DEVELOPMENT OF INTERNET OF THING BASED GPS TRACKING SYSTEM FOR A MOBILE CARGO ASSETS

AHMAD JAZMI BIN MUKHTAR
(141031706)

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

SCHOOL OF MICROELECTRONIC ENGINEERING
UNIVERSITI MALAYSIA PERLIS

JUNE 2018

UNIVERSITI MALAYSIA PERLIS (UniMAP)

SCHOOL OF MICROELECTRONIC ENGINEERING

DECLARATION OF REPORT

Author's full name	:	AHMAD JAZMI BIN MUKHTAR
Date of birth	:	12 FEBRUARY 1995
Title	:	DEVELOPMENT OF INTERNET OF THING BASED GPS TRACKING SYSTEM FOR A MOBILE CARGO ASSETS
Academic Session	:	2017/2018

I hereby declare that the report becomes the property of Universiti Malaysia Perlis (UniMAP) and to be placed at the library of UniMAP. This report is classified as:

☐ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

☐ **RESTRICTED** (Contains restricted information as specified by the organization where research was done)*

☐ **OPEN ACCESS** I agree that my report is to be made immediately available as hard copy or on-line open access (full text)

I, the author, give permission to the UniMAP to reproduce this report in whole or in part for the purpose of research or academic exchange only (except during a period of _____ years, if so requested above).

Certified by:

______________________	______________________
SIGNATURE	**SIGNATURE OF SUPERVISOR**
950212-03-5379	**IR DR SOHIFUL ANUAR BIN ZAINOL MURAD**
(NEW IC NO. / PASSPORT NO.)	**NAME OF SUPERVISOR**
Date: _________________	Date: _________________

***NOTES**: If the report is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentially or restriction.

UNIVERSITI MALAYSIA PERLIS (UniMAP)

SCHOOL OF MICROELECTRONIC ENGINEERING

DECLARATION SHEET

I, AHMAD JAZMI BIN MUKHTAR, declare that the report entitled DEVELOPMENT OF INTERNET OF THING BASED GPS TRACKING SYSTEM FOR A MOBILE CARGO ASSETS and the work presented in the report are both my own, and have been generated by me as the result of my own original research. I confirm that:

Signature :

Name : AHMAD JAZMI BIN MUKHTAR

Date :

This report titled DEVELOPMENT OF INTERNET OF THING BASED GPS TRACKING SYSTEM FOR A MOBILE CARGO ASSETS was prepared and submitted by AHMAD JAZMI BIN MUKHTAR (Matrix Number: 141031706) and has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirement for the Bachelor of Engineering (Hons) (Eectronic Engineering) in Universiti Malaysia Perlis (UniMAP).

Checked and Approved by

……………………….

(IR DR SOHIFUL ANUAR BIN ZAINOL MURAD)

Supervisor

School of Microelectronic Engineering

Universiti Malaysia Perlis

(Date: ....)

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere appreciation and gratitude is extended to my supervisor, Ir Dr Sohiful Anuar Bin Zainol Murad for his guide to the right direction while I am doing my final year project. He is very generous in sharing his great knowledge from starting stage of this project until now. I was always welcome to talk about my project whenever I faced any problem. I really enjoyed doing my final year project under his supervision.

Apart from that, I would like to thanks my senior Muhammad Aidil Faiz bin Ismail for lending me a hand when I faced any problem in hardware design. I will be having a hard time to complete this project without his help.

Last but not least, I also would like to thanks my parents and course mate for their endless support for me to finish this project.

UNIVERSITI MALAYSIA PERLIS (UniMAP)
PUSAT PENGAJIAN KEJURUTERAAN MIKROELEKTRONIK

ABSTRAK

PENCIPTAAN SISTEM PENGESAN GPS BERDASARKAN KEPADA OBJEK RANGKAIAN INTERNET UNTUK BARANGAN KARGO YANG BERGERAK

Ahmad Jazmi Bin Mukhtar

Teknologi GPS akan menjadikan sistem pengawasan lebih berkuasa jika digabungkan dengan teknologi internet (IoT). Isu-isu yang membawa kepada penciptaan projek ini adalah anggota syarikat yang perlu mengetahui lokasi aset mereka dan masalah jenayah yang berkaitan dengan aset kargo yang perlu dikurangkan. Projek ini dijalankan adalah bertujuan untuk melaksanakan perkakasan sistem pengesanan GPS berdasarkan IoT yang terdiri daripada modul GPS dan modul WiFi sebagai komponen utama sistem, untuk membangunkan perisian android untuk sistem pengesanan GPS berdasarkan IoT yang membolehkan pengguna untuk mengesan lokasi aset mereka melalui telefon pintar, untuk menghasilkan integrasi perisian dan perkakasan yang mencapai hasil yang diinginkan yang dapat melihat lokasi item yang dikesan dalam aplikasi telefon pintar dan untuk mengesan kedudukan barang bergerak dalam jarak yang jauh. Pada asasnya, hasil projek ini diperoleh berdasarkan reka bentuk dan pembangunan perkakasan menggunakan Arduino Uno dan pembangunan perisian menggunakan MIT App Inventor 2. Projek ini telah mencapai objektif iaitu dapat mengesan laluan item yang dikesan dari titik permulaan yang telah disimpan ke lokasi masa nyata barangan tersebut.

UNIVERSITI MALAYSIA PERLIS (UniMAP)
SCHOOL OF MICROELECTRONIC ENGINEERING

ABSTRACT

DEVELOPMENT OF INTERNET OF THING BASED GPS TRACKING SYSTEM FOR A MOBILE CARGO ASSETS

Ahmad Jazmi Bin Mukhtar

GPS technology will make the monitoring system even more powerful if combined with the internet of things (IoT) technology. The issues that lead to the development of this project is company members that need to aware the location of their assets and the crime problem related to cargo assets that need to be reduced. This project is carried out purposely to implement an IoT GPS tracking system hardware which consist of GPS module and WiFi module as a main component of the system, to develop an android software for IoT GPS tracking system which enable a user to track the location of their goods and facilities through their smartphone, to produce an integration of software and hardware that achieve a desired result which can view the location of tracked items in the smartphone apps and to track the position of moving items in a long distance. Basically, the result of this project is obtained based on the design and the development of hardware using Arduino Uno microcontroller board and development of software using MIT App Inventor 2. The project has achieve the objective which can tracked the route of the tracked items from the save starting point to its real time location.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Thing
GPS	Global Positioning System
GSM	Global System for Mobile Communication

Chapter 1
Introduction

1.1 Research background

Due to the overwhelming problem of losing property in today's world, Global Positioning System or better known as GPS has been introduced to the world community. GPS is a type of system comprising a network of 24 orbiting satellites that transmit a precise and accurate microwave signal. This kind of technology enables the receiver on Earth to accurately calculate its position [1]. Many problems with loss of property such as loss of vehicles, loss of exports and imports of manufactured goods and even a human loss have also been solved using GPS technology. This is indirectly helpful in solving many more criminal and kidnapping crime cases more easily.

GPS technology will make the monitoring system even more powerful if combined with the internet of things (IoT) technology. IoT is a new thing in the field of information and communications technology that allows all sensors to interact smartly with users [2]. Latitude and longitude information tracked by GPS will be sent to apps in smartphones via WiFi as data transmission medium. The reason why WiFi is chosen as the medium of transmission is that the distance between transmission and data reception for WiFi is far more than RFID and consume less power compared to GSM. Furthermore, the WiFi is more suitable to be implemented in the cargo trucks which it is possible to have the wifi connection inside the vehicles nowadays.

Arduino programming language is selected to be applied in this IoT tracking system to enable the hardware tracker system read the data of the tracked real-time location in terms of latitude and longitude. Those data will be sent to the database via Wi-Fi. Besides, Arduino programming is widely used in developing the IoT based hardware system such as in the field of mobility, healthcare, industry, energy and so on [3]. Arduino is a middle-level language that enables programmer, developer, and makers to compiles interprets and edits the operating system and embedded programming [4]. Even the Intel development board such as Up board, Edison board, Galileo board and Joule board also enable the makers to use Arduino programming in developing IoT base hardware system [5].

The laptop will be used to connect to the Arduino Uno board and eventually to upload the sketches programs into the board. Besides, it is based on the WiFi module and small size development board which will consume lower power consumption and of course will use less money to develop the hardware tracker system. This tracker will help in monitoring the real-time location of the cargo which will make it easier for the cargo owner to track the location of their goods.

The tracker system as illustrated in Figure 1.1 can help to monitor the real-time location of the tracked item. The data of the position is in term of latitude and longitude. The principles of satellite navigation are based on the signals and data sent from satellite to the receiver [10]. The receiver is located inside the satellite purposely to have its orbit determined [10]. This receiver is able to measures the travel time of the signal and then calculates the distance between the receiver and the GPS satellite [10]. In case of the clocks are synchronized, the position is enough to be obtained by the measurements from the three GPS satellites [10]. When the data are obtained by the GPS receiver module, it will be sent to the database via wifi [10].

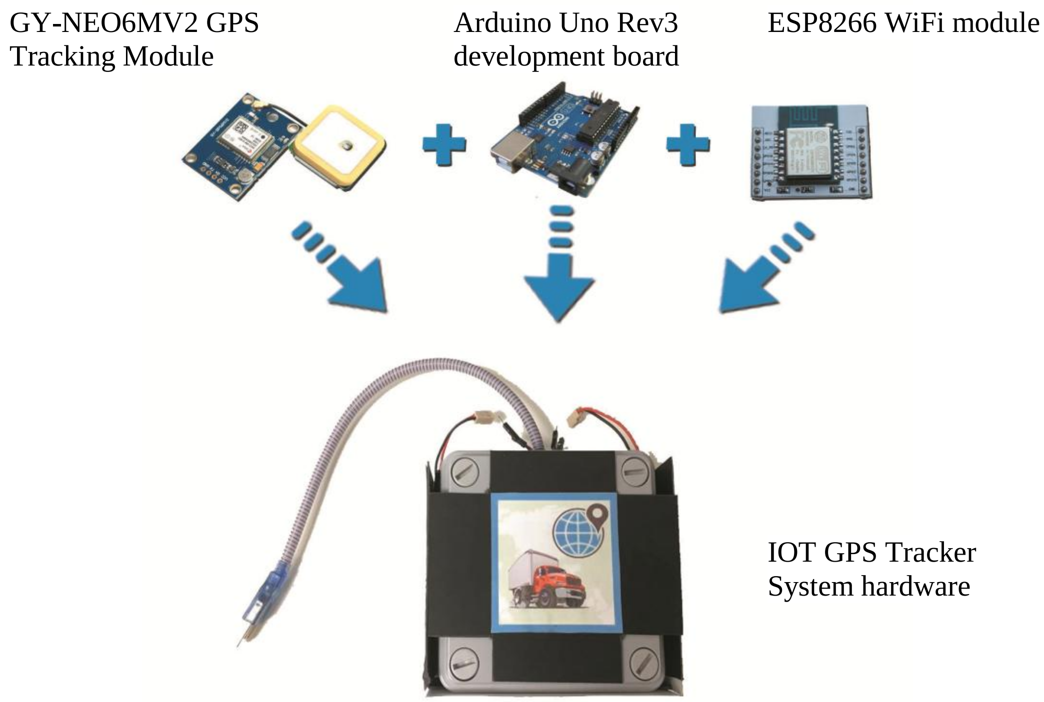


Figure 1.1: Basic structure of IOT GPS Tracker System

1.2 Research problem

Based on research explored, the logistics and transportation industry in Malaysia has turned into noteworthy and generator for production line and distribution channel. Without logistics and transportation, a process in a production line and distribution channel could be deadened.

The expansion in logistics administration was viewed as a source of competitive power. Its successful utilize gives potential to cost diminishment and an open door for drawing in the investors. The component in logistics has dependably been essential to the manufacturing, stockpiling and development of merchandise and items. Presently, the role of logistics was a noteworthy part in the achievement of a wide range of operations and associations. Logistics can likewise be referred as "physical web" for its talent to interface every one of the products from supplier to the clients with unsurprising, auspicious, and financially savvy way. Its ability makes logistics to develop into a key determinant of nation's intensity.

