PERFORMANCE ANALYSIS OF IOT DEVELOPMENT BOARDS FOR GPS APPLICATIONS

THAM YEE YANT

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

PERFORMANCE ANALYSIS OF IOT DEVELOPMENT BOARDS FOR GPS APPLICATIONS

THAM YEE YANT

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2019/2020

DECLARATION

I declare that this report entitled "Performance Analysis of IoT Development Boards for GPS Applications" is the result of my own work except for quotes as cited in the references.

Signature :

Author : THAM YEE YANT

Date : 26/6/2020

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name : Radi H Ramlee

Date : 26/6/2020

DEDICATION

A specially dedicated to my beloved family, friends, lecturers and supervisor who always support me and guide me to complete my final year project successfully.

ABSTRACT

Internet of Things (IoT) is defined as a network devices comprise with electronic sensor and software product which communicate with each other to make our physical world become real and smarter. Hence, smaller size of IoT development boards or IoT microcontroller units (MCU) perform various applications in an IoT projects due to computer system on a chip, processor and programmable input and output peripherals which allow interact and exchange the data in cloud. Technology become the rescue and made the revolution changes which enable our life more convenience, easier and innovations. As we know that Global Positioning System (GPS) technology often utilized mostly in our daily life as we depend mostly to pinpoint the position of location based satellite system. Therefore, the performance of IoT development boards based GPS application is implemented and analysed. As part of a research, IoT development boards is studied and implemented with GPS sensor to test and analyse their performance based on the parameters on accuracy of locations latitude, longitude, altitude, speed, distance and duration time between two locations. Real time GPS data is received and stored in Thingspeak cloud for further research. The best performances of IoT development boards is verify based on the results compared to the Google Map applications.

ABSTRAK

Internet Perkara (IoT) ditakrifkan sebagai peranti rangkaian terdiri daripada sensor elektronik dan produk perisian yang berkomunikasi antara satu sama lain untuk menjadikan dunia fizikal kita menjadi nyata dan lebih bijak. Oleh itu, saiz lebih kecil papan pengembangan IoT atau unit mikrokontroller IOT (MCU) melaksanakan pelbagai aplikasi dalam projek IoT kerana sistem komputer pada cip, pemproses dan input dan keluaran perisian yang boleh diprogramkan yang membolehkan berinteraksi dan menukar data dalam awan. Seperti yang kita tahu bahawa Sistem Penentududukan Global (GPS) sering digunakan terutamanya dalam kehidupan seharian kita kerana kebanyakan bergantung pada teknologi GPS untuk menentukan kedudukan sistem satelit berasaskan lokasi. Oleh itu, pelaksanaan aplikasi berasaskan papan pemajuan IoT dilaksanakan dan dianalisis. Sebagai sebahagian daripada penyelidikan, papan pengembangan IoT dikaji dan dilaksanakan dengan sensor GPS untuk menguji dan menganalisis prestasi mereka berdasarkan parameter mengenai ketepatan latitud, longitud, ketinggian dan tempoh masa antara lokasi. Data GPS masa nyata diterima dan disimpan di cloud Thingspeak sebagai rujukan. Persembahan terbaik lembaga pembangunan IoT disahkan berdasarkan keputusan berbanding dengan aplikasi Peta Google.

ACKNOWLEDGEMENTS

I am pleasure with having a valuable chance and opportunity to express my gratitude to those who supporting and guiding me to complete my final year project in Universiti Teknikal Malaysia Melaka (UTeM). First and foremost, I would like to appreciate to all my family members for their tremendous support and encouragement all the times. Moreover, I would like to give a sincere thanks to my supervisor, Mr. Radi Husin Bin Ramlee for his abundant advices, valuable suggestion and motivation by providing me all the information and knowledge about this project. Last but not, least, special thanks to all my friends who lend their hand for completing my project. Thankyou.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

IOT	:	Internet of Things
MCU	:	Microcontroller Units
GPS	:	Global Positioning System
IT	:	Information Technology
CAGR	:	Compound Annual Growth Rate
SoC	:	System on Chip
WLAN	:	Wireless Local Area Network
UWB	:	Ultra Wideband
RSSI	:	Received Signal Strength Indicator
AoA	:	Angle of arrival
ToF	:	Time of flight
NMEA	:	National Marine Electronics Association
TDOA	:	Time-difference of arrival
IDE	:	Integrated Development Environment
CAGR	:	Compound Annual Growth Rate
SoC	:	System on Chip
WLAN	:	Wireless Local Area Network
GPRS	:	General Packet Radio Service

GSM	:	Global System for Mobile Communications
SQL	:	Structured Query Language
ASP	:	Active Server Page
JS	:	JavaScript
USB	:	Universal Serial Bus
HDMI	:	High-Definition Multimedia Interface
ARM	:	Advanced RISC Machines
BCM	:	Broadcom
GPIO	:	General Purpose Input/Output
CPU	:	Central Processing Unit
SRAM	:	Static Random-Access Memory
SDRAM	:	Synchronous Dynamic Random-Access Memory
EEPROM	:	Electrically Erasable Programmable Read-Only Memory
IDLE	:	Integrated DeveLopment Environment
SD	:	Secure Digital
DC	:	Direct Current
ASCII	:	American Standard Code for Information Interchange
DGPS	:	Differential Global Positioning System
RS & MS	:	Reference Station & Mobile Station
GUI	:	Graphical User Interface
SMS	:	Short Message Service
MSE	:	Mean Squared Error
UART	:	Universal Asynchronous Receiver-Transmitter
TX & RX	:	Transmitter & Receiver
I^2C	:	Inter-Integrated Circuit

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

Global positioning system (GPS) is a powerful outdoor location positioning technology in the world as everyone of us depend on GPS to locate their exact locations based satellite system. There are various innovative project of Internet of Things (IoT) devices based GPS tracker dedicated GPS application arise in different sector. However, GPS tracker (GPS sensor) performance is varied with real GPS (google map) when implement with smaller size IoT devices. Therefore, IoT development boards had been studied and the performance of GPS sensor application on IoT development boards is analyzed and evaluated. Project background, problem statement, objectives, scope of project and overview of the chapter is presented in this chapter.

1.1 Project Background

In this universal modernization decade, rapid growth of innovation technology have enhance our world with high technology services. Commonly, Global Positioning System (GPS) becomes high demands application technologies which prompt users navigate around the world as well as to track the information and location of people during this Covid-19 pandemic. Global positioning system (GPS) such as Waze, Google Maps, HERE maps and etc are developed based on GPS application. Due to powerful and advance technology of GPS, it becomes high demand for various sectors such as information technology (IT), transportation and telecommunication in the market. GPS also incorporated to form navigation devices or apps in order to enhance security and safety of user.

Nowadays, there are wide range of smaller size IoT development boards developed by company for developer to build up their most standalone prototype and project easily. This is due to their on board specification such as processor, plug-in peripherals and sensor accessibility. According to global market size, product of microcontrollers with an estimated USD 20.82 billion in 2019 and is projected to grow from 2020 to 2027 at a Compound Annual Growth Rate (CAGR) of 10.8%. The most commonly used of IoT development boards to build a project are Arduino, NodeMCU, Raspberry Pi and etc. Basically, implement of IoT development boards and various sensor are connect together to form a smart system, mobile and app through the concept of Internet of Things (IoT) most importantly for innovating, analyzing and collecting data for further research.

Hence, performance of IoT development boards with GPS application is implemented to support computer programming, collect data by firmware and transfer the data to cloud server. Precision and the best achievements based on research results is showed and analyzed according to an accuracy parameters such as locations latitude, longitude, altitude, speed, distance and duration of time between two locations.

1.2 Problem Statement

Nowadays, an innovative and intelligent prototype and project provide a new outcome and advanced service which benefit to most of the sectors. This is because development and growth of variety of smaller size, efficient yet portable IoT development boards become a dominant demand trend in globally. Furthermore, most specification of IoT devices different in term of System on Chip (SoC) processor, peripherals connectivity and core of the devices. Hence, this research is conducted to investigate and analyze the performance of IoT development boards for GPS application based on the precision parameter on accuracy position of locations latitude, longitude, altitude, speed, distance and duration time between two locations. As we know that, GPS becomes important part of our life as GPS help us to track and pinpoint a location all around the world based navigation system. However, GPS data received from satellite when implement with different IoT Microcontroller units (MCU) become an issues among the GPS receiver part. Nevertheless, factor such as the block of signal in indoor and outdoor position, building and atmospheric conditions will affect the precision of GPS. As a result, the best performance of IoT development boards is verified for further researchers.

1.3 Objectives

Below are the three objectives that need to be achieved:

i. To analyze IoT development boards among NodeMCU and Raspberry Pi 3 for Global Positioning System (GPS) application using two different software.
ii. To evaluate the performance analysis of two different IoT development boards and update the device location in cloud database platform.
iii. To verify the best performance of the IoT development board based on parameter accuracy of locations latitude, longitude, altitude, speed, distance and duration time between two locations.

1.4 Scope of Work

Primary scope of this project is limited for GPS application of Internet of Things (IoT). Limited only on two different IoT development board that is commonly available in the market. GPS module GY-NEO6MV2 is used to implement with IoT development Boards which are Arduino NodeMCU and Raspberry Pi 3 Model B. The experiment is conducted from the locations Taman Raub Jaya 2 to KFC Raub, Pahang. Data received from GPS sensor interface with two different IoT development board is shown using Thingspeak cloud database which running on software Arduino IDE and OS Raspbian using C/C++ and Python Language. Performance result based precision locations latitude, longitude, altitude, speed, distance and duration time is analyzed and compared with Google Map application. All in all, data analysis of data is used to determine the best performance of IoT devices for GPS applications.

1.5 Project Significant

Aim of this project is predominantly to analyze the performance of IoT development boards for GPS application. This developed project can act as a reference for further research. As the precision and accuracy of GPS data from the receiver part is varied when interface with different IoT development boards. Apart from that, implementation of this system is low cost, efficient, reliable and protect human's security and health due to an incredible invention of GPS technology which used globally in various sector by navigation and tracking the locations all around the world.

Today, our life become easier due to the smarter Internet of Things. Various IoT project and prototype is build using IoT development boards while the specification and infrastructure of hardware and software embedded in IoT development boards need to manage and consider before building up a projects. This is based on computer system on a chip, processor and programmable input and output peripherals which allow interact and exchange the data in cloud. Therefore, specification of two different IoT development boards is studied to implement with GPS module. Lastly, analysis data is carried out to verify the best performance of IoT development boards based parameter precision locations latitude, longitude, altitude, speed, distance and duration time.

1.6 Thesis Outline

There are five chapter covered in this thesis. Introduction of the project is presented in Chapter 1. Chapter 2 discusses the background study on previous project proposed which includes overview Global Positioning System (GPS), Internet of Things (IoT), capability of the microcontroller board of Arduino Node Mcu, Raspberry Pi 3 and implement of Internet of Things (IoT) platforms. In Chapter 3, project methodology and flow chart design to implement the process is further discussed. Chapter 4 presented and discussed about the results obtained from an experiments. Lastly, recommendation for future work is summarized in Chapter 5.

CHAPTER 2

BACKGROUND STUDY

This chapter will discuss the theory and background of indoor and outdoor positioning system, Global Positioning System (GPS) and Internet of Thing (IoT). To perform and compare an analysis of this project, various journal and research paper from previous research based IoT development boards with GPS is studied and discussed in this chapter. Moreover, the performance measurement accuracy of GPS data from previous research are also discussed in this chapter.

2.1 Indoor and Outdoor Positioning System

Demand of positioning system become vital in part of our life. An ideal positioning system provide an accurate service which able to track our locations in anytime and anywhere, indoor as well as outdoor. There are numerous positioning systems developed which includes of GPS positioning system, WLAN positioning system and

UWB (ultra wideband) positioning as shown in Figure 2.1. GPS is an excellent which provide the most accurate position and operate in outdoors. Range of accuracy is within 10 meters with an accurate timing but it reached about 50 meters in urban centres due to blockage of signal and multipath of signal reflection on large buildings. However, the advanced GPS technology have made success the navigation application based outdoor positions[1].

Wireless local area network (WLAN) is an indoor positioning system which able to determine the position location of the user instead of GPS. Most WLAN positioning system have also been developed for outdoor environment based GPS positioning system in order to improve the accuracy of indoor and outdoor positioning system. Techniques apply in WLAN consists of technology of RSSI, Fingerprinting based RSSI, Angle of arrival (AoA) and Time of flight (ToF). Accuracy for indoor positioning using WLAN covered from 5 to 15 meters. Therefore, indoor positioning based WLAN are possible and proved to provide more accurate services. Moreover, Ultra-wideband (UWB) provide high accurate locations due to characteristic of the short impulse transmissions signal. This system consider high cost but low power consumption as modulators, demodulators and IF stages are not required by transmitters and receivers of UWM. UWM contributes in many applications especially in positioning location and radar system [2].

		Outdoors				
	Indoor	Downtown	Residential	Suburban	Countryside	Highways
GPS	0% -	100% 27m	100% 3m	100% 2m	100% 5m	100% 5m
GSM Cell-ID	100% same as outdoors	100% 219m	100% 384m	100% 438m	100% 941m	100% 704m
GSM Centroid	100% ~5m^a	100% 177m	100% 301m	100% 318m	100% 680m	100% 572m
WLAN	98% ~5m^a	100% 26m	100% 27m	92% 45m	46% 77m	73% 155m
Bluetooth	31% ~2m^a	0% -	0% -	0% -	0% -	0% -

Figure 2.1: Range and Accuracy of Location Positioning system [5]

2.1.1 Global Positioning System (GPS)

In this universal modernization decade, rapid growth of technology have enhance our world with high technology services shown in Figure 2.2. Commonly, Global Positioning System (GPS) becomes high demands which prompt users utilize as part of daily life in every aspect and navigate around the world. Due to powerful and advance technology of GPS, it becomes high demand for various sectors such as information technology (IT), transportation and telecommunication in the market. GPS incorporated to form navigation devices or apps in order to enhance security and safety [3].

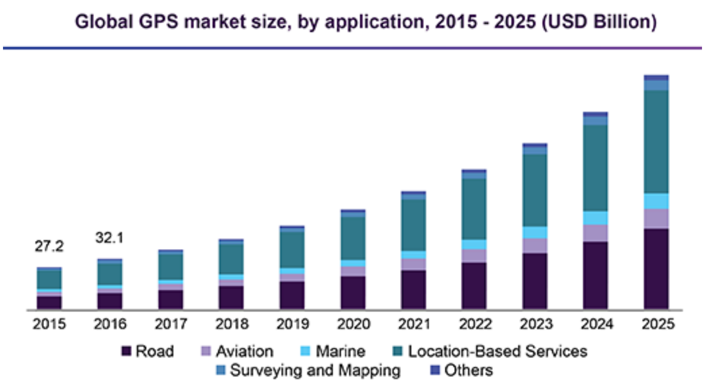

Figure 2.2: Trend of Global GPS market size [3]

2.1.2 How GPS Work

Hoque,(2016) [4] described the modern technology of Global Positioning System (GPS) is the most widely-used satellite-based navigation system due to rapid growth development of technologies in this 21th century. The satellite able to transmit all the signals and received by GPS receiver. Then, GPS receiver received the signal information and users able to track their own locations in the Earth as shown in Figure 2.3 below. The importance of GPS contribute to different field of research and development and thus rapidly increase for further GPS application [5].

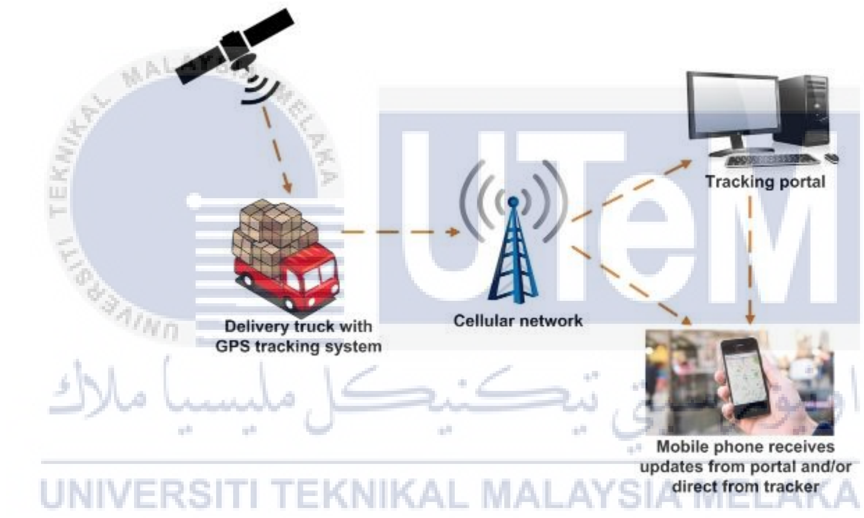

Figure 2.3: Real-time GPS tracking system [4]

2.1.3 Features of Global Positioning System

Hammami, (2018) [6] stated that the main features of GPS provides the accuracy position location within 2 centimetre when the specialized equipment is placed closely to the receiver. Distance between GPS receiver and satellites about the position on the Earth is known through the process of triangulation. Besides that, GPS provide a precision of time due to an atomic clocks on each satellites which have the accuracy within a nanosecond while the Quartz clock in the receivers which have a property of constant synchronization with the satellite's atomic clock cause the GPS consist precise timing. Moreover, GPS includes of space segment, control segment and user segment as shown in Figure 2.4 below. Transmission of navigation signal are all occurs in space segment which is controlled by which 24 satellites orbiting around the Earth. Function of control segment in GPS is to track the location, analyze and update the navigation data in order to maintain the stable satellite constellation. An intelligence GPS receiver that received the GPS signal to retrieve the highly accurate position and time is known as user segment and which reveal the major part concern in this project. The blending of this three segment become free and open source system which adopted by user and applied in every sector around the world as mentioned by Yin & Arellano, (2012) [7].

