Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF PARKING GUIDANCE AND INFORMATION SYSTEM FOR MULTI-STORY BUILDING

NURUL AZARINA BINTI ABD RAHIM

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

2021

DEVELOPMENT OF PARKING GUIDANCE AND INFORMATION SYSTEM FOR MULTI-STORY BUILDING

NURUL AZARINA BINTI ABD RAHIM

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I declare that this project report entitled entitled “Development of parking guidance and information system for multi-story building” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : azarina

Student Name : Nurul Azarina Binti Abd Rahim

Date : 10 JAN 2022

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

Signature : 

Supervisor Name : TS. ZAHARIAH BINTI MANAP

Date : 11-01-2022

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

To my beloved mother, Tuminah Bt Suriff, and father, Ghazali Bin Mohd Ariff,

and

To dearest family

ABSTRACT

The urbanisation in developing countries has resulted an increase in intensity of population in urban and downtown area. This situation has forced rapid development of multi-story buildings to accommodate the need of living spaces and commercial premises such as hotels, apartments, shopping centers, offices, hospitals, and airports. Subsequently, the number of vehicles is also increasing tremendously which leads to inefficient parking slot searching time and traffic congestions at the buildings' parking space. Therefore, a systematic parking management for multi-storey buildings is needed to overcome the issues, which in turn will provide a betterment in modern lifestyle. Most of existing parking management systems implement an automatic gate and a parking ticket. However, there are limited information about the parking slot availability provided, commonly using of LED display at the parking entrance. The integration between conventional parking systems with Internet of Things (IoT) technology is a good solution to provide more efficient and flexible parking management systems. This project aims to develop a smart parking system that provides an efficient parking space utilization using IoT technology. The developed system consists of two main components which are an on-site device and mobile application. The on-site device updates the information about parking availability and its location in a cloud server. The data can be accessed by the user through a mobile application. This system informs the user availability of parking slots in that building with specified level. The accuracy for the IoT based Smart Parking System is analysed through a series of real time testing. By comparing the actual condition of the on-site device and the result shown in the Blynk mobile application, the percentage of accuracy of IoT based Smart Parking System is calculated. All of the actual results of the on-site device are synchronized with the result in Blynk mobile application. This system allows the smart parking system and the user to interact. Which will allow the user to check the vacant parking slots at their hands.

ABSTRAK

Pembandaran di negara-negara membangun telah menyebabkan peningkatan intensiti penduduk di kawasan bandar dan pusat bandar. Keadaan ini telah memaksa pembangunan bangunan bertingkat yang pesat untuk menampung keperluan tempat tinggal dan premis komersial seperti hotel, pangsapuri, pusat membeli-belah, pejabat, hospital, dan lapangan terbang. Selepas itu, jumlah kenderaan juga meningkat dengan pesat yang menyebabkan masa pencarian slot tempat letak kenderaan yang tidak cekap dan kesesakan lalu lintas di tempat parkir bangunan. Oleh itu, pengurusan tempat letak kenderaan yang sistematik untuk bangunan bertingkat diperlukan untuk mengatasi masalah tersebut, yang seterusnya akan memberikan peningkatan dalam gaya hidup moden. Sebilangan besar sistem pengurusan tempat letak kereta yang ada menggunakan pintu automatik dan tiket tempat letak kereta. Namun, terdapat sedikit maklumat mengenai ketersediaan slot tempat letak kereta yang disediakan, biasanya menggunakan paparan LED di pintu masuk tempat letak kereta. Integrasi antara sistem tempat letak kereta konvensional dengan teknologi Internet of Things (IoT) adalah penyelesaian yang baik untuk menyediakan sistem pengurusan tempat letak kereta yang lebih cekap dan fleksibel. Projek ini bertujuan untuk membangunkan sistem tempat letak kenderaan pintar yang menyediakan penggunaan tempat letak kereta yang cekap menggunakan teknologi IoT. Sistem yang dibangunkan akan terdiri daripada dua komponen utama iaitu peranti di lokasi dan aplikasi mudah alih. Peranti di lokasi mengemas kini maklumat mengenai ketersediaan tempat letak kereta dan lokasinya di pelayan awan. Data dapat diakses oleh pengguna melalui aplikasi mobile. Sistem ini memberitahu ketersediaan slot tempat letak kenderaan di bangunan tersebut dengan tahap yang ditentukan. Ketepatan untuk Sistem Parkir Pintar berasaskan IoT akan dianalisis melalui satu siri ujian masa nyata. Dengan membandingkan keadaan sebenar peranti di lokasi dan hasil yang ditunjukkan dalam aplikasi mudah alih Blynk, peratusan ketepatan Sistem Parkir Pintar berasaskan IoT dikira. Semua hasil sebenar peranti di lokasi diselaraskan dengan hasil dalam aplikasi mudah alih Blynk. Sistem ini membolehkan sistem parkir pintar dan pengguna berinteraksi. Yang akan membolehkan pengguna memeriksa slot tempat letak kereta kosong di tangan mereka.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, madam Zahariah Binti Manap for her precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my parent for the financial support which enables me to accomplish the project. Not forgetting my fellow course mate for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study.

Finally, I would like to thank all the staffs at the UTeM and classmates as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

V	-	Voltage
mA	-	Current
KB	-	Kilobyte
MHz	-	Megahertz
mm	-	Millimetre
g	-	Gram
vcc	-	Voltage Common Collector

LIST OF ABBREVIATIONS

V	-	Voltage
IoT	-	Internet of things
i.e.	-	That is

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Many different types of vehicles are produced and used by people in this century of modernism, especially in urban areas. The increased number of vehicles can have a significant serious effect on the environment. Particularly because of the increased need for public parking infrastructure at multi-story building. For example, find vacant parking in a shopping mall or a hospital facility.

Only an automatic gate and a parking ticket are used in a conventional parking lot system and the availability of the parking lot is not presented to the user. They have located it on their own in order to park their vehicles. In this situation, finding a parking spot and a vacant parking lot takes time for the user. Besides, they had no idea whether the parking lots were occupied or still vacant. The parking lot system needs to be refined to make it more systematic, efficient, and reliable for users. To overcome the following difficulties while also meeting demand for more parking spaces and improved services, parking management organizations are competing to create solutions that will result in a more efficient parking experience.

The goal of this project is to create an IoT based application for a smart parking system by using Arduino. Wi-Fi wireless communication technology is chosen because it can easily save cable costs and internet access anywhere. The proposed system assists a user in knowing parking spaces are available. The parking systems are designed to provide users with facilities such as finding, allocating car park available to a user in a given level. Such

systems require the deployment of efficient sensors in the parking areas for occupancy monitoring as well as quick data processing units to gain practical insights from data collected from different sources.

1.2 Problem Statement

When it comes to parking lots especially in multi-story buildings like hospital, Shopping centers. There is a slew of difficulties to consider. In a traditional parking lot system, only an automatic gate and a parking ticket are provided and the user is not informed about the parking lot's availability. Finding a vacant parking lot by each level takes time for the user in this case. Furthermore, they had no clue if the parking lots were full or empty.

To solve this issue, a smart parking infrastructure is being built to handle the task using the latest technologies. However, most of the existing smart parking implementations only provide on-site information about parking availability. For example, a smart parking system at a shopping mall displays the number of available parking slots at the parking entrance. Currently, the integration of a smart parking system with IoT devices is still a new area to venture into. This approach is predicted to provide a far convenient way of parking searching especially in high population cities.

This project aims to build an IoT-based smart parking system that saves driver time and achieves an accessible and user-friendly system. This is for promoting traffic movement inside the car park. The primary purpose of this is to develop a mobile application that will allow users to check the availability of parking within their area. A model of parking system is designed to illustrate the parking function, and consumers can use the mobile application to check parking slots by each level availability in multi-story building.

1.3 Project Objective

The aim of this project is to develop a parking guidance and information system for multi-story building. Specifically, the objectives are as follows:

a) To develop an on-site device for a smart parking system by using Arduino.

b) To create a smart parking mobile application that can interact with the on-site smart parking device.

c) To evaluate the performance of the developed system.

1.4 Scope of Project

My project's scope is to develop a parking guidance and information system for multi-story building that can access the on-site device that had installed. My project will separate into two parts which are the development of an on-site device for smart parking systems and implementation of smart parking mobile applications. In this parking system, it will have 3 floor and each floor has parking slot is detected and the data is sent to the Arduino to be processed. The data is uploaded to the Blynk cloud via Wi-Fi. In order to interact with the on-site smart parking device, the mobile application is developed on the Blynk IoT platform and serves as an interface for communication with the device by end-users. The central server Blynk serves as a directory to hold all information and end-users connected to the parking area who have access to the network. So that, the apps show number of available parking spaces by each level to the users. The data stored on the cloud is backed up and updated continuously to make sure that the information is the latest for the users.