The fast advancement of the logistics business has sadly welcomed the risk of cargo crime, which as of late has turned into a troublesome issue. There are few sorts of cargo crime such as hijacking, warehouse theft, lost in transit and many more.

Since cargo crime is a wide theme and issues, the investigation is concentrating on the cargo crime on the road particularly in hijacking and theft of legal shipments during transit. Cargo crime on the road can be characterized as any burglary with shipment committed during its street transportation or inside a warehouse or premise. The most mainstream cargo crime on the road is hijacking. Vehicle hijacking of trucks with cargo is one of the appearances thefts, which made the hijacking, is under an indistinguishable crime from burglary. This statement has concurred when the hijacking trucks characterized as the unlawful and purposeful strong expulsion and suitable of a truck.

The prime focuses of cargo crime are cargo compartments, trucks, and railcars that cross the country every day. It is not shocking if the cargo crime continues expanding every year as evaluated RM 90 to RM 150 billion in cargo is stolen worldwide every year.

In Malaysia particularly, the general crime index in the initial nine-month in 2010 has declined 16% around the world, yet there was not eased up in the quantity of thefts premises and lorry hijackings. The quantity of cargo crime was adaptable as it can be expanded and decay anytime without our notice. This was demonstrated when the most recent report by Transported Asset Protection Association - Asia (TAPA) evaluated RM7.7 million worth of products were stolen from Malaysian ports, airplane terminal, stockrooms, and trucks from 2007 to 2010. Indeed, the initial four months of 2011, there have been 21 episodes of freight crime were reported. In 2012, Malaysia was recorded as the second most elevated amount of cargo crime in Asia Pacific area in term of the estimation of products stolen. While the primary most abnormal amount in Hong Kong.

Besides the road accident related to cargo trucks is one of the reason the monitoring system of the assets brought by the cargo truck needs to exist. The Transport Statistic Malaysia 2016 which is the latest transport statistic shown in Official Portal Ministry of Transport Malaysia reveals that from 2007 until 2018, the Total Motor Vehicles Involved in Road Accidents gradually increase which will affect the profit of the company if the accident happened on the cargo trucks that bring the company assets. The company members should know at least the location of their assets at real time so that fast action can be taken to avoid the company suffered from the severe losses.

1.3 Research Objective

This project has a very concise aim to design a functional internet of things GPS tracking system that able to observe the position and the route of the tracked items. This is the envisaged at the end point of the project in order to get a real time location data in term of latitude and longitude to be viewed in mobile application software.

The objectives of this project are:

i. To implement an IoT based GPS tracking system hardware which consist of GPS module and WiFi module as a main component of the system.
ii. To develop an android software for IoT based GPS tracking system which enable a user to track the location of their goods and facilities through their smartphone.
iii. To produce an integration of software and hardware that achieve a desired result which can view the location of tracked items in the smartphone apps.
iv. Track the position of moving items in a long distance.

1.4 Project scope

The scope of this internet of thing based gps tracker system includes

i. The arduino software and MIT App Inventor 2 software to develop the internet of thing based gps tracker system.
ii. The hardware of internet of thing based gps tracker system which consist of Arduino UNO microcontroller board, ESP8266 WiFi module and gy-gps6mv ublox gps module.

iii. The development of application software for internet of thing based gps tracker system which can link to google map.
iv. The prototype of internet of thing based gps tracker system.

1.5 Thesis organization

The methods proposed in this research, investigate the position parameters which is latitude and longitude of the tracked items. This report had been divided into several parts according to chapters.

Chapter 1 consists of introduction about the research that have been carried out and methods to calibrate it. It consists of the requirement of the tracking system and apply the IOT in tracking system in an efficient way, problem statement of the research, the objectives of the project, scope of work and thesis organization.

Chapter 2 elaborate the literature review that describes the gps module which is the important part in this project. Basically the GPS module is used to get the data of the position of the tracked items which can be analysed by referring to the articles and the journals based on the interrelated chapters.

Chapter 3 is about the methodology, including the techniques used to transfer the data to the database and to send the data to the mobile apps. In addition, the techniques also include the materials used for this project.

Chapter 4 shows the result obtained in this implemented project and the discussion on the result. The result obtained from the hardware circuit design indicated the positioning sensor which is GPS module. The sensor determining by the data appeared in the database which is Thingspeak and in the mobile apps to show the role of monitoring function.

Chapter 5 is the summary done for conclusions based on the objectives in order to achieve the aims. Other than that, this chapter also deliberate on the recommendation for improvement of this research in the future.

Chapter 2
Literature review

2.1 Introduction

IOT GPS Tracker system for cargo goods is a system to provide a position of a cargo truck. The position data is base on latitude and longitude. Both of this parameter is referring to the angles which uniquely define the points on this sphere which together, the points include facilities conspire that can find or recognize geographic positions on the surfaces of planets, for example, the earth [11].

For instance, the biggest influence of real-time tracking system nowadays is the transfer media that utilized in a system. The ideal transfer media for real-time tracking system is depending on where the tracker system is going to be used. The larger the coverage covered by the system, the higher the suitability rate to be used as the medium of transmission within this system. If the system covers a limited area, it tends to make the system fail to work as it has been proposed.

Moreover, the position data of the tracked item which in term of latitude and longitude is also played a major impact on the system. For accuracy purposes, degrees of longitude and latitude have been separated into minutes (') and seconds ("). There is an hour in every degree. Every moment is separated into 60 seconds. Seconds can be additionally separated into tenths, hundredths, or even thousandths [11].

2.1.1 Comparison the techniques of previous work

A. RF Control Based Mobile Robotic System for Search Mission with GPS Tracker [12]

This journal proposed that robot can complete a work effortlessly, which is by all accounts outlandish for a man and it turns out to be more useful on the off chance that one can control it remotely [12]. The idea of utilizing an automated framework for scan mission has been proposed for seeking of survivors if there should arise an occurrence of cataclysmic event like tremor, tropical storm, wave and it can likewise be utilized as a GPS beacon as it additionally has inbuilt GPS gadget which can give the correct current area of the robot [12]. There are sure highlights of the automated framework which makes a total and flexible framework for look mission like it can be controlled remotely utilizing RF method which is thought to be the best as for range and obstruction, this framework can likewise detect the earth with the assistance of various sensors, it has inbuilt camera which can send the live recordings remotely to remote gadget, GPS is additionally appended with the framework which can send the present area to remote gadget any time of the time [12]. Every one of these highlights makes this mechanical framework equipped for doing things which appears to be extremely troublesome for the human [12].

B. Application of GPS Tracker Technology for Identification of Road Trafficking Conditions [13]

This journal proposed that continuous following framework innovation has been made conceivable by incorporating three innovations, to be specific worldwide situating framework (GPS), database innovations, for example, geographic data framework (GIS) and portable media communications innovations, for example, general parcel radio benefit (GPRS) [13]. This paper has proposed a vehicle following component in light of GPS tracker to construct a continuous movement data framework [13]. A GPS server is worked to process information of position and speed of the vehicle for additionally prepared into vehicle activity data

[13]. The Server and GPS tracker is intended to convey utilizing GPRS benefits progressively [13]. Moreover, the server forms the information from the GPS tracker into activity data, for example, the road turned parking lot, thick, medium and easily [13]. Test outcomes demonstrated that the GPS server can picture the genuine position of the vehicle and can choose the class of activity data progressively [13].

C. GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization [14]

This journal proposed adding an area to the accessible data empowers another classification of uses [14]. With the compelled battery on mobile phones, vitality proficient limitation turns into an essential challenge. In this paper we present a low-vitality calibration-free limitation conspire in light of the accessible interior sensors in a large number of the present telephones [14]. We begin by vitality profiling the distinctive sensors that can be utilized for restriction. In light of that, we propose GAC: a half and half GPS/accelerometer/compass conspire that depends chiefly on utilizing the low-vitality accelerometer and compass sensors and utilizations the GPS rarely for synchronization [14]. We actualized our framework on Android-empowered cell telephones and assessed it in both roadways and intra-city driving conditions [14]. Our outcomes demonstrate that the proposed half breed conspire has an exponential sparing in vitality, with a direct misfortune in exactness contrasted with the GPS exactness [14]. We likewise assess the impact of the diverse parameters on the vitality exactness tradeoff [14].

2.2 The technique of IOT GPS tracker system

By implementing this real-time IOT base GPS tracker system, the selection of utilized module in a system plays a very important role in order to invent the most efficient way to track the items in a long distance via IOT. GPS module is a very suitable module to be used since it gets the real-time data from the satellite and the only things to be done is find out the best medium to view the location from the data obtained. Besides, the wifi module is also really suitable to be used since the tracker system will be implemented at a location that covers a wifi network. A brief explanation regarding the technique used in the real-time IOT base tracker system is as followed.

2.2.1 Technique 1: System integration with gps module

Figure 2.1 shows that the Ublox GPS module which built-in Arduino UNO Rev3 microcontroller board. The satellite data is sent by satellite as the transmitter and read by GPS module as a receiver. The working operation of Global positioning system depends on the 'trilateration' numerical rule [13]. The position is resolved from the separation estimations to satellites [13]. The four satellites are utilized to decide the position of the receiver on the earth [13]. The objective area is affirmed by the fourth satellite [13]. What's more, three satellites are utilized to follow the area put [13]. A fourth satellite is utilized to affirm the objective area of every one of those space vehicles [13]. The worldwide situating framework comprises of satellite, control station and screen station and recipient [13]. The GPS collector takes the data from the satellite and uses the technique for triangulation to decide a correct position of the tracked items [13]. The microcontroller board which is Arduino UNO then processed the data to get the result.

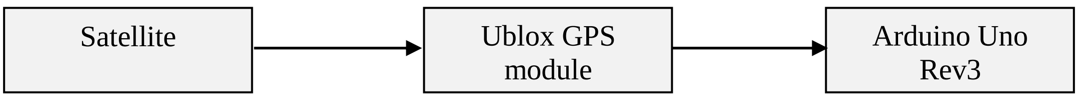


Figure 2.1: Work operation of GPS.

2.2.2 Technique 2: System integration with wifi module

Figure 2.2 shows that that the data stored in Arduino memory will be sent to the cloud database via wifi. Wi-Fi is a fast web association and system association without utilization of any links or wires [14]. The remote system is working three basic components that are radio signs, reception apparatus and switch [14]. The radio waves are keys which make the Wi-Fi organizing conceivable [14]. Wi-Fi similarity has been utilizing another creation to constituent inside the ground associated with group organize [14]. The genuine communicate is associated with an arrangement in actuality it is finished by a method for stereo framework surf and additionally the value of wires with a screen to characterization inclined [14]. Wi-Fi enables the individual keeping in mind the end goal to gain admittance to the web wherever in the genuine gave region [14]. This enables the users to produce a framework inside Resorts, library, schools, universities, grounds, individual foundations, and additionally coffee stores and also on the open spot to make the organization substantially more lucrative and in addition communicate with their own client at whatever point [14]. Wi-Fi similarity can influence surf with gaze to the organization to utilizing their rousing digital TV much a littler sum drive down [14]. The radio signs are transmitted from reception apparatuses and switches that signs are gotten by Wi-Fi collectors [14]. At whatever point the PC gets the signs within the scope of 100-150 feet for the switch it associates the gadget quickly [14]. The scope of the Wi-Fi relies on nature, indoor or outside extents [14]. The speed of the gadget utilizing Wi-Fi association increments as the PC draws nearer to the fundamental source and speed is diminished PC escapes [14].