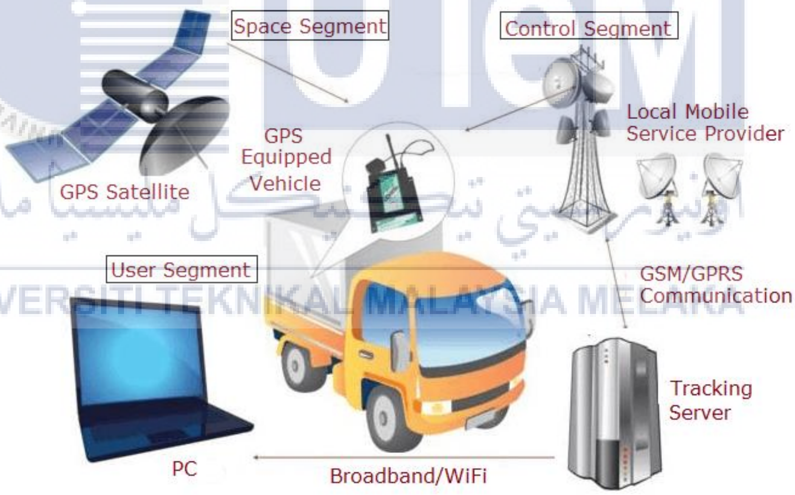

Figure 2.4: GPS Space Segment, Control Segment and User Segment [7]

Furthermore, Computer programming and manufacturer GPS support the standard format National Marine Electronics Association (NMEA). It is used to transmit and receive the data locations in GPS receiver. Structure message of NMEA is formed in standard line sentences that is independently from other sentences. "GP" is the characterize prefix for GPS receiver. Content of the sentences is described by the following three sequence letter. Every sentences start with a '$' and ends with carriage return. Data items covered inside the sentences is separated by commas for a program to read the data. Figure 2.5 below show NMEA data from GPS receiver [8].

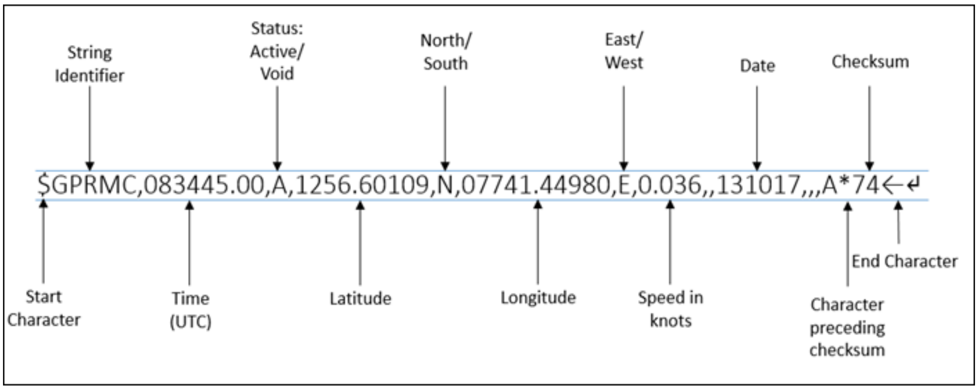

Figure 2.5: Read GPS Data from NMEA standard [8]

2.1.4 Principle of Triangulation

GPS system broadcast our location and time which have the same concepts of the techniques of triangulation process. Geometric properties of triangle is applicable in principle of triangulation process to measure the distances. Based on the Figure 2.6 below, there are three main points which are N_1 *and* N_2 as two references points with (T) as a target point. φ_1 *and* φ_2 are obtain based on the intersection line from N_1 *to* T and N_2 *to* T [9].

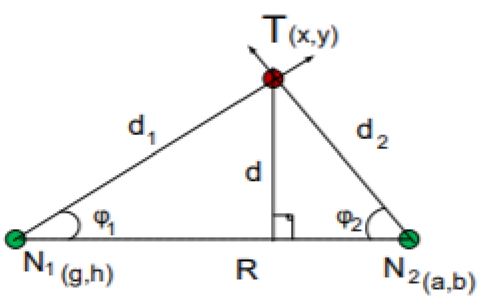

Figure 2.6: Theory of Triangulation [9]

Target position is proved to calculate by using trigonometric functions based on the equation below.

$$R = \frac{d}{\tan\varphi_1} + \frac{d}{\tan\varphi_2} \qquad (2.1)$$

$$d = \frac{R * \sin\varphi_1 * sin\varphi_2}{sin(\varphi_1 + \varphi_2)} \qquad (2.2)$$

While d is the perpendicular line between T and R.

$$\varphi_1 = tan^{-1}\left(\frac{h-y}{g-x}\right) \qquad (2.3)$$

φ_1 is the angle between $d_1 and\ R$. Coordinate of $N_1\ is\ defined\ as\ (g, h)$.

$$\varphi_2 = tan^{-1}\left(\frac{b-y}{a-x}\right) \qquad (2.4)$$

φ_2 is the angle between $d_2 and\ R$. Coordinate of $N_2\ is\ defined\ as\ (a, b)$.

Moreover, coordinate of (x, y) from T can be derived from equation below.

$$x = \frac{b - h - a\tan\varphi_2 + g\tan\varphi_2}{\tan\varphi_1 - \tan\varphi_2} \qquad (2.5)$$

$$y = x\tan\varphi_2 + (b - a\tan\varphi_1 \qquad (2.6)$$

Therefore, distance is determined,

$$d_1 = \| g - x \| = \sqrt{(g-x)^2 - (h-y)^2} \qquad (2.7)$$

$$d_2 = \| a - x \| = \sqrt{(a-x)^2 - (b-y)^2} \qquad (2.8)$$

The researchers suggested RSS (Received Signal Strength) method, AOA (Angle of Arrival) method and TOA (Time of arrival) and TDOA (Time-difference of arrival) methods to determine the position location as shown in Figure 2.7 below. It showed that RSS have high accuracy as time synchronization is not required in this method but it is necessary for TOA methods. However, it is achieved with minimum three reference node to locate the position and produce inaccurate location due to multipath of interference. Furthermore, AOA methods which have the same principle of triangulation is efficient to track the location with only two reference point compared to RSS methods. Meanwhile, accuracy location decreased as the distance increased since the propagation of received signal is changed. As mentioned by the researchers, TOA and TDOA depends on propagation of time travelled to measure the position. TOA and TDOA can be achieve high accuracy position and the accuracy calculation is not affected as long as the distance increased compared with AOA and RSS. Nevertheless, multipath propagation in TOA and TDOA generate the error of positioning location due to the changing of time of signal [10][11].

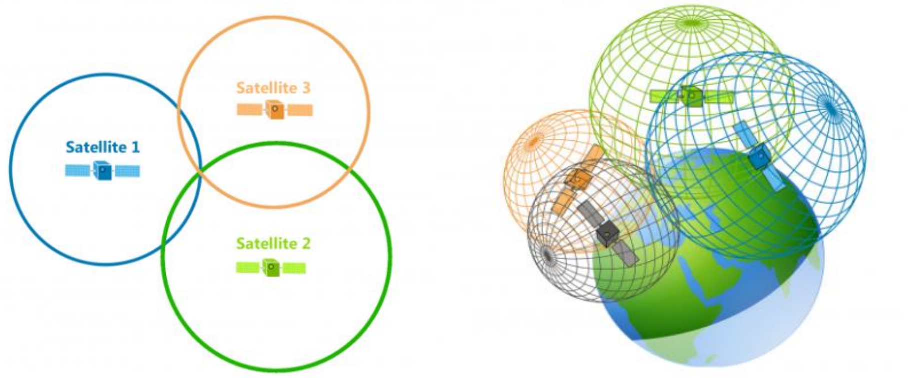

Figure 2.7: GPS based Triangulation Principle [10][11]

2.2 Internet of Things (IoT)

Bansal et al. (2019) [12] supported rapid development of technology have prompted the huge demand of IoT technology in our world as shown in Figure 2.8 below. It was appreciable that IoT nowadays are unique that able to transform the virtual world into the real world with the working of an internet. Internet of things is defined as the network system which link all the electronic sensors, devices and an objects with interrelation within internal state and external environment via automatically. IoT system delivers high performance and functionality to the end-user which enable the development of an application and services which support the innovated IoT product [13].

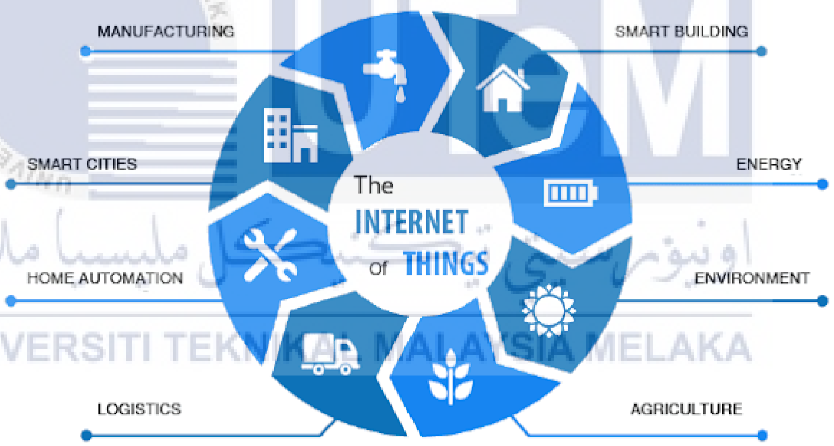

Figure 2.8: Internet of Things (IoT) [12][13]

2.2.1 Implement of IoT development boards with GPS module

Nowadays, the used of microcontroller rising in the market due to well-equipped ARM processor and control units. Microcontrollers is powerful and highly efficient as it consists of multitude integrated peripheral yet smaller in size compared to the computer or laptop [14][15]. Previous research based performance of IoT development boards with GPS module is studied and discussed in this chapter.

2.2.1.1 Arduino Node MCU

NodeMCU shown in Figure 2.9 below is the system on chip (SOC) which come from ESP8266 family. This board capable to read an input to function any electronic device and transfer it into an output to activate the devices using programming language of Arduino software which is Integrated Development Environment (IDE). NodeMCU has an embedded Wi-Fi system support unlike Arduino Uno which needs other connections.

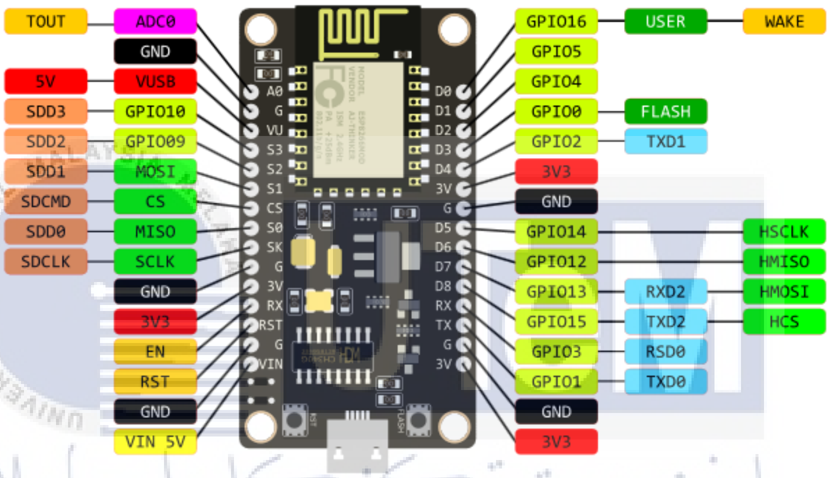

Figure 2.9: Node MCU Pinout Diagram [16]

The researchers Boddapati Venkata [16] had implemented a vehicle monitoring and tracking system based on microcontroller Arduino Node MCU with the GPS module. GPS module is utilize to detect the vehicles distance and the live location of the vehicles can be track. Node MCU is used as it has a build-in Wi-Fi module that able to collect all the data from all the sensors. The data collected then will be send to the cloud through internet for storage and further analysis. The main reasons where NodeMCU is selected as it consumed less power (3.3V), efficiency cost compared to the other microcontroller and sensors used only digital pins and analog pins. According to Ngui & Lee, (2019) [17] global positioning system and indoor positioning system can be implemented with the low power consumption mode applied. The author's state that weakness of GPS caused by the blockage of the GPS signal and high power consumption of GPS. Therefore, a portable and energy save module is built to track the location travelled using Arduino pro mini, GPS module, GSM module and GPRS module to transmit the data when the signal is received. RFID and accelerometer is function to trigger the indoor positioning system in order to calculate the location once the GPS signal is malfunction.

2.2.1.2 Raspberry Pi

Raspberry Pi is launched in 2012 and Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3 and now even Raspberry Pi 4 have been released today. This low cost microcontroller, credit card size computer as shown in Figure 2.10 run in Linux operating system designed for us to build and explore IoT projects yet in industrial application.

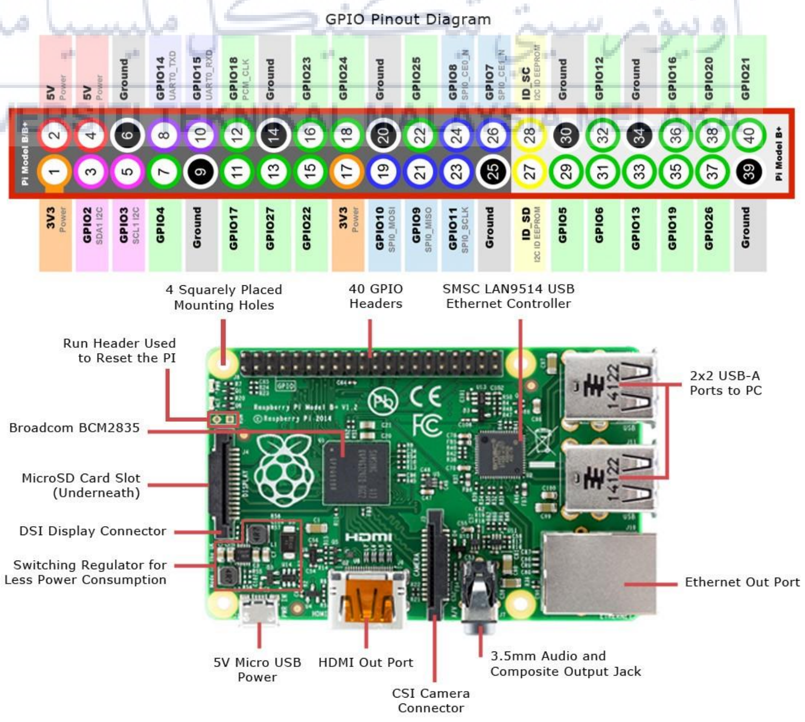

Figure 2.10: Raspberry Pi Pinout Diagram [18]

Shinde & Mane, (2015) [18] had proposed the vehicle monitoring tracking system based on the microcontroller Raspberry Pi (Embedded Linux board) using SIM908 Module which embedded GPS, GPRS and GSM shown in Figure 2.11 below. Alert message is received when the wrong path is detected and the speed is over the range. The vehicle’s position is determined on the web page in real time. It had proved that Raspberry Pi as an excellent features of MySQL database system for storing and updating the data of location, date, time and speed in a real-time.

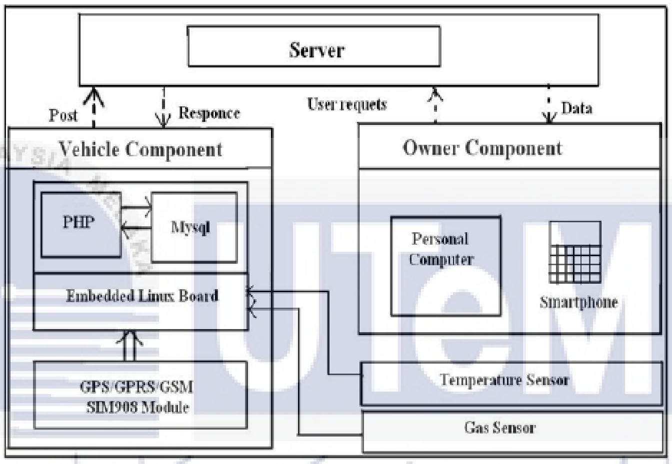

Figure 2.11: System Block Diagram [18]

According to Vokorokos et al.,(2018) [19] the researchers had conducted a study based on Raspberry Pi 3 with the GPS and GSM modules for education process. Furthermore, web application of ASP.NET and AngularJS were developed for manage the data from tracker and verified by the user by tested through the real data. Compared to Arduino, Raspberry Pi has built in USB ports, internet, Bluetooth and HDMI port, Python and Scratch as main programming languages and most important is it is a running operating system based Linux which lead to a powerful microcontroller compared to Arduino which studied by Desai & Phadke, (2017) [20]. The author implement raspberry pi with GPS module to track vehicle location and stored the data in the server as shown in Figure 2.12 and Figure 2.13 below.

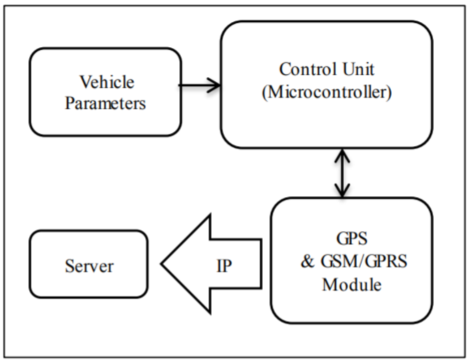

Figure 2.12: Proposed System for Vehicle Tracking [20]

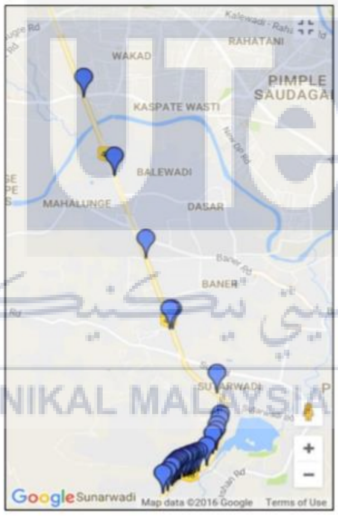

Figure 2.13: Vehicle location display on Google map [20]

Saha et al., 2017 [21] indicated to implement the GPS smart spy surveillance robotic system using Raspberry Pi available for remote sensing and security application as shown in Figure 2.14 below. The efficiency of the functional microprocessor of Raspberry Pi 3, Gripper, Pi NoIR camera module, ultrasound distance sensor, GPS module and Antenna SKG13C are interfaced and implemented with each other to function the robotic system. They found that single board of model

Raspberry Pi 3 have the advantages in very powerful in-built BCM2837 64 bit ARMv8 processor and has a great performance than the previous Model, embedded with Wi-Fi, Bluetooth and run on Linux operating system.