1.5 Project Significance

In terms of commercialization, this project aspires to provide a parking guidance and information system for multi-story buildings that are in great demand. The commercial potential is enticing because this system has minimal equipment costs, minimal maintenance costs, and a simple infrastructure. The project gives users a lot of benefits to avoid wasting a lot of precious time finding the parking spaces for each level manually especially in terms of locating parking spaces in multi-story buildings such as shopping malls, hospitals, hotels, and other similar structures and allows users to instantly locate the best available parking spot because of the information provided by the mobile application.

In addition, users can avoid going to the level that already fully occupied car parks so that the traffic congestion rate can be reduced. As a result, rather than searching manually, this system can provide available parking. This can help save time and improve quality of life.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The daily urbanization of developed countries has led to a rise in the demand for automobiles. This has in turn culminated in the emerging of new vehicles into the market. There has been a spike in the number of cars, but there has been no proportional increase in the amount of parking space available. As a result, traditional parking systems may suffer from a number of negative consequences, including traffic congestion, noise, fuel usage, and time consumption. Furthermore, anyone looking for a parking spot must circle the parking lot multiple times and wasting valuable time. As a result, malls and other public places need the most effective solution to this issue. The online parking information system is critical to improving the parking system. This lead the demand for automated parking systems, online parking slot reservations, multi-story parking buildings, and other alternatives are increased. This chapter reviews articles and works from previous research on parking guidance and information systems in multi-story buildings.

2.2 Overview of parking guidance and information system

A detail survey on the development of parking guidance and information system has been found in [1], [2], [3]. The authors outlined the many technologies, methodologies, and algorithms that have been used to create a parking guidance and information system. GPS, GSM module, Wi-Fi, Bluetooth, RFID, Arduino, Raspberry Pi, and sensors like Pi camera, photodiode sensor, magnetometer sensor (MAG3110), Xbee, Ultrasonic, and Infrared (IR) Sensor are some of the most common and regularly used technologies

discussed in the articles. Parking guidance and information system technologies can certainly be a well identified innovation because based on the various technologies that support the parking guidance and information system. Although there are a massive range of technologies that have been used this day, there are a few commonly used technologies. In this section, parking guidance and information system applications in different sensors technologies are reviewed.

2.2.1 Infrared sensors Technologies

An infrared sensor is an electronic device that emits infrared rays to detect certain features of the environment. An infrared sensor can detect motion as well as measure the heat of an object. This sensor exclusively detects infrared radiation emitted by an object. These kinds of thermal radiations are often invisible to the naked eye, but they can be detected by an infrared sensor. Figure 2.1 show basic principle of IR sensor. The primary operating idea is when an infrared LED emits light, the photodiode senses it.

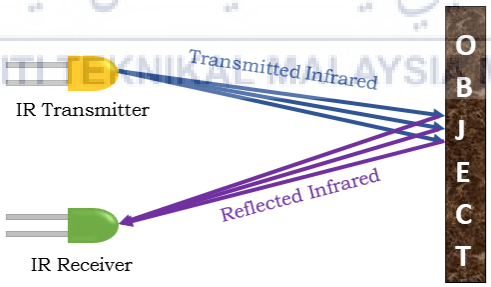


Figure 2.1 Basic working principle of IR sensor

The authors in [4] defined that Real-time sensor reading seem to be the most promising way to inform about the status of car parking (busy or free). The implementation of the prototype of car parking system has been done in this work. The real-time readings of the infra-red sensors are sent wirelessly to the gateway server where they will be saved and analyzed.

The authors in [5] proposed the IR technology is used for vehicle detection. The IR sensor will be placed at the entry point to help eliminate multiple check-ins and parking lot traffic congestion. The infrared obstacle sensor, which is mounted at each floor's entry and exit and whose duty is to transfer data to the Arduino, which processes the sensor and sends data to the database, is shown in [6]. The database is utilized to record data and sensor values that will be displayed in a Web browser on a monitor on the ground floor to alert the driver of an empty parking spaces as to produce efficiency BBM and time efficiency.

In addition, the author in [7] created the main processing unit is an Arduino microcontroller. Which receives information from IR sensors to assist the user. When sensors detect a vehicle, the corresponding output is sent to the cloud, via protocol and various layers of the OSI model. So, the data can be accessed on a user's mobile device via an app or on a computer via an html page. Then user can view the parking lot of any geographical area to find an empty parking slot.

The authors in [1], designed an availability of parking lots based on the IR sensor linked with an online monitoring system. Those infrared sensors have established a wireless connection. Through the LCD counter display the parking system operator may regulate and monitor the state of each parking lot. This indoor parking assistance system will provide the driver with significant advantages.

In [8], the authors proposed when the application is not in use the RFID readers are utilized to control the entry and exit gates and the servo motors serve as the entry and exit gates. Then, like in-spot sensors IR reflective sensors monitor the condition of the spot (free or occupied by a parked car) with all the IR sensors connected to a single Interface kit. RFID, Servos and interface kit is directly connected to a laptop acting as a server that collects information about the car park infrastructure and occupancy.

2.2.2 Ultrasonic sensor Technology

An ultrasonic sensor is an electronic device that uses ultrasonic sound waves to detect the distance between a target item and converts the reflected sound into an electrical signal. Ultrasonic waves travel at a faster rate than audible sound waves (i.e., the sound that humans can hear). Figure 2.2 shows ultrasonic sensors include two primary components. A transmitter (which uses piezoelectric crystals to make sound) and a receiver (which encounters the sound after it has travelled to and from the target).

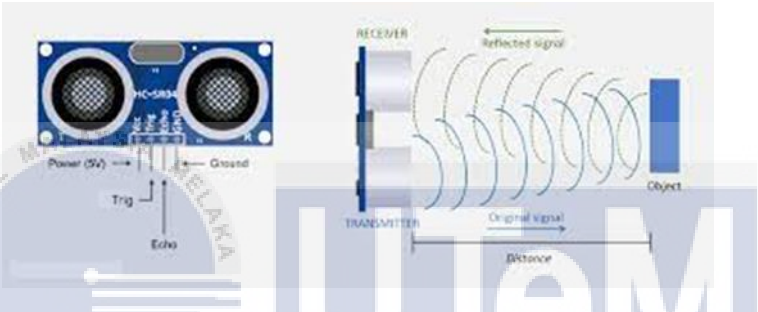


Figure 2.2 basic working principle of ultrasonic sensor

The author in [9] and [10] implemented Ultrasonic sensors are installed in the parking slots to detect cars entering and exiting the slots. Author in [9], an ultrasonic sensors to convey information about empty and full slots to the Arduino. When a car approaches the sensor, the sensor's trigger emits a high-frequency sound wave that is reversed. That signal is processed by the Arduino, which then displays a green or red LED. The LED will turn red if all slots are occupied. The LED at the parking entry will be green if it isn't.

In addition, the authors in [11] introduce HC-SR04 ultrasonic module to detect parking slot. When HC-SR04 ultrasonic module output is sent to the Arduino Nano, and data is transferred to the server using esp8266 Wi-Fi module. Further, the data is processed by the server, which then displays the information in the form of a parking area map on both web and desktop applications. This system will display available space on a website and on a computer with a few second delay and a simple wiring system.

This testing in Figure 2.3 shows that each parking space has a sensor module installed. The distance data changes between the parking ceiling and floor, as well as the vehicle are acquired by the sensor module. Using a potentiometer, sensor module may vary the distance between the ceiling and the floor to a maximum of 4 meters.

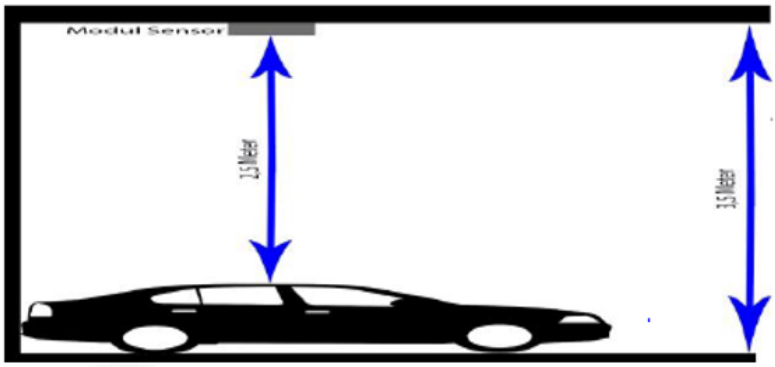


Figure 2.3 Installation Of The Sensor Module In The Parking Slot

In [12], for identifying the vacant slot, an ultrasonic parking sensor is proposed. It uses the aforementioned echo-location method to find a parking spot. The authors claim that the mechanism is extremely powerful, making the system more promising.