Figure 2.2: Working operation of wifi module

2.3 Introduction of Arduino Uno Rev3 microcontroller board

Arduino Uno is a microcontroller board in perspective of the ATmega328P. It has 14 propelled data/yield pins (of which 6 can be used as PWM yields), 6 basic information sources, a 16 MHz quartz valuable stone, a USB affiliation, a power jack, an ICSP header and a reset get. It contains everything anticipated that would help the microcontroller; essentially interface it to a PC with a USB connection or power it with an AC-to-DC connector or battery to start. "Uno" infers one in Italian and was meant the landing of Arduino Software (IDE) 1.0. The Uno board and frame 1.0 of Arduino Software (IDE) were the reference variations of Arduino, now progressed to additional state-of-the-art releases.

The Arduino Uno has a resettable polyfuse that shields PC's USB ports from shorts and overcurrent. Yet most PCs give their own inside affirmation, the breaker gives an extra layer of security. If more than 500 mA is associated with the USB port, the breaker will normally break the relationship until the point that the short or over-load is ousted.

The Uno shifts from each and every going before board in that it doesn't use the FTDI USB-to-serial driver chip. Or maybe, it incorporates the Atmega16U2 (Atmega8U2 up to variation R2) modified as a USB-to-serial converter.

The Arduino Uno board can be controlled through the USB affiliation or with an external power supply. The power source is picked normally. External (non-USB) power can come either from an AC-to-DC connector (divider wart) or battery. The connector can be related by ceasing a 2.1mm concentration positive associate with the board's vitality jack. Leads from a battery can be installed in the GND and Vin stick headers of the POWER connector. The board can chip away at an external supply from 6 to 20 volts. If gave under 7V, regardless, the 5V stick may supply under five volts and the block may wind uncertain. In case using more than 12V, the voltage controller may overheat and hurt the board. The endorsed go is 7 to 12 volts. The power pins are according to the accompanying:

i. Vin. The data voltage to the Arduino/Genuino board when it's using an outside power source (rather than 5 volts from the USB affiliation or other oversaw control source).
ii. 5V.This stick yields a coordinated 5V from the controller on the board. The board can be given control either from the DC control jack (7 - 12V), the USB connector (5V), or the VIN stick of the board (7-12V). Giving voltage by methods for the 5V or 3.3V pins evades the controller, and can hurt the board.
iii. 3V3. A 3.3 volt supply created by the on-board controller. Most prominent current draw is 50 mA.
iv. GND. Ground pins.
v. IOREF. This stick on the Arduino/Genuino board gives the voltage reference which the microcontroller works. A truly planned shield can read

the IOREF stick voltage and select the best possible power source or enable voltage translators on the respects work with the 5V or 3.3V.

The ATmega328 has 32 KB (with 0.5 KB required by the bootloader). It in like manner has 2 KB of SRAM and 1 KB of EEPROM (which can be scrutinized and created with the EEPROM library).

Each one of the 14 propelled sticks on the Uno can be used as an information or yield, using pinMode(), digitalWrite(), and digitalRead() limits. They work at 5 volts. Each stick can give or get 20 mA as recommended working condition and has an inward draw up resistor (withdrew as usual) of 20-50k ohm. A most extraordinary of 40mA is the regard that must not be outperformed on any I/O stick to keep up a vital separation from constant mischief to the microcontroller. Furthermore, a couple of pins have particular limits:

i. Serial: 0 (RX) and 1 (TX). Used to get (RX) and transmit (TX) TTL serial data. These pins are related with the relating pins of the ATmega8U2 USB-to-TTL Serial chip.
ii. Outer Interrupts: 2 and 3. These pins can be intended to trigger a thwart on a low regard, a rising or falling edge, or a modification in regard.
iii. PWM: 3, 5, 6, 9, 10, and 11. Outfit 8-bit PWM yield with the analogWrite() work.
iv. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins reinforce SPI correspondence using the SPI library.
v. Driven: 13. There is a worked in LED driven by cutting edge stick 13. Right when the stick is HIGH regard, the LED is on, when the stick is LOW, it's off.
vi. TWI: A4 or SDA stick and A5 or SCL stick. Reinforce TWI correspondence using the Wire library.

The Uno has 6 straightforward information sources, named A0 through A5, each one of which give 10 bits of assurance (i.e. 1024 novel characteristics). As per normal procedure they measure from ground to 5 volts, anyway is it possible to change the upper end of their range using the AREF stick and the analogReference() work. There are a few distinct sticks on the board:

i. AREF. Reference voltage for the straightforward wellsprings of information. Used with analogReference().
ii. Reset. Pass on this line LOW to reset the microcontroller. Regularly used to add a reset catch to shields which upset the one on the board.

Arduino/Genuino Uno has different workplaces for talking with a PC, another Arduino/Genuino board, or distinctive microcontrollers. The ATmega328 gives UART TTL (5V) serial correspondence, which is open on cutting edge pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial correspondence over USB and appears as a virtual com port to programming on the PC. The 16U2 firmware uses the standard USB COM drivers, and no external driver is required. Regardless, on Windows, an .inf record is required. The Arduino Software (IDE) consolidates a serial screen which empowers clear printed data to be sent to and from the board. The RX and TX LEDs on the board will streak when data is being transmitted by methods for the USB-to-serial chip and USB relationship with the PC (anyway not for serial correspondence on pins 0 and 1).

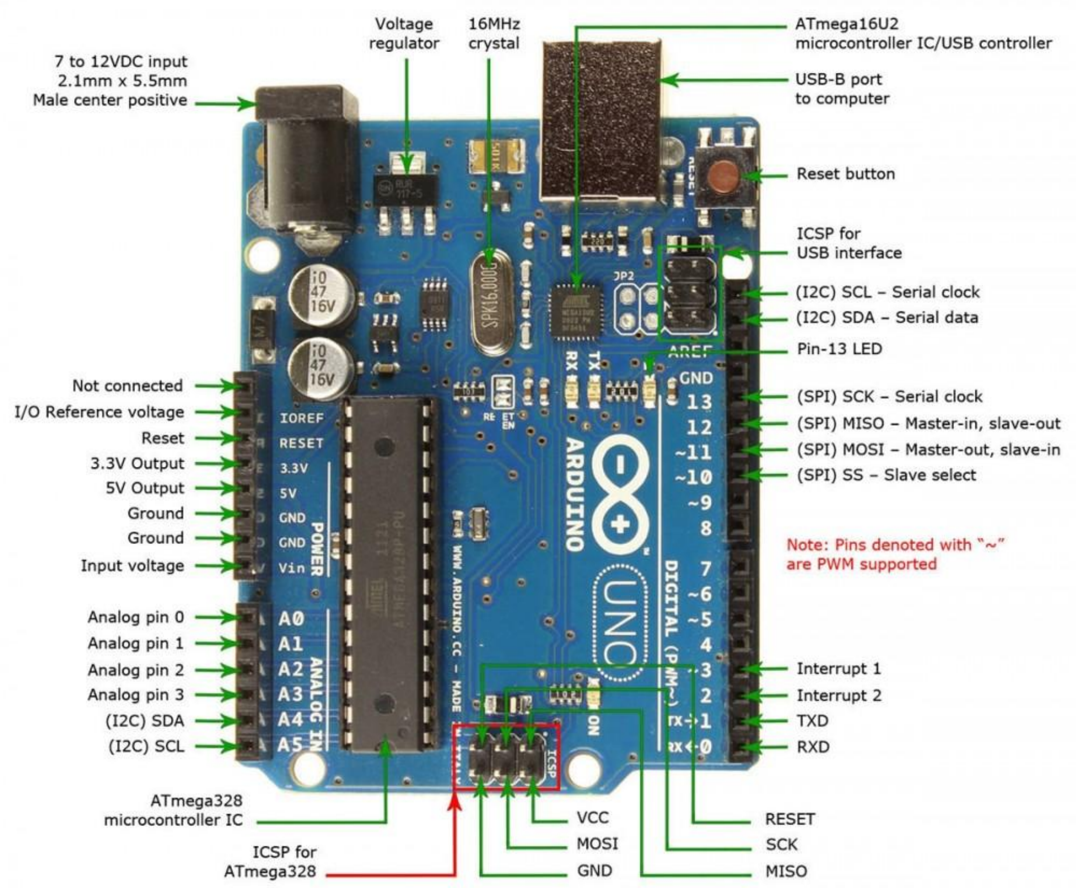


Figure 2.3: Arduino Uno Rev3 microcontroller board

2.3.1 Interface between module and Arduino Uno Rev3 board

The figure 2.4 illustrates above is the module interface with the TX pin and RX pin of the Arduino board. The parameter is based on the position data received by the GPS receiver modules from the satellite. The wifi module then sends the data to the cloud database.

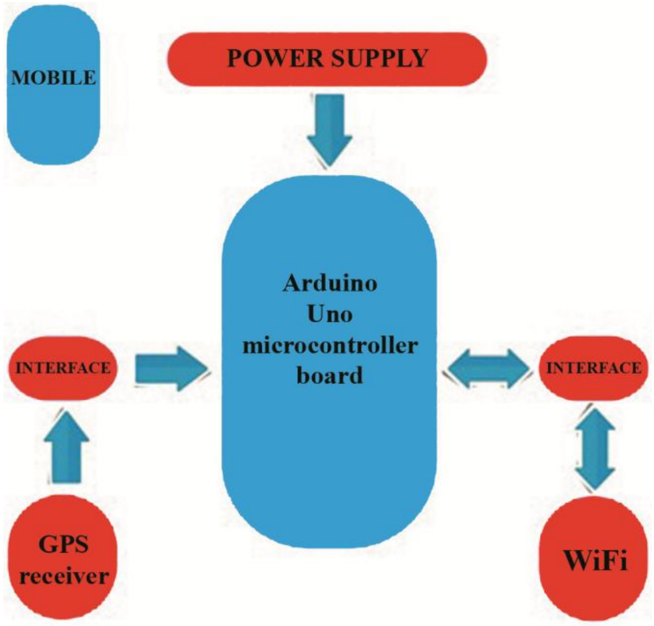


Figure 2.4: Modules interface with Arduino Uno microcontroller board

2.3.2 Arduino IDE Software

Figure 2.5 demonstrates the Arduino IDE downloader webpage. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a content manager for composing code, a message zone, a content support, a toolbar with catches for regular capacities and a progression of menus [16]. It associates with the Arduino and Genuino equipment to transfer programs and speak with them [16]. Projects composed utilizing Arduino Software (IDE) are called draws [16]. These portrayals are composed in the word processor and are spared with the record augmentation .ino [16]. The editorial manager has highlighted for cutting/sticking and for seeking/supplanting content [16]. The message territory gives criticism while sparing and trading and furthermore shows mistakes [16]. The support shows content yield by the Arduino Software (IDE), including complete blunder messages and other data [16]. The base righthand corner of the window shows the arranged board and serial port [16]. The toolbar catches enable clients to confirm and transfer programs, make, open, and spare draws, and open the serial screen [16].

Figure 2.5: Arduino IDE downloader site

Figure 2.6 demonstrates an Arduino IDE programming. Here is the place the code is run and transfer into the advancement board.