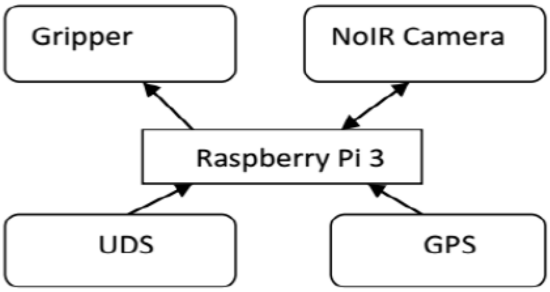

Figure 2.14: Block diagram of smart spy surveillance robotic system [21]

2.2.1.3 Comparative IoT Development Boards

Table 2.1: Comparison Features between NodeMCU, Raspberry Pi 3 Model B

Microcontroller	**NodeMCU**	**Raspberry Pi 3 Model B**
Operating Voltage	3.3V	5V
Processor	Tensilica 32-bit RISC CPU Xtensa LX106	64-bit quad-core ARM Cortex-A72
Clock Speed	80MHz	1.2GHz
System memory	64KB, SRAM	512MB, SDRAM
GPIO	16	40
Operating system	XTOS	Linux distribution
CPU	ESP8266 (LX106)	ARM11 @700MHz
Storage	Flash, EEPROM	Micro-SD
Integrated Development Environment	Arduino IDE	Scratch, IDLE, Squeak, Python, Java, Linux
Embedded Wi-Fi	Yes	Yes
Developer	ESP8266 Opensource Community	Android developer

2.2.1.4 GPS Module Neo6MV2

Two important part which are Neo-6M U-Blox on-board memory chip with a ceramic patch antenna able to implement with a wide range of microcontroller to build a system effectively based GPS application as shown in Figure 2.15 below. Range of the DC input is 3.3V to 5V with the baud rate 9600 and it has the high sensitivity for indoor applications.

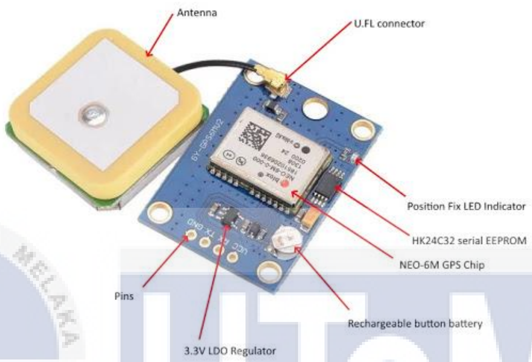

Figure 2.15: GPS Module Neo6MV2 [22][23]

NMEA is defines as National Marine Electronics Association based on ASCII with the standard format of NMEA-0183. All the information data received through the GPS receiver are compute in NMEA format which show the complete position, velocity and time information in the form of their standard sentences which supported by Shi, et al.,(2013) [22][23]. Table 2.2 below show some of the GPS NMEA sentences with descriptions.

Table 2.2: Description of GPS NMEA Sentences

Sentences	Description
$GPGGA	Global Positioning System Fix Data.
$GPGLL	Geographic position, latitude/longitude

$GPGSA	GPS DOP and active satellites.
$GPRMC	Recommended minimum specific GPS/Transit data.
$GPTRF	Transit Fix Data.
$GPSTN	Multiple Data ID.
$GPVBW	Dual Ground/Water Speed
$GPVTG	Track made good and ground speed
$GPXTE	Cross-track error, Measured
$GPZDA	Date & Time

2.3 GPS Location (Latitude and Longitude), distance, speed and time

Trend of GPS technology has growth into the world recently. Most of user rely on GPS apps such as Google map, waze and etc which able to track the locations on Android, record and display the route map of journey with detail speed time and distance travelled. Different algorithm can be applied for determine the distance location. According to Winarno et al. (2018) [24], the researchers proved and implemented Haversine method which is the great circle distance to obtain the distance between two points on sphere. Haversine formula is important for navigation used as the shortest distance at any two points can be calculated based on GPS data of latitudes and longitudes obtained. Latitude is defined as angle between straight of the location that measured in phi while Longitude is defined as position of angle points (West or East) by Prime Meridian or Greenwich meridian. To test the calculation of distance, the latitude and longitude known are converted to the radian value by multiplied each with the constant 0.0174532925. Therefore, it can be proved to calculate the distance among two points by using Harvesine formula shown below:

$$d = R\sqrt{x * x + y * y} \tag{2.9}$$

$$x = (\Delta long2 - \Delta long1) * \cos\left(\frac{\Delta lat1 + \Delta lat2}{2}\right) \tag{2.10}$$

$$y = \Delta lat2 - \Delta lat1 \tag{2.11}$$

According to Jiménez-Meza et al.,(2013) [25], ground distance based on latitude and longitude between two point locations can be implemented by using the simple math function. Efficiency of using simple math equations able to reduce programming and processing computation. Distance can be calculated by

$$d = \sqrt[2]{A^2 + B^2} * 1609.344 \tag{2.12}$$

d is the distance in meters, 1609.344 is the constant that convert miles to meters.

$$A = 69.1 * (lat2 - Lat1) \tag{2.13}$$

$$B = 69.1 * (Lon2 - Lon1) * COS\left(\frac{Lat1}{57.3}\right) \tag{2.14}$$

69.1 and 57.3 are the constants used to convert degree into ground distance in miles. It is showed that one degree of latitude is equal to 69.1 miles while one degree of longitude is equal to 57.3 miles.

Moreover, [26] suggested to calculate the speed, km/h between two different location points from GPS data using

$$v = \frac{d}{t} * 3.6 \tag{2.15}$$

d is the distance measured in meters, time, t in seconds and the constant 3.6 is applied as to transform m/s to km/h.

$$t = (ct2 - ct1) * 86400 \qquad (2.16)$$

t defined the time travel between two points in seconds, ct1 and ct2 are the respective time in GPS coordinate 1 and coordinate 2. Constant 86400 is multiplied as the format of 24 hours in seconds.

2.4 Performance of GPS Precision and Accuracy

Global Positioning System (GPS) is used military in almost industry sector as well as transport sector. Time of bus arrival is hard to estimate and this are the main problems faced by most of the public transport users. T.P.Fowdur and M.N.Khodabacchus proposed a real-time cloud based bus tracking system based different prediction algorithm shown in Figure 2.16 below [27][28]. Adafruit GPS Module, Arduino Uno microcontroller and SIM900 GSM module is used to build the bus tracking system based on different predicted algorithm. Based on this journal, estimated arrival time and speed of vehicle can be calculated based on the formula below:

$$Estimated\ arrival\ time = \frac{Distance\ from\ destination}{Vehicle\ current\ speeed} \qquad (2.17)$$

$$Speed\ of\ vehicle\ (Kmh^{-1}) = \frac{Distance\ between\ two\ GPS\ points}{Time\ Taken} \qquad (2.18)$$

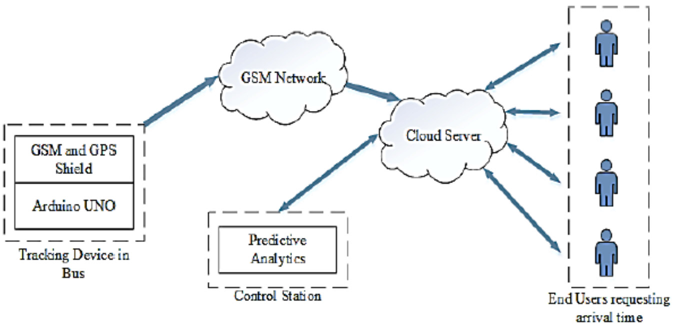

Figure 2.16: Proposed bus tracking system based Predictive Analytics [27][28]

Furthermore, GPS technology is important for parent to track the location of their children to ensure safety purpose. Based on statistics from Police Malaysia, there are exactly 834 cases children missing since 2014 from reported. M. afnan Dzulkifi and S.Sulaiman [29][30][31] proposed the tracking system based GPS and GSM module to obtain the GPS coordinate data such as latitude and longitude while the data location imformation will received by mobile phone. Accuracy GPS data received is verified and compared with the Google Map website. Figure 2.17, Figure 2.18, Figure 2.19, Figure 20 and Figure 21 below show the proposed tracking system and the result of the data.

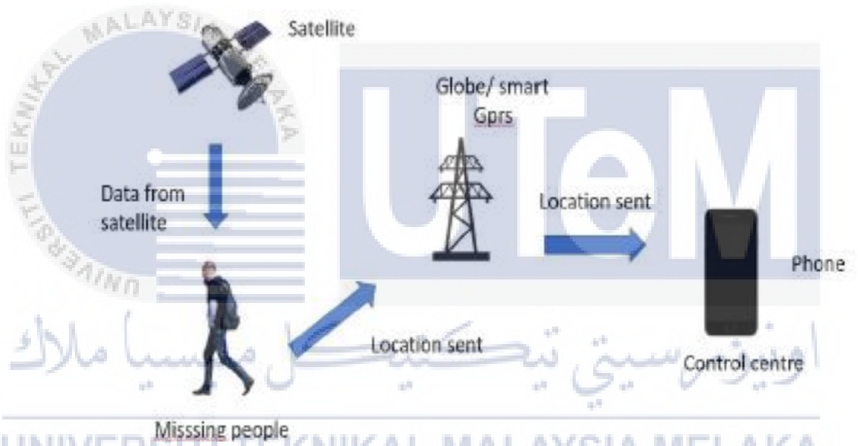

Figure 2.17: Proposed GPS tracking system GPS and GSM module [29][30][31]

Location	Tracking Number	Delay
(Stadium Shah Alam)	Latitude:3.083387 Longitude:101.545421	35 seconds
(KFC Shah Alam)	Latitude:3.087091 Longitude:101542151	42 seconds
(Terminal Bas Seksyen13)	Latitude:3.085355 Longitude:10154671	50 second

Figure 2.18: Result accuracy GPS data [29][30][31]

Range of satellite- based navigation system GPS is about 15-25meter and it is made up about 24 satellites. However, to improve and increases the accuracy positioning data, Differential Global Positioning System (DGPS) is utilized for reduce the errors of GPS signal with only the position data. The author [32] Asaad and Majeed proposed a system of real time DGPS using SMS services of GSM network based on Reference Station (RS) and Mobile station (MS). The author proved that the designed system based SMS service achieved the horizontal accuracy of 0.44m and vertical accuracy of 0.55m. Data of latitude, longitude and Altitude is calculated as formula below:

$$Latitude\ (\lambda) = tan^{-1}\frac{X}{Y} \tag{2.19}$$

$$Longitude\ (\varphi) = tan^{-1}\frac{Z}{\sqrt{(X^2+Y^2)(1-(2-f))(f)}} \tag{2.20}$$

$$Altitude\ (h) = \frac{\sqrt{(X^2+Y^2)}}{cos\ (\varphi)-N} \tag{2.21}$$

Where

$$X = (N + h)\cos(\varphi)\cos(\lambda)$$

$$Y = (N + h)\cos(\varphi)\sin(\lambda)$$

$$Z = ((1 - f)^2 . N + H)\sin(\varphi)$$

$$N = \frac{a}{\sqrt{(1 - e^2 sin^2\varphi)}}$$

Φ is the Longitude in decimal degree, λ is the latitude in decimal degree, h is the altitude, and N is the radius of curvature , a = 6378137.0m

Position No.	Longitude (E)	Latitude (N)	Altitude
P1	45° 26' 37.32''	35° 33' 58.51''	876.56 m
P2	45° 26' 37.39''	35° 33' 58.65''	876.81 m
P3	45° 26' 36.85''	35° 33' 59.52''	877.18 m
P4	45° 22' 56.16''	35° 34' 48.2''	757 m
P5	45° 23' 30.77''	35° 34' 17.67''	776.5 m
P6	45° 19' 26.21''	35° 33' 39.35''	733.82 m
P7	45° 19' 36.17''	35° 33' 36.32''	731.56 m

Figure 2.19: Manually Calculated position of GPS latitude, longitude and altitude [30][31]

Experiment No.	Average of horizontal error before correction (meters)	Average of horizontal error after correction (meters)	Percentage of error correction %
ES1	2.60	0.25	90.0
ES2	2.80	0.35	87.3
ES3	2.10	0.35	83.3
ES4	2.49	0.79	68.0
ES5	2.96	0.46	84.4
ES6	2.25	0.24	88.0
ES7	2.30	0.64	72.0

Figure 2.20: Percentage error result of average horizontal error [30][31]

Experiment No.	Average of vertical error before correction (meters)	Average of vertical error after correction (meters)	Percentage of error correction %
ES1	9.5	0.24	97.0
ES2	9.72	0.38	96.0
ES3	9.18	0.24	97.0
ES4	9.32	0.80	96.0
ES5	4.80	0.30	93.7
ES6	2.25	0.24	95.0
ES7	6.24	2.08	68.0

Figure 2.21: Percentage error result of average vertical error [30][31]

Accuracy and precision of GPS technology plays an essential role for tracking locations or an objects. There are grows of various GPS receiver module in the market, while the GPS receiver module we applied in our project usually will affect the accuracy and precision of GPS results. Single point location error, two point's location error and area location error are the sample of GPS location error. Yong He and Haiyong et.al proposed a study on GPS location error while different method location measurement and mathematics is proved to regulate the GPS accuracy. The author show that measurement error can be reduced by using two points' relative location methods. Figure 2.22 below show the analysis error orientation of two points conduct by the author which show that the accuracy is below 1.2 meters for two points relative location [33][34][35].

Britain Greenwich observatory time	Latitude of point A (degree. Second)	Longitude of point A (degree. Second)	Latitude of point B (degree. Second)	Longitude of point B (degree. Second)	Distance (m)	Distance Error (m)	Average Error (m)
065442.28	30.161997	120.114654	30.162485	120.114576	91.146	1.146	
065448.28	30.161997	120.114655	30.162483	120.114567	90.909	0.909	
065450.37	30.161997	120.114655	30.162483	120.114565	91.017	1.017	
065454.28	30.161997	120.114655	30.162483	120.114565	91.017	1.017	1.108
065458.25	30.161997	120.114655	30.162483	120.114566	91.019	1.019	
065500.28	30.161997	120.114655	30.162484	120.114564	91.271	1.271	
065504.28	30.161997	120.114655	30.162483	120.114561	91.378	1.378	

Figure 2.22: Analysis result based distance error [33][34][35]

2.5 Benchmark of GPS analysis performance approaches

Table 2.3: Performance analysis table for different approach

Author	Method and explanation	Performance
T.P.Fowdur and M.N.Khodabacchus [27][28]	Predict accuracy of bus arrival time based prediction algorithm. (Moving Average, Linear Regression, Polynomial	Adaptive algorithm achieved the lowest MSE (average MSE of 0.0297) which provide maximum

	Regression, Autoregressive Integrated moving average, K-Nearest neighbours, Neural network and adaptive algorithm)	accuracy for bus arrival prediction time.
M. Afnan Dzulkifi [29][30][31]	Implement GPS and GSM technology for emergency used (track missing peple). Accuracy of result data (latitude, longitude) is compared via Google Map and mobile phone.	There some delay time due to a network of GSM service and the delay time is about 30 seconds to 1 minute for a particular location.
Asaad et al. [32]	Algorithm of Differential GPS (DGPS) is applied to determine the position location based Reference station (RS) and Mobile Station (MS) using GPS, GSM and SMS service.	Based on the experiment conducted, DGPS technique reduced the average of horizontal and vertical error with 81% and 91% achievement for horizontal and vertical error.
Yong He and Haiyong et.al [33][34][35]	Improved accuracy of GPS measurement using two point relative location measurement.	Reduced measurement error in measurement and hence proved to enhance the measuring accuracy.

2.6 Chapter Summary

As an overview from the literature review proposed by previous researchers, background study of Global Positioning System based Internet of Things had been studied. Method to conduct the related projects and the specification of each microcontroller are further discussed. Method to conduct performance analysis based GPS application with Microcontroller units from previous research are discuss briefly. Besides, ability of GPS to track location with speed, distance and time are proved and presented respectively.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will discussed in detail the methodology performance of IoT development boards based GPS application. Flowcharts, implementation and installation of Hardware and Software part, troubleshooting process, parameters measured and project management will be covered in this project.

3.2 Flowcharts

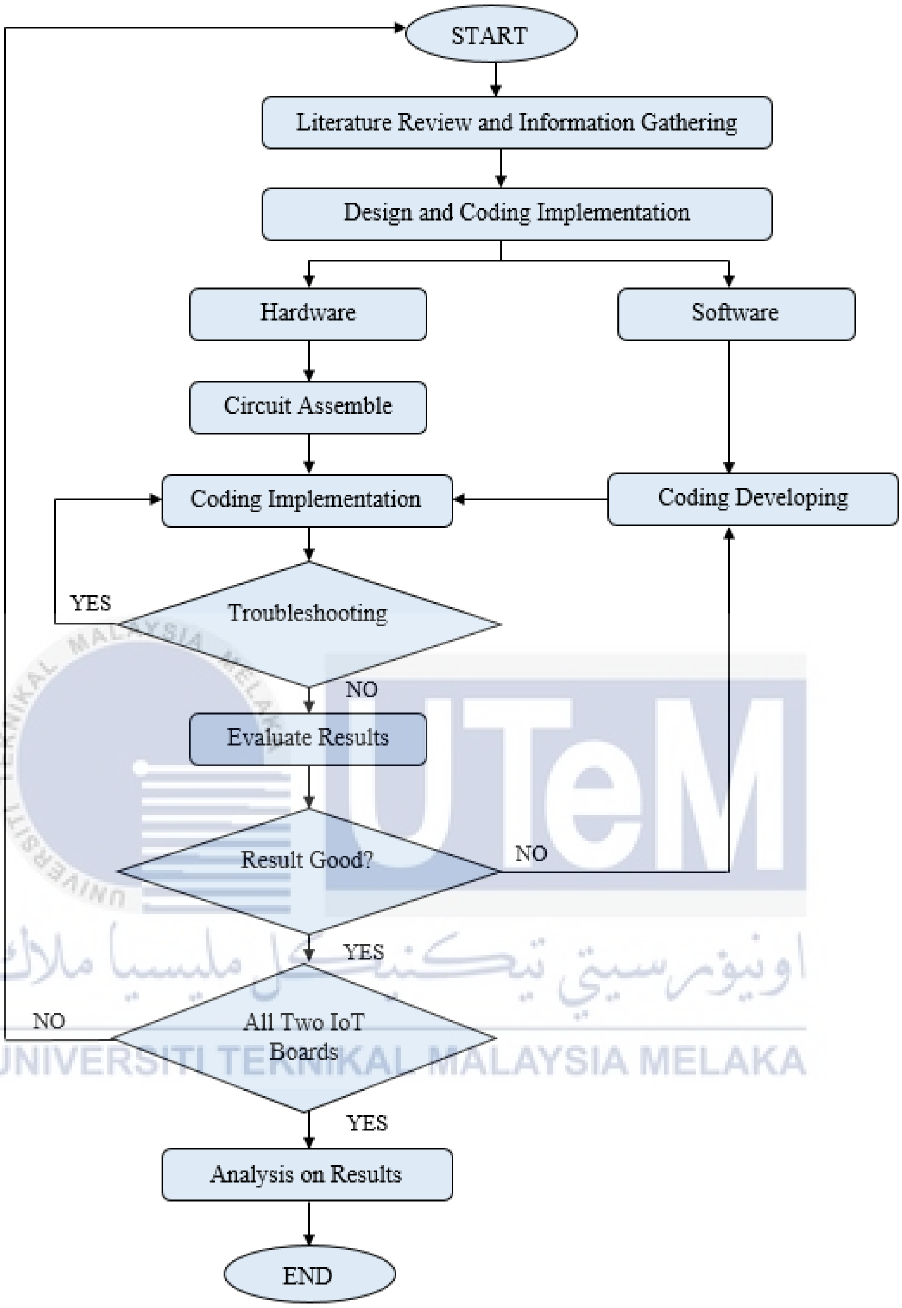

Figure 3.1: Overall flowchart

There are two main part covered in this project which are hardware and software part. Figure 3.1 above show the overall flow chart in this project. Coding implementation is executed and troubleshoot along with hardware IoT development boards based GPS module. Hence, system application is built and run on hardware and software part. Moreover, result data from GPS receiver will stored in cloud Thing speak as an indicator references for further research. After implement and troubleshoot process of software and hardware part, the result will be compared to Google Maps to analyze the precision accuracy and error. Then, final result will be analyzed and discussed in this final project report.