The author in [13] implemented the Ultrasonic Range Detection Sensor is utilized with Arduino to indicate the empty slot. Drivers can identify an empty parking place by determining the area using an ultrasonic sensor, which helps the driver identify the place quickly and reduces the time spent searching

2.2.3 Pi-camera Technology

Figure 2.4 show the Raspberry Pi Camera Board is a custom-built add-on module for the Raspberry Pi. It uses a modified CSI interface to connect to the Raspberry Pi. The sensor has native resolution of 5 megapixels in still capture mode. In video mode, it can capture video at resolutions up to 1080p at 30 frames per second.

The authors in Intelligent Parking System Using IOT [14] The features of the Raspberry Pi 2 have been proposed, as well as the fact that it is equipped with a pi camera. Where the Raspberry Pi camera is put on the top of streetlight lamp posts or the ceiling of indoor parks. As a result, the camera can continuously inspect each parking place in a parking lot to determine whether it is occupied or vacant.

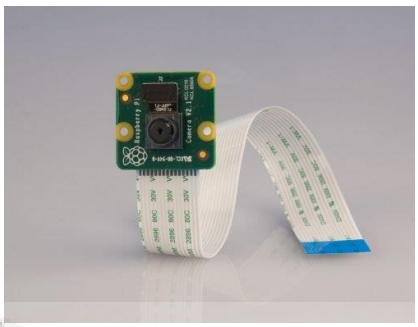


Figure 2.4 Raspberry pi camera module

2.2.4 Magnetometer Sensor (MAG3110) Technology

A magnetometer is a scientific equipment that detects and measures the strength and direction of magnetic fields in the vicinity. Because of changes in the Earth's magnetic field created by different rock types and interactions between charged particles from the sun and the magnetosphere of a planet. Magnetism varies from location to place. The authors in [15] propose the application of magnetometer sensor (MAG3110) . When a vehicle passes over the sensor, the magnetometer detects a disruption in the magnetic field and takes the appropriate action. Several testings of the magnetometer were carried out throughout the semester. When no car was present, the authors conducted tests in open fields and parking lots to obtain reliable readings of the earth's magnetic field. The result shows the sensor network can reasonably detect the presence of a vehicle.

2.3 Summary of technologies used in the parking guidance and information system

A broad variety of technologies for parking guidance and information system are being applied these days, according to the various papers that are being studied. Table 2.1 summarizes the precision, functionality and limitation of the technologies reviewed in the literature.

Table 2.1 Comparison between articles based on sensor technologies

Comparison between articles based on sensor technologies				
Article	Technologies			
	Infrared sensors	ultrasonic sensor	Pi-camera	magnetometer sensor (MAG3110)
[2]	/	X	X	X
[3]	X	/	X	X
[4]	/	X	X	X
[5]	/	X	X	X
[6]	/	X	X	X
[7]	/	X	X	X
[8]	/	X	X	X
[9]	X	/	X	X
[10]	X	/	X	X
[11]	X	/	X	X
[12]	X	/	X	X
[13]	X	/	X	X
[14]	X	X	/	X
[15]	X	X	X	/

2.4 Summary of the chapter

This chapter reviewed and analyzed the various applications and technologies that are frequently used in the development of parking guidance and information systems. Following a review of the relevant literature, it can be concluded that each of the technologies and methodologies used has a role to play in the completion of the parking guidance and information system, each with its own complement of advantages and disadvantages.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The objective of this project is to provide a parking guidance and information system for a double-story structure. This chapter describes the work processes, methodologies, and techniques used to attain the project goal. This chapter also addresses the components used to develop the system. The next section will provide a summary of the work cycle followed by a thorough description of the development of hardware and software.

3.2 Project overview

The flowchart of Figure 3.1 describes the flow of process of this project. A detailed literature review focusing on the parking guidance and information system starts off the project. Many research concerns relating to parking guidance and information systems are described in articles and conference papers. According to the literature, the majority of researchers in the parking guidance and information system are interested in vehicle tracking. According to our knowledge gain from the research. Infrared technology is one of the most relevant and commonly utilized technologies for object tracking in parking guidance and information systems. Therefore, we design the proposed system by using infrared technology to provide a solution to detect vehicle.

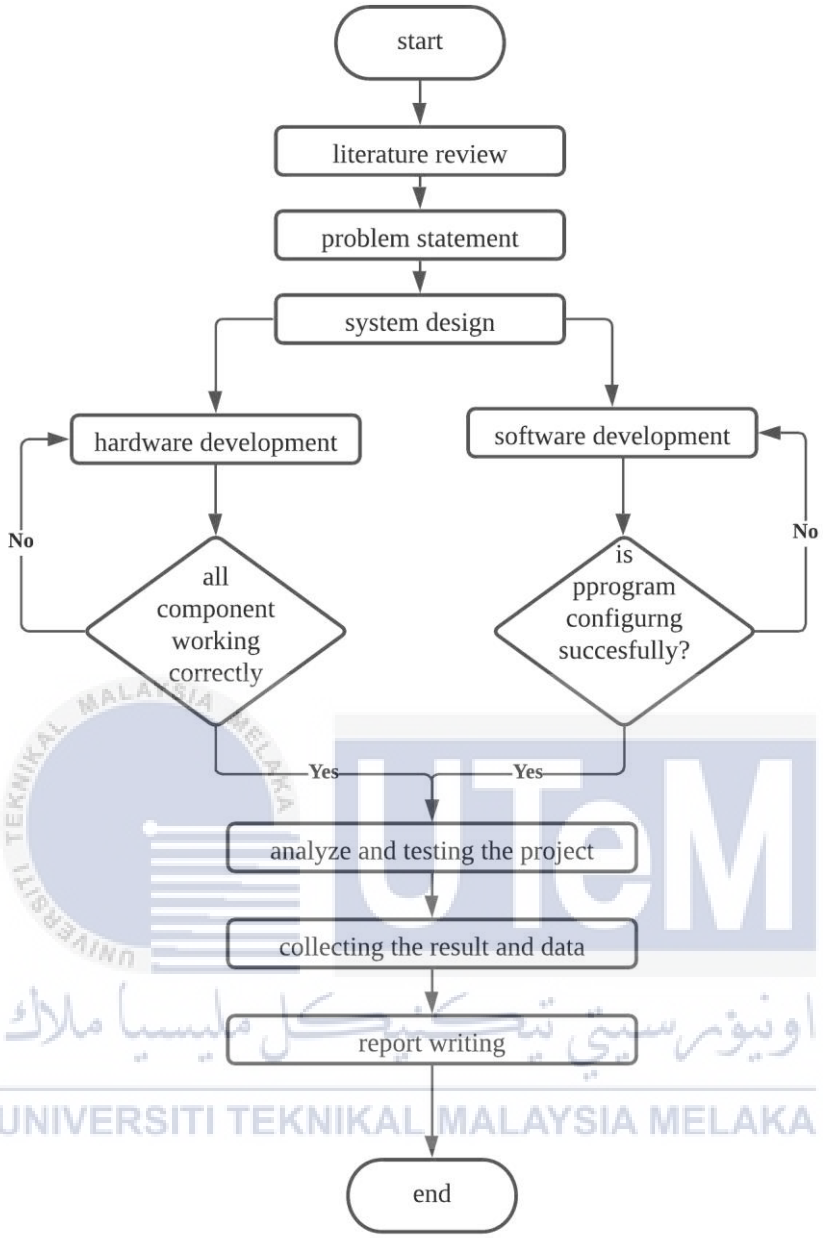


Figure 3.1 Flowchart for project

Both the hardware and software development are implemented in Figure 3.2 are suitable components used in the smart parking system are described in the hardware development. In the software development section, the software and the algorithm used are discussed. Finally, the performance of the developed system is evaluated through a series of test. The result and data are collected to analyze parking system accuracy.

3.3 System Design

Figure 3.2 shows the block diagram of the developed system. The counter receives and displays the number of parking spaces available in its particular parking level. The vacant parking slot information is acquired from the infrared sensor installed in each parking space and forwarded to the Arduino module for processing. This project's brain is an Arduino board The Arduino microcontroller controls the operation of the infrared sensor, which determines whether a parking space is occupied or not. The data on available parking spaces will be uploaded to the cloud and updated in the database via the Wi-Fi module.

When the user needs to check the best parking place, user will use a mobile phone to open the smart parking system application and access the cloud database. The Graphical User Interference programmed which interacted with the Arduino module via the Wi-Fi module, will display the number of vacant parking spaces and the level of floor of the parking lot from the user's location.