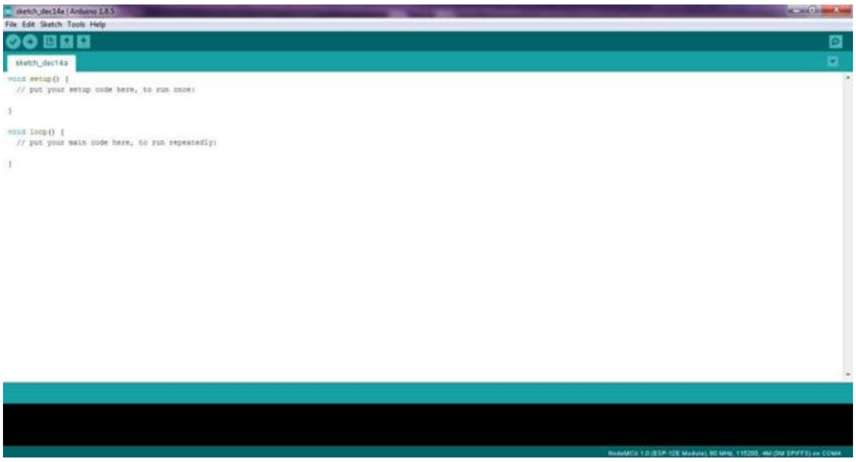


Figure 2.6: Arduino IDE software

2.3.3 MIT App Inventor 2 Software

Figure 2.7 shows the MIT App Inventor. Application Inventor for Android is an open- source web application initially gave by Google, and now kept up by the Massachusetts Institute of Technology (MIT) [17]. It enables newcomers to utilize C syntax programming to make programming applications for the Android working framework (OS). It utilizes a graphical interface, fundamentally the same as Scratch and the StarLogo TNG UI, which enables clients to relocate visual articles to make an application that can keep running on Android gadgets. In making App Inventor, Google drew upon huge earlier research in instructive registering, and additionally work done inside Google in online improvement environments. Application Inventor and the tasks on which it is based are educated by constructionist learning speculations, which underscores that programming can be a vehicle for connecting with intense thoughts through dynamic learning. All things considered, it is a piece of a continuous development in PCs and instruction that started with crafted by Seymour Papert and the MIT Logo Group in the 1960s and has additionally shown itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo. MIT App Inventor is likewise upheld with the firebase database expansion. This enables individuals to store information on google's firebase. The Android software is developed using MIT App Inventor.

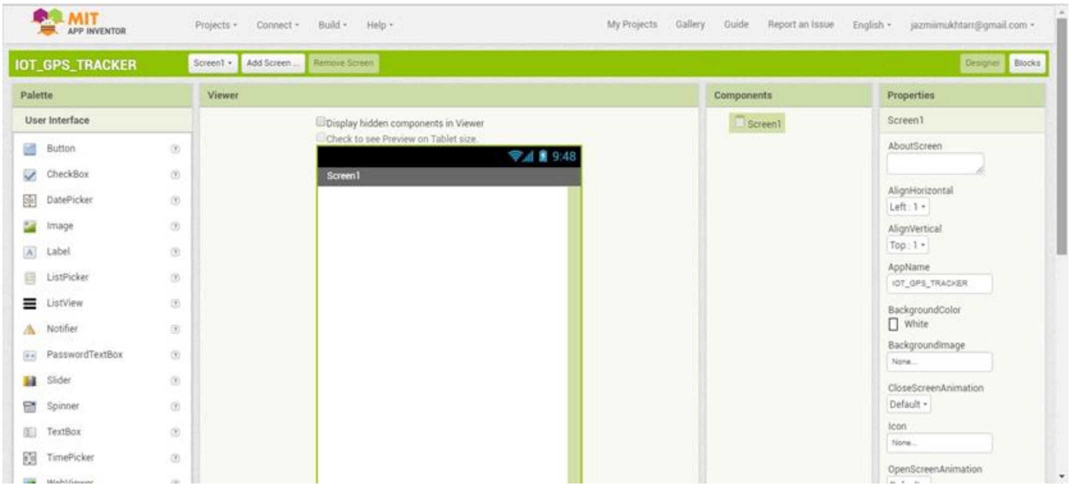


Figure 2.7: MIT App Inventor 2 online software

2.3.4 Thingspeak database

Figure 2.8 demonstrates the Thing Speak online database programming. "ThingSpeak is an open source Internet of Things (IoT) application and API to store and recuperate data from things using the HTTP tradition over the Internet or by methods for a Local Area Network [16]. ThingSpeak engages the arrangement of sensor logging applications, region following applications, and a casual association of things with takes note" [16]. ThingSpeak was at first moved by ioBridge in 2010 as an organization with the help of IoT applications [16]. ThingSpeak has facilitated assistance from the numerical handling programming MATLAB from MathWorks [16]. Empowering ThingSpeak customers to analyze and envision exchanged data using Matlab without requiring the purchase of a Matlab allow from Mathworks [16]. ThingSpeak has a comfortable relationship with Mathworks, Inc [16]. Honestly, most of the ThingSpeak documentation is intertwined into the Mathworks' Matlab documentation website page and despite enabling selected Mathworks customer accounts as real login accreditations on the ThingSpeak webpage [16]. The terms of organization and security approach of ThingSpeak.com are between the agreeing customer and Mathworks, Inc. ThingSpeak has been the subject of articles specifically "Designer" destinations like Instructables, Codeproject, and Channel 9.

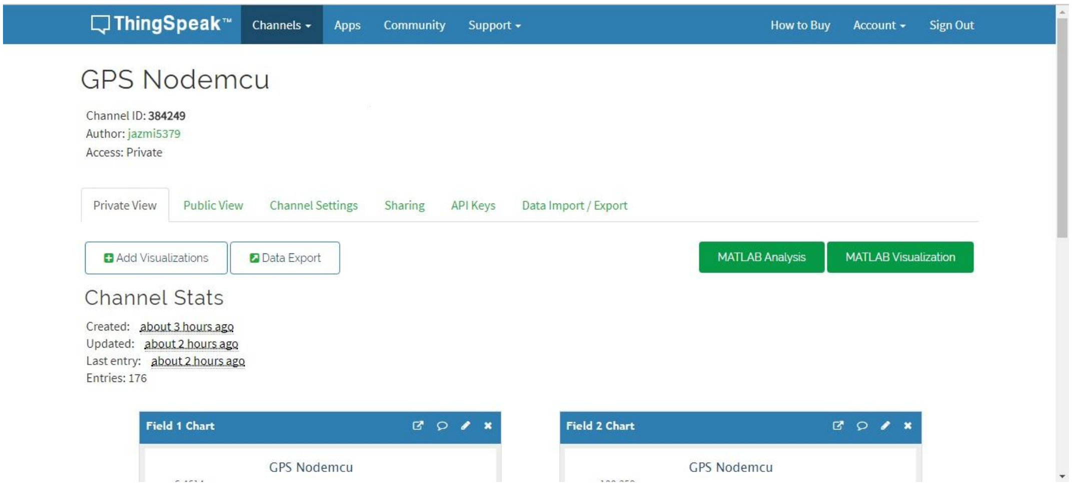


Figure 2.8: Thingspeak online database

2.4 Research of existing GPS Tracker System

Some research regarding the existing tracker system in terms of coverage of transfer media have been done by referring to some trusted resources such as a journal. The power consumption has the data transfer module also carried out using the same method.

2.4.1 Coverage range of wifi network and gsm network

From table 2.1 we can see that GPRS cover the largest range of network while wifi covers the lowest range of the network. However, the transmitter of GPRS and gsm to their receiver respectively is impossible to be very close to implementing such as both transmitter and receiver cannot be in the vehicle while wifi is possible to do so.

Tracking system	Utilize Network	Coverage Range
RF Control Based Mobile Robotic System for Search Mission with GPS Tracker [12]	WiFi	1 kilometers from the transmitter to the receiver
Application of GPS Tracker Technology for Identification of Road Trafficking Conditions [13]	General packet radio service (GPRS)	57 kilometers from the transmitter to the receiver
Energy-Efficient Hybrid GPS-Accelerometer Compass GSM Localization [14]	GSM	20 kilometers from the transmitter to the receiver

Table 2.1: Classification of network by different tracking system

2.5 List of Significant Components

Table 2.2 shows the essential components to build the IOT GPS tracker hardware system. The features and the usage of the components were listed out clearly how its implement the hardware design. Each component above performs a useful individual function.

Type of components	**Specification**	**Function**
 Arduino Uno Rev3	• Microcontroller: ATmega328P • Operating Voltage: 5V • Input Voltage (recommended): 7-12V • Input Voltage (limit): 6-20V • Digital I/O Pins: 14 (of which 6 provide PWM output)	Act as the main board to connect all the utilized modules.

	• PWM Digital I/O Pins: 6 • Analog Input Pins: 6 • DC Current per I/O Pin: 20 mA • DC Current for 3.3V Pin: 50 mA • Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader • SRAM: 2 KB (ATmega328P) • EEPROM: 1 KB (ATmega328P) • Clock Speed: 16 MHz • LED_BUILTIN: 13 • Length: 68.6 mm • Width: 53.4 mm • Weight: 25 g • Refer Appendix B for the details of Arduino Uno board.	
 ESP8266 WiFi module	• Operating Voltage: 3.3V • Wireless Standard: 802.11 b/g/n 2.4GHz • Integrated: TCP/IP protocol stack • Interface: RS232(TTL level, 3.3V for 'H', 0V for 'L') • Default baud rate: 115200bps • No. of GPIO: 9 • Refer Appendix C for pin assignment.	Act as the data transfer module to the cloud data base.

 GY-NEO6MV2 GPS module	• Operating Voltage : 3.3v - 5v • Model : GY-NEO6MV2 /GY-GPS6MV2 • Build in EEPROM for saving data when powered off • Baud rate : 9600 • Backup Battery • LED Signal indicator • Antenna Size : 25*25mm • Module Size : 25mm * 35mm • Mouting Holes on Board : M3 size *4	Act as the GPS receiver to get the data in terms of latitude and longitude from the satellite.

Table 2.2: List of significant components

2.6 Conclusion

The IOT GPS Tracker system can be implemented through an Arduino Uno rev3 microcontroller board with the coding upload to the board and interface with the modules to get the data of the tracked items and transfer the data to the database. The Thingspeak database also created online to receive the data via wifi which then can be sent to the mobile application which is MIT App Inventor 2.

Chapter 3
Methodology

3.1 Flow Chart

The research methodology can be divided into four main phases.

PHASE I: Literature Review

In this phase, the survey of tracking system has been done in order to understand the requirement of the existing tracking system nowadays.

PHASE II: Designing and modification development

The source code will be created using Arduino software to interface with the hardware implementation. The code also created in the MIT app inventor online software for the mobile application of the real-time IOT base GPS tracker system. At the same time, the hardware part of the system will be developed.

PHASE III: Implementation and verification of correctness

The modified source code will be tested with hardware part in order to verify its correctness.

PHASE IV: Documentation

In this phase, documentation of this project will be completed. These four phases are shown in Figure 3.1 to illustrate its relation. This project will be carried out by following the flowchart in figure 3.1. In order to accomplish the system, there are few testing be carried out to make sure the system is working smoothly.