3.3 Overall Project Block Diagram

GPS Module GYNEO6MV2 sensor are used as an input to receive the GPS data from satellite. Research based on implementation of two different IoT development boards with GPS module are compared and analyzed in compliance with the parameter measured in this project. Furthermore, data information obtained is stored in real time Thing Speak cloud database as a references indicator. Then, the results obtained will be analyzed by comparing the result with Google Maps Web. Figure 3.2 below show block diagram in this project.

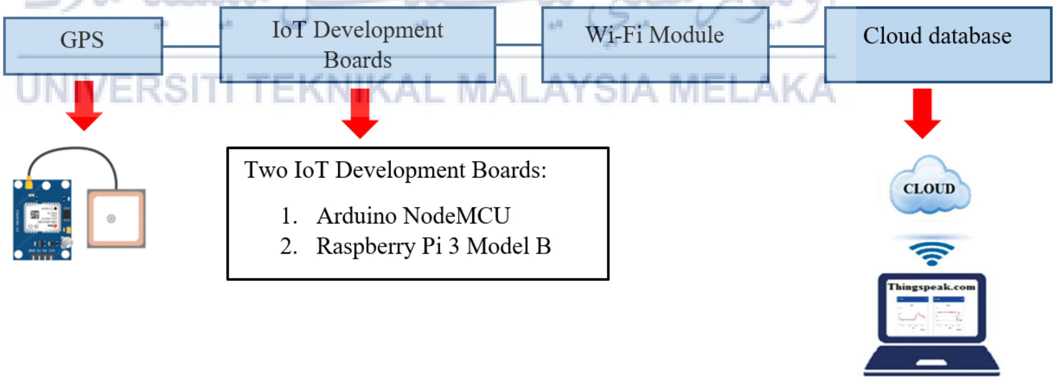

Figure 3.2 Block Diagram

3.4 Hardware Implementation Requirement

Table 3.1 below show the Hardware required to implement the project with Software part.

Table 3.1 Overall lists of hardware tools

No.	Item	Description
1.	NodeMCU	Implement with GPS sensor.
2.	Raspberry Pi 3 Model B	Implement with GPS sensor.
3.	GPS-GYNEO6MV2	Receive GPS signal.
4.	USB Cable	Supply the power and transfer data between laptop and IoT boards.
5.	Ethernet Cable	Connect Raspberry Pi with Laptop to enable the display show in laptop.
6.	Female to Female /Male to Female Jumper Wire	Allow electric current to pass through the circuit between IoT boards and sensor.
7.	SD card	Install Raspbian operating system to function Raspberry Pi.
8.	Micro SD card reader USB 2.0	Enable laptop to read, copy and process the data from SD card.
9.	Laptop	Process and show the results from Iot boards and sensor.
10.	Mobile Phone	Supply Wifi data.

3.4.1 GPS Module GYNEO6MV2

Figure 3.3: GPS Module GY-NEO6MV2

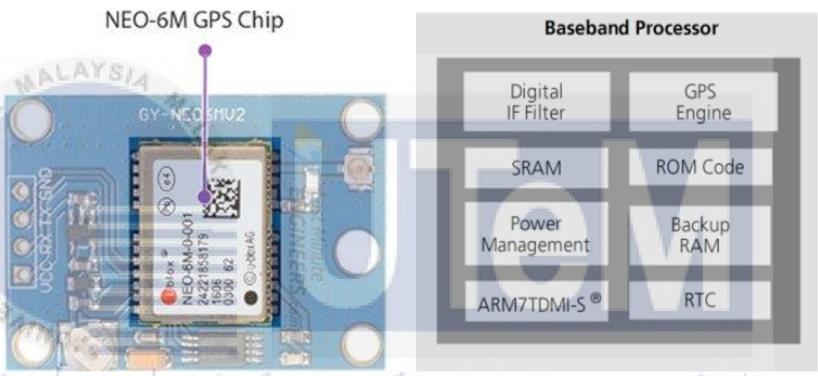

Figure 3.4: Block Diagram of U-Blox NEO-6M GPS Module [22][23]

GPS module GYNEO6MV2 is work perfectly with microcontroller include of Arduino and Raspberry Pi those consists 5V operating voltage as the GPS module operate within 3.3V to 5V. This powerful GPS receiver is based on U-blox Neo-6M GPS module with the build-in EEPROM, on-board battery and external ceramic antenna for better signal reception yet it is low cost efficiency. Communication interface in this GPS module is UART TTL with 9600bps. In addition, there are four pin-out interface which comprise of VCC, RX those receive data in, TX which transmit data out and GND as a logic and power ground. Therefore, GPS NEO6MV2 shown in Figure 3.3 and Figure 3.4 above is used to implement with microcontrollers which include NodeMCU and Raspberry Pi 3.

3.4.2 Node MCU

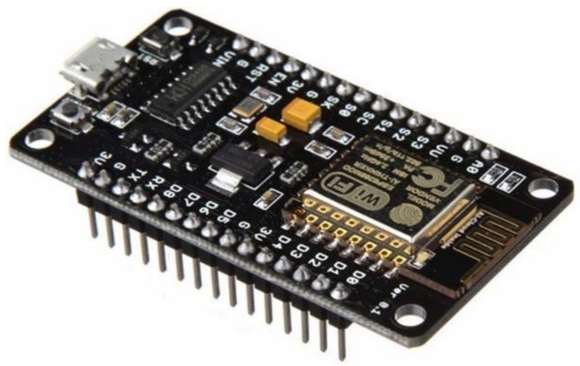

Figure 3.5: Arduino NodeMCU

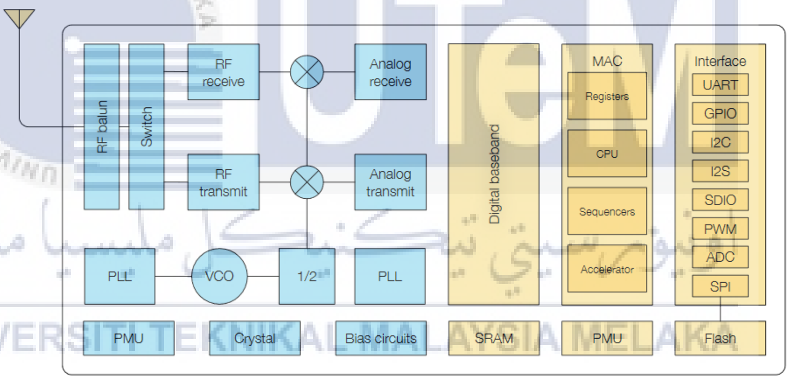

Figure 3.6: Architecture Arduino NodeMCU [16]

A highly programmable System on Chip (SoC), NodeMCU ESP-12E microcontroller is utilized as an IoT device board with GPS sensor in this project. It is a stand-alone 32-bit low power CPU, ROM and RAM microcontroller operate at 3.3V embedded with a Wi-Fi transceiver which allow us to programed and build an IoT project with Arduino IDE (C\C++). Moreover, it contributes 2.4GHz Wi-Fi, 13 GPIO, Inter-Integrated Circuit (I^2C), 10-bit ADC, Serial Peripheral Interface (SPI), UART and pulse-width modulation (PWM) as shown in Figure 3.5 and Figure 3.6 above.

3.4.3 Raspberry Pi 3 Model B

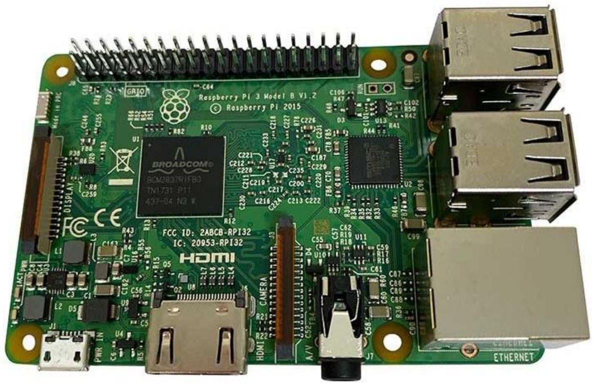

Figure 3.7: Raspberry pi 3 model B

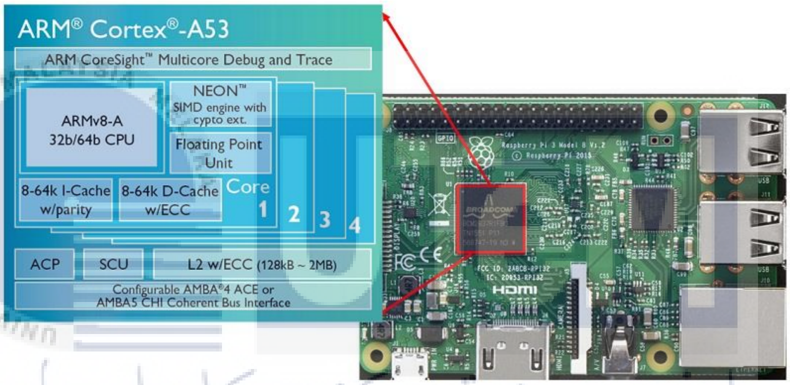

Figure 3.8: Architecture ARM Cortex-A53 [18]

In this project, Raspberry Pi 3 Model B is utilized as it can be described as a credit-card sized computer which run on Linux that we used to explore the Internet of Things (IoT) and enhance programming skills. Raspberry Pi is different from other IoT development boards as it can plug in to computer monitor. Therefore, program and working principle of computer are able to learn and understand in this project. Unlike Arduino which run only on single program at a time, Raspberry Pi able to run on multiple programs at a time. Main chip of BCM2837 System on a Chip (SoC) embedded in Raspberry Pi 3 Model B which running at 1.2GHz based quad-core ARM Cortex-A53 processor are effectively capable when implement with GPS module.

Furthermore, Raspberry Pi 3 model B composed of ARM CPU or GPU, GPIO Pins, USB port and HDMI connector, SD card slot, Internet, Ethernet and Bluetooth function as shown in Figure 3.7 and Figure 3.8 above.

3.5 Operating System and Software Installation Requirement

To implement the system, operating system play an important role in computer hardware and user as a user-friendly interface (GUI). Coding is developed using operating system and software installation to code the program into microcontrollers to perform the function well. Consequently, troubleshooting process will carried out while the errors occurs from programming code. Once the coding debug successfully, the results will display on the terminal based on each software installed. Hence, coding development and implementation process among two microcontrollers with GPS sensor is repeated until to get the best performance result or else repeating the same process until two microcontroller done perfectly with zero error. Table 3.2 below show the Software tools required to implement the project with hardware part.

Table 3.2 Overall lists of Software tools

No.	Item	Description
1.	Arduino IDE	As an operating System for Node MCU
2.	C/C++ programming language	Language used (coding) to implement the function in Arduino IDE
3.	Remote Desktop Connection	Access display in laptop remotely from Raspberry Pi 3
4.	Raspbian	Operating system used in Raspberry Pi 3
5.	Advanced IP scanner	Scan the IP address of Raspberry Pi 3
6.	Python	Programming language used to implement Raspberry Pi 3 with GPS sensor in Terminal

3.5.1 Arduino Software

Figure 3.9: Arduino Software

Open-source Arduino Software (IDE) is the platform to write a code and upload to the microcontroller NodeMCU board as shown in Figure 3.9 above. Programming code C/C++ is run and compile in this software. Menu bar to activate the Arduino IDE software is shown in Figure 3.10 below.

Figure 3.10: Menu Bar in Arduino IDE

Once connect the NodeMCU to computer using USB cable, NodeMCU 1.0 (ESP-12E Module) in Arduino software is selected. Then, Installed “ESP8266 by ESP8266 community” into Arduino IDE. Tiny GPS library and ThingSpeak library is also download and installed in Arduino IDE. Therefore, the code is uploaded to NodeMCU board once the port is activated from tools. Method to execute the code and accomplish the task are shown in Figure 3.11 below.

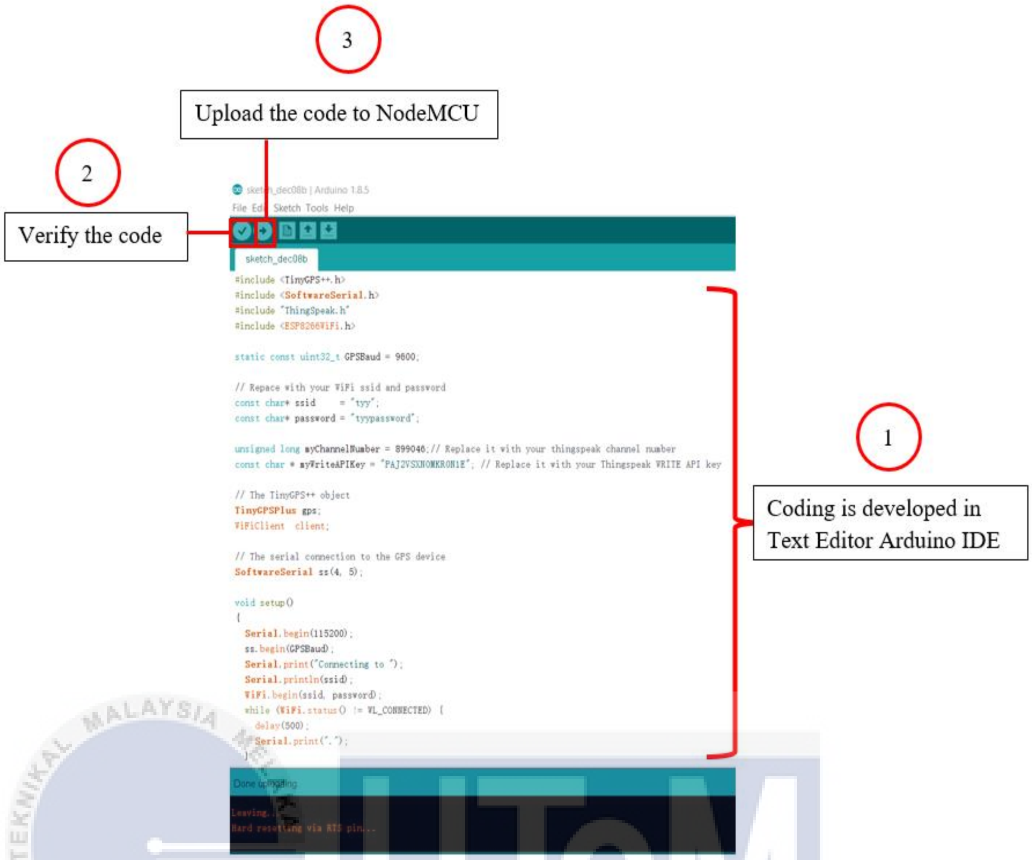

Figure 3.11: Step to execute the code in Arduino IDE

3.5.2 Python

Figure 3.12: Python Software

The powerful high-level programming language, Python is start implement by Guido Van Rossum in 1989. Python programming language mostly used by computer and software engineer, developer and researcher to build web applications, graphic user interface (GUI) based applications, Gamming applications, business applications and etc. The open source-code python language able to run on different platform which include of Windows, Linux, Unix, Macintosh and etc. Moreover, various version of python is arises while the most commonly used version is Python 2 and Python 3. Python language is understandable and readable as Standard English keywords is applied and is suitable for beginner’s learner. As a result, programming language python shown in Figure 3.12 above which support C, C++, JAVA and etc is used and implement with Raspberry Pi 3 in this project.

3.5.2.1 Set-up Raspberry Pi 3 Display in Laptop

The low cost Raspberry Pi 3 is different from other IoT development boards as Raspberry Pi 3 is same as the smaller pocket size computer that able to plug into TV or computer monitor. To setup the display Raspberry pi 3 Model B to laptop, materials and tools required includes of SD card, Ethernet cable and USB cable. Operating system of Raspbian is installed within the SD card before connecting the Raspberry pi to laptop. To show Raspberry pi 3 display in Remote Desktop Connection, Ethernet cable is connected directly among the Raspberry pi 3 with Laptop once the power is supply to Raspberry pi 3 by connecting the USB cable with Laptop. Figure 3.13 below show the Setup of Raspberry Pi 3.

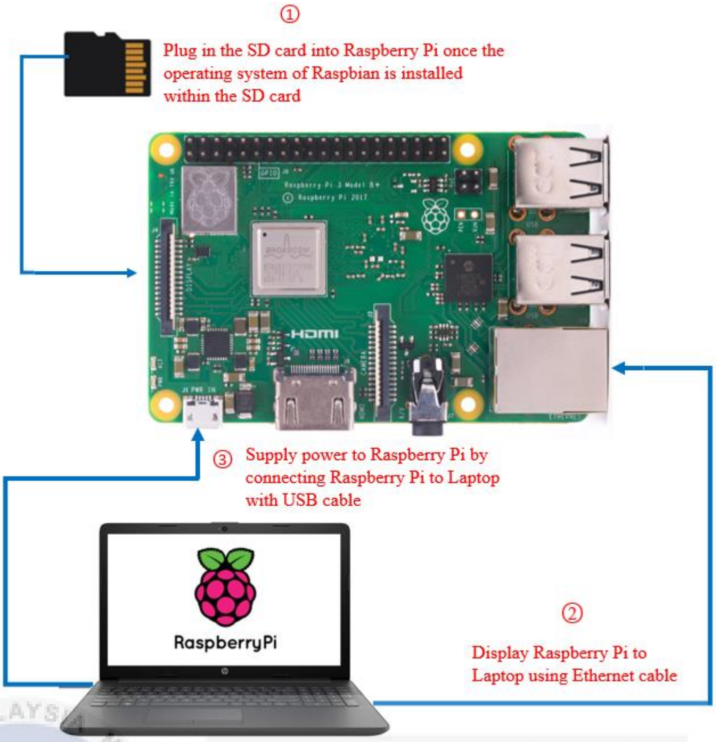

Figure 3.13: Set up of Raspberry Pi 3 Model B

In order to remote the display from Raspberry Pi to Laptop, Raspberry Pi 3 Model B is connected over Remote Desktop Connection application on Window operating system using Ethernet and USB cable. When the local IP address of Raspberry Pi 3 Model B is found by IP scanner devices, then enter raspberry Pi's local IP address in **Computer** and click the **Connect** button. Then, followed by enter **username** and **password** to login to raspberry pi. Therefore, implement coding between GPS module with Raspberry Pi 3 Model B can be troubleshoot by using Python code. Figure 3.14, Figure 3.15 and Figure 3.16 below show the Remote Desktop Connection application for laptop display.

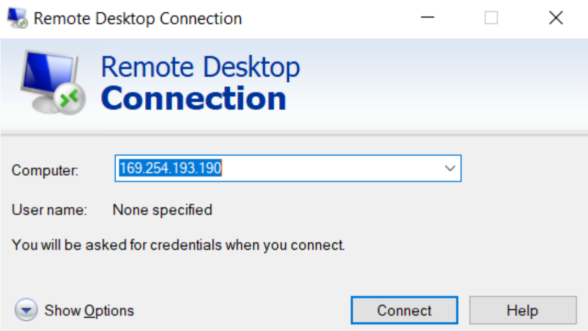

Figure 3.14: Connect to the Raspberry Pi over Remote Desktop Protocol

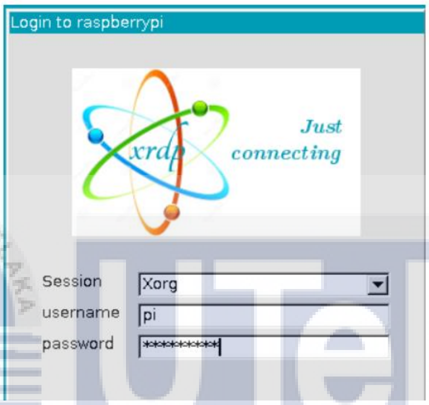

Figure 3.15: Login to Raspberry pi by entering the username and password

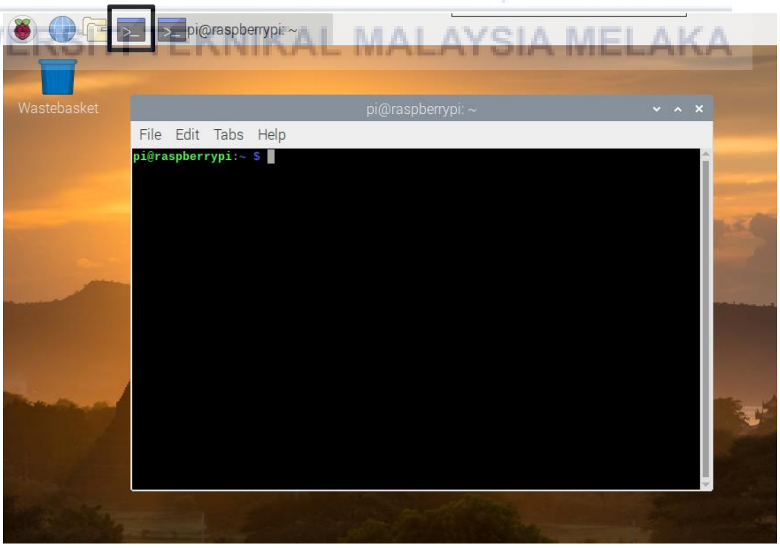

Figure 3.16: Implement Python code using terminal in Raspberry Pi Desktop

3.5.3 ThingSpeak Cloud

Figure 3.17: ThingSpeak Software

To store the received GPS data, an open cloud data IoT platform, ThingSpeak as shown in Figure 3.17 above is used in this project. It is an efficient platform capable to analyse and visualize the data obtained from GPS sensor in real time to carry out further research. First of all, sign up an account for ThingSpeak and the channel is created as shown in Figure 3.18, Figure 3.19 and Figure 3.20 below.

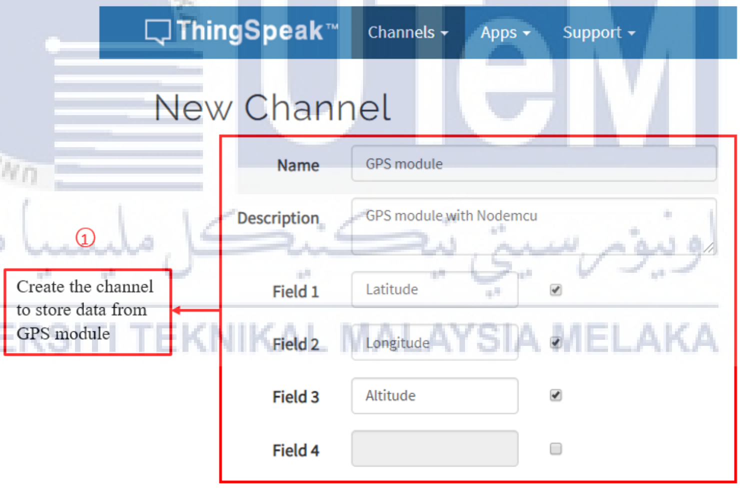

Figure 3.18: Create the ThingSpeak channel

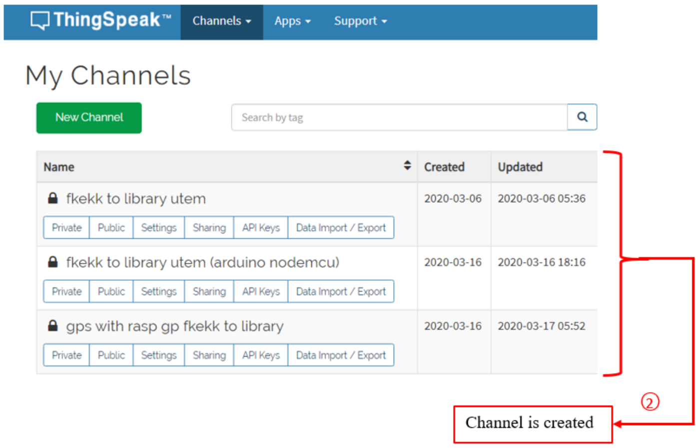

Figure 3.19: Channel is created

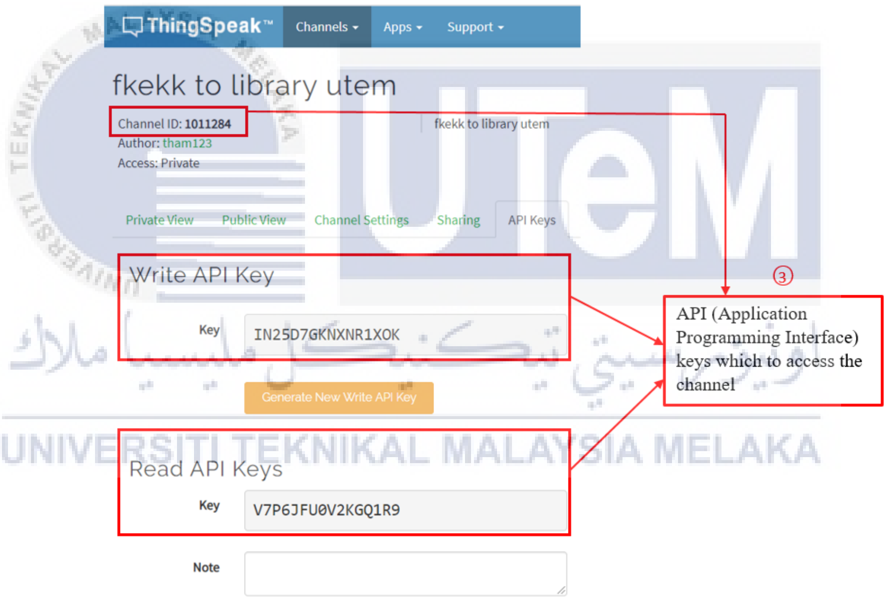

Figure3.20: Channel ID and API keys

3.5.4 Google Map (as a reference benchmark)

Figure 3.21: Google Maps Application

A user friendly mapping application known as Google Map used by most citizens for the ability to locate the places around the world. A high resolution images with an exact coordinates positioning taken from satellites, aerial shots or street view shots reveals the efficiency of Google Maps. Hence, in this project, Google Maps application as shown in Figure 3.21 above is used as a benchmark for searching and comparing the precise locations with GPS module which act as an indicator for further research.

3.6 Hardware Connections

Once the software is installed successfully, hardware connection between GPS module sensor and IoT development boards are set up immediately based on block diagram and circuit diagram as shown in Figure 3.22, Figure 3.23, Figure 3.24 and Figure 3.25 below.

3.6.1 Node MCU with GPS sensor

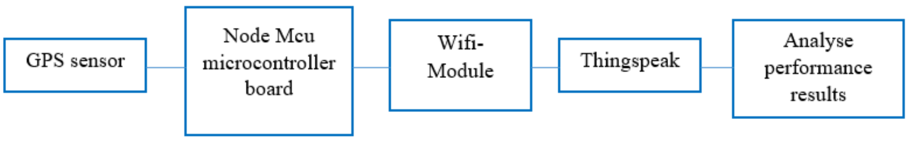

Figure 3.22: Block diagram NodeMCU with GPS sensor

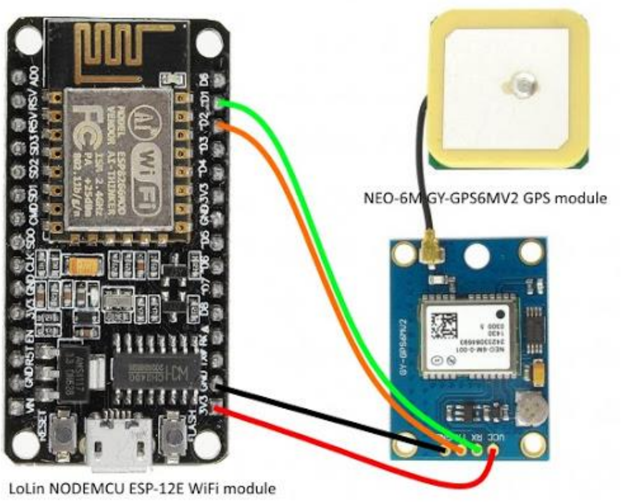

Figure 3.23: Circuit diagram NodeMCU with NEO-6MGY-GPS6MV2

3.6.2 Raspberry Pi 3 Model B with GPS sensor

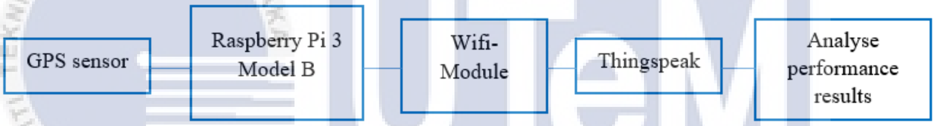

Figure 3.24: Block diagram Raspberry Pi 3 Model B with GPS sensor

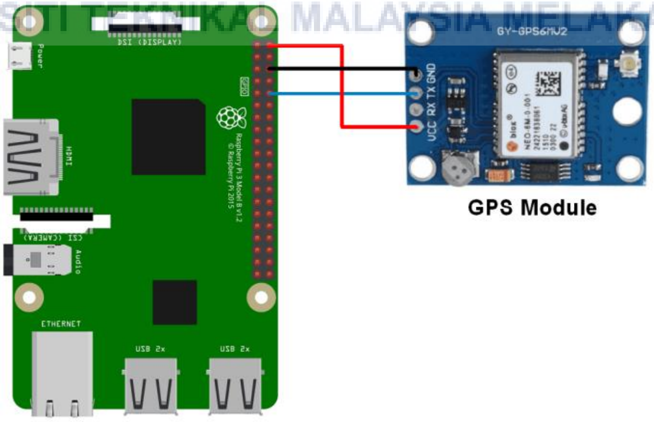

Figure 3.25: Circuit diagram NodeMCU with NEO-6MGY-GPS6MV2

3.7 Troubleshoot and Analyze

Troubleshoot process is important to carry out in this project to ensure to obtain the good results. Therefore, troubleshoot process among coding development and hardware implement is carried out to verify the results performance. This project is proceed with two different IoT development boards which are Node MCU and Raspberry Pi 3 Model B with GPS sensor, GY-NEO6MV2. Data information obtained based locations latitude, longitude, alttitude, speed, distance and duration time are compared with Google Maps Website. Hence, final results will be analyze and discussed as shown in Figure 3.26 below.

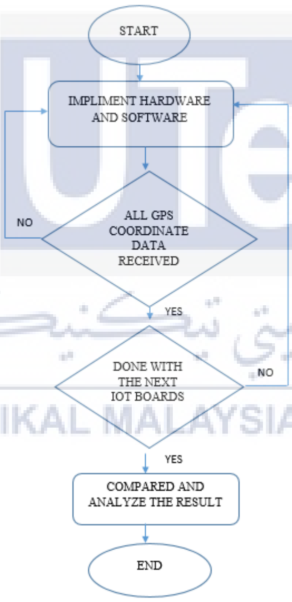

Figure 3.26: Troubleshoot process flow chart

3.8 Project Management

Overall budget cost of this project is reasonable and affordable. All the materials, tools and IoT development boards are available from supervisor and FKEKK UTeM except the GPS sensor module. Hence, the overall budget in this project cost about RM46.50 as shown in Table 3.3 below.

Table 3.3 show the overall budget cost in this project

No	Apparatus	Quantity	Price (RM)	Cost (RM)
1	Raspberry Pi 3 Model B	1	RM 165.00	Free (available in FKEKK store)
2	Arduino NodeMcu ESP8266	1	RM 27.50	Free (available in FKEKK store)
3	GPS module(GY-NEO6MV2)	1	RM 46.50	RM 46.50
		Total	RM 239.00	**RM 46.50**

3.9 Project Significance

As we know that everyone of us depend mostly on GPS to locate the position locations all around the world. GPS application is useful and efficient as position locations can be track based latitude, longitude, altitude, speed, distance and time based satellite system. Since, different kind of IoT development boards arise in the market due to on-board specification features as processor, plug-in peripherals and their operating system architecture. Hence, the performance of IoT development boards for GPS application carried out in this project able to verify the different precision and error which become a reference for further research. Besides, low cost of this project compared to the complex and expensive GPS technology is sustainability and environmental friendly. The GPS sensor integrated with IoT boards is sufficient to apply in almost all industry as well as to protect human's health and security by tracking their locations.

3.10 Chapter Summary

In conclusion, methodology to conduct this project is discussed briefly in this chapter. GY-NEO6MV2 implement with NodeMCU and Raspberry Pi 3 Model B microcontroller board are coded using Arduino IDE and OS Raspbian Software. Every task and procedure must be well organized in order to achieve the aim of this project.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter will discuss and compare all the results obtained from the experiment. The experiment conducted in this project is to observe and compare the performance of two different IoT development boards for GPS applications. GPS GY-NEO6MV2 is used to obtained the results when implement with NodeMCU and Raspberry Pi 3 Model B based on location selected from start point to destination point. Moreover, all the collected data for each IoT development boards is updated and stored in the Thing Speak cloud for further research and analyze. Hence, data obtained is compared to actual location shown in Google Map App. Research on the performance analysis based accuracy and precision locations for GPS application from each IoT boards will be discussed in table and bar graph form.

4.1 Experiment set-up for GPS sensor with IoT development boards

4.1.1 GPS GY-NEO6MV2 with NodeMCU

In this section, a module of 3.3V GPS NEO6MV2 with 9600 baud rate is integrated with NodeMCU as shown in Figure 4.1 shown below. In order to receive the GPS signal from the receiver, GPS sensor need a hot start and wait for some time for GPS sensor to fix the signal for a particular location until GPS start blinking as shown in Figure 4.2 below. The blue LED from GPS module started blinking when the GPS module sensor receive the signal by external antenna connected by the module. This showed that the GPS signal is fixed and received the location GPS signal. Once the GPS signal is received, GPS data (Latitude, Longitude, Altitude, Date and Time) are display out by each IoT board's terminal. At the same time, all the GPS data are updated in Thing Speak cloud at real time for further analysis.

Figure 4.1: Set up of hardware connection circuit for GPS GY-NEO6MV2 with NodeMCU

Figure 4.2: Blue LED from GPS sensor module started blinking

Programming Arduino with C/C++ is used for NodeMCU. The reading of GPS signal which obtained from the GPS sensor is shown in serial monitor as shown in Figure 4.3 below. At the same time, the reading of GPS data is updated to Thing speak cloud for further analyzed.

Figure 4.3: Data received shown in Serial Monitor in Arduino IDE Software

4.1.2 GPS GY-NEO6MV2 with Raspberry Pi 3 Model B

Once the Raspbian operating system (OS) is installed, IP address is obtained and able connected to Remote Desktop, hardware connection circuit of Microcontroller Raspberry Pi 3 Model B with GPS module is set up for the purpose of communicate with GPS sensor module via UART as shown in Figure 4.4 and Figure 4.5 below.

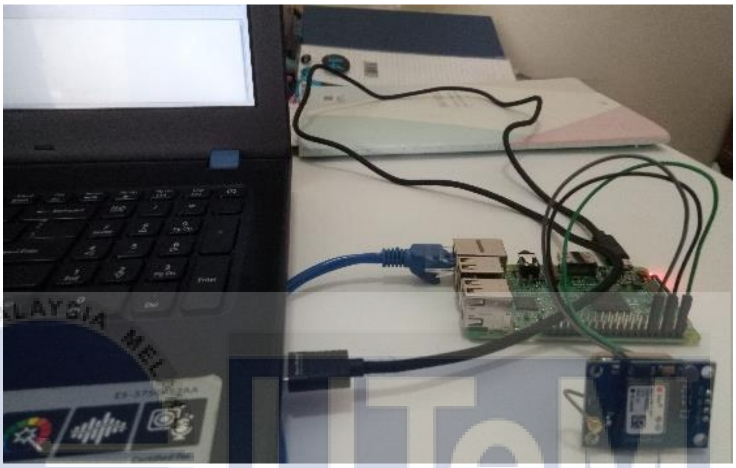

Figure 4.4: Set up of hardware connection circuit for GPS GY-NEO6MV2 with Raspberry Pi 3 Model B

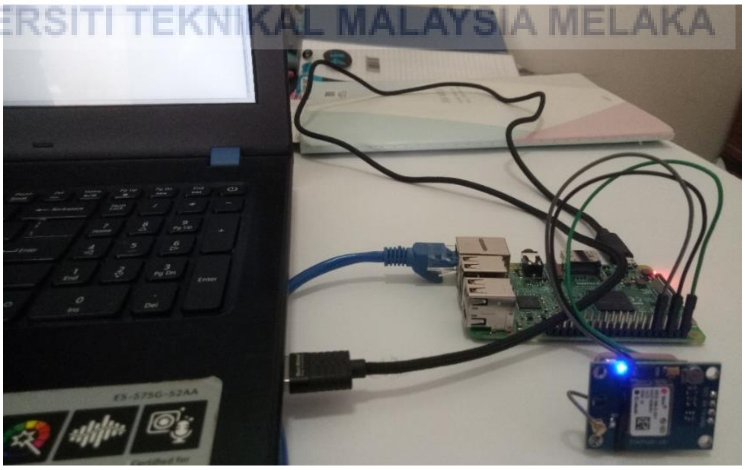

Figure 4.5: Blue LED from GPS sensor module started blinking

Programming Python is used to implement with Raspberry Pi 3 and GPS sensor. Therefore, reading of GPS signal which obtained from the GPS sensor is shown in terminal as shown in Figure 4.6 below. At the same time, the reading of GPS data is display to Thingspeak cloud for further analyzed.

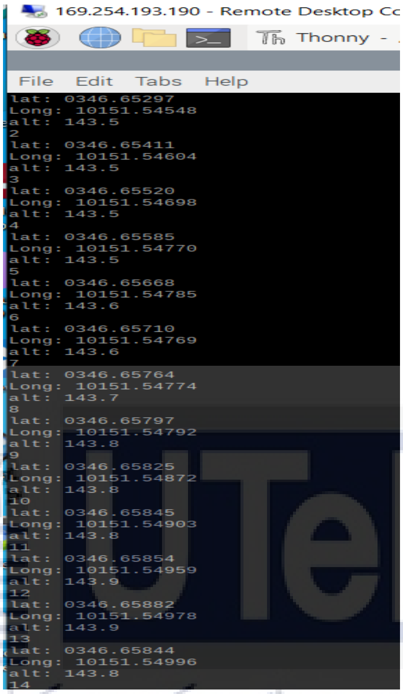

Figure 4.6: Data received shown in Terminal in Raspberry Pi Software using Remote Desktop Connection

4.2 Results of GPS data display in Thing Speak Cloud

4.2.1 Reading of GPS data from NodeMCU

The GPS module sensor is started with a hot start for about 20 minutes for GPS receiver to fix the data in order to obtain the data values. First of all, data is received from the location, Taman Raub Jaya 2 as a starting point. Figure 4.7 below show the value of latitude received from GPS receiver. Based on the reading shown below, value of Latitude is stable which show 3.777787 at 16:25pm.

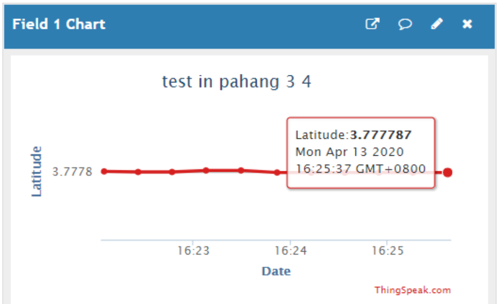

Figure 4.7: Latitude show 3.777787 at 16:25p.m in Thingspeak

Figure 4.8 below show the value of longitude received from GPS receiver. Longitude value obtained is unstable and there is an abnormal change at the beginning. Then, the longitude reading become stable at 101.859096 at 16:25pm.

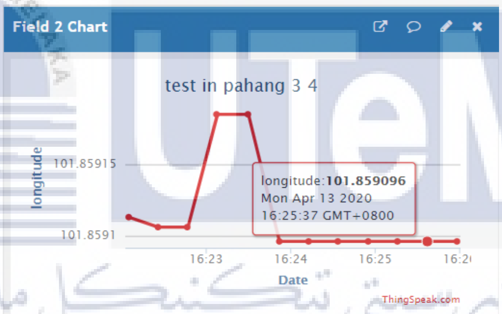

Figure 4.8: Longitude show 101.859096 at 16:25pm in Thingspeak

The altitude value is obtained at the stable value of 137.8 meters at 16:25pm as shown in figure 4.9 below.

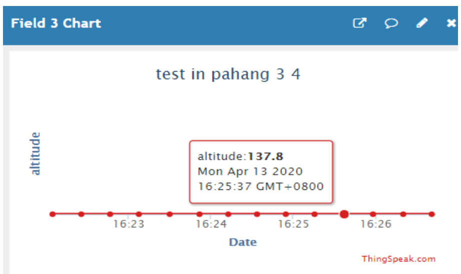

Figure 4.9: Altitude show 137.8 meters at 16:25pm in Thingspeak

Data is continued to be received until the destination location is reached by driving a car along with GPS module sensor and NodeMCU. As a result, data is received when reached in KFC, Raub. Figure 4.10 below show the reading of latitude 3.789112 when reached the location at 16:28pm. Based on the reading obtained, value is increased gradually from the starting point 16:25pm to 16: 28pm. The whole duration had taken 3 minute to reach KFC, Raub from the starting point.

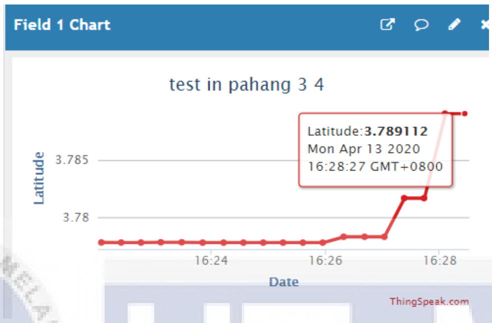

Figure 4.10: Latitude show 3.789112 at 16: 28pm in Thing speak

Figure 4.11 below show the longitude value 101.858064 at 16:28pm. There is an unstable changed value shown in thing speak graph when the destination location is reached.

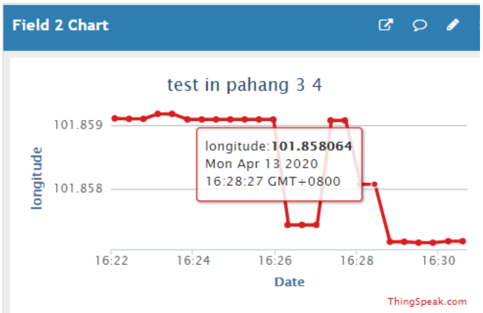

Figure 4.11: Longitude show 101.858064 at 16: 28pm in Thingspeak

However, the reading of altitude obtained remained constant at 137.8 meters as shown in Figure 4.12 below.

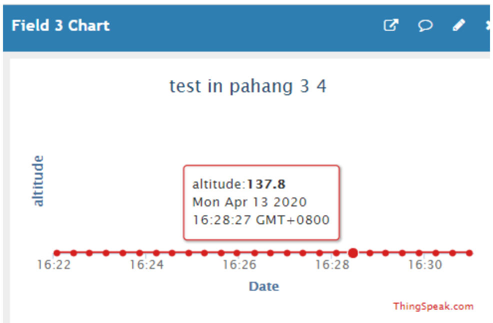

Figure 4.12: Altitude show 137.8 meters at 16: 28pm in Thingspeak

4.2.2 Reading of GPS data from Raspberry Pi 3 Model B

The GPS module sensor is started with a hot start for about 20 minutes for GPS receiver to fix the data in order to obtain the data values. First of all, data is received from the location, Taman Raub Jaya 2 as a starting point. Figure 4.13 below show the value of latitude received from GPS receiver. Based on the reading shown below, value of Latitude is gradually increased which show 346.65771 at 14:12pm.

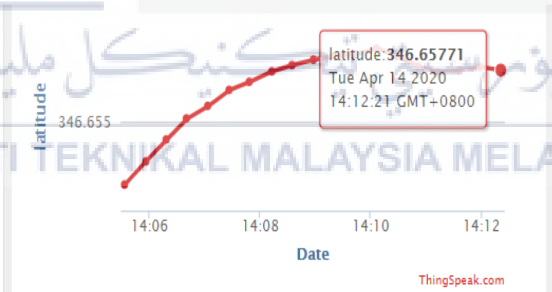

Figure 4.13: Latitude show 346.65771 at 14:12 pm in Thing speak

Figure 4.14 below show the value of longitude received from GPS receiver. Longitude value obtained in Thing speak graph is unstable and there is an abnormal changed. Then, the longitude reading is obtained with 10151.54932 at 14:12pm.

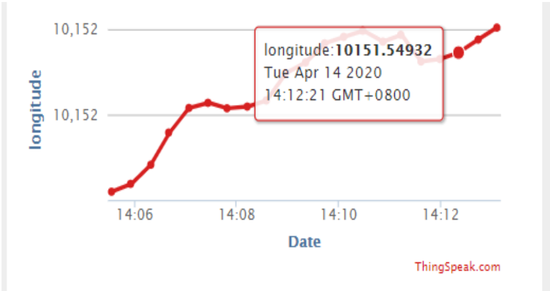

Figure 4.14: Longitude show 10151.54932 at 14:12pm in Thing speak

The altitude value obtained from thing speak graph show in Figure 4.15 below is 143.4 meters at 14:12pm. Based on the graph obtained, the reading show unstable changed compared to altitude reading obtained from microcontroller NodeMCU.

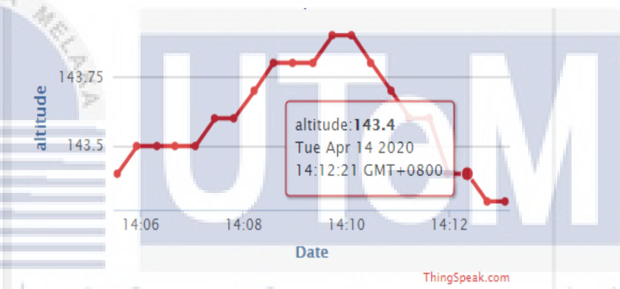

Figure 4.15: Altitude show 143.4 meters at 14:12pm in Thingspeak

Data is continued to be received until the destination location is reached by driving a car along with GPS module sensor and Raspberry Pi 3. As a result, data is received when reached in KFC, Raub. Figure 4.16 below show the reading of latitude 346.66069 when reached the location at 14:15pm. Based on the reading obtained, value is increased gradually from the starting point 14:12pm to 14: 15pm. The whole duration had taken 3 minute to reach KFC, Raub from the starting point.

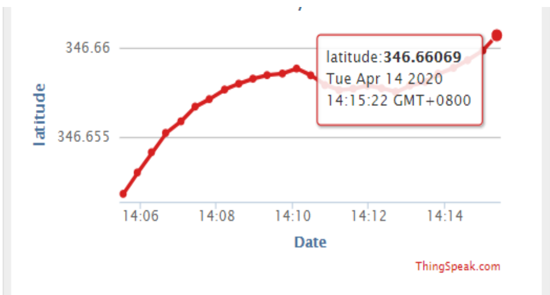

Figure 4.16: Latitude show 346.66069 at 14:15pm in Thing speak

Figure 4.17 below show the longitude value 10151.55174 at 14:15pm. There is an unstable changed value shown in thing speak graph when the destination location is reached.

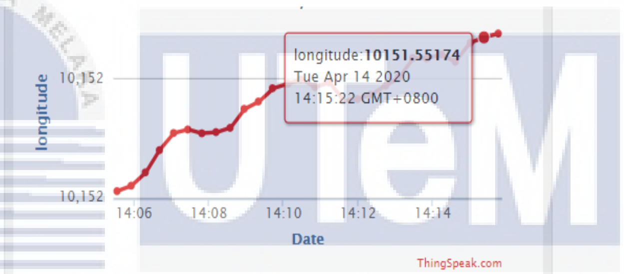

Figure 4.17: Longitude show 10151.55174 at 14:15pm in Thing speak

However, the reading of altitude obtained is remained unstable at 143.5 meters as shown in Figure 4.18 below.

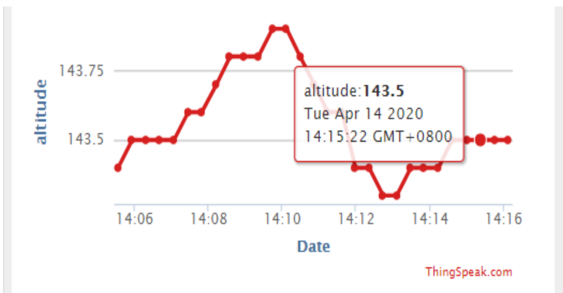

Figure 4.18: Altitude show 143.5 meters at 14:15p.m in Thing speak

4.3 Actual GPS location show in Google Map

As we know today, Google Map provides a better features for the user especially the street view image which allow us to define the position place virtually. Moreover, high technology of this virtual tool give a platform for us to carry out the research with high a demonstration quality which increase the quality of impression. This efficiency and reliable mapping service help us to navigate and locate our location accurately based satellite imagery with a faster travel time [36]. As a result, Google Map App is the suitable choice of a virtual tool to navigate and show the actual location for the particular location in this project to compare the results obtained from GPS sensor.

4.3.1 Location of Taman Raub Jaya 2 show in Google Map

Figure 4.19 Actual location of Taman Raub Jaya 2 from Google Map

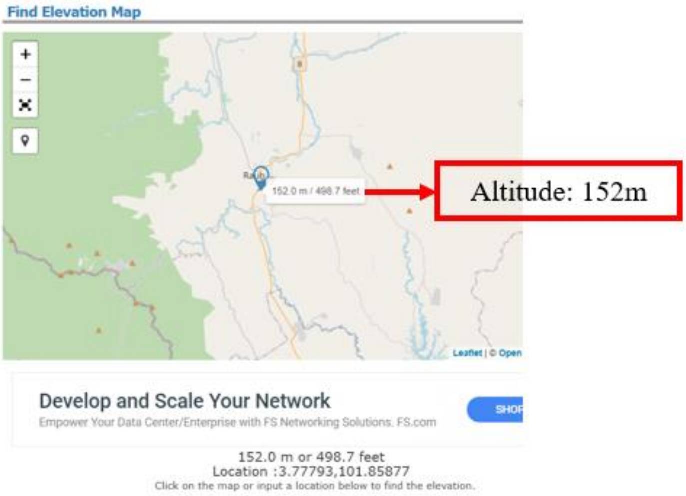

Figure 4.20: Actual altitude show in elevation map

4.3.2 Location of KFC, Raub show in Google Map

Figure 4.21: Actual location of KFC, Raub from Google Map

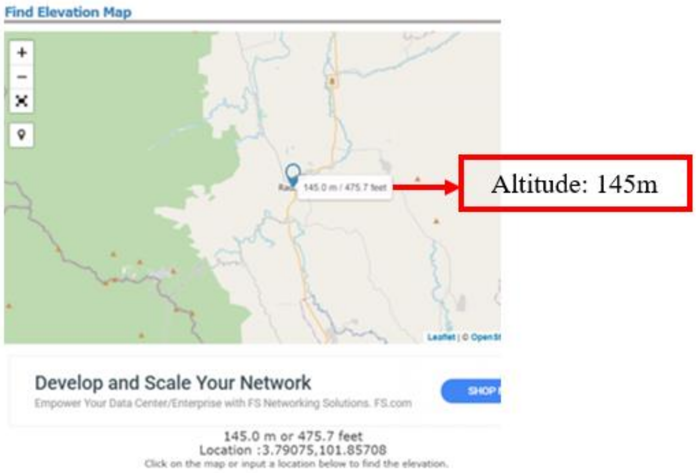

Figure 4.22: Actual altitude show in elevation map

4.3.3 Actual distance from Taman Raub Jaya 2 to KFC, Raub show in Google Map

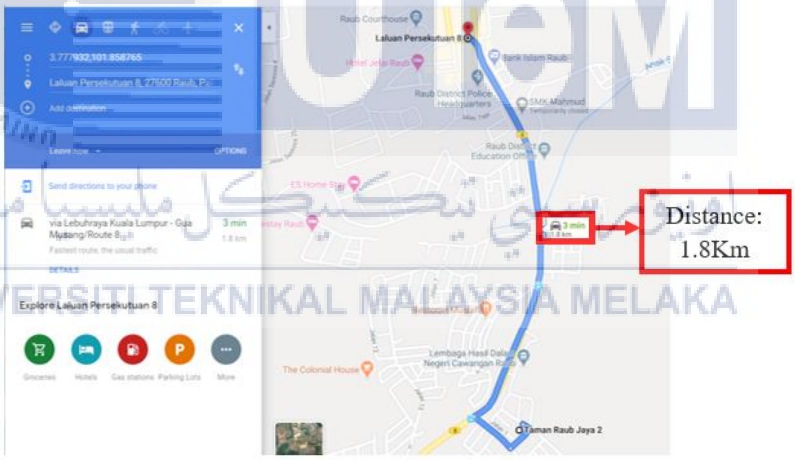

Figure 4.23: Actual distance from Taman Raub Jaya 2 to KFC, Raub in Google Map

4.4 Remarks for Precision and Accuracy Table

In this session, data obtained from each IoT development boards with the GPS sensor is compared and analyzed based on an actual location shown in Google Map App. Research on the performance of accuracy and precision locations for GPS application of each IoT development boards will be presented and discussed in table and bar graph form. Hence, remarks parameter based accuracy locations, distance and speed from each IoT development boards are discussed in this part.

First of all, latitude and longitude collected from GPS sensor which display in Thingspeak cloud based NodeMCU was in decimal format and the results has been analyzed in the Table 4.3 and Table 4.4. Since, latitude and longitude collected displayed on Thingspeak based Raspberry Pi 3 are not in decimal format, the data must first converted to decimal format according to the step shown below in Table 4.1 and Table 4.2 before analyzed in Table 4.5 and Table 4.6 as shown below.

Table 4.1: Data collected from location Taman Raub Jaya 2 based Raspberry Pi 3 convert to decimal format

	NMEA format	Decimal format
Latitude	346.65771	3 + (46.65771/60) = 3.7776285
Longitude	10151.54932	101 + (51.54932/60) = 101.8591553

Table 4.2: Data collected from location KFC, Raub based Raspberry Pi 3 convert to decimal format

	NMEA format	Decimal format
Latitude	346.66069	3 + (46.66069/60) = 3.77768167
Longitude	10151.55174	101 + (51.55174/60) = 101.8591957

Furthermore, algorithm to measure the distance based on Table 4.7 and Table 4.8 below can be defined based on Equation 4.1 and Equation 4.2 below.

$$\text{Distance, D} = \sqrt[2]{A^2 + B^2} * 1609.34 \quad (4.1)$$

$$\text{A} = 69.1 * (\text{lat2} - \text{Lat1})$$

$$\text{B} = 69.1 * (\text{Lon2} - \text{Lon1}) * COS\left(\frac{Lat1}{57.3}\right)$$

D is the distance in meters, 1609.344 is the constant that convert miles to meters while 69.1 and 57.3 are the constants used to convert degree into ground distance in miles. Hence, the speed can be measured based on the equation 4.2 below.

$$Speed = \frac{Distance}{Time} \quad (4.2)$$

All measurement consist of errors and each errors can be analyzed based accuracy and precision. Precision is defined as how the measured value based GPS sensor are close to each other while the accuracy is defined as the closes between the measured value and actual value of the location. Parameter is defined as set of measureable things for the purpose of performance result. To analyze the performance of the results, precision and accuracy of the locations, distance and speed is obtained by Equation 4.3 shown below.

$$Precision\ and\ accuracy(\%) = \left|\frac{actual\ value - experiment\ value}{actual\ value}\right| x100\% \quad (4.3)$$

4.5 Performance Results on IoT development boards

4.5.1 Result of locations based on NodeMCU boards

Table 4.3: Precision and accuracy results from the location, Taman Raub Jaya 2

	Experimental Value	Actual value	Precision and accuracy (%)
Latitude	3.777787	3.777932	99.62
Longitude	101.859096	101.858765	99.97
Altitude	137.8 meters	152.0 meters	90.66
Time	16:25p.m	16:25p.m	None

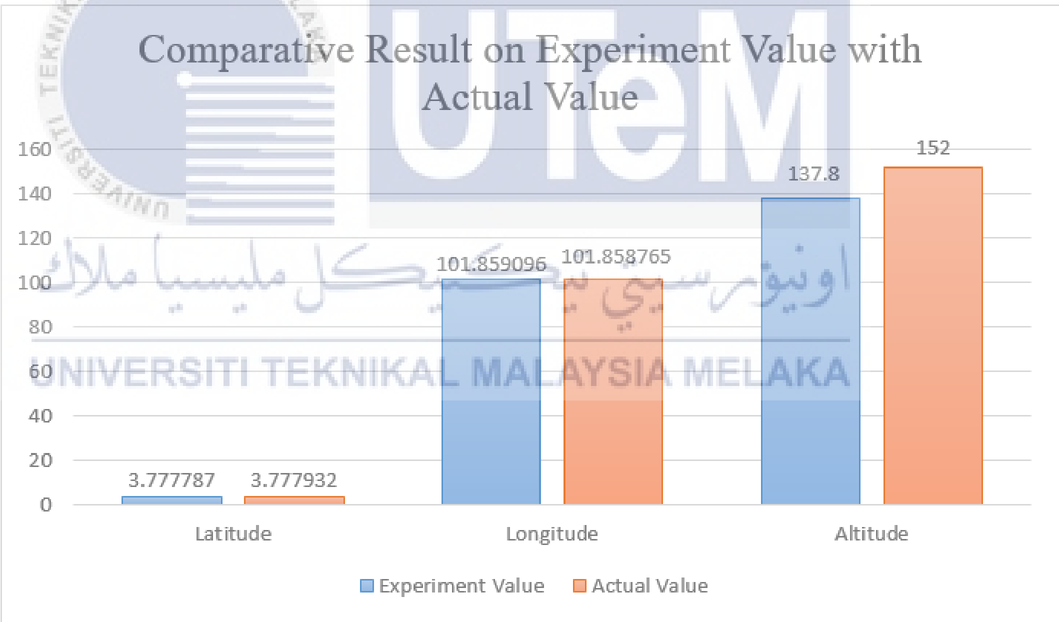

Figure 4.24: Comparative results based Experiment value and Actual value

Based on Table 4.3 and Figure 4.24, all the result based latitude, longitude and altitude are collected from the location of Taman Raub Jaya 2 which based on microcontroller NodeMCU implement with GY-NEO6MV2 GPS sensor. The result based latitude, longitude and altitude from experimental value show slightly different value compared to actual location based Google Maps app. We can observed that measurement errors for latitude and longitude is low compared to altitude. Hence, precision for latitude and longitude is higher than altitude while measurement errors for altitude show the highest which achieving the lowest accuracy of 90.66% based on the result shown in Table 4.3 and histogram plotted in Figure 4.24. The data collected from GPS sensor module need a hot start to receive the signal data and hence the receiver consume a longer period to obtain the data compared to the Google Map Apps. Therefore, period of time to receive the signal caused the changes of result from data collected based on sensor and microcontroller utilized.

Table 4.4: Precision and accuracy results reached in location KFC, Raub

	Experimental Value	Actual value	Precision and accuracy (%)
Latitude	3.789112	3.790755	95.67
Longitude	101.858064	101.857083	99.9
Altitude	137.8 meters	145.0 meters	95.03
Time	16:28p.m	16:28p.m	None

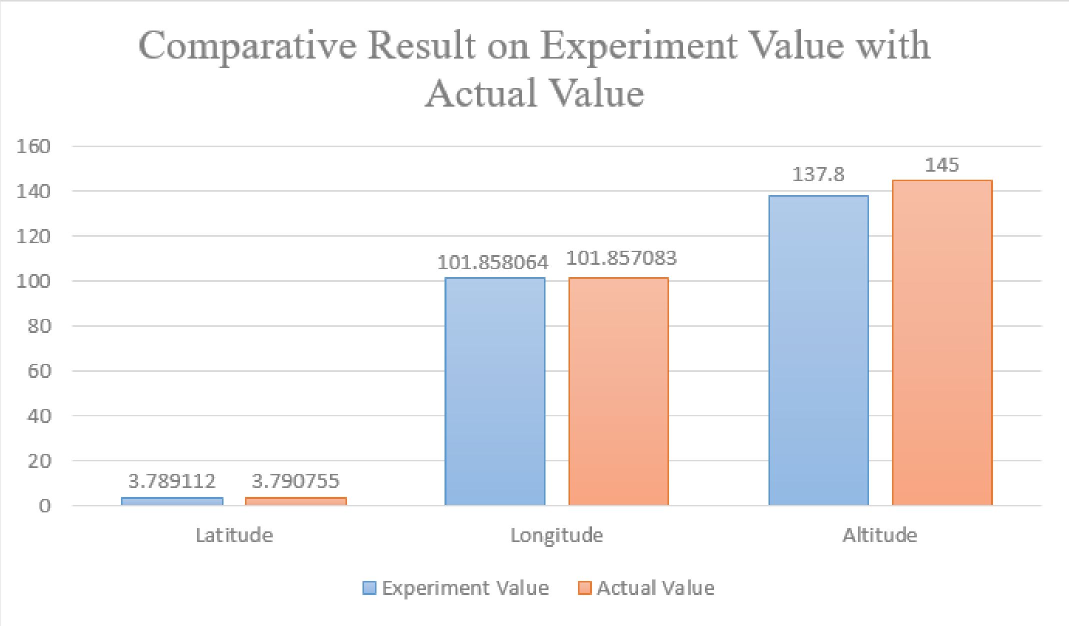

Figure 4.25: Comparative results based Experiment value and Actual value

Based on Table 4.4 and Figure 4.25, all the result based latitude, longitude and altitude are collected from the location KFC in Raub which based on microcontroller NodeMCU implement with GY-NEO6MV2 GPS sensor. The results show the obvious different changes from the actual location based Google Maps app. Measurement errors for latitude and altitude is higher compared to longitude. Hence, accuracy for latitude and altitude each achieved 95.67% and 95.03% which is lower than longitude. Based on the result obtained, we can observed that the results were not good as compared to previous result in Table 4.3. In this part, the data are collected by running the GPS sensor in the car until reached the destination. As a results, when the car are travel along the destination, there were the blockage of the signal where the signal had been absorb or reflect by environment building, houses, humans and cars. This probably cause the delay of time for sensor and the IoT devices to process and received the signal.

4.5.2 Result of locations based on Raspberry Pi 3 Model B

Table 4.5: Precision and accuracy results from the location, Taman Raub Jaya 2

	Experimental Value	Actual value	Precision and accuracy (%)
Latitude	3.7776285	3.777932	99.2
Longitude	101.8591553	101.858765	99.96
Altitude	143.4 meters	152.0 meters	94.64
Time	14:12 p.m	14:12 p.m	None

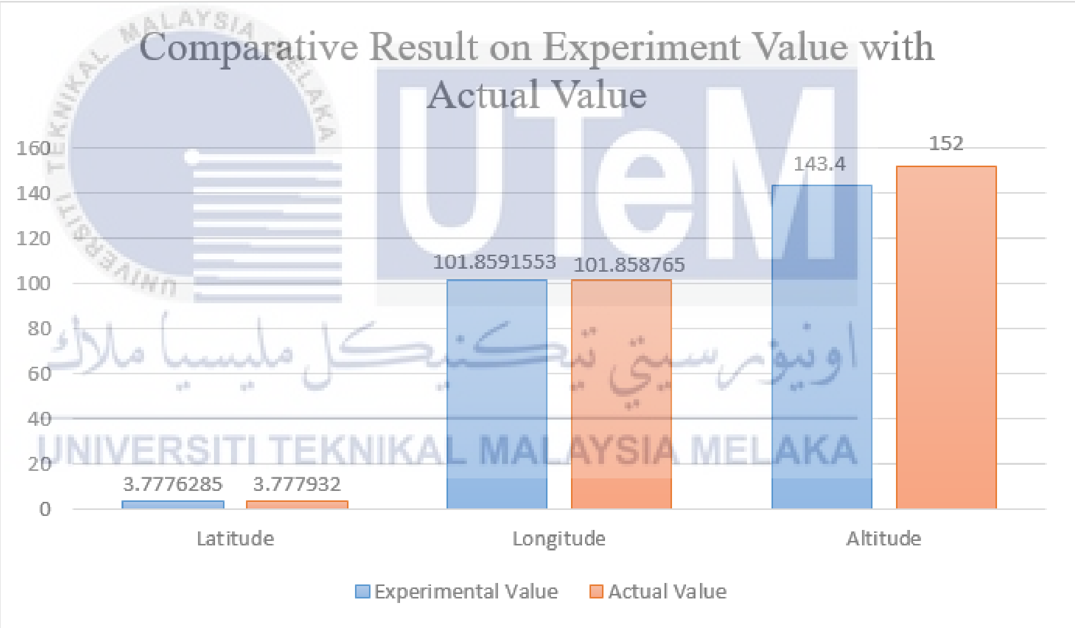

Figure 4.26: Comparative results based Experiment value and Actual value

Based on Table 4.5 and Figure 4.26, all the result based latitude, longitude and altitude are collected from the location of Taman Raub Jaya 2 which based on microcontroller Raspberry Pi 3 Model B implement with GY-NEO6MV2 GPS sensor. According to the result obtained, the performance of the results show the good result and there were not much changes in measurement errors. This may due to the great signal received by the sensor at the static starting point when the car are not moving. Therefore, devices of Raspberry Pi 3 which have a better function of processor able to receive and display the data value efficiently. So, the results collected have higher accuracy compared to previous result in Table 4.3 and Table 4.4. Each accuracy achieved by latitude, longitude and altitude by using raspberry pi 3 are 99.2%, 99.96% and 94.64%.

Table 4.6: Precision and accuracy results when reached in location KFC, Raub

	Experimental Value	Actual value	Precision and Accuracy
Latitude	3.777678167	3.790755	65.50
Longitude	101.8591957	101.857083	99.80
Altitude	143.5 meters	145.0 meters	98.97
Time	14:15p.m	14:15p.m	None

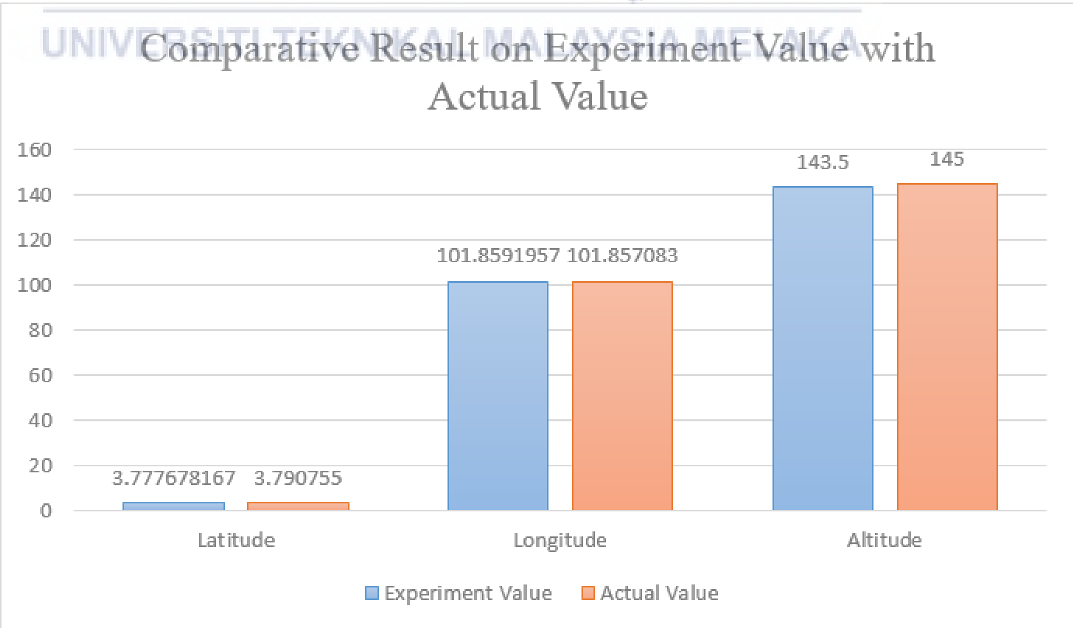

Figure 4.27: Comparative results based Experiment value and Actual value

Based on Table 4.6 and Figure 4.27, all the result based latitude, longitude and altitude are collected from the location KFC in Raub which based on microcontroller Raspberry Pi 3 implement with GY-NEO6MV2 GPS sensor under the same location. The results show the highly obvious different changes from the actual location based Google Maps App from previous results shown in Table 4.3, Table 4.4 and Table 4.5. Measurement errors for latitude is the highest compared to longitude and altitude. Accuracy achieved by latitude is the lowest which is only about 65.50% as shown in table 4.6. Based on the result obtained, we can observed that the changed of signal and the signal received is sensitive to environment changes. During the data collection process, the sky was cloudy and raining. Besides, the data are collected by running the GPS sensor in the car until reached the destination. As a results, the changes of the environment condition probably caused the great influence of the accuracy results. So, delay of time for sensor and the IoT devices to process and received the signal caused the high measurement errors occurred.

4.5.3 Result of Distance and Speed based on NodeMCU

Table 4.7: Speed and distance from location A to B based Node MCU

Location A to B	Experiment Value	Actual Value	Precision and Accuracy (%)
Distance (Km)	1.30	1.80	72.22
Speed (km/h)	19.5	36	54.17

In this part, distance and speed can be measured and obtained based on the results collected from location A (Taman Raub Jaya 2) to location B (KFC, Raub) based GPS sensor and microcontroller of NodeMCU. Algorithm to measure the distance and speed can be developed as shown in Table 4.7 with the unit of kilometres for distance and kilometre per hours for speed. Based on the results in Table 4.7, there are the changes of precision of error compared to an actual value which the precision and accuracy for the distance and speed is below 80% but the precision and accuracy is higher than performance result in Raspberry Pi 3. This may due to the issue arises from the previous experiment in Table 4.3, Table 4.4, Table 4.5 and Table 4.6 as GPS need a hot start, blockage of signal and an environmental changes caused the delay time to receive the signal data.

4.5.4 Results of Distance and speed based on Raspberry Pi 3 Model B

Table 4.8: Speed and distance from location A to B based Raspberry Pi 3 model B

Location A to B	Experiment Value	Actual Value	Precision and Accuracy (%)
Distance (Km)	0.00712	1.80	0.4
Speed (km/h)	0.1068	36	17.75

In this part, distance and speed are measured and obtained based on the results collected from location A (Taman Raub Jaya 2) to location B (KFC Raub) based sensor and microcontroller of Raspberry Pi 3. Algorithm to measure the distance and speed can be developed as shown in Table 4.8 with the unit of kilometres for distance and kilometre per hours for speed. Based on the results in Table 4.8, the result from experiment value and actual value are totally different and this show the performance result were not accurate. Precision and accuracy for distance and speed is just only 0.4% and 17.75%. Hence, the results conclude that the result of latitude and longitude collected from the previous experiment affect the distance and speed.

4.6 Overall performance results on IoT development boards

Table 4.9: Overall Precision and Accuracy on IoT development boards

	Overall Precision and Accuracy (%)	
	IoT devices NodeMCU	IoT devices Raspberry Pi 3 Model B
Location A	99.80	99.58
Location B	97.785	82.65
Altitude A	90.66	94.64
Altitude B	95.03	98.97
Distance A to B	72.22	0.4
Speed A to B	54.17	17.75

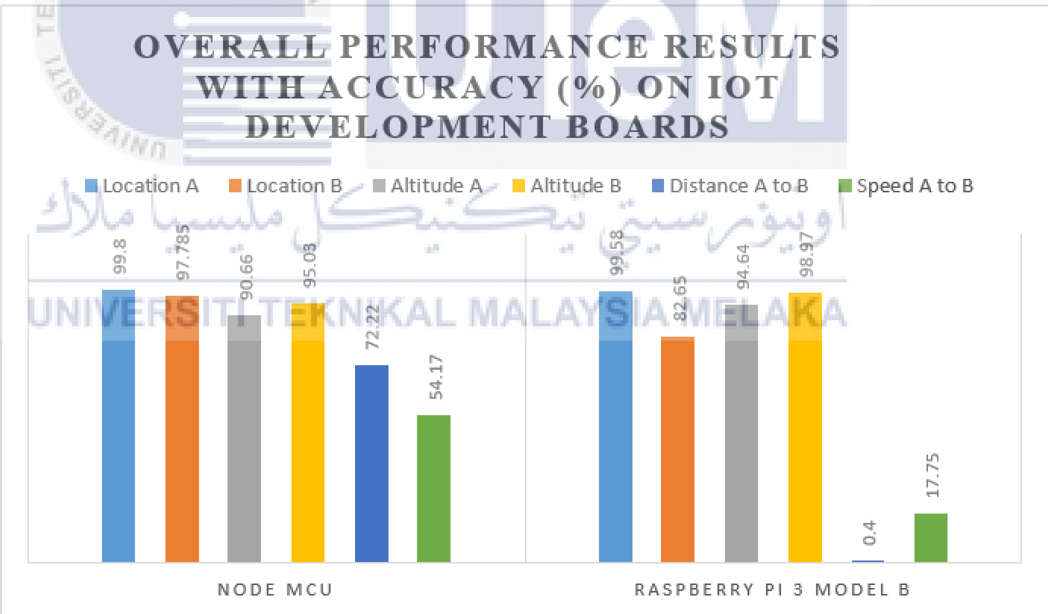

Figure 4.28: Overall performance results with accuracy (%) on IoT development boards

The overall precision and accuracy for IoT development boards for NodeMCU and Raspberry Pi 3 Model B were analyzed and plotted in Table 4.9 and Figure 4.28.

According to this results, microcontroller NodeMCU perform well and efficiently compared to Raspberry Pi 3 when implement with GPS sensor. NodeMCU achieved highest accuracy for Location A (average precision and accuracy of latitude and longitude in Taman Raub Jaya 2) and Location B (average precision and accuracy of latitude and longitude in KFC, Raub) compared to Raspberry Pi 3 Model B. Location A and location B based NodeMCU each achieved 99.80% and 97.785% higher than Raspberry Pi 3. Besides, accuracy distance and speed from A to B are higher than Raspberry Pi 3 which achieved 72.22% and 54.17%. However, the overall accuracy for altitude A and B using NodeMCU is slightly lower than Raspberry Pi 3 which achieved 90.665 and 95.03%. This may due to the issue of changes of signal and the environment conditions. Based on this experiment, performance results of Raspberry Pi 3 with GPS sensor were not good compare to NodeMCU as the accuracy of distance and speed travel from A to B had changes obviously which achieved only 0.4% and 17.75% compare to NodeMCU. This may due to the result obtained from the experiment affect the measurement accuracy of distance and speed. Therefore, difference performance from IoT developments boards had been arises due to different conditions when integrate with GPS sensor. As GPS sensor were sensitive to the signal received, blockage of the signal and the reflection of signal through the building, sky, humans and objects caused the receiver failed to receive the original signal based on the location. In addition, algorithm to conduct and measure the experiment also become an issues which caused different performance in this project. As a conclusion, experiment is carried out in this project and the results showed that IoT development boards for NodeMCU show the best performance when implement with GPS sensor as the results achieved the highest accuracy based on the locations, altitude, distance and speed compared to Raspberry Pi 3.

4.7 Chapter Summary

Based on first part in this chapter, experiment are set up embedded with the hardware and software part. Each IoT development boards, Node MCU and Raspberry Pi 3 are implement with the GPS sensor to track and trace the locations from starting point to the final destination. Route of location is travel by using car along with the IoT devices and GPS sensor which start from Taman Raub Jaya 2 and reach destination in KFC restaurant in Raub. In second part of this chapter, all the data collected from the GPS sensor are access and display to the IoT cloud, Thingspeak by applying the principle and knowledge of the coding for further analysis. Hence, research on the performance results of IoT developments boards for GPS application is implemented and verified while the objectives in this project had been achieved and succeed.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

Rapid growth of innovation technology have enhance our world with high technology services as Global Positioning System (GPS) becomes high demands application technologies which prompt users utilize as part of daily life and navigate around the world especially to track infected people and confirmed cases during this Covid-19 Pandemic. Due to wide range of standalone and portable IoT development boards develops in the market, aim to analyze the performance of IoT development boards for GPS application was achieved in this research project.

Outdoor tracking system based on the locations from the starting point, Taman Raub Jaya 2 and reached in location KFC, Raub was successfully implemented and developed. Data is collected from NodeMCU and Raspberry Pi 3 Model B which is implemented with GPS sensor module, GY-NEO6MV2 which displayed in IoT cloud, Thingspeak. Performance results according the research experiment has been analyzed and discussed in this thesis.

As a conclusion, IoT development boards for NodeMCU show the best performance where overall accuracy location A and location B achieved by NodeMCU was 99.8% and 97.785% higher than Raspberry Pi 3 when compared with actual location from Google Map App. However, overall accuracy for altitude A and B using NodeMCU is slightly lower than Raspberry Pi 3 which achieved 90.66% and 95.03%. Apart from that, accuracy based the algorithm measurement for the distance and speed from NodeMCU is higher than Raspberry Pi 3 which achieved 72.22% and 54.17%. Hence, the results show that difference performance from IoT developments boards had been arises due to different conditions when integrate with GPS sensor. Blockage of signal, environment factors, weather conditions and the algorithm method and measurement caused the different performance arises in the results. Last but not least, the aim and objectives in this project had been achieved successfully.

5.2 Future Recommendation

Implement of deep learning become more popular in this recent year. Performance analysis of this project can be improved by applying technique deep learning application for localization and positioning system. Therefore, a deep learning based image recognition can be detect the locations more efficiently and accurately for the future used. Due to limitation of GPS signal in urban areas which includes of building structure and environment conditions, performance of the signal can be improved by applying technique of Received signal strength (RSS) to overcome accuracy performance compared to Google Map. Furthermore, there are only one GPS module sensor apply in this project. In order to increase the performance of analysis, different GPS module sensor can be implement and analyze for further research. In addition, this project can be further improved to an advanced app by using mobile phone. As a result, details of location positioning system and performance can be track in everywhere in anytime especially in this Covid-19 outbreak which spread globally. Hence, a portable and multifunction app that provide the latest information can be develop which provide convenience for the end user.

REFERENCES

[1] L. Yu, Y. Liu, T. Chi, and L. Peng, "An iBeacon-based indoor and outdoor positioning system for the fire emergency command," *2017 Forum Coop. Position. Serv. CPGPS 2017*, pp. 326–329, 2017.

[2] L. Pei, R. Chen, Y. Chen, H. Leppäkoski, and A. Perttula, "Indoor/outdoor seamless positioning technologies integrated on smart phone," *Proc. - 2009 1st Int. Conf. Adv. Satell. Sp. Commun. SPACOMM 2009*, pp. 141–145, 2009.

[3] J. Whipple, W. Arensman, and M. Starr, "A Public Safety Application of GPS-Enabled Smartphones and the Android Operating System," no. October, pp. 2059–2061, 2009.

[4] M. Z. Hoque, "Basic Concept of GPS and Its Applications," *IOSR J. Humanit. Soc. Sci. (IOSR-JHSS*, vol. 21, no. 3, pp. 31–37, 2016.

[5] C. J. Hegarty, J. M. Foley, and S. K. Kalyanaraman, "Global positioning system," *Digit. Avion. Handbook, Third Ed.*, vol. 46, no. 2, pp. 4-1-4–24, 2017.

[6] A. Hammami, "Person Tracking System based on Arduino Microcontroller and Web Technologies," *2018 Int. Conf. Smart Appl. Commun. Networking,*

SmartNets 2018, pp. 1–4, 2018.

[7] M. L. Yin and R. Arellano, “A risk analysis framework on GPS User Range Accuracy,” *Proc. - Annu. Reliab. Maintainab. Symp.*, pp. 0–5, 2012.

[8] M. Shoab, K. Jain, M. Anulhaq, and M. Shashi, “Development and implementation of NMEA interpreter for real time GPS data logging,” *Proc. 2013 3rd IEEE Int. Adv. Comput. Conf. IACC 2013*, pp. 143–146, 2013.

[9] M. Thesis, A. M. H. Khalel, and A. Mohammed, “Position Location Techniques in Wireless Communication Systems,” *Electr. Eng.*, no. October, pp. 1–53, 2010.

[10] J. Zhu and G. D. Durgin, “Indoor = outdoor location of cellular handsets based on received signal strength ELECTRONICS LETTERS 6th January 2005,” vol. 41, no. 1, pp. 1–3, 2005.

[11] F. Brachmann, “A comparative analysis of standardized technologies for providing indoor geolocation functionality,” pp. 20–23, 2012.

[12] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, “Smart Energy Efficient Home Automation System Using IoT,” *Proc. - 2019 4th Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2019*, pp. 1–4, 2019.

[13] C. Lee, “Internet of Things Security - Multilayered Method For End to End Data Communications Over Cellular Networks,” *2019 IEEE 5th World Forum Internet Things*, pp. 24–28, 2019.

[14] M. Bansal, “IoT Based Development Boards for Smart Healthcare

Applications," *2018 4th Int. Conf. Comput. Commun. Autom.*, pp. 1–7, 2018.

[15] A. Raza and A. J. Ikram, "A Review of Low Cost and Power Efficient Development Boards for IoT Applications," no. December, pp. 786–790, 2016.

[16] B. V. S. Padmaja, V. R. Kolluru, and S. S. Kota, "IoT based implementation of vehicle monitoring and tracking system using node MCU," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 6, pp. 446–450, 2019.

[17] M. H. Ngui and W. K. Lee, "Low power wearable device with GPS and indoor positioning system," *Proc. - 2019 7th Int. Conf. Green Hum. Inf. Technol. ICGHIT 2019*, pp. 125–127, 2019.

[18] P. A. Shinde and Y. B. Mane, "Advanced vehicle monitoring and tracking system based on Raspberry Pi," *Proc. 2015 IEEE 9th Int. Conf. Intell. Syst. Control. ISCO 2015*, pp. 1–6, 2015.

[19] L. Vokorokos, M. Uchnar, and E. Pietrikova, "The raspberry Pi in an education process," *SACI 2018 - IEEE 12th Int. Symp. Appl. Comput. Intell. Informatics, Proc.*, pp. 87–90, 2018.

[20] M. Desai and A. Phadke, "Internet of Things based vehicle monitoring system," *IFIP Int. Conf. Wirel. Opt. Commun. Networks, WOCN*, pp. 1–3, 2017.

[21] S. Saha *et al.*, "GPS based smart spy surveillance robotic system using Raspberry Pi for security application and remote sensing," *2017 8th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2017*, pp. 705–709, 2017.

[22] K. Shi *et al.*, "A novel file carving algorithm for National Marine Electronics Association (NMEA) logs in GPS forensics," *Digit. Investig.*, vol. 23, pp. 11–21, 2017.

[23] S. N. Parmar, S. Nainan, K. Bakade, and D. Sen, "An efficient mobile GPS navigator, tracker and altimeter system for location based services," *2013 Int. Conf. Adv. Technol. Eng. ICATE 2013*, no. 150, pp. 3–6, 2013.

[24] E. Winarno, W. Hadikurniawati, and R. N. Rosso, "Location based service for presence system using haversine method," *Proc. - 2017 Int. Conf. Innov. Creat. Inf. Technol. Comput. Intell. IoT, ICITech 2017*, vol. 2018-Janua, pp. 1–4, 2018.

[25] A. Jiménez-Meza, J. Arámburo-Lizárraga, and E. de la Fuente, "Framework for Estimating Travel Time, Distance, Speed, and Street Segment Level of Service (LOS), based on GPS Data," *Procedia Technol.*, vol. 7, pp. 61–70, 2013.

[26] W. Pu, J. J. Lin, and L. Long, "Real-Time Estimation of Urban Street Segment Travel Time Using Buses as Speed Probes," pp. 81–89.

[27] T. P. Fowdur and M. N. Khodabacchus, "Performance Analysis of a Real-Time Cloud Based Bus Tracking System with Adaptive Prediction," vol. 7, no. 25, pp. 3454–3473, 2017.

[28] D. Bhardwaj, S. Daphal, and A. Nerkar, "Predicting Bus Arrival Time with GPS on Android Application," vol. 5, no. 1, pp. 2014–2017, 2016.

[29] M. Afnan, "Tracking System for Missing Person," pp. 57–60, 2017.

[30] A. Chaudhari, S. Bohra, H. Karma, and A. Dhupadale, “GPS / GSM Enabled Person Tracking System,” pp. 981–986, 2015.

[31] M. R. Mosavi, “An Adaptive Correction Technique for DGPS using Recurrent Wavelet Neural Network,” pp. 3029–3033, 2007.

[32] A. Al-hindawi, “Performance of Differential GPS Based on a Real-Time Algorithm Using SMS Services of GSM Network,” no. November, 2015.

[33] Y. He, H. Yu, and H. Fang, “Study on Improving GPS Measurement Accuracy,” no. May, pp. 17–19, 2005.

[34] M. G. Wing and J. Frank, “Vertical measurement accuracy and reliability of mapping-grade GPS receivers,” *Comput. Electron. Agric.*, vol. 78, no. 2, pp. 188–194, 2011.

[35] R. K. S. H. B. Hablani, *High-Accuracy GPS-Based Aircraft Navigation for Landing using Pseudolites and Double-Difference Carrier Phase Measurements*, vol. 47, no. 1. IFAC, 2014.

[36] P. Verma and J. . Bhatia, “Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring,” *Int. J. Comput. Sci. Eng. Appl.*, vol. 3, no. 3, pp. 33–40, 2013.

APPENDICES

Appendix A

```
#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

static const uint32_t GPSBaud = 9600;

// Repace with  WiFi ssid and password

const char* ssid     = "tyy";

const char* password = "tyypassword";

unsigned long myChannelNumber = 1020837; // Replace it with your thingspeak channel number

const char * myWriteAPIKey = "1FM4VKUG0HBQR161"; // Replace it with your Thingspeak WRITE API key
```

```
// The TinyGPS++ object

TinyGPSPlus gps;

WiFiClient  client;

// The serial connection to the GPS device

SoftwareSerial ss(4, 5);

void setup()

{

  Serial.begin(115200);

  ss.begin(GPSBaud);

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

   Serial.println("");
```

```
  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

  Serial.print("Netmask: ");

  Serial.println(WiFi.subnetMask());

  Serial.print("Gateway: ");

  Serial.println(WiFi.gatewayIP());

  ThingSpeak.begin(client);

  }

void loop()

{

  // This sketch displays information every time a new sentence is correctly encoded.

  while (ss.available() > 0)

    if (gps.encode(ss.read()))

      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)

  {
```

```
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
}
void displayInfo()
{
  if (gps.location.isValid())
  {
    double latitude = (gps.location.lat());
    double longitude = (gps.location.lng());
    double altitude = (gps.altitude.meters());
    String latbuf;
    latbuf += (String(latitude, 6));
    Serial.println(latbuf);
    String lonbuf;
    lonbuf += (String(longitude, 6));
    Serial.println(lonbuf);
```

```
  String altbuf;

  altbuf += (String(altitude));

  Serial.println(altbuf);

  ThingSpeak.setField(1, latbuf);

  ThingSpeak.setField(2, lonbuf);

  ThingSpeak.setField(3, altbuf);

  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  delay(20000);

    }

else

{

  Serial.print(F("INVALID"));

}

Serial.print(F("  Date/Time: "));

if (gps.date.isValid())

{

  Serial.print(gps.date.month());
```

```
  Serial.print(F("/"));

  Serial.print(gps.date.day());

  Serial.print(F("/"));

  Serial.print(gps.date.year());

}

else

{

  Serial.print(F("INVALID"));

}

Serial.print(F(" "));

if (gps.time.isValid())

{

  if (gps.time.hour() < 10) Serial.print(F("0"));

  Serial.print(gps.time.hour());

  Serial.print(F(":"));

  if (gps.time.minute() < 10) Serial.print(F("0"));

  Serial.print(gps.time.minute());
```

```
    Serial.print(F(":"));

    if (gps.time.second() < 10) Serial.print(F("0"));

    Serial.print(gps.time.second());

    Serial.print(F("."));

    if (gps.time.centisecond() < 10) Serial.print(F("0"));

    Serial.print(gps.time.centisecond());

  }

  else

  {

    Serial.print(F("INVALID"));

   }

 Serial.println();

}
```

Appendix B

```
import sys

from urllib import urlopen   // change from urllib.request to urllib, because python 2 only support urllib

from time import sleep

import serial

import string

import time

import pynmea2

port="/dev/ttyAMA0"

myAPI = 'IN25D7GKNXNR1XOK'  // From thingspeak API

// URL where we will send the data, Don't change it

baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI

ser = serial.Serial(port, baudrate = 9600, timeout = 0.5)

while 1:

    try:

            data = ser.readline()
```

```
  except:
    print("loading")
  if data[0:6] == '$GPGGA':
    msg = pynmea2.parse(data)
    latval = msg.lat
// latitude value = latval
    concatlat = "lat: " + str(latval)
    print (concatlat)
    longval = msg.lon
// longtitude value = longval
    concatlon = "Long: " + str(longval)
    print (concatlon)


    altval = msg.altitude
    //altitude value = altval
    concatalt = "alt: " + str(altval)
    print (concatalt)
```

time.sleep(1)

conn = urlopen(baseURL + '&field1=%s&field2=%s&field3=%s' % (latval, longval, altval)) // Set the variable to upload the thingspeak

print (conn.read())

conn.close()

sleep(20)