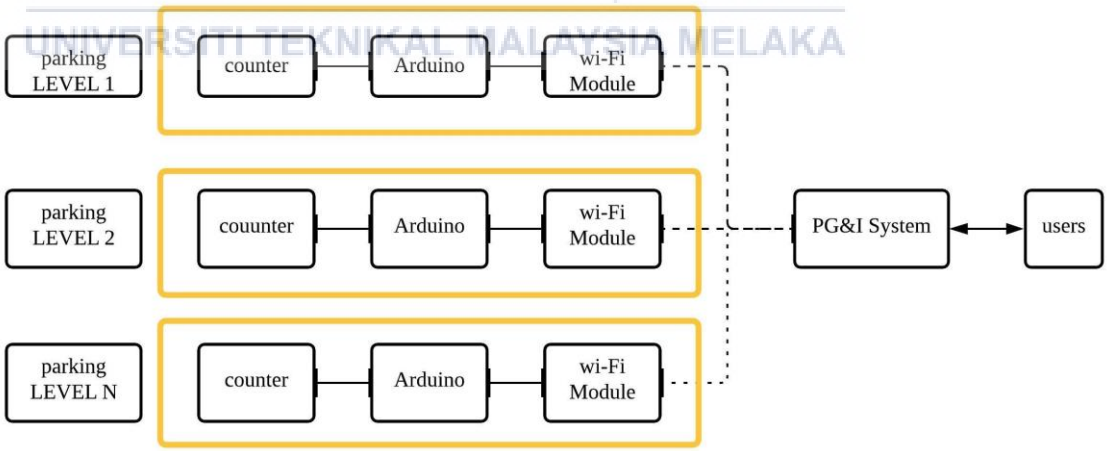


Figure 3.2 Block diagram of the project

3.4 Hardware Development

Hardware is important to the achievement of this project's study. In order to do this research, each piece of hardware has its own role. The hardware to develop this project is shown in Figure 3.2. Three main hardware components forming the on-site device of the developed system are Arduino Mega, ESP8266 Wi-Fi module, and Infrared sensor. This section explains how the components are integrated together. Figure 3.2 show in this project, All the component such as infrared sensor and Wi-Fi module will be connected to the Arduino board. The function of the Arduino is used for controlling the whole process with the Wi-Fi module and the Blynk platform.

3.4.1 Arduino Mega 2560

The Arduino Mega 2560 microcontroller board is based on the ATmega2560. A 16 MHz crystal oscillator, 54 digital input/output pins, 16 analogue inputs, 4 UARTs (hardware serial ports), a USB connection, a power jack, an ICSP header, and a reset button are all included. In this project, IR sensors, servo motor, LCD, ESP8266 Wi-Fi Module is connected and controlled by Arduino Mega 2560. It controls based on the programming input that inserts in the Arduino. The programming is executed in the Arduino. The IR sensor is installed in each parking slot on each floor, and its function is to provide data to the Arduino, which processes the sensor by sending data to the database. The database is used to store data and sensor values in the database and send data to Blink apps. Which is displayed an empty parking slots for each floor. At on site, it display the output from IR sensors on the LCD.

Table 3.1 Arduino Mega specification

Parameter	Specification

Microcontroller	ATmega2560
Operating voltage	5V
Input voltage (recommended)	7-12V
Input voltage (limit)	6-20V
Digital i/o pins	54 (of which 15 provide PWM output)
Analog input pins	16
DC current per I/O pin	20 mA
DC current for 3.3v pin	50 mA
flash memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock speed	16 MHz
Length, Width, Weight	101.52 mm, 53.3 mm, 37 g
LED_Builtin	13

The reasons why Arduino Mega used it in this project instead of another microcontroller because this project used 11 IR sensor to detect car, ESP8266 Wi-Fi Module, servo motor and LCD. It used many pins and Arduino Mega has a simple operating system, easy to configure, cheap, has may pin and it offers a large library in software. Besides that, Arduino is ideals to used for controlling the whole process with the Wi-Fi module and the Blynk platform.

3.4.2 Infrared sensor

This project also uses an infrared sensor obstacle detection sensor module in addition to the Arduino Mega module. The simple principle of IR (infrared) obstacle detection is to send an IR signal (radiation) in one direction, and when the IR radiation bounces back from the object's surface, a signal is received at the IR receiver. In general, all objects emit some form of thermal radiation in the infrared spectrum. Although such radiations are undetectable to the naked eye, infrared sensors may detect them. Infrared sensors (IR Sensors) are Non-Contacting Sensors.

There are two main parts of the IR sensor or infrared sensor. Infrared transmitter and receiver The job of the IR transmitter, also known as an infrared transmitter, is to transmit infrared waves, while the purpose of the IR receiver is to receive these infrared waves. The IR receiver continually provides digital data in the 0 or 1 format to the sensor's voice pin. Figure 3.3 depicts the OUT pin, which is used to link the sensor to the microcontroller's input or output port. Features of an Infrared Sensor 5VDC I/O pins are compatible with 5V and 3.3V operating voltages. Ambient Light Sensor 20mA supply current IR sensor is built-in. Aside from that, AC coupled response ranges from 30 kHz to 60 kHz, and all data formats are supported. The frequency range is limited to 55 kHz if the IR signal power is greater than 1000 mW/m2 (distance less than 0.35 m with a common IR remote control). An IR sensor is made comprised of an IR LED and an IR Photodiode, which are referred to as a Photocoupler or Optocoupler when used together. The IR transmitter continues to emit IR light, while the IR receiver continues to look for reflected light. The IR receiver receives the light if it is reflected back by hitting any object in front of it. In the case of the IR sensor, the item is detected in this manner. A potentiometer is the blue knob. The effective distance range of 2cm to 80cm and the range can be control from how far want to detect the object by changing the value of the potentiometer.

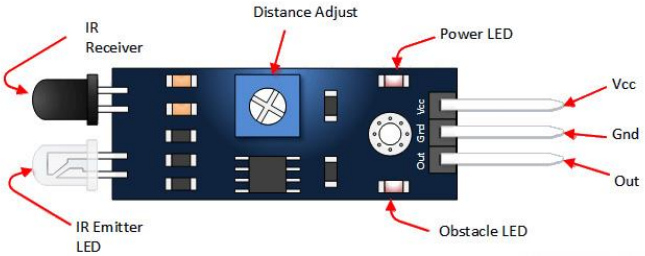


Figure 3.3 Ultrasonic sensor module pin diagram

In this project, the status of the parking lot is detected when the receiver detects the bounced back infrared light of LED that means the vehicle is parked in the parking lot and gets near to the infrared sensor. The information will be sent to the microcontroller to be

processed. Besides that, it cheap, small and enough to covered up all the system region of the project.

3.4.3 ESP8266 Wi-Fi Module

The ESP8266 Wi-Fi Module is a self-contained SOC with a built-in TCP/IP protocol stack that can enable access to the Wi-Fi network to any microcontroller. It can instantly retrieve data and upload it to the Internet, simplifying the Internet of Things. Often, you can use the API to capture data from the internet so that the project can access some information on the internet, making you smarter. The ESP8266 module is programmable using the Arduino IDE with AT-commands and allows serial communication. For the ESP8266, there is a total of 8 pinouts.

Figure 3.4 show the ESP8266 Wi-Fi module has one Vcc pin which plug into 3.3 V and ground pin that connected to circuit ground. The TX pin linked to RX pin of microcontroller to import the program and serves an Input/Output pin for general-purpose when not used as TX while the RX pin and GPIO-2 pin act as I/O pin for general use. The CH_EN pin is used to disable or enable the Chip-Effective High. The GPIO-0 used to take module into serial programming when it is kept small at initialization.

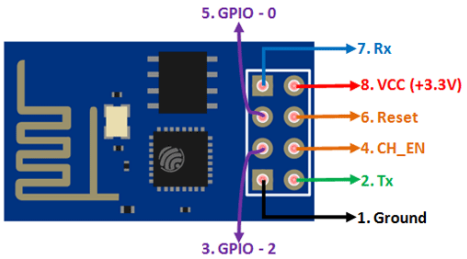


Figure 3.4 ESP8266 Wi-Fi Module

3.4.4 Micro Servomotor

A servo is an error-sensing feedback control that is used to improve the performance of a system. It also demands a complex controller, which is typically a separate module designed specifically for servomotors. Servo motors are DC motors that control angular position with precision. They're DC motors with gears that slow them down gradually. A revolution cutoff ranges from 90° to 180° on servo motors.

A standard DC motor, a gear reduction unit, a position sensor, and a control circuit make up a servo motor. The DC motor is attached to a gear mechanism that provides feedback to a position sensor, which is commonly a potentiometer. The servo spline in the gearbox transfers the motor's output to the servo arm. Regular servo motor gear is normally constructed of plastic, whereas high-power servo motor gear is usually made of metal.

In this project, micro servomotor act as barrier gate parking. When IR sensor detect a car at entrance parking model, the micro servomotor will rotate 90°, then car will enter through micro servomotor. Figure 3.5 show a servo motor consists of three wires- a brown wire connected to the ground, an orange wire connected to the control unit, and a red wire connected to the power supply.

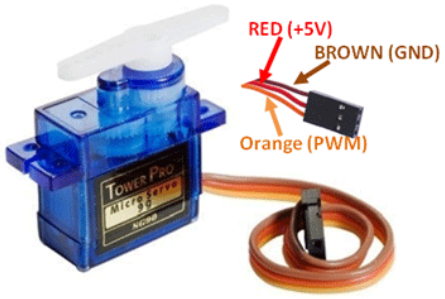


Figure 3.5 Micro Servomotor

3.5 Software Development

The data of availability of parking can access the network with a smartphone via a Blynk-developed app. Blynk is an IoT platform with customizable mobile devices, a private cloud, a system control rules engine, and a display that is hardware agnostic. It can work with a variety of hardware platforms, including Arduino, Raspberry Pi, and others. In addition, Blynk provides a variety of interfaces for connecting the microcontroller board to the Blynk cloud. Blynk server is conceited for all communication between the mobile device and the hardware running the Blynk application. Blynk libraries allow server communication and handle all incoming and outgoing Blynk applications and hardware commands.

Figure 3.6 flowchart based on how the smart parking system application works show. The user will enter the apps using QR code that has been generate in this project. Once the data from all parking sites received, the Apps will display the information to the user. The user able to see the available parking slot for each level in the apps. Driver can directly access the parking at any level that available. The cloud will now be updated with the latest information about parking spaces.

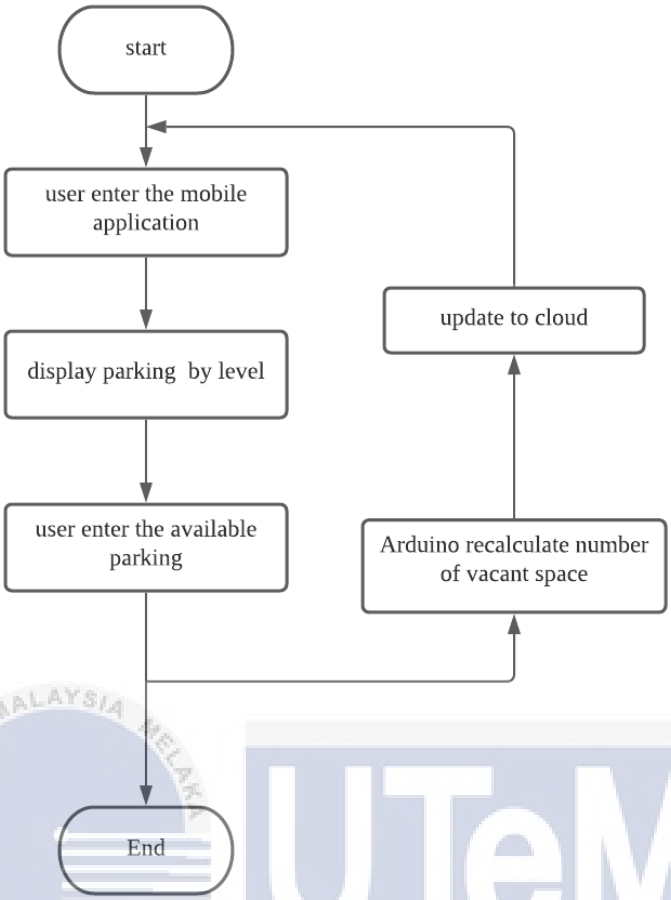


Figure 3.6 Flowchart based on how the smart parking system application works

3.6 Performance Testing

The development system will be tested in term of hardware and software test. Each 3 level of the parking sites of multi-story building is modeled as a circuit of on-site device consisting of Arduino Mega, ESP-8622 Wi-Fi Module, servo motor and three Infrared Sensors for each level. The infrared sensors represent the parking slots available at the parking sites. The sensors will be given the power source and the power indicator turn on. The output indicator of sensors lights up when object placed in the effective range of detection. The effective range detection will be test less than 3 times to ensure it can detect car without obstacle.

After success testing the sensors. Sensor will detect car and the data will be send to the Arduino Uno then to the ESP-8622 Wi-Fi module. Connection of ESP-8622 Wi-Fi module will be tested less than 3 times to avoid wrongly connected wire. Because ESP-8622 Wi-Fi Module is important part that used to transmit data from the microcontroller to the Blynk Server. Blynk app is used to show the availability of parking slots via Wi-Fi module when the data transmission is successful. Lastly, all parking slots will be tested. Each parking level will be filled and emptied with car. The purpose is to test whether the Blynk app and equipment are working properly for each floor of multi-story building.

3.7 Summary

From this chapter, the systems of parking guidance and information system is built by using methods and processes that already being discussed. Arduino Uno, Infrared sensors and ESP8266 Wi-Fi Module are chosen as the main component hardware for this project. The block diagram and layout of the project are designed in order to gain overall knowledge of project concept. Moreover, the flowchart is divided into two parts with are flow chart for how the system works and flowchart for software development. Both of this is related to each other. When the infrared sensor receiver receives the LED's bounced back infrared light, it implies the vehicle is parked in the parking lot and is getting close to the infrared sensor, the parking lot status is identified. The data will be transmitted to the microcontroller, which will process it. Then, Wi-Fi module is used to communicate with the Blynk platform and send the information to the user in a long distance.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

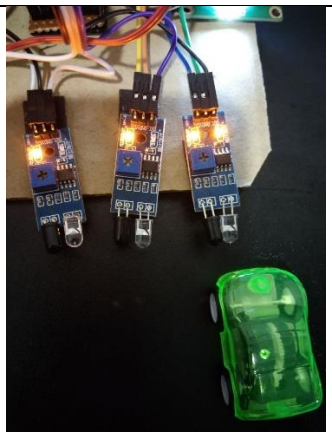
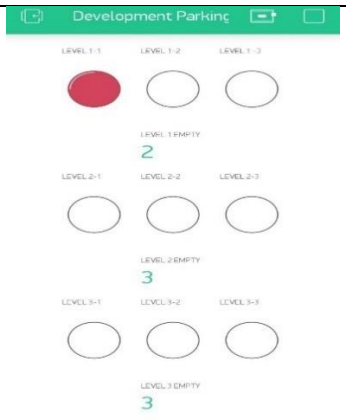
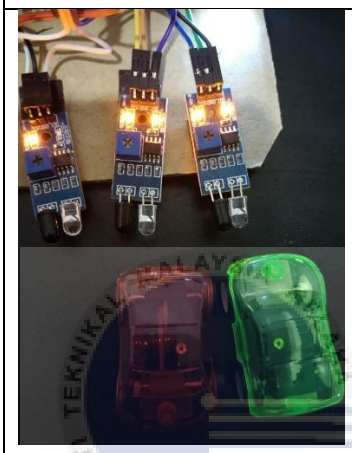
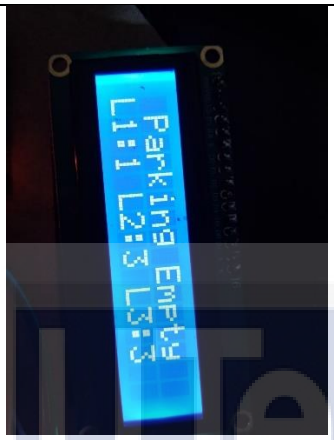
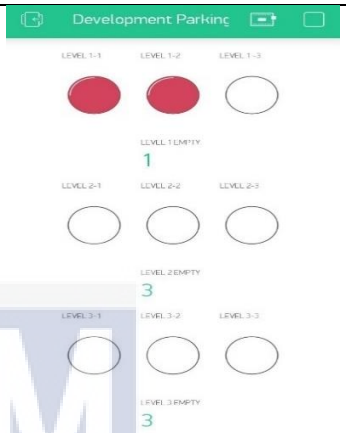
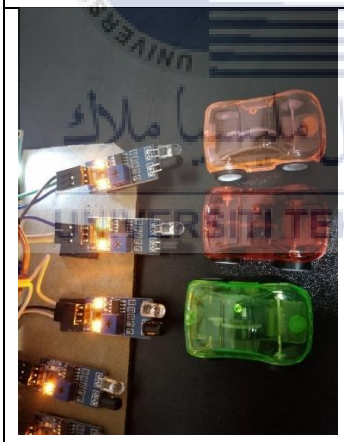
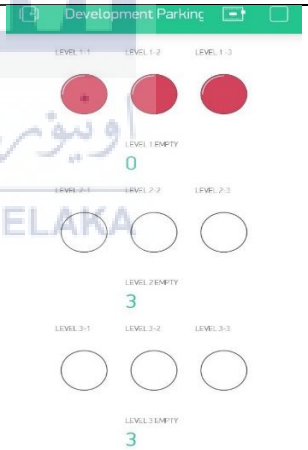
This project aims to develop the parking guidance and information system for multi-story building. Is the overall accomplishment of infrared sensor detecting vehicle parked in the parking lot. Then the data will be transmitted to the microcontroller, which will process it. Then, Wi-Fi module is used to communicate with the Blynk platform and send the information of availability of parking slots through a mobile application. This chapter discusses the project's outcomes and analyses the performance of the parking guidance and information system for multi-story building.