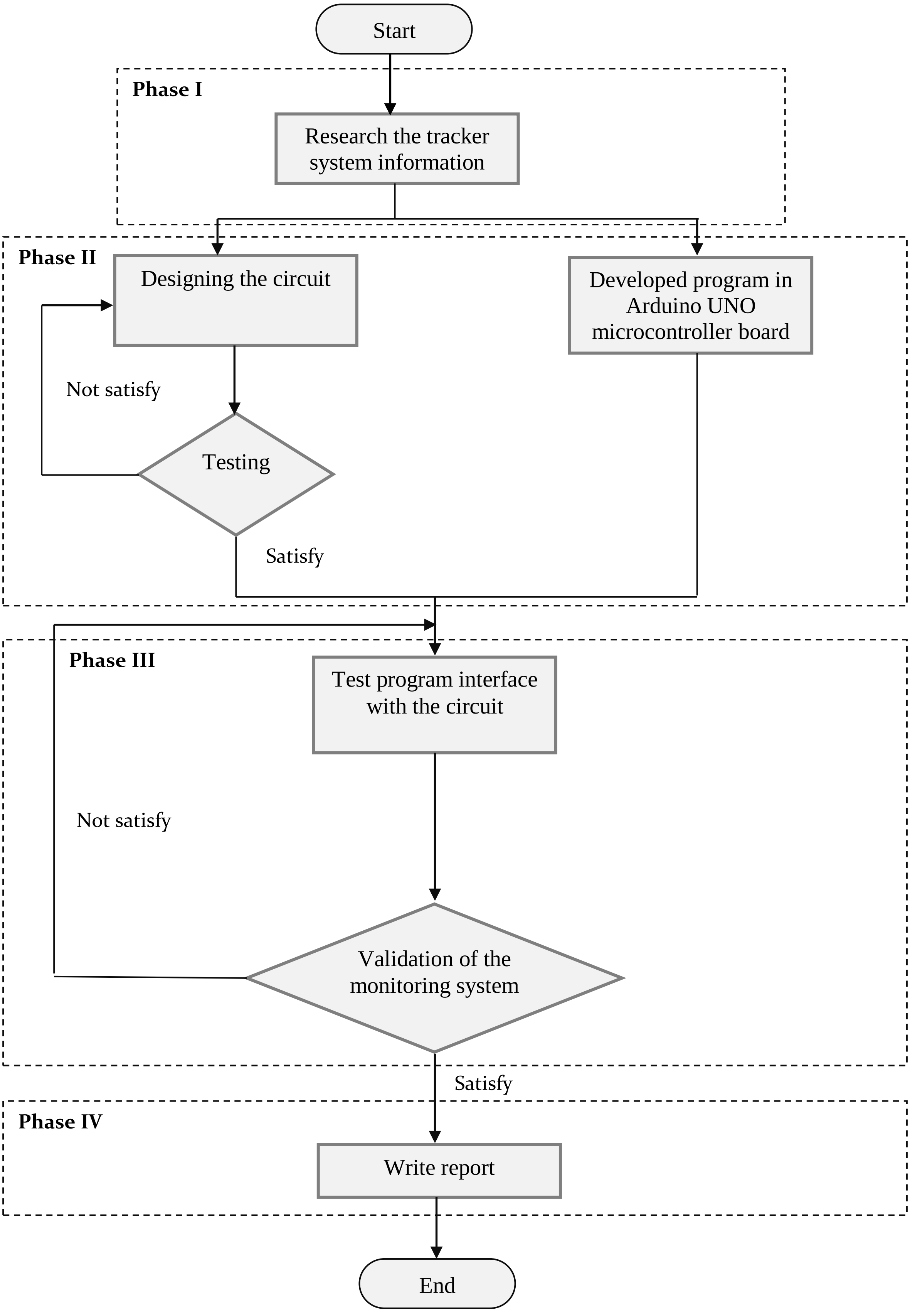


Figure 3.1: The overall project flow.

3.2 Design of Internet of Thing based GPS Tracker System

There are two major parts to be designed in this project. The parts included hardware design part and software design part. Both of this part is very important to make sure the IOT GPS tracker system can be implemented successfully. The hardware tracker system will be placed in the cargo trucks which want to be tracked while the software tracking system will be used to view the real-time location of the tracked items.

3.2.1 Hardware design

Figure 3.2 shows the flowchart of hardware tracker system. The Arduino is connected to the power supply. When the Arduino is turned on, all the module which is WiFi module and GPS module will also be turned on. If the Arduino still turns off, check the power supply which is lipo battery. The module can be confirm to be turn on by observing the led at both module. If the led is blinking, those modules are turn on. If the led still turn off at both modules, check the connection of both modules to the Arduino power source. Then the location in terms of latitude and longitude can be check at the serial monitor of the Arduino software. If the location appear at the serial monitor in real time, the GPS module is actually working properly. Then we can check the wifi module whether it is transferring data or not by observing the Thingspeak database. If the data of latitude and longitude appear in a real-time, the wifi is working properly.

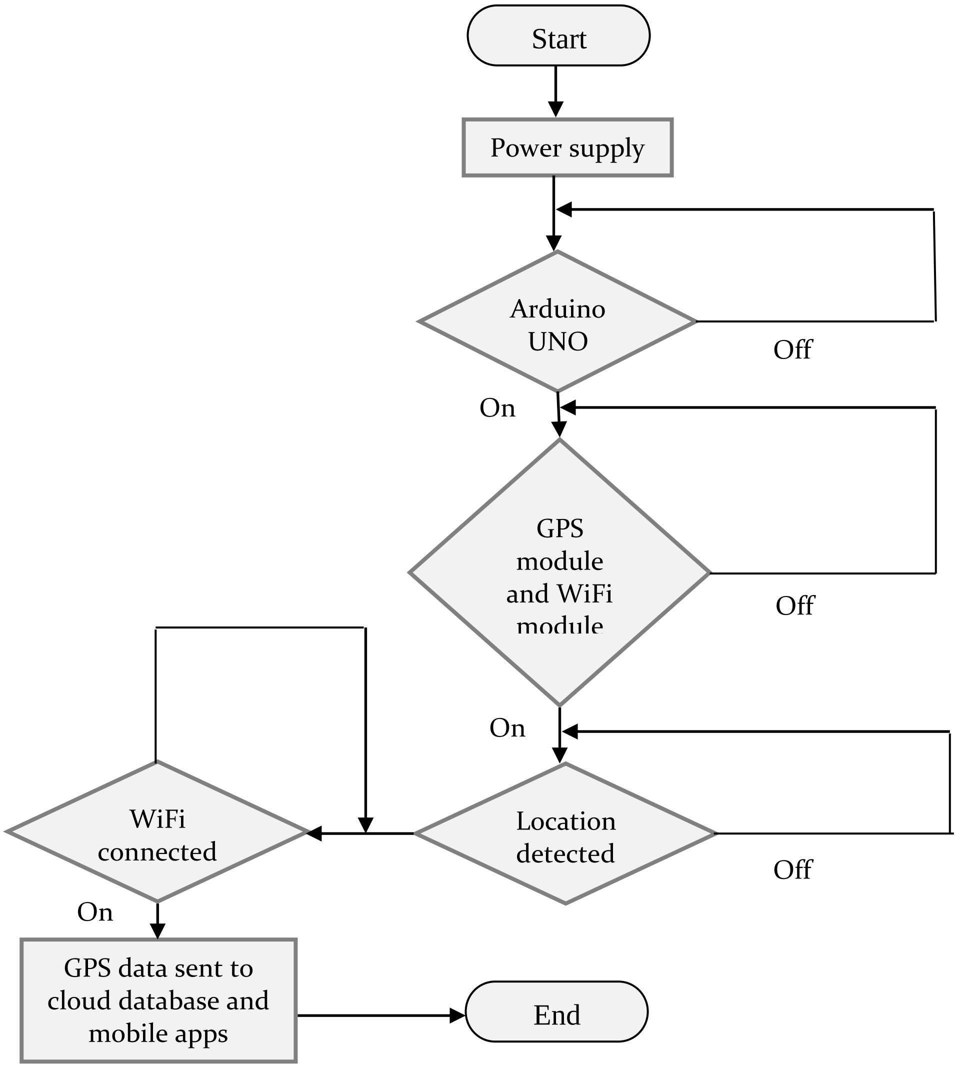


Figure 3.2: Flowchart of the hardware system

Figure 3.3 shows the Block diagram of the internet of thing based GPS tracker system hardware. It consists of one microcontroller board which is Arduino Uno Rev3 and two modules used in this design which is ESP8266 WiFi module and GY-NEO6MV2 GPS module. Table 3.1 shows the pin assignment of Arduino with wifi module and GPS module.

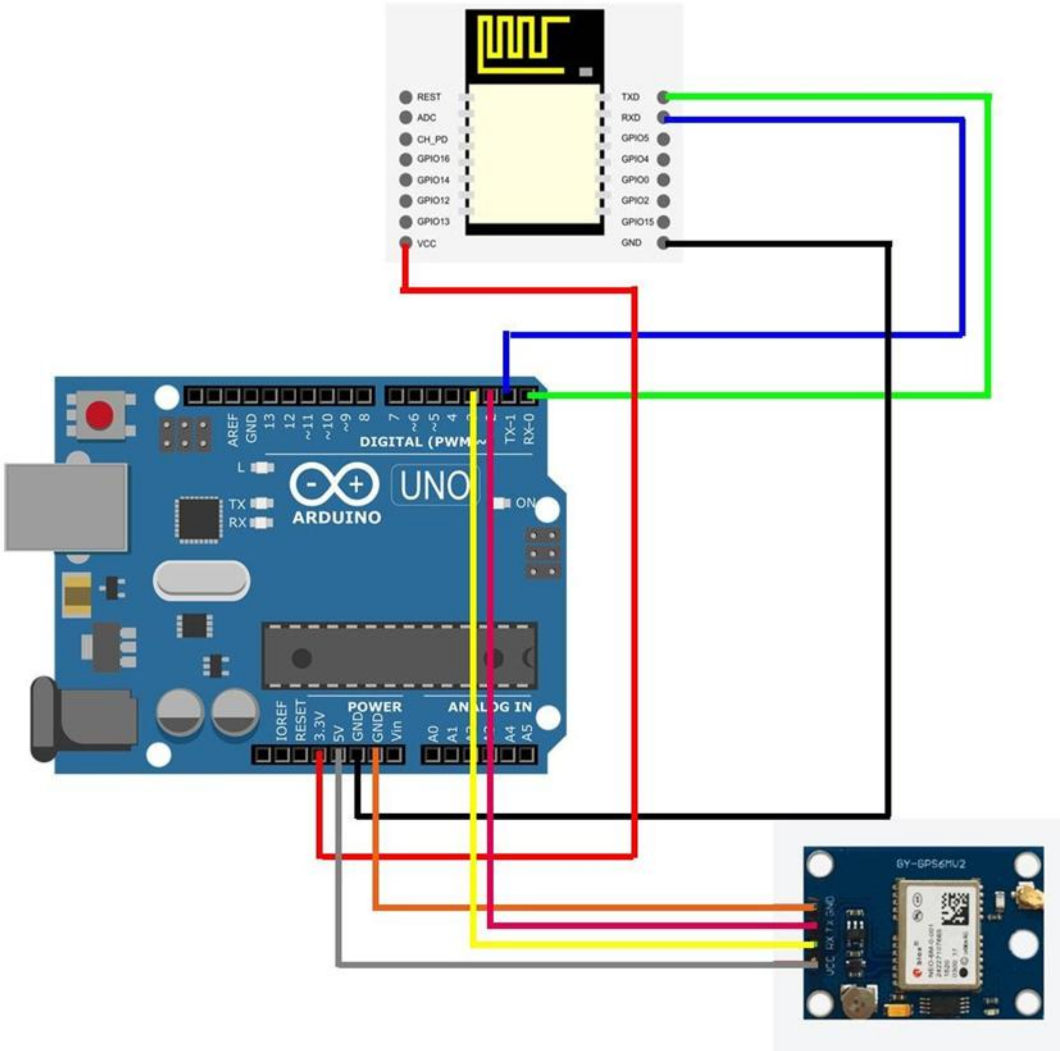


Figure 3.3: Block diagram of internet of thing based gps tracker system hardware

Arduino microcontroller board	ESP8266 WiFi module	GY-NEO6MV2 GPS module
GND	GND	GND
5V	-	VCC
3V3	VCC	-
TX	RX	-
RX	TX	-
D2	-	TX
D3	-	RX

Table 3.1: Pin assignment for arduino Uno microcontroller with ESP8266 Wifi module and GY-NEO6MV2 GPS module

3.2.2 Software design

The software of this tracker system is created using MIT App Inventor 2 online software purposely to monitor the location of the item in real-time. This software consists of two part which is design and block. The design is the basic part of this software where the user can choose what to be put in their mobile application. There is a lot of interfaces that user can put in their software such as image, text label, textbox, password textbox and many more.

Based on figure 3.4, for IOT GPS Tracker system mobile application, ten labels and two buttons inside the user interface palette are utilized. The first label is used to label "STARTING POINT", second label is used to label GPS, third label is used to label value of latitude, fourth label is used to label the comma to separate latitude and longitude value, fifth label is used to label the longitude, sixth label is used to label the "CURRENT LOCATION" and seventh to the eighth label is as same as the third to fifth label. The latitude and longitude value will change base on the code that will be made during the block code part of this software.

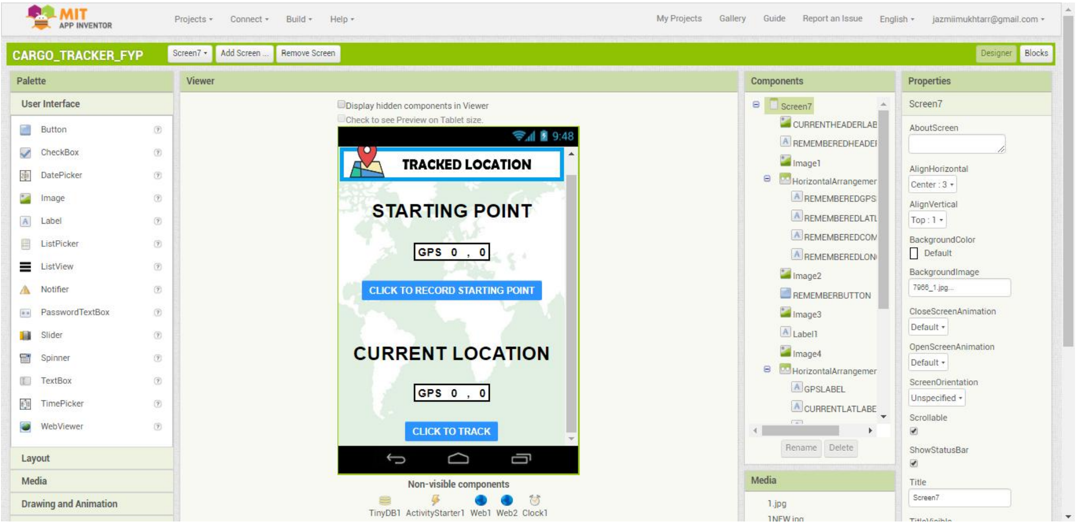


Figure 3.4: Design part in MIT App Inventor 2 online software for IOT GPS Tracker System

Besides, the two button utilizes in this application is a set the starting point button and the second button purposely to view the route from the starting point to the real-time tracked location. Figure 3.5 until figure 3.9 below show how the code is created to make this mobile application run its functionality as desired. Based on figure 3.5 above, the code start when the screen initialize. When the screen initializes, the value of latitude and longitude of starting point will be based on the current value of latitude and longitude detect by the hardware.

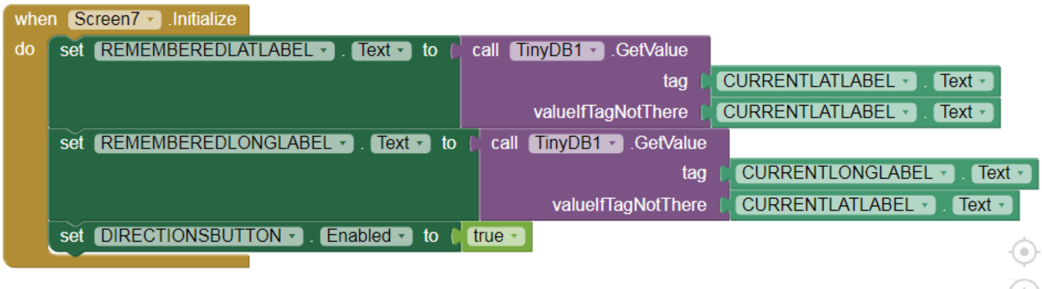


Figure 3.5: First part of block in MIT App Inventor 2 online software for IOT GPS Tracker System

Based on figure 3.6, when web 1 got text data, the will be inserted to the current latitude value. Then it will enable the remembered button which to set the starting point. Next, when web 2 got text data, the data will be inserted to the current longitude value. Then the remembered button to set the starting point will be enabled.

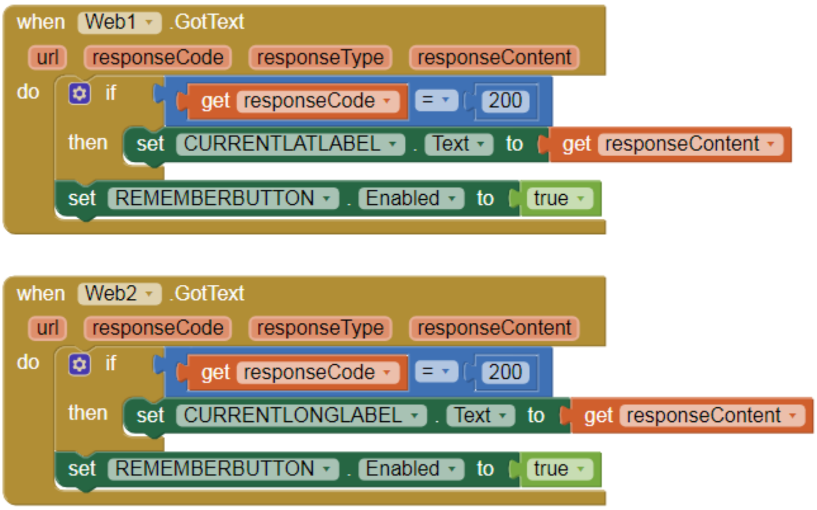


Figure 3.6: Second part of block in MIT App Inventor 2 online software for IOT GPS Tracker System

Based on figure 3.7, when the remembered button is clicked, it will take the data from the current latitude and longitude tracked previously. The directions button is then set to enable.

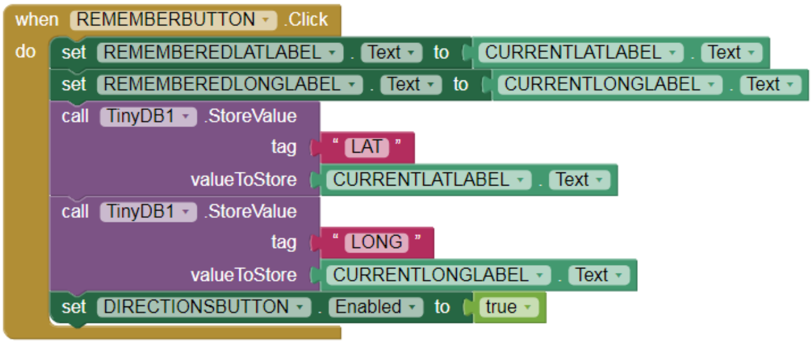


Figure 3.7: Third part of block in MIT App Inventor 2 online software for IOT GPS Tracker System

Based on figure 3.8, when direction button is clicked, it will be linked to google map. It is based on the intent that we declare inside the code. The "android.intent.action.VIEW" shows a list of the considerable number of notes under "content://com.google.provider.NotePad/notes", which the client can browse through and see the points of interest on. By referring to the above figure, it will bring the users to the google map. The "&daddr=" is a command to show the direction of two different points in google map.

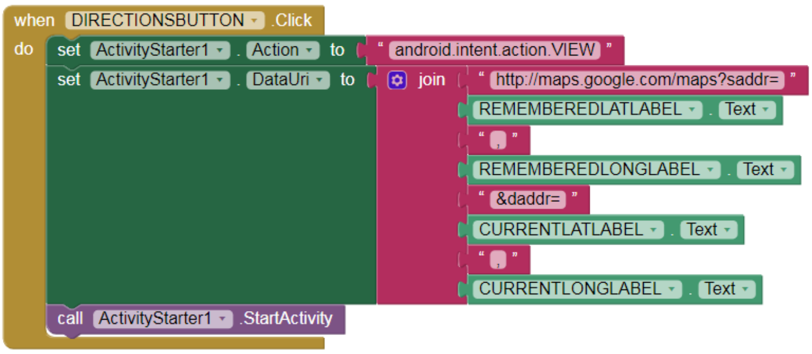


Figure 3.8: Fourth part of block in MIT App Inventor 2 online software for IOT GPS Tracker System

Based on figure 3.9, web1 and web2 are direct to the field1 and field2 respectively inside the thingspeak database. It will get the latest data from the thingspeak.

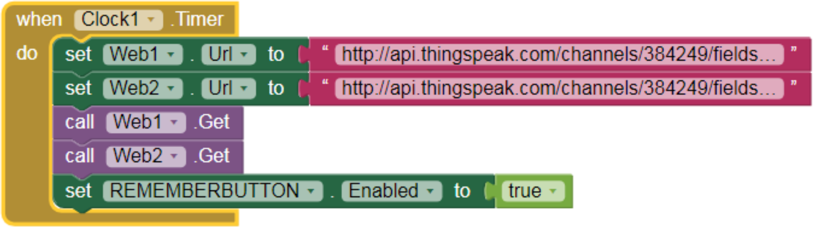


Figure 3.9: Fifth part of block in MIT App Inventor 2 online software for IOT GPS Tracker System

3.2.3 Overall System Design

The overall system design of IOT GPS Tracker system for cargo goods is shown in figure 3.10. The satellite sends the data to the GPS receiver module. When the data process by the microcontroller board, the data will be sent to the cloud database via wifi. The data is in term of latitude and longitude. The data from the database is then sent to the mobile application, to view the real-time location of the tracked items.

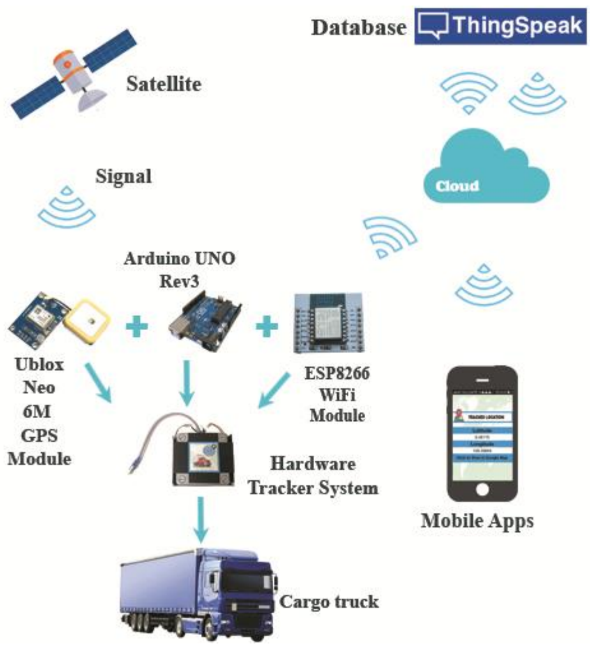


Figure 3.10: System design of IOT GPS Tracker system for cargo goods

3.3 Evaluation

Evaluation will be done by checking the output of a system which responds to the feedback and determines whether it is expected output with any condition. If the result is not satisfactory, the source code will be improved until it meets the requirement.

3.4 Conclusion

In conclusion, the IOT based GPS Tracker system is designed for the usage of cargo trucks that bring the company assets and the truck must have a wifi signal. The design is built unique and special with the ability to track the route of the tracked items from the starting point. This invention will make it easier for a company member to track their assets.

Chapter 4
Results and discussion

4.1 Introduction

This chapter discusses the result obtained from the software and hardware development. For the hardware implementation, the result will be based on the modules build which related to the power, current, and voltage consumed. Moreover, the software implementation is built with Arduino programming which simulates and compile using Arduino software. Besides, the mobile application software is built online using MIT App Inventor 2 software.

4.2 Testing

Testing of the project is done to troubleshoot the tracker system circuit which consists of GY-NEO6MV2 GPS module and ESP8266 WiFi module and makes sure the circuit is functioning. The connection of every single component must be in the correct way. Then the input of the modules must be interfaced with the output port of the Arduino Uno Rev3 microcontroller board and the feedback of the modules required to send to the cloud database and mobile application so that the output is able to be monitor.

4.2.1 Testing result 1: Tracked hardware viewed in mobile application

Figure 4.1 shows the mobile application for the system while figure 4.2 shows the hardware used in this system. By referring to the mobile application shown in figure 4.1, when the "CLICK TO RECORD STARTING POINT" button is clicked, the value of latitude and longitude from the current location will be saved. The value of latitude and longitude of the current location will always change in real-time based on the latest value obtained inside the database. When the "CLICK TO TRACK" button is clicked, the mobile application will directly link to google maps. It will show the direction from the starting point to the current location tracked in a real-time. Based on figure 4.1, the latitude and longitude of the starting point are 6.46260 and 100.35277 respectively while the latitude and longitude of the current location are 6.42974 and 100.28692 respectively. Google map image in figure 1 is the sample result tested from UniMAP main campus to Arau, Perlis.

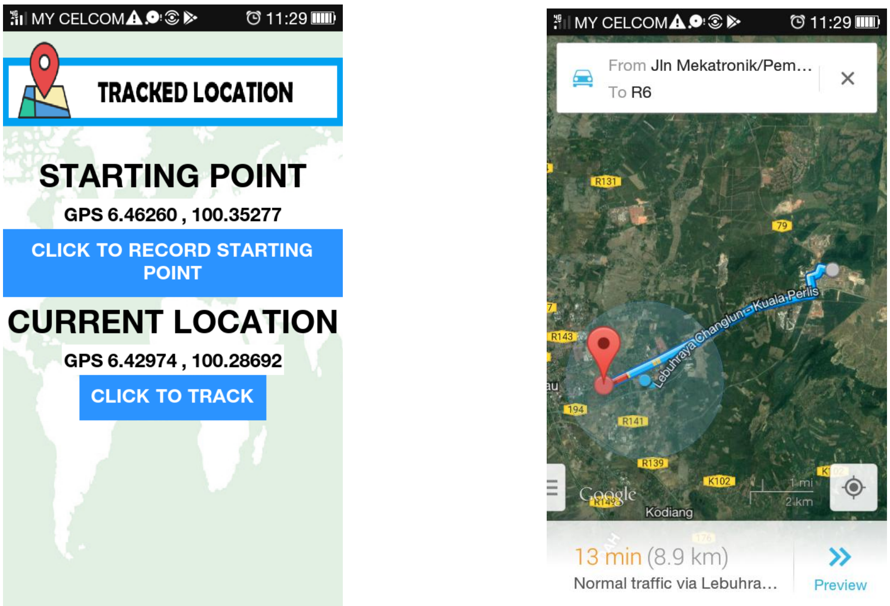


Figure 4.1: Mobile application for internet of thing based GPS Tracker System

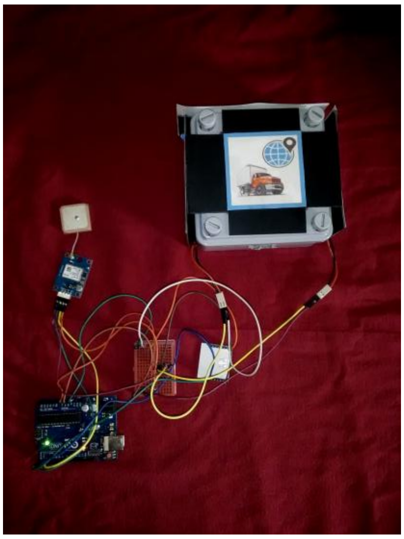
 

Figure 4.2: Hardware for internet of thing based GPS Tracker System

4.2.2 Testing result 2: Tracked hardware viewed in Google Earth software

Figure 4.3 shows the tracked location generate using Google Earth software. The longitude and latitude data are obtained from the Thingspeak database. The data is in the CSV format. It then converted to the kmz format using online file converter software which is GPS Visualizer. The data then uploaded into the Google Earth software manually. The test location is from UniMAP main campus to the Arau, Perlis.

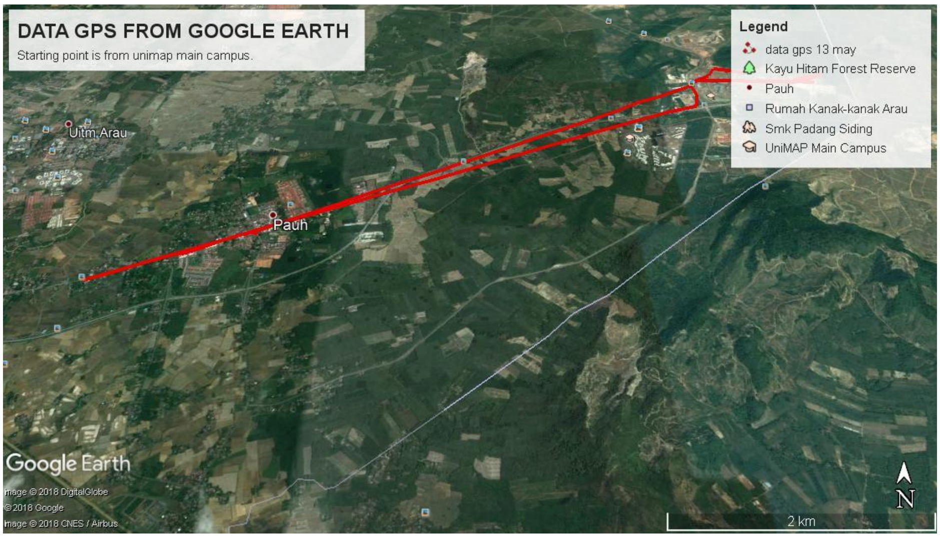


Figure 4.3: Tracked location generate using Google Earth software

4.3 Conclusion

The results of the experiment conducted to obtain the location from starting point to the real-time location of the tracked items can be view using mobile apps which can be linked to google map. The direction command is used during designing the code in MIT App Inventor 2 software to enable the items tracked by using the route from starting point to end point.

Besides, the users also able to track each point of tracked location by manually view it in a Google Earth software. The data of the GPS can be obtained from the thingspeak database and converted to the suitable file which is kmz file using online converter which is GPS Visualizer.

In addition, the reason why WiFi module is selected to be used in this system as compared to GSM module is that the WiFi module consumes less power which makes the battery used as a power supply to stand longer and can support during the long journey.

Chapter 5
Conclusion and future work

5.1 Summary

This project is mainly about the internet of thing global positioning tracking system to enable the user monitor their assets in a long distance. In general this system provide the GPS system with a WiFi as it transfer media. This system consist of hardware tracker system that connected via WiFi to the mobile application for the tracking system. This system utilizing the internet of thing concept which connected a data sensed by hardware and send back to the database and the mobile application.

This system mainly automated using Arduino. Arduino configuration is normally specified using arduino programming. The utilize modules is connected first to the microcontroller board after uploading the sketch. This purposely to ensure that the functionality of the system meets all the specification of IOT based hardware tracker system.

Next, the data obtained by the GPS module from the satellites is processed by the microcontroller and send to the database which is Thingspeak via wifi. In the database, the collected data is in term of latitude and longitude. The latest data in the database will be sent to the mobile apps in real-time.

The system is based on the design for usage of cargo trucks that bring the company assets inside it which have the wifi connection inside it. This system is analysed to select the best data transfer media to get the optimized power consumption to be used in this tracker system.

The objective of this project has been achieved which to implement an internet of thing based GPS tracking system which consist of GPS module and wifi module as a main component of the system. Besides, this project which purposely to develop an android software for IoT based GPS tracking system which enable a user to track the location of their goods and facilities through their smartphone have also been achieved. Development of this project which to produce and integration of software and hardware that achieve a desired result which can view the location of tracked items in the smartphone apps have also been achieved. The final objective of this project which to track the location of moving items in a long distance also has been achieved.

5.2 Future work

In general, a tracking system in Malaysia still does not utilize the internet of thing concept to track the location of the cargo truck to track the assets own by a company. Means that the existing tracker nowadays unable to view the route from the starting point to the real-time end point of the tracked items. However, there is a lot of thing to be improved in future to make this tracking system become more efficient to be used.

5.2.1 Tracking module

Currently utilize tracker sensor modules might cover only limited area and the data might not really precise. So for the future improvement the gps system will be used a better gps chip which make the detect location become more precise and the system indirectly become more efficient.

5.2.2 Data transfer module

Currently used data transfer module is ESP8266 wifi module. In the next stage of improvement, the system will be utilizing a more powerful wifi module so that the rate of data transfer to the database become more faster. This indirectly make this internet of thing base gps tracker system become more efficient to be used.

5.2.3 Mobile application software

Currently built mobile application only able to view the real-time location of the items from it starting point. For the future improvement, the history of the tracked items in term of location and time can be viewed using the mobile application.

5.2.4 Hardware improvement

Currently developed hardware is utilizing Arduino Uno microcontroller board making the hardware become quite large. For the future improvement, arduino mini pro will be consider to be used since it is much smaller and consume less power as compared to the currently utilized microcontroller board.

Appendix A System Coding

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 16, 2);

#define DEBUG FALSE //comment out to remove debug msgs

//*-- Hardware Serial
#define _baudrate 115200

//*-- Software Serial
//
#define _rxpin      2
#define _txpin      3

//*-- IoT Information
#define SSID "OPPO R831L"
#define PASS "12345678"
#define IP "api.thingspeak.com" // ThingSpeak IP Address: 184.106.153.149

long lat, lon;
float flat, flon;
int speed_gps;
unsigned long age, date, time, chars;
int year;
byte month, day, hour, minute, second, hundredths;
```

```
unsigned short sentences, failed;
String slat = "";
String slon = "";
String TempStringVal = "";                //Temporary String Val


int nrf_digital = 6;
int indic = 7;
int nrf_val = 0;
int state = 0;


SoftwareSerial debug( _rxpin, _txpin ); // RX, TX
SoftwareSerial mySerial(2, 3);
TinyGPS gps;
void gpsdump(TinyGPS &gps);
void printFloat(double f, int digits = 2);


// GET /update?key=[THINGSPEAK_KEY]&field1=[data 1]&field2=[data 2]...;
String GET = "GET /update?key=ZLEMFAMKZKUOWO2D";


//
String GET_PB = "GET /pushingbox?devid=v4A5EA631E93EC37"; //Token utk
post data ke database bersama API key
#define IP_PB "api.pushingbox.com" //Target URL HOST yang nak connect satgi
<--------------------- Ni nama IP tu



void setup() {
  Serial.begin( _baudrate );
```

```
  debug.begin( _baudrate );
    mySerial.begin(9600);
    pinMode(indic,OUTPUT);
      // initialize the LCD
  lcd.begin();

  // Turn on the blacklight and print a message.
  lcd.backlight();
    delay(1000);
pinMode (nrf_digital,INPUT);
    connectWiFi();
delay(3000);
}



void loop() {

  bool newdata = false;
  unsigned long start = millis();
  // Every 5 seconds we print an update
  while (millis() - start < 15000)
  {
    if (mySerial.available())

    {
      char c = mySerial.read();
      //Serial.print(c);  // uncomment to see raw GPS data
      if (gps.encode(c))
      {
        newdata = true;
        break;  // uncomment to print new data immediately!
      }
```

```
    }
  }
  if (newdata)
  {
    gps.f_get_position(&flat, &flon, &age);
    int speed_gps = gps.f_speed_kmph();
    slat = float2string(flat);
    slon = float2string(flon);
    String speed_string = String(speed_gps);
    lcd.setCursor(0,0);
    lcd.print("lat: ");
    lcd.print(slat);
    lcd.setCursor(0,1);
    lcd.print("long: ");
    lcd.print(slon);
    Serial.print(slat);
    Serial.print(",");
    Serial.println(slon);
    update_location(slat,slon);
     delay(15000);
  }
}

void runpushingbox()
{
  // ESP8266 Client
  String cmd = "AT+CIPSTART=\"TCP\",\"";// Steeing connection wifi ke internet guna TCP connection
  cmd += IP_PB; // "+=" tu dia akan tambah string cmd tu dgn string IP (tgok IP kt ataih)
  cmd += "\",80"; //port 80
  sendDebug(cmd); //Try connect ngn URL host
  delay(2000);
```

```
  if( Serial.find( "Error" ) ) //Kalau tak boleh connect dia akan keluaq error
  {
    debug.print( "RECEIVED: Error\nExit1" );
    return;
  }

  cmd = GET_PB + "&turbidity=" + " HTTP/1.1\r\nHost:
api.pushingbox.com\r\nContent-Length: 0\r\n\r\n"; //Kalau boleh connect ke URL
host(api.pushingbox.com)..Post data yang

//temperature ngn humidity yang dh convert ke string tu

  Serial.print( "AT+CIPSEND=" ); //Send request guna GET method
  Serial.println( cmd.length() ); //Tgok berapa suma lenght URL tu
  if(Serial.find( ">" ) ) //untuk debug
  {
    debug.print(">");
    debug.print(cmd);
    Serial.print(cmd);
  }
  else
  {
    sendDebug( "AT+CIPCLOSE" );//close TCP connection
  }
  if( Serial.find("OK") )
  {
    debug.println( "RECEIVED: OK" );
  }
  else
  {
    debug.println( "RECEIVED: Error\nExit2" );
  }
}
```

```
//----- update the  Thingspeak string with 3 values
void update_location(String slat,String slon)
{
  // ESP8266 Client
  String cmd = "AT+CIPSTART=\"TCP\",\"";// Setup TCP connection
  cmd += IP;
  cmd += "\",80";
  sendDebug(cmd);
  delay(2000);
  if( Serial.find( "Error" ) )
  {
    debug.print( "RECEIVED: Error\nExit1" );
    return;
  }

  cmd = GET + "&field1=" + slat +"&field2="+ slon +"\r\n";
  Serial.print( "AT+CIPSEND=" );
  Serial.println( cmd.length() );
  if(Serial.find( ">" ) )
  {
    debug.print(">");
    debug.print(cmd);
    Serial.print(cmd);
  }
  else
  {
    sendDebug( "AT+CIPCLOSE" );//close TCP connection
  }
  if( Serial.find("OK") )
  {
```

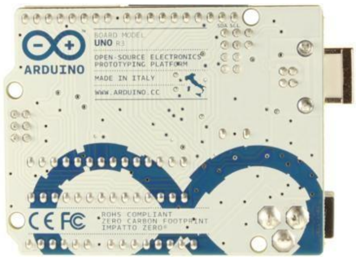
```
    debug.println( "RECEIVED: OK" );
  }
  else
  {
    debug.println( "RECEIVED: Error\nExit2" );
  }
}

void sendDebug(String cmd)
{
  debug.print("SEND: ");
  debug.println(cmd);
  Serial.println(cmd);
}

boolean connectWiFi()
{
  Serial.println("AT+CWMODE=1");//WiFi STA mode - if '3' it is both client and
AP
  delay(2000);
  //Connect to Router with AT+CWJAP="SSID","Password";
  // Check if connected with AT+CWJAP?
  String cmd="AT+CWJAP=\""; // Join accespoint
  cmd+=SSID;
  cmd+="\",\"";
  cmd+=PASS;
  cmd+="\"";
  sendDebug(cmd);
  delay(5000);
  if(Serial.find("OK"))
  {
    debug.println("RECEIVED: OK");
    return true;
```

```
  }
  else
  {
    debug.println("RECEIVED: Error");
    return false;
  }

  cmd = "AT+CIPMUX=0";// Set Single connection
  sendDebug( cmd );
  if( Serial.find( "Error") )
  {
    debug.print( "RECEIVED: Error" );
    return false;
  }
}

  String float2string (float Temp2){
  TempStringVal = "";
  TempStringVal+=String(int(Temp2))+ "."+String(getDecimal(Temp2));
  return TempStringVal;
}

long getDecimal(float val)
{
  int intPart = int(val);
  long decPart = 100000*(val-intPart); //I am multiplying by 100000 assuming that
the foat values will have a maximum of 5 decimal places
  if(decPart>0){delay(1); return(decPart); } //return the decimal part of float
number if it is available
  else if(decPart<0){delay(1); return((-1)*decPart);} //if negative, multiply by -1
  else if(decPart=0){delay(1); return(00); } //return 0 if decimal part of float number
is not available
}
```

Appendix B Arduino Uno Datasheet

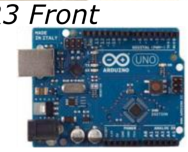
Arduino Uno

Arduino Uno R3 Front



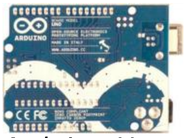
Arduino Uno R3 Back

Arduino Uno R2 Front



Arduino Uno SMD

Arduino Uno Front



Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 Hz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an
ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying

voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools** > **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following halfsecond or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

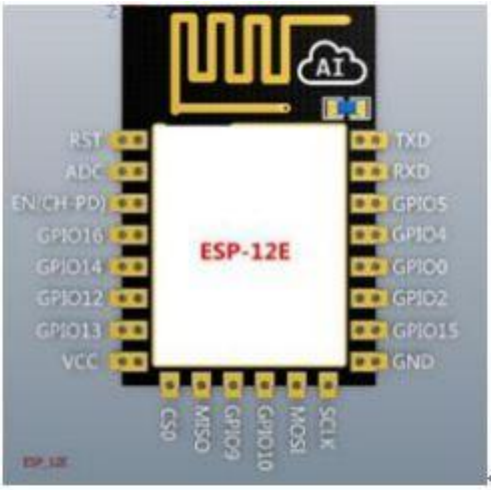
USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Appendix C ESP-12E WiFi Module Pin Description



NO.	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion result.Input voltage range 0-1v,scope:0-1024
3	EN	Chip enable pin.Active high
4	IO16	GPIO16; can be used to wake up the chipset from deep sleep mode.
5	IO14	GPIO14; HSPI_CLK
6	IO12	GPIO12; HSPI_MISO
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input

11	IO9	GPIO9
12	IO10	GBIO10
13	MOSI	Main output slave input
14	SCLK	Clock
15	GND	GND
16	IO15	GPIO15; MTDO; HSPICS; UART0_RTS
17	IO2	GPIO2; UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD; GPIO3
22	TXD	UART0_TXD; GPIO1

REFERENCES

[1] McMinn, "Evaluation of the Trackstick™ Super GPS Tracker for Use in Walking Research", Research Quarterly for Exercise and Sport, 2012.

[2] I. Widyantara and N. Sastra, "Internet of Things for Intelligent Traffic Monitoring System: A Case Study in Denpasar", International Journal of Computer Trends and Technology, vol. 30, no. 3, pp. 169-173, 2015.

[3] "UP Squared Grove IoT development kit", Up Board | Power Up Your Ideas!, 2018. [Online]. Available: http://www.up-board.org/upkits/up-squared-grove-iot-development-kit/. [Accessed: 30- Apr- 2018].

[4] "Why every programmer should learn C", Pluralsight.com, 2018. [Online]. Available: https://www.pluralsight.com/blog/software-development/why-every-programmer-should-learn-c. [Accessed: 30- Apr- 2018].

[5] "Comparison UP versus Edison,Galileo & Joule", Up Board | Power Up Your Ideas!, 2018. [Online]. Available: http://www.up-board.org/up/comparison-up-versus-edisongalileo-joule/. [Accessed: 30- Apr- 2018].

[6] "Arduino Low Power - How To Run ATmega328P For a Year On Coin Cell Battery", Home Automation Community, 2018. [Online]. Available: http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/. [Accessed: 30- Apr- 2018].

[7] "Using the Arduino Pro Mini 3.3V - learn.sparkfun.com", Learn.sparkfun.com, 2018. [Online]. Available: https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v. [Accessed: 30- Apr- 2018].

[8] "Difference between NFC, RFID, bluetooth, wifi-NFC vs RFID vs bluetooth vs wifi", Rfwireless-world.com, 2018. [Online]. Available: http://www.rfwireless-world.com/Terminology/NFC-vs-RFID-vs-Bluetooth-vs-wifi.html. [Accessed: 30- Apr- 2018].

[9] A. Y. Fu, H.-P. Chou, C. Spence, F. H. Arnold, and S. R. Quake, "An integrated microfabricated cell sorter," Analytical Chemistry, vol. 74, pp. 2451-2457, 2002.

[10] A. Chiaradia, H. Kuga and A. de Almeida Prado, "Onboard and Real-Time Artificial Satellite Orbit Determination Using GPS", Mathematical Problems in Engineering, vol. 2013, pp. 1-8, 2013.

[11] "What is latitude and longitude? - Definition from WhatIs.com", WhatIs.com, 2018. [Online]. Available: https://whatis.techtarget.com/definition/latitude-and-longitude. [Accessed: 19- Apr- 2018].

[12] A. Sharma and M. Balamurugan, "RF Control based Mobile Robotic System for Search Mission with GPS Tracker", Indian Journal of Science and Technology, vol. 8, no. 19, 2015.

[13] "How Does a Global Positioning System (GPS) Tracking System Works?", ElProCus - Electronic Projects for Engineering Students, 2018. [Online]. Available: https://www.elprocus.com/how-gps-system-works/. [Accessed: 21- Apr- 2018].

[14] "Wi-Fi (Wireless Technology) Working Principles, Types and Applications", ElProCus - Electronic Projects for Engineering Students, 2018. [Online]. Available: https://www.elprocus.com/how-does-wifi-work/. [Accessed: 21- Apr- 2018].

[15] ESP8266 Arduino Core Documentation. Ivan Grokhotkov, 2017, pp. 5-6.

[16] "Learn More - ThingSpeak IoT", Thingspeak.com, 2018. [Online]. Available: https://thingspeak.com/pages/learn_more. [Accessed: 21- Apr- 2018].

[17] L. Gooch, "For Companies, Risk of Cargo Theft in Asia", Nytimes.com, 2017. [Online]. Available: http://www.nytimes.com/2011/06/23/business/global/23cargo.html. [Accessed: 19 Oct- 2017].

[18] "Cargo Theft From Containers", Nepia.com, 2017. [Online]. Available: http://www.nepia.com/insights/signals-online/cargo/containers/cargo-theft-fromcontainers/. [Accessed: 19- Oct- 2017].

[19] "Economic and Statistics - GPS", Sites.google.com, 2017. [Online]. Available: https://sites.google.com/a/cortland.edu/gps/economic-or-statistics. [Accessed: 19- Oct-2017].

[20] T. Mai, "Global Positioning System", NASA, 2014. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html. [Accessed: 12- Nov- 2017].

[21] T. Mai, "Global Positioning System History", NASA, 2017. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History. html. [Accessed: 12- Nov- 2017].

[22] Blackman, S.S. Multiple-Target Tracking with Radar Applications; Artech House, Inc.: Dedham, MA, USA, 1986; Volume 1, p. 463.

[23] Smith, D.; Singh, S. Approaches to multisensor data fusion in target tracking: A survey.IEEE Trans. Knowl. Data Eng. 2006, 18, 1696–1710.

[24] Luo, R.C.; Yih, C.C.; Su, K.L. Multisensor fusion and integration: Approaches, applications, and future research directions. IEEE Sens. J. 2002, 2, 107–119.

[25] Gui, C.; Mohapatra, P. Power conservation and quality of surveillance in target tracking sensor networks. In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, 26 September–1 October 2004; pp. 129–143.

[26] Benfold, B.; Reid, I. Stable multi-target tracking in real-time surveillance video. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011; pp. 3457–3464.

[27] Nakabo, Y.; Ishikawa, M.; Toyoda, H.; Mizuno, S. 1 ms column parallel vision system and its application of high speed target tracking. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 650–655.

[28] Zhou, K.; Roumeliotis, S.I. Multirobot active target tracking with combinations of relative observations. IEEE Trans. Robot. 2011, 27, 678–695.
[29] Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part I. Dynamic models. IEEE Trans. Aerosp. Electron. Syst. 2003, 39, 1333–1364.
[30] Li, X.; Jilkov, V.P. Survey of maneuvering target tracking. Part II: Motion models of ballistic and space targets. IEEE Trans. Aerosp. Electron. Syst. 2010, 46, 96– 119.
[31] Hue, C.; Le Cadre, J.P.; Pérez, P. Sequential Monte Carlo methods for multiple target tracking and data fusion. IEEE Trans. Signal Process. 2002, 50, 309–325.
[32] McGinnity, S.; Irwin, G.W. Multiple model bootstrap filter for maneuvering target tracking. IEEE Trans. Aerosp. Electron. Syst. 2000, 36, 1006–1012.
[33] Blackman, S.S. Multiple hypothesis tracking for multiple target tracking. IEEE Aerosp. