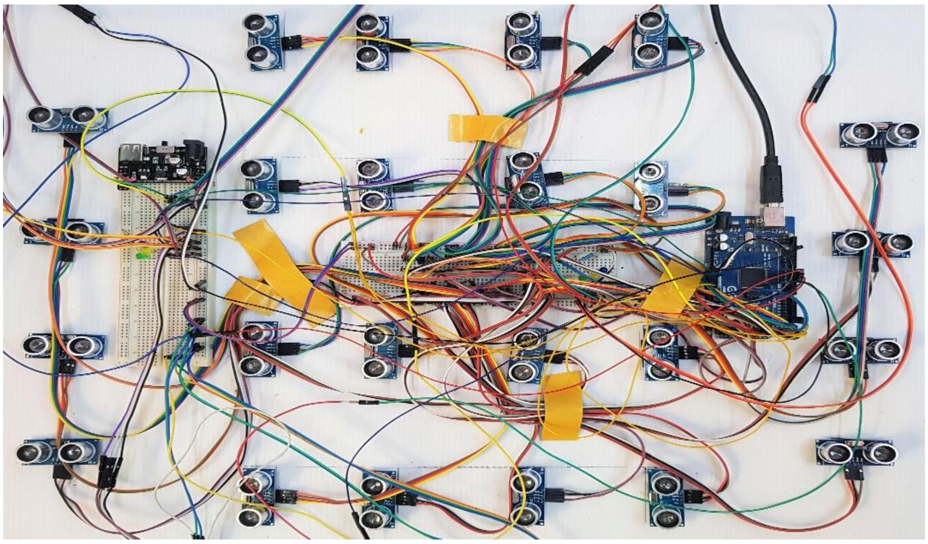

**Figure 4.12 Prototype Arduino Connects to Sensors**

The last part is for exits. The exits are simple. There are two sensors for the two exits. The sensors detected any objects arriving at the gate and open the gate.

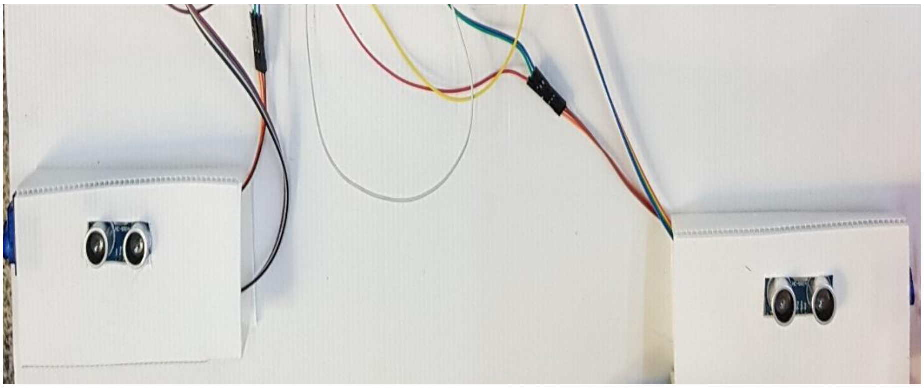

**Figure 4.13 Prototype for Exit Two Gates**

## 4.3 System Experiments

Many methods from the shortest path problem use Graph theory. Calculation for shortest path show in Figure 4.14.

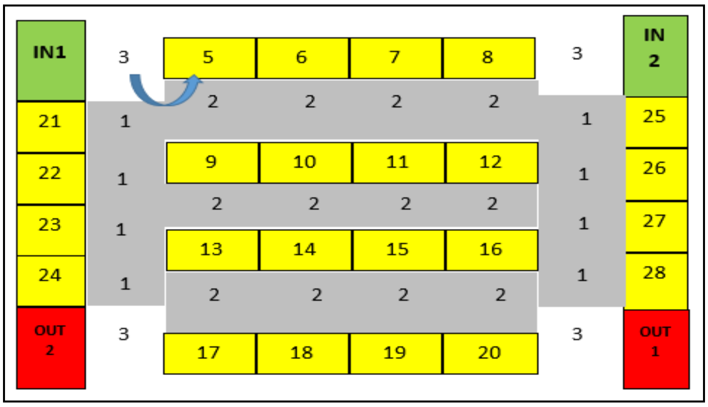


**Figure 4.14 Calculate for Weight Value**

Theoretically, the three floors are calculated with different weights. There are 24 cars on each floor. The car park downstairs has a weight value of 3 for the two entrances. To go up to the second floor, to add the weight value for the roundabout to the four weight values and must have a weight value of 7. For the last floor, to add the weight value for the roundabout to the four weight values and must have a weight value of 11. While it is going from node1 (IN1) to node5, the first weight value is 5, the second weight value is 23 and the last weight value is 27, then choose for the shortest path is weight value 5.

While it is going from node5 to exit (OUT1), the first weight value is 15, the second weight value is 18 and the last weight value is 17, then choose for the shortest path is weight value 12 show in Figure 4.15.

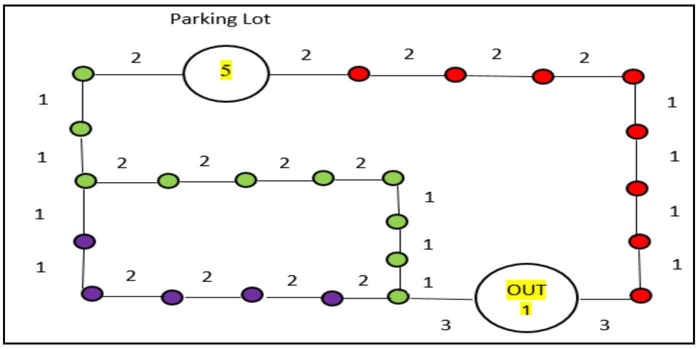


**Figure 4.15 Three Paths with Different Weight Values of a Node 5 to OUT1**

**Table 4.1 Result for First Floor Plan of the Shortest Path Numbers**

| Arv/Dsb | IN1 | IN2 | OUT1 Line1 | OUT1 Line2 | OUT1 Line3 |
|---|---|---|---|---|---|
| IN1 | - | - | - | | |
| IN2 | - | - | - | | |
| OUT1 | - | - | - | - | - |
| OUT2 | - | - | - | - | - |
| 5 | - | - | 15 | 18 | 17 |
| 6 | - | - | 13 | 20 | 19 |
| 7 | - | - | 11 | 11 | 21 |
| 8 | - | - | 9 | 24 | 23 |
| 9 | - | - | 15 | 18 | 17 |
| 10 | - | - | 13 | 20 | 19 |
| 11 | - | - | 11 | 11 | 21 |
| 12 | - | - | 9 | 24 | 23 |
| 13 | - | - | 11 | 14 | 19 |
| 14 | - | - | 9 | 12 | 21 |
| 15 | - | - | 7 | 10 | 23 |
| 16 | - | - | 5 | 8 | 25 |
| 17 | - | - | 11 | 19 | 21 |
| 18 | - | - | 9 | 21 | 23 |
| 19 | - | - | 7 | 23 | 25 |
| 20 | - | - | 5 | 25 | 27 |
| 21 | - | - | 15 | 16 | 16 |
| 22 | - | - | 14 | 15 | 17 |
| 23 | - | - | 13 | 16 | 18 |
| 24 | - | - | 12 | 17 | 19 |
| 25 | - | - | 7 | 25 | 24 |
| 26 | - | - | 6 | 26 | -25 |
| 27 | - | - | 5 | 27 | 26 |
| 28 | - | - | 4 | 28 | 27 |
| | IN1 | IN2 | OUT1 | OUT1 | OUT1 |

This data is taken the calculated weights and directions shown in Table 4.1. Theoretically, the graph is used to check the nearest car park with exit 3 (OUT1) or exit 4 (OUT2).

If the space is available, it will indicate the destination. In theory, Weight and trajectory for the shortest route are calculated using Matlab. It is then written in Arduino code with an if-then rule, depending on its weight.

The shortest path for car parking search can be done using Java, Python, C++, C, Matlab, C# and Java script programming. Only the required values will be calculated in matrix format. Many methods from the shortest path problem use Graph theory.

### 4.3.1 Calculation for First Floor

First floor plan of the parking system and calculation for shortest path show in Figure 4.16.

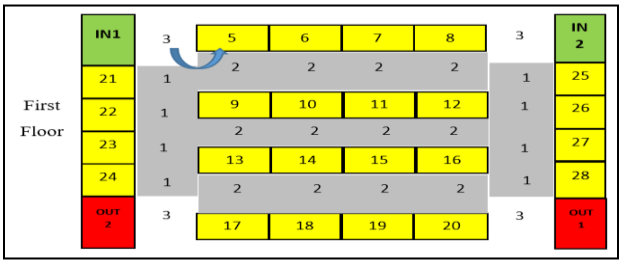


**Figure 4.16 First Floor Plan of the Parking System**

**Table 4.2 Shortest Path Number Needed is taken for First Floor**

| Arv\Dst | IN1 | IN2 | OUT1 | OUT2 |
|---|---|---|---|---|
| IN1 | - | - | - | - |
| IN2 | - | - | - | - |
| OUT1 | - | - | - | - |
| OUT2 | - | - | - | - |
| 29 | - | - | 19 | 13 |
| 30 | - | - | 17 | 15 |
| 31 | - | - | 15 | 17 |
| 32 | - | - | 13 | 19 |
| 33 | - | - | 19 | 13 |
| 34 | - | - | 17 | 15 |
| 35 | - | - | 15 | 17 |
| 36 | - | - | 13 | 19 |
| 37 | - | - | 15 | 9 |
| 38 | - | - | 13 | 11 |
| 39 | - | - | 11 | 13 |
| 40 | - | - | 9 | 15 |
| 41 | - | - | 15 | 9 |
| 42 | - | - | 13 | 11 |
| 43 | - | - | 11 | 13 |
| 44 | - | - | 9 | 15 |
| 45 | - | - | 19 | 11 |
| 46 | - | - | 18 | 10 |
| 47 | - | - | 17 | 9 |
| 48 | - | - | 16 | 8 |
| 49 | - | - | 11 | 19 |
| 50 | - | - | 10 | 18 |
| 51 | - | - | 9 | 17 |
| 52 | - | - | 8 | 16 |
| | IN1 | IN2 | OUT1 | OUT2 |

## 4.3.2 Calculating for Second Floor

Second floor plan of the parking system and calculation for shortest path show in Figure 4.17.

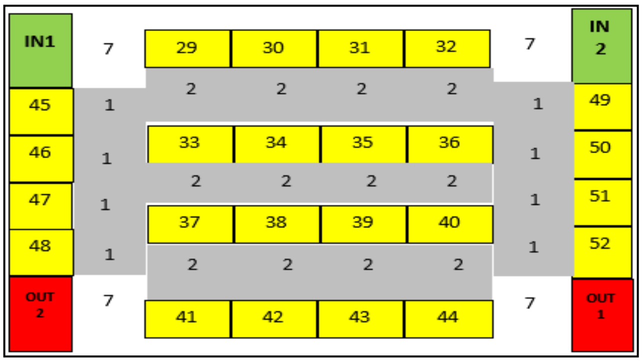


**Figure 4.17 Second Floor plan of the Parking System**

**Table 4.3 Shortest Path Number needed is taken for Second Floor**

| Arv\Dst | IN1 | IN2 | OUT1 | OUT2 |
|---|---|---|---|---|
| IN1 | - | - | - | - |
| IN2 | - | - | - | - |
| OUT1 | - | - | - | - |
| OUT2 | - | - | - | - |
| 29 | - | - | 19 | 13 |
| 30 | - | - | 17 | 15 |
| 31 | - | - | 15 | 17 |
| 32 | - | - | 13 | 19 |
| 33 | - | - | 19 | 13 |
| 34 | - | - | 17 | 15 |
| 35 | - | - | 15 | 17 |
| 36 | - | - | 13 | 19 |
| 37 | - | - | 15 | 9 |
| 38 | - | - | 13 | 11 |
| 39 | - | - | 11 | 13 |
| 40 | - | - | 9 | 15 |
| 41 | - | - | 15 | 9 |
| 42 | - | - | 13 | 11 |
| 43 | - | - | 11 | 13 |
| 44 | - | - | 9 | 15 |
| 45 | - | - | 19 | 11 |
| 46 | - | - | 18 | 10 |
| 47 | - | - | 17 | 9 |
| 48 | - | - | 16 | 8 |
| 49 | - | - | 11 | 19 |
| 50 | - | - | 10 | 18 |
| 51 | - | - | 9 | 17 |
| 52 | - | - | 8 | 16 |
| | IN1 | IN2 | OUT1 | OUT2 |

## 4.3.3 Calculating for Third Floor

Third floor plan of the parking system and calculation for shortest path show in Figure 4.18.

Third Floor

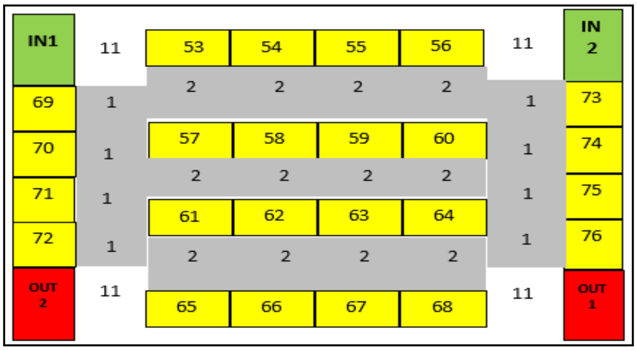


**Figure 4.18 Third Floor Plan of the Parking System**

**Table 4.4 Shortest Path Number Needed is taken for Third Floor**

| Arv\Dst | IN1 | IN2 | OUT1 | OUT2 |
|---|---|---|---|---|
| IN1 | - | - | - | - |
| IN2 | - | - | - | - |
| OUT1 | - | - | - | - |
| OUT2 | - | - | - | - |
| 53 | - | - | 23 | 17 |
| 54 | - | - | 21 | 19 |
| 55 | - | - | 19 | 21 |
| 56 | - | - | 17 | 23 |
| 57 | - | - | 23 | 17 |
| 58 | - | - | 21 | 19 |
| 59 | - | - | 19 | 21 |
| 60 | - | - | 17 | 23 |
| 61 | - | - | 19 | 13 |
| 62 | - | - | 17 | 15 |
| 63 | - | - | 15 | 17 |
| 64 | - | - | 13 | 19 |
| 65 | - | - | 19 | 13 |
| 66 | - | - | 17 | 15 |
| 67 | - | - | 15 | 17 |
| 68 | - | - | 13 | 19 |
| 69 | - | - | 23 | 15 |
| 70 | - | - | 22 | 14 |
| 71 | - | - | 21 | 13 |
| 72 | - | - | 20 | 12 |
| 73 | - | - | 15 | 23 |
| 74 | - | - | 14 | 22 |
| 75 | - | - | 13 | 21 |
| 76 | - | - | 12 | 20 |
| | IN1 | IN2 | OUT1 | OUT2 |

## 4.4 Experimental Result

The first architecture had two entrances IN1 (Entrance1) and IN2 (Entrance2), two exits OUT1 (Exit3) and OUT2 (Exit4), and the parking lot had 24 cars.

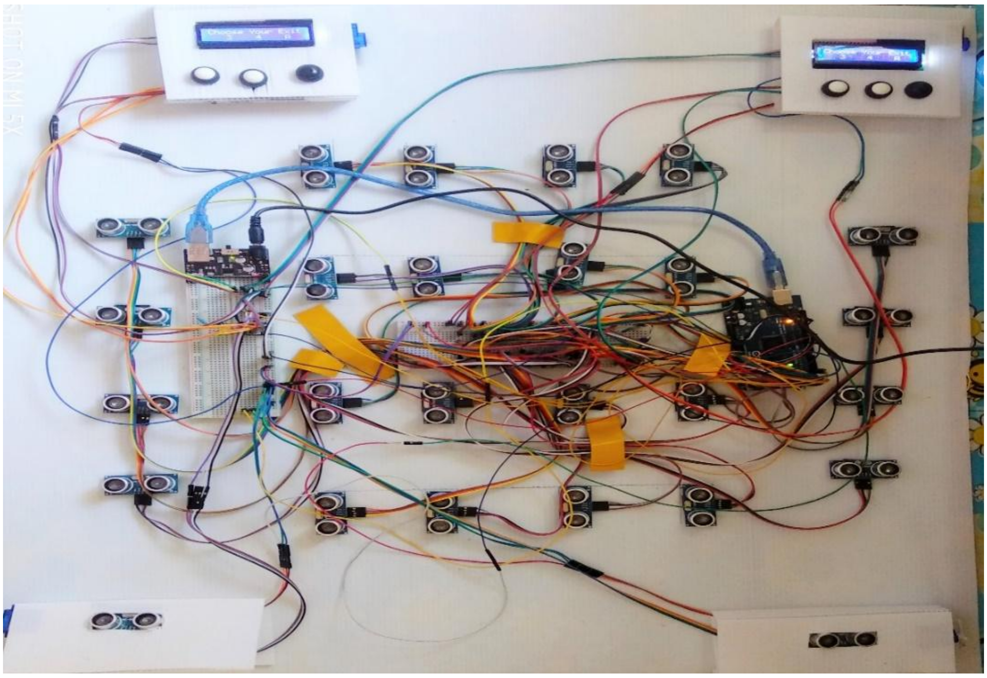

**Figure 4.19 Complete Circuit Diagram for Smart Car Parking System**

The input will be selected from the push buttons. The entrance numbers are IN1 and IN2, and the exit numbers are OUT1 and OUT2. Each entrance port will have an LCD screen and three buttons (3, 4 and A). This is a system that points to the parking lot closest to the exit when entering the car. Selecting push button number OUT1 (Exit3) will find the shortest path closest to OUT1 (Exit3) or select push button number OUT2 (Exit4) to find the shortest path closest to OUT2 (Exit4). Alternatively, if the user selects the A button, the best shortest path will be auto-selected as shown in Figure 4.20.


**Figure 4.20 Display the First Steps**

The system has two entrances, but if there are less than five cars, IN2 (Entrance2) will shut down automatically and the system can only enter through IN1 (Entrance1), as shown in Figure 4.21.

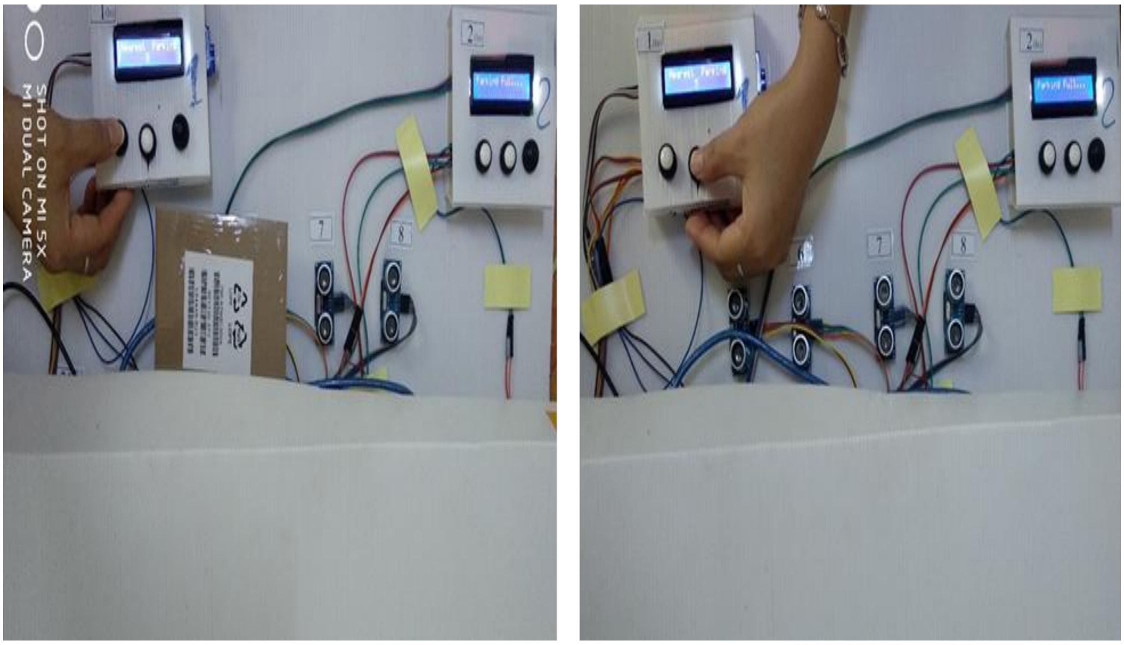


**Figure 4.21 Display for Entrance 2 will be Automatically Closed**

Only parking numbers 9,10,11,12,13,14,15 and 16 are left. Press push button exit3 (OUT1) to calculate the minimum weight value and get result number 16 show in figure 4.22.

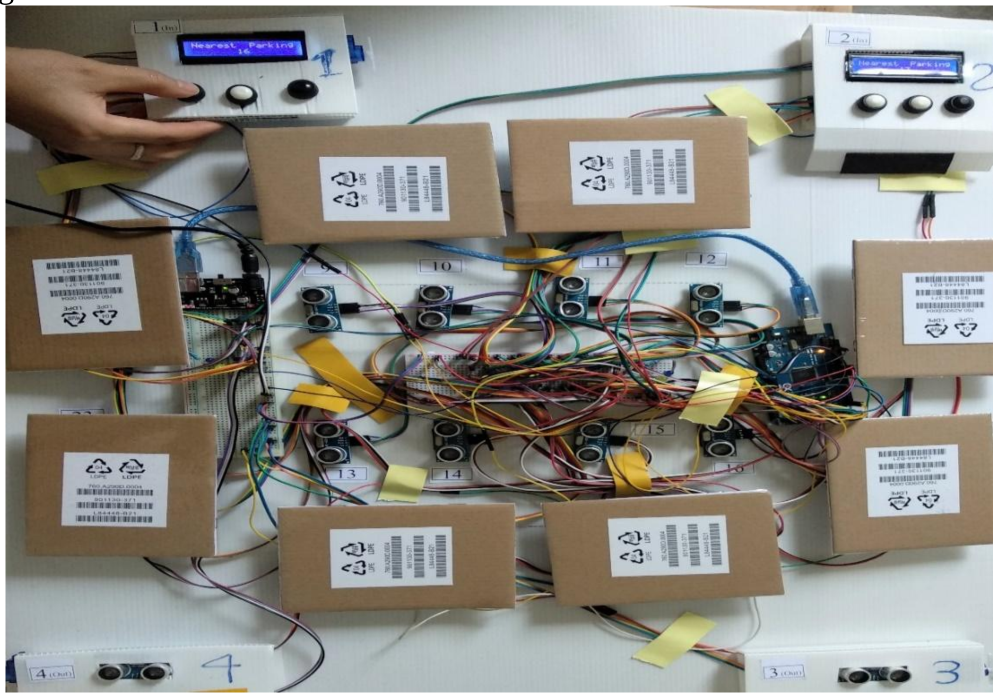


**Figure 4.22 Prototype for 8 Parking lot Left Display the Shortest Path Number**

When all the parking spaces are full, both LCDs will show parking full, as shown in Figure 4.23.

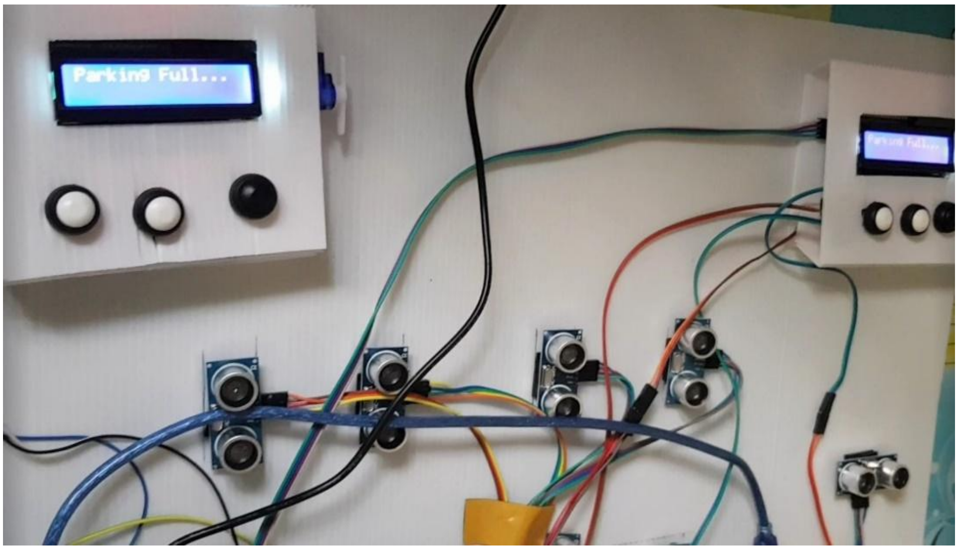


**Figure 4.23 LCDs will Show Parking Full**

If there are more than five parking spaces, both entrances will be reopened show in figure 4.24.

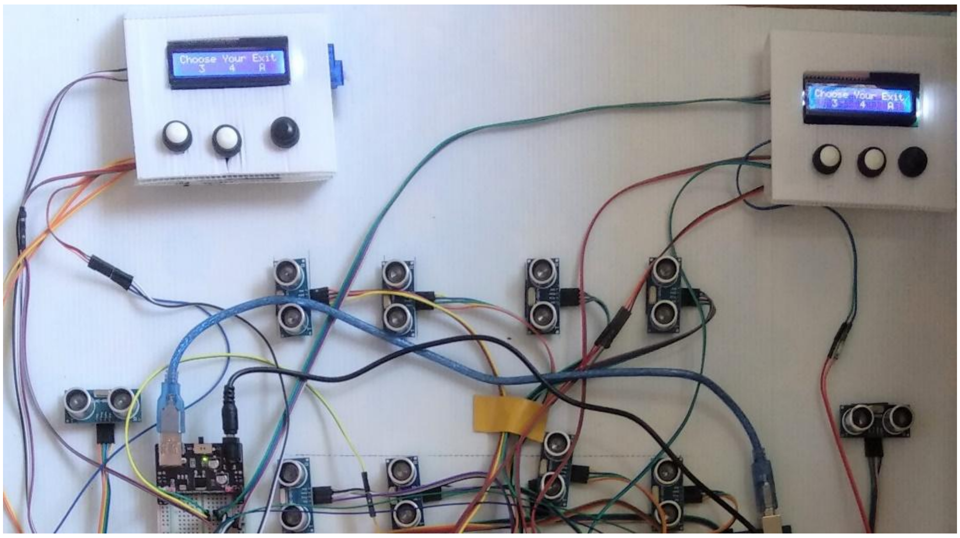

**Figure 4.24 Prototype for Two Entrances will be Reopened**

If the user selects OUT1, the parking lot closest to the exit will be displayed as shown in Figure 4.25.

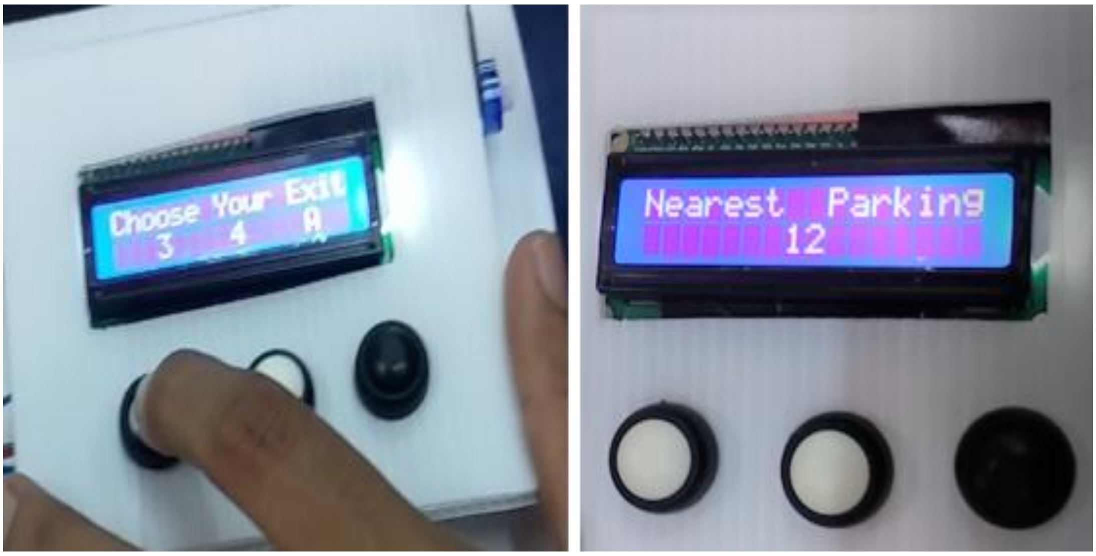


**Figure 4.25 LCDs will Show Nearest Parking**

## 4.5 Analysis for Car Parking Searching First Floor

In this system, analysis the best route from the multiple routes to the exit (OUT1) and its time duration as shown in Figures 4.26.

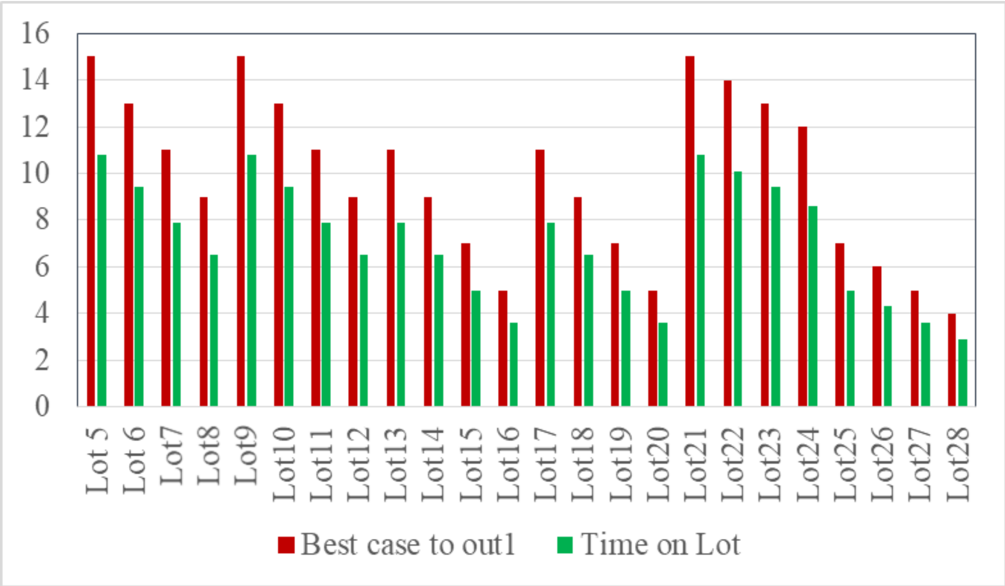


**Figure 4.26 Analysis for First Floor OUT 1 (Exit3)**

Analysis the best route from the multiple routes to the exit (OUT2) and its time duration as shown in Figures 4.27.

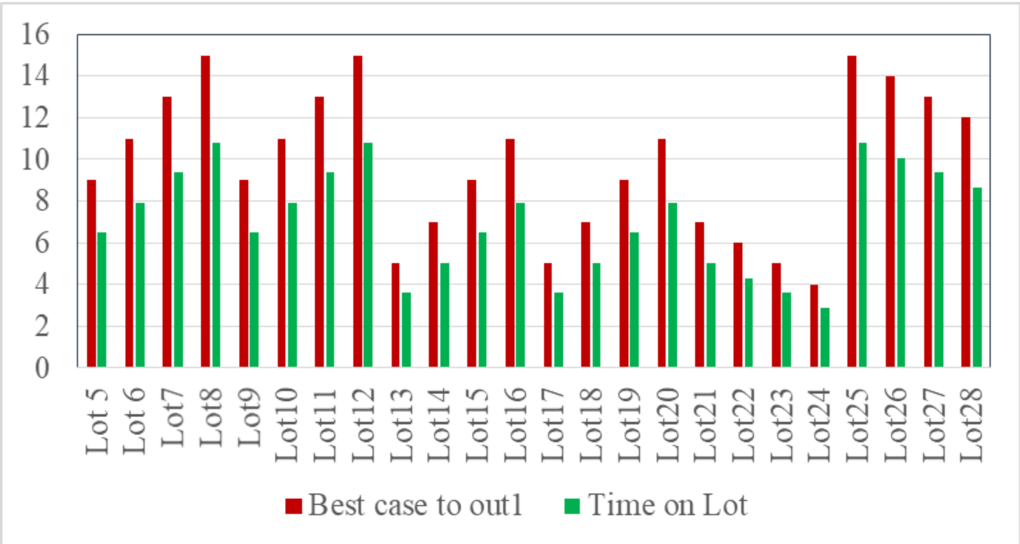


**Figure 4.27 Analysis for First Floor OUT 2 (Exit4)**

## 4.6 Analysis for Car Parking Searching Second Floor

Analysis the best route from the multiple routes to the exit (OUT1) and its time duration as shown in Figures 4.28.


**Figure 4.28 Analysis for Second Floor OUT 1 (Exit3)**

Analysis the best route from the multiple routes to the exit (OUT2) and its time duration as shown in Figures 4.29.


**Figure 4.29 Analysis for Second Floor OUT 2 (Exit4)**

## 4.7 Analysis for Car Parking Searching Third Floor

Analysis the best route from the multiple routes to the exit (OUT1) and its time duration as shown in Figures 4.30.


**Figure 4.30 Analysis for Third Floor OUT 1 (Exit3)**

Analysis the best route from the multiple routes to the exit (OUT2) and its time duration as shown in Figures 4.31.


**Figure 4.31 Analysis for Third Floor OUT 2 (Exit4)**

## 4.8 Distance Measuring Experiment

In experiment the distance measured by the ultrasonic range finder measurement distance is different in the actual distance. These differences are shown in Table 4.5.

**Table 4.5 The Differences Between Actual and Measurement Distance**

| Actual distance (cm) | Detected distance (cm) | Ratio of differences |
|---|---|---|
| 20 | 17 | 0.85 |
| 40 | 35 | 0.87 |
| 60 | 55 | 0.91 |
| 80 | 75 | 0.93 |
| 100 | 93 | 0.93 |
| 120 | 116 | 0.96 |
| 140 | 136 | 0.97 |
| 160 | 154 | 0.96 |
| 180 | 174 | 0.97 |
| 200 | 192 | 0.96 |
| 220 | 204 | 0.93 |
| 240 | 218 | 0.91 |

All sonar field systems cause sonar reflection problems. Most objects reflect light with light. The roughness surface of an object is large compared to the wavelength of light, which causes the light source to scatter. For very smooth surfaces (like mirrors) the reflector becomes targeted.

Ultrasonic waves naturally reflect almost all large surfaces. The amount of energy returned depends on the incident angle of the sound energy. Figure 4.37 is the differences between actual and detected distance chart. According to experimental results, the rangefinder can proximally measure the distance between 60 and 180cm. Maximum actual detected length is about 400cm.

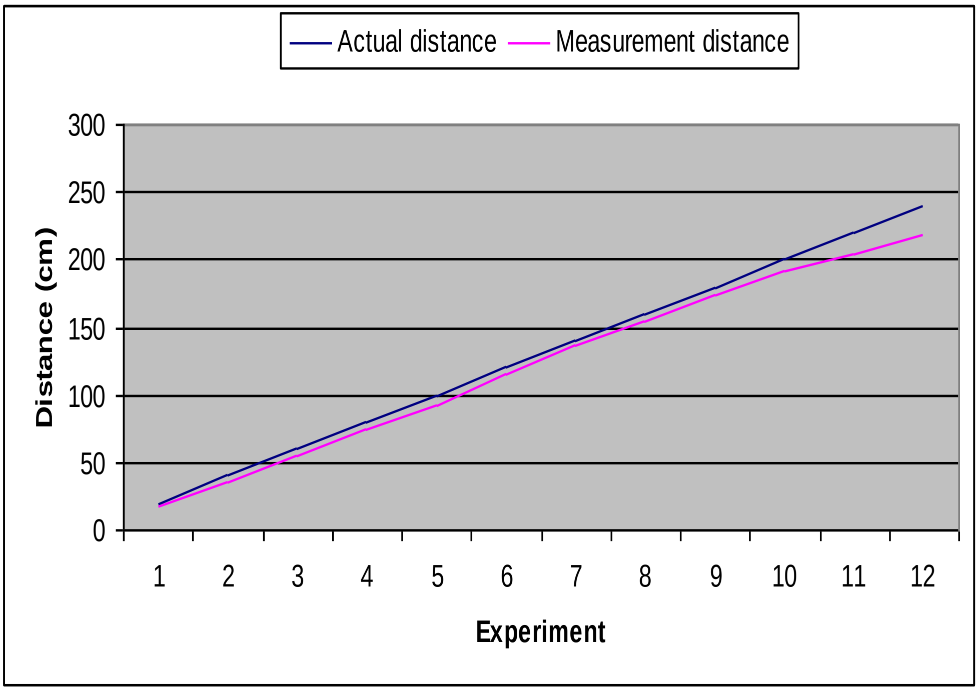


**Figure 4.32 Differences between Actual and Detected Distance**

CHAPTER 5

# CONCLUSION

In this thesis, the circuit module and mechanical structure for the smart car parking lot management system has been successfully constructed and demonstrated with toy vehicle.

Smart Car Parking Lot Management System using modified Graph Theory (or Graph Algorithm) has been successfully developed. The system provides a more convenient parking system for users. Users can stop without looking for the nearest place at the entrance of the condo.

The priority is given to the driver who presses the button first and the input signal from the ultrasonic sensor located at the nearest of exit gates. The parking number will be displayed on the LCD to find the parking space and the driver will be able to search in a short time.The parking lot is processing to assign the nearest parking lot based on Graph Theory. If all users park at the assigned lot, the system will utilize completely.

## 5.1 Advantages

The system focuses on the study of smart car parking system with graph theory. According to the results finding the graph theory is applicable to the path or lot finding for the parking system. The system can be extended name in the realistic world. This thesis can provide the way how to find the parking lot with simulation study and implement with devices.

This system has advantages. These are stated as follows:

- Graph Theory can be used to find the shortest path.
- Optimized parking.
- Reduced traffic.
- Reduced pollution.
- Enhanced User Experience.
- Increased Safety.
- Real-Time Data and Trend Insight.
- Decreased Management Costs.

## 5.2 Limitation

In this thesis, among many microcontrollers, use for Arduino Mega 2560. Using an Arduino simplifies the amount of hardware and software development to get a system running. On the software side, Arduino provides multiple libraries to make microcontroller programming easier. The simplest of functions control and read the I / O pins rather than having to fiddle with the bus / bit masks normally which it used to interface with the Atmega I / O. The system has two entrances and two exits, so it is not suitable for a condo in the middle of the street. It can only be used for a condo on the corner. When entering the parking lot, it will only be directed to the nearest parking lot with the exit, so it will not be able to point to the nearest entrance to the condo.

## 5.3 Further Extension

More useful are things such as being able to set I/O pins to PWM at a certain duty cycle using a single command or doing serial communication. Personally, the greatest advantage is having the hardware platform set up already, especially the fact that it allows programming and serial communication over USB and another work will be network connectivity. This work can be extended to autonomous car parks where the display will be used to detect the availability of parking lots as well as accepting different payment methods. In this system, it would be better for drivers to add sensors and lights and arrows along the lane at the entrance to the parking lot.

# REFERENCES

[1] H. JAAFAR*, M. H. ZABIDI, A. C. SOH, T. P. HOONG, S. SHAFIE, S. A. AHMAD: Intelligent Guidance Parking System Using Modifies Dijkstra's Algorithm, October 2014.

[2] K.I. Mohammad Ata*, A. Che Soh*, A. J. Ishak, H. Jaafar, N. A. Khairuddin: SmartIndoor Parking System Based on Dijkstra's Algorithm, April 2019.

[3] Mr. kush Shah, Ms. Priya Chaudhari, Arduino Based Smart Parking System, International Research Journal of Engineering and Technology, January 2017.

[4] Md. Sifat Hassan, Nushrat Naima Islam, Asim Md. Ahsan Fahim, Tanvir Hasan Turja, Shahrin Chowdhury: Automated Parking System using Graph Algorithm, Janaury 2020.

[5] Maher Hassan Kadhim, Arduino Smart Parking Manage System based on Ultrasonic Internet of Things (IOT) Technology, October 2018.

[6] Richard C.Dorf & Robert H.Bishop Modern Control System Ninth Edition

[7] https://www.heavy.ai/technicalglossary/embedded-systems

[8] https://www.arduino.cc/en/Main/arduinoBoardMega2560

[9] https://www.twinschip.com/Push_Button_Switch_Normally_Open_22mm

[10] https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor

[11] https://www.google.com/search?channel=nrow5&client=firefox-b-d&q=What+is+servo+motor

[12] https://www.mantech.co.za/datasheets/products/A000047.pdf

[13] http://www.appdigitech.com/WhatPWM.pdf

[14] https://learn.sparkfun.com/tutorials/pulse-width-modulation/all

[15] https://www.google.com/search?channel=nrow5&client=firefox-b-d &q=what+is+lcd+display

[16] https://ozeki.hu/p_1564-oz-ultra-sonic-controller-test-circuit.html

[17] https://robotdyn.com/breadboard-power-supply-5v-3-3v-1a.html

[18] https://elonics.org/servo-motor-controller-tester-circuit-using-555-ic/

[19] https://www.quora.com/What-is-an-embedded-system

[20] https://en.wikipedia.org/wiki/Embedded_system

# LIST OF PUBLICATIONS

[1] Nan Chaw Su Kyi, K.K.K.Theint, " Smart Car Parking System Using Graph Algorithm, Parallel and Soft Computing, UCSY, 2022.