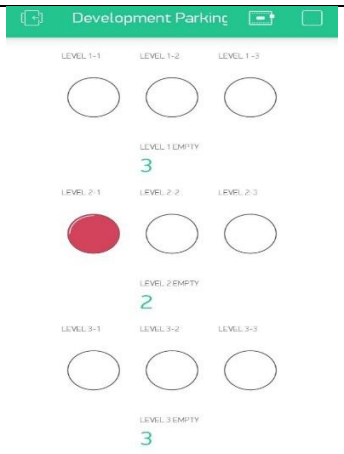
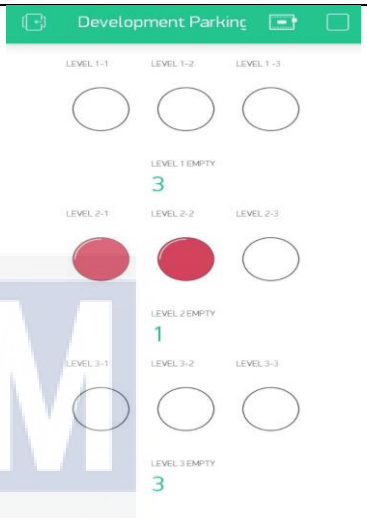
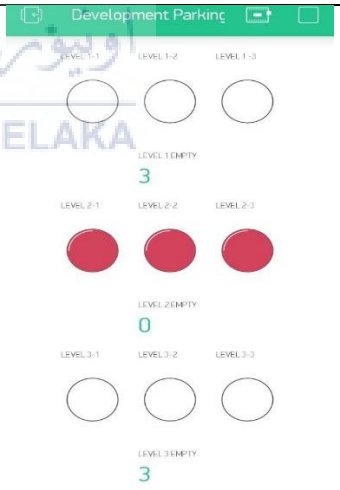
4.2 Results and Analysis

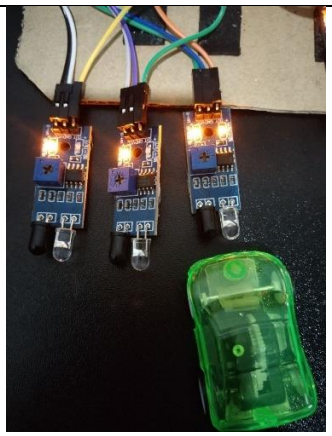
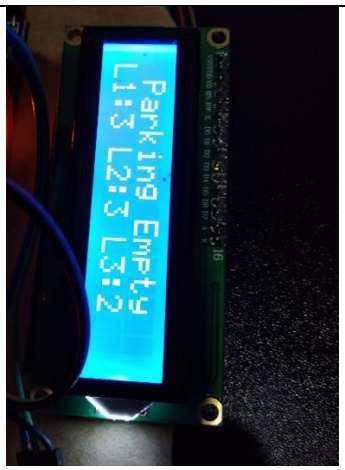
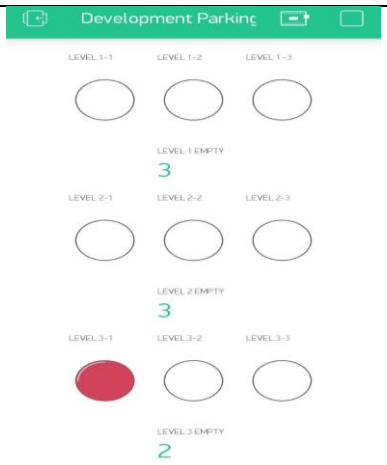
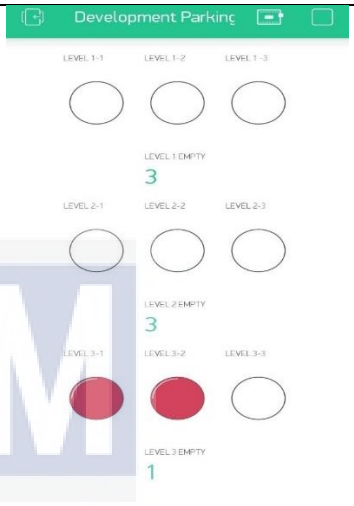
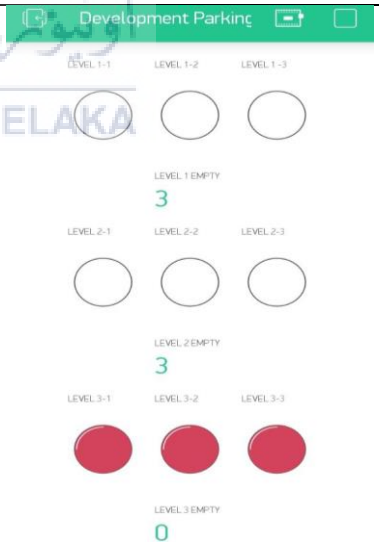
The aim of this section below is used to discuss the performance of the smart parking system. The accuracy for the parking guidance and information system for multi-story building that has been observed by comparing the actual condition of the on-site device, and the result shown in the Blynk mobile application.

Table 4.1 The comparison of the actual result and the result in the Blynk mobile application

Parking level	The actual result of the on-site device	LCD result on-site device	Result in Blynk mobile application

Level 1			
			
			

Level 2			
			
			

Level 3			
			
			

Based on Table 4.1, the table shows the comparison between the actual result of the on-site device, and the result in the Blynk mobile application. The percentage of accuracy of IoT based Smart Parking System is 100%. All of the actual results of the on-site device are

synchronized with the result in Blynk mobile application. The speed of the refresh rate was about 1.5s and the speed can also be affected by the upload speed of the Wi-Fi network connected.

4.3 Summary

From this chapter, the analysis of this project was obtained to estimate the respond to the parking system in the operation of transmission data from the on-site device to the Blynk IoT platform. In this project, the Blynk application chosen as the IoT platform. Several Blynk widgets are added to the Blynk application to display the number and the level of empty parking spaces. The application's user can enter any empty parking space from Apps. The LCD screen will display the availability of vacant slots by each level. From the Blynk widget, the off LEDs is empty parking spaces, while red LEDs is occupied parking spaces. The LEDs is divided by level and under the LEDs, there is display of vacant parking in that level. As the result, user can easily find the parking spaces without go to each level parking in the building to find empty parking spaces increase since the user can check the information of parking availability without reaching the parking area.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter discusses the conclusion and the future work for of the parking guidance and information system for multi-story building. The performance of the parking guidance and information system for multi-story building. has been determined and recommendation for this project was suggested in this chapter

5.2 Conclusion

. This project consists of hardware and software development. The hardware part is constructed by using Arduino Mega, ESP-8266 Wi-Fi module, LCD, and Infrared Sensor. The hardware part can also know as the on-site device in this project. For the software part, the programming language used in this project is C++. Wi-Fi communication technology that provides high-speed internet and long-range distance is chosen to develop the parking guidance and information system for multi-story building. The ESP-8266 Wi-Fi module was used to be the communication devices in this project to receive and transmit the data to the IoT platform. Blynk platform is letting the user spot parking spaces for each level. parking guidance and information system for multi-story building is a real-time system which can observe all of the condition of parking slots whether when the parking slots are occupied or when the parking slots are empty.

Besides, the performance of the smart parking system had been seen accurate in real time. The data received in the apps base on Wi-Fi access speed. The objective of this project had been achieved successfully.

5.3 Future Works

This section determines the suggestions for future improvements of the parking guidance and information system for multi-story building.

i) Extend the parking guidance and information system for multi-story building software to include booking parking spaces in apps.

ii) Change ESP-8266 Wi-Fi module that can only reach more than 100 meters line of-sight to LoRa module because the coverage up to three miles (five kilometers) in urban areas, and up to 10 miles (15 kilometers) or more in rural areas (line of sight) while minimizing current consumption.

iii) Consider the accuracy in detecting the vehicle, suggest to use Ultrasonic Sensor because the object recognition system can be implemented into the Ultrasonic Sensor where the only certain object can be detected by the system. Compare to Infrared Sensor that can detect all kinds of an object when there are other objects or obstacles instead of the vehicle are step or block in front of the Infrared Sensor

REFERENCES

[1] K. I. Ata *et al.*, "Smart Indoor Parking System Based on Dijkstra's Algorithm A Smart Guidance Indoor Parking System Based onDijkstra's Algorithm and Ant Colony Algorithm View project Carbon-based Flexible Sensor View project Smart Indoor Parking System Based on Dijkstra's Algorithm," 2019. [Online]. Available: https://www.researchgate.net/publication/339209665.

[2] A. Rahman and A. Alkharabsheh, "AN INTELLIGENT COOPERATIVE MULTI-AGENTS BASED PARKING SYSTEM: DESIGN AND IMPLEMENTATION," *J. Theor. Appl. Inf. Technol.*, vol. 31, no. 10, 2018, [Online]. Available: www.jatit.org.

[3] P. Sadhukhan, "An IoT-based E-parking system for smart cities," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-Janua, no. September, pp. 1062–1066, 2017, doi: 10.1109/ICACCI.2017.8125982.

[4] K. N. Utpala, N. S. Kumar, K. Praneetha, D. H. Sruthi, and K. S. A. Varma, "Authenticated iot based online smart parking system with cloud," *Pramana Res. J.*, vol. 9, no. 4, pp. 197–204, 2019.

[5] P. Otchere and W. Opare, "Design and Construction of an Authorized Car Parking Alert System," *Saudi J. Eng. Technol.*, vol. 04, no. 12, pp. 522–534, 2019, doi: 10.36348/sjeat.2019.v04i12.006.

[6] A. Wisaksono and C. A. Ragil, "Design and Development of Parking Motor Parking Information System at Muhammadiyah University, Sidoarjo," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, no. 1, 2020, doi: 10.1088/1757-899X/874/1/012015.

[7] C. Jambotkar, "IoT Based Smart Car Parking System," no. April, 2019.

[8] P. Melnyk, S. Djahel, and F. Nait-Abdesselam, "Towards a smart parking management system for smart cities," *5th IEEE Int. Smart Cities Conf. ISC2 2019,*

no. September, pp. 542–546, 2019, doi: 10.1109/ISC246665.2019.9071740.

[9] M. O. B. Sabbea, "Design and Development of a Smart Parking System," *J. Autom. Control Eng.*, vol. 6, no. 2, pp. 66–69, 2017, doi: 10.18178/joace.6.2.66-69.

[10] S. S. Bhagat, A. D. Bagul, P. N. Patil, and S. A. Dahale, "Bhagat, S. S., Bagul, A. D., Patil, P. N., & Dahale, S. A. (n.d.). Perceptive Car Parking Booking System With IOT Technology. International Research Journal of Engineering and Technology. Retrieved from http://esatjournals.net/ijret/2014v03/i02/IJRET20140," *Int. Res. J. Eng. Technol.*, [Online]. Available: http://esatjournals.net/ijret/2014v03/i02/IJRET20140.

[11] D. N. Bagenda and C. Parulian, "Online Information of Parking Area Using Ultrasonic Sensor through Wifi Data Acquisition," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 175, no. 1, 2018, doi: 10.1088/1755-1315/175/1/012011.

[12] A. Rahman, "Radio frequency identification based smart parking system using Internet of Radio frequency identification based smart parking system using Internet of Things," no. March, pp. 41–50, 2021, doi: 10.11591/ijra.v10i1.pp41-50.

[13] S. Nandyal, S. Sultana, and S. Anjum, "Smart Car Parking System using Arduino UNO," *Int. J. Comput. Appl.*, vol. 169, no. 1, pp. 13–18, 2017, doi: 10.5120/ijca2017914425.

[14] Sushma, S. M, R. R. A N, Z. Fathima, and S. B M, "Intelligent Parking System Using IOT," Jun. 2018, pp. 172–177, doi: 10.21467/proceedings.1.29.

[15] E. Munguia, D. (2019). S. P. I. (S.). Skaza, MichaelMunguia, E., Skaza, M., Donadio, A., & Perry, A. Donadio, and D. Perry, "Smart Parking Infrastructure (SPI)," 2019.

APPENDICES

Appendix A Main coding

```
#include <SoftwareSerial.h>
SoftwareSerial ESP8266_WIFI(14,15);// pin tx -2 pin rx -3
#include <Servo.h> //includes the servo library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
Servo myservo;
#define ir_enter 6 //servo motor naik
#define ir_back 7 //servo motor Turun
int ir_car1 = 31; // set pin IR sensor to arduino
int ir_car2 = 25; // set pin IR sensor to arduino
int ir_car3 = 27; // set pin IR sensor to arduino
int ir_car4 = 29; // set pin IR sensor to arduino
int ir_car5 = 33; // set pin IR sensor to arduino
int ir_car6 = 35; // set pin IR sensor to arduino
int ir_car7 = 37; // set pin IR sensor to arduino pin37
int ir_car8 = 41; // set pin IR sensor to arduino pin41
int ir_car9 = 45; // set pin IR sensor to arduino pin45
int S1=0, S2=0, S3=0, S4=0, S5=0, S6=0,S7=0, S8=0, S9=0; //declare kosong sensor
int flag1=0, flag2=0;
int slot = 9; // total tapak
int total;
//----------------LEVEL1-----------------------
int slotL1 = 3; // slot l1 ade 3 tapak
int L1_TOTAL;
int flag3=0, flag4=0, flag5=0; // flag define keadaan flag 0 , detect 1. flag boleh
tukar nama
//-----------------------------------------------
//---------------LEVEL2-------------------------
int slotL2 = 3;
int L2_TOTAL;
int flag6=0, flag7=0, flag8=0;
//-----------------------------------------------
//---------------LEVEL3-------------------------
int slotL3 = 3;
int L3_TOTAL;
int flag9=0, flag10=0, flag11=0;
//-----------------------------------------------
String sensor1; // declare sensor untuk panggil hantar maklumat ke esp
String sensor2;
String sensor3;
```

```
String sensor4;
String sensor5;
String sensor6;
String sensor7;
String sensor8;
String sensor9;
String cdata =""; // complete data, consisting of sensors values
void setup(){
Serial.begin(9600);
ESP8266_WIFI.begin(9600); //start declare esp coding untuk esp wifi
pinMode(ir_car1, INPUT); // declare sensor as input
pinMode(ir_car2, INPUT);
pinMode(ir_car3, INPUT);
pinMode(ir_car4, INPUT);
pinMode(ir_car5, INPUT);
pinMode(ir_car6, INPUT);
pinMode(ir_car7, INPUT);
pinMode(ir_car8, INPUT);
pinMode(ir_car9, INPUT);
pinMode(ir_enter, INPUT);
pinMode(ir_back, INPUT);
myservo.attach(8); // declare pin kaki utk servo motor
myservo.write(90); // declare servo motor 90 degree
lcd.init(); // initialize the lcd
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(0,0); // First line
lcd.print(" Development Of ");
lcd.setCursor(0,1); // move to the begining of the second line
lcd.print(" Parking System ");
delay(3000);
lcd.clear();
//------------------------------
int L1_TOTAL =S1+S2+S3;
slotL1=slotL1-L1_TOTAL ;
int L2_TOTAL =S4+S5+S6;
slotL2=slotL2-L2_TOTAL ; // slot program counter
int L3_TOTAL =S7+S8+S9;
slotL3=slotL3-L3_TOTAL ;
//------------------------------
}
void loop(){
p1slot1(); //panggil program ke atas. sbb program kat bawah semua.
p1slot2();
p1slot3();
p1slot4();
```

```
p1slot5();
p1slot6();
p1slot7();
p1slot8();
p1slot9();
digitalWrite(ir_car1, HIGH); //define sensor high
digitalWrite(ir_car2, HIGH);
digitalWrite(ir_car3, HIGH);
digitalWrite(ir_car4, HIGH);
digitalWrite(ir_car5, HIGH);
digitalWrite(ir_car6, HIGH);
digitalWrite(ir_car7, HIGH);
digitalWrite(ir_car8, HIGH);
digitalWrite(ir_car9, HIGH);
Read_Sensor();
lcd.setCursor (0,0); // paras 1
lcd.print(" Parking Empty ");
lcd.setCursor (0,1); // baris dua
lcd.print("L1:");
lcd.print(slotL1);
lcd.setCursor (5,1); // petak ke berapa
lcd.print("L2:");
lcd.print(slotL2);
lcd.setCursor (10,1); // petak ke berapa
lcd.print("L3:");
lcd.print(slotL3);
lcd.print(" ");
if(digitalRead (ir_enter) == 0 && flag1==0) // prgram untuk kereta masuk (pintu
masuk)
{
if(slot>0)
{
flag1=1;
if(flag2==0)
{
myservo.write(90); //pintu tutup
slot = slot-1;
}
}
else
{
//lcd.setCursor (0,0);
//lcd.print(" Sorry Parking Full ");
//delay(1500);
}
}
if(digitalRead (ir_back) == 0 && flag2==0){ //sensor 2
```

```
flag2=1;
if(flag1==0){
myservo.write(0);
slot = slot+1;
}
}
if(flag1==1 && flag2==1){
delay (500);
myservo.write(90);
flag1=0, flag2=0;
}
delay(1);
cdata = cdata + sensor1 +"," + sensor2 + ","+ sensor3 +","+ sensor4+"," + sensor5
+ ","+ sensor6 +","+ sensor7+ ","+ sensor8 +","+ sensor9+","+slotL1+","+slotL2+",
"+slotL3; // comma will be used a delimeter
Serial.println(cdata);
ESP8266_WIFI.println(cdata);
delay(500); // 100 milli seconds
cdata = "";
}
void Read_Sensor(){
S1=0, S2=0, S3=0, S4=0, S5=0, S6=0, S7=0, S8=0, S9=0;
if(digitalRead(ir_car1) == 0){S1=1;}
if(digitalRead(ir_car2) == 0){S2=1;}
if(digitalRead(ir_car3) == 0){S3=1;}
if(digitalRead(ir_car4) == 0){S4=1;}
if(digitalRead(ir_car5) == 0){S5=1;}
if(digitalRead(ir_car6) == 0){S6=1;}
if(digitalRead(ir_car7) == 0){S7=1;}
if(digitalRead(ir_car8) == 0){S8=1;}
if(digitalRead(ir_car9) == 0){S9=1;}
}
//--------------------------------------Level
1-----------------------------------------------
void p1slot1() // parkng 1 slot1
{
if( digitalRead(ir_car1) == LOW && flag3==0)
{
if(slotL1>0)
{
flag3=1;
slotL1 = slotL1-1;
sensor1 = "255";
delay(200);
}
}
if( digitalRead(ir_car1) == HIGH && flag3==1)
```

```
{
slotL1 = slotL1+1;
flag3=0;
sensor1 = "0";
delay(200);
}
}
void p1slot2() // parkng 1 slot1
{
if( digitalRead(ir_car2) == LOW && flag4==0)
{
if(slotL1>0)
{
flag4=1;
slotL1 = slotL1-1;
sensor2 = "255";
delay(200);
}
}
if( digitalRead(ir_car2) == HIGH && flag4==1)
{
slotL1 = slotL1+1;
flag4=0;
sensor2 = "0";
delay(200);
}
}
void p1slot3() // parkng 1 slot1
{
if( digitalRead(ir_car3) == LOW && flag5==0)
{
if(slotL1>0)
{
flag5=1;
slotL1 = slotL1-1;
sensor3 = "255";
delay(200);
}
}
if( digitalRead(ir_car3) == HIGH && flag5==1)
{
slotL1 = slotL1+1;
flag5=0;
sensor3 = "0";
delay(200);
}
}
```

```
//------------------------------level2--------------------------------
void p1slot4() // parkng 1 slot1
{
if( digitalRead(ir_car4) == LOW && flag6==0)
{
if(slotL2>0)
{
flag6=1;
slotL2 = slotL2-1;
sensor4 = "255";
delay(200);
}
}
if( digitalRead(ir_car4) == HIGH && flag6==1)
{
slotL2 = slotL2+1;
flag6=0;
sensor4 = "0";
delay(200);
}
}
void p1slot5() // parkng 1 slot1
{
if( digitalRead(ir_car5) == LOW && flag7==0)
{
if(slotL2>0)
{
flag7=1;
slotL2 = slotL2-1;
sensor5 = "255";
delay(200);
}
}
if( digitalRead(ir_car5) == HIGH && flag7==1)
{
slotL2 = slotL2+1;
flag7=0;
sensor5 = "0";
delay(200);
}
}
void p1slot6() // parkng 6 slot1
{
if( digitalRead(ir_car6) == LOW && flag8==0)
{
if(slotL2>0)
{
```

```
flag8=1;
slotL2 = slotL2-1;
sensor6 = "255";
delay(200);
}
}
if( digitalRead(ir_car6) == HIGH && flag8==1)
{
slotL2 = slotL2+1;
flag8=0;
sensor6 = "0";
delay(200);
}
}
//-----------------------------------------------level 3
void p1slot7() // parkng 7 slot1
{
if( digitalRead(ir_car7) == LOW && flag9==0)
{
if(slotL3>0)
{
flag9=1;
slotL3 = slotL3-1;
sensor7 = "255";
delay(200);
}
}
if( digitalRead(ir_car7) == HIGH && flag9==1)
{
slotL3 = slotL3+1;
flag9=0;
sensor7 = "0";
delay(200);
}
}
void p1slot8() // parkng 8 slot1
{
if( digitalRead(ir_car8) == LOW && flag10==0)
{
if(slotL3>0)
{
flag10=1;
slotL3 = slotL3-1;
sensor8 = "255";
delay(200);
}
}
```

```
if( digitalRead(ir_car8) == HIGH && flag10==1)
{
slotL3 = slotL3+1;
flag10=0;
sensor8 = "0";
delay(200);
}
}
void p1slot9() // parkng 7 slot1
{
if( digitalRead(ir_car9) == LOW && flag11==0)
{
if(slotL3>0)
{
flag11=1;
slotL3 = slotL3-1;
sensor9 = "255";
delay(200);
}
}
if( digitalRead(ir_car9) == HIGH && flag11==1)
{
slotL3 = slotL3+1;
flag11=0;
sensor9 = "0";
delay(200);
}

}
```

Appendix B Coding of ESP-8266 Wi-Fi

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>
char auth[] = " d3jEhEAztQTkGO9U_Nx3pzK7QKYxNn5e";// set autoken
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "caina";// set wifi name
char pass[] = "azacun97"; // set password
SimpleTimer timer;
String myString; // complete message from arduino, which consistors of snesors data
char rdata; // received charactors
int firstVal, secondVal,thirdVal; // sensors
intl ed1,led2,led3,led4,led5,led6,led7,led8,led9,total,data,data2;
// This function sends Arduino's up time every second to Virtual Pin (1).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void myTimerEvent()
{
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V0, millis() / 1000);
}
void setup()
{
// Debug console
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
timer.setInterval(1000L,sensorvalue1); //timer utk led dtg ke blink, 1000L=
10 second
timer.setInterval(1000L,sensorvalue2);
timer.setInterval(1000L,sensorvalue3);
timer.setInterval(1000L,sensorvalue4);
timer.setInterval(1000L,sensorvalue5);
timer.setInterval(1000L,sensorvalue6);
timer.setInterval(1000L,sensorvalue7);
timer.setInterval(1000L,sensorvalue8);
timer.setInterval(1000L,sensorvalue9);
timer.setInterval(1000L,sensorvalue10);
timer.setInterval(1000L,sensorvalue11);
timer.setInterval(1000L,sensorvalue12);
}
```

```
void loop()
{
if (Serial.available() == 0 ) //
{
Blynk.run();
timer.run(); // Initiates BlynkTimer
}
if (Serial.available() > 0 )
{
rdata = Serial.read();
myString = myString+ rdata;
// Serial.print(rdata);
if( rdata == '\n')
{
Serial.println(myString);
// Serial.println("fahad");
// new code
String l = getValue(myString, ',', 0); //value yg data dg dr arduino
String m = getValue(myString, ',', 1);
String n = getValue(myString, ',', 2);
String o = getValue(myString, ',', 3);
String p = getValue(myString, ',', 4);
String q = getValue(myString, ',', 5);
String r = getValue(myString, ',', 6);
String s = getValue(myString, ',', 7);
String t = getValue(myString, ',', 8);
String u = getValue(myString, ',', 9);
String v = getValue(myString, ',', 10);
String w= getValue(myString, ',', 11);
// these leds represents the leds used in Blynk application
led1 = l.toInt();
led2 = m.toInt();
led3 = n.toInt();
led4 = o.toInt();
led5 = p.toInt();
led6 = q.toInt();
led7 = r.toInt();
led8 = s.toInt();
led9 = t.toInt();
total =u.toInt();
data = v.toInt();
data2 = w.toInt();
myString = "";
// end new code
}
}
}
```

```
void sensorvalue1() // sensorvalue is define dlm blink. led kelip. dia akan
tunjuk la pin mana pakai.
{
int sdata = led1;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V1, sdata); //define pin kat blink.
}
void sensorvalue2()
{
int sdata = led2;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V2, sdata);
}
void sensorvalue3()
{
int sdata = led3;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V3, sdata);
}
void sensorvalue4()
{
int sdata = led4;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V4, sdata);
}
void sensorvalue5()
{
int sdata = led5;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V5, sdata);
}
void sensorvalue6()
{
int sdata = led6;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V6, sdata);
}
void sensorvalue7()
{
int sdata = led7;
// You can send any value at any time.
```

```
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V7, sdata);
}
void sensorvalue8()
{
int sdata = led8;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V8, sdata);
}
void sensorvalue9()
{
int sdata = led9;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V9, sdata);
}
void sensorvalue10()
{
int sdata = total;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V10, sdata);
}
void sensorvalue11()
{
int sdata = data;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V11, sdata); //define pin kat blink.
}
void sensorvalue12()
{
int sdata = data2;
// You can send any value at any time.
// Please don't send more than 10 values per second.
Blynk.virtualWrite(V12, sdata); // define pin kat blink.
}
String getValue(String data, char separator, int index) // comman program utk esp.
wajib ade.
{
int found = 0;
int strIndex[] = { 0, -1 };
int maxIndex = data.length() - 1;
for (int i = 0; i <= maxIndex && found <= index; i++) {
if (data.charAt(i) == separator || i == maxIndex) {
found++;
```

```
strIndex[0] = strIndex[1] + 1;
strIndex[1] = (i == maxIndex) ? i+1 : i;
}
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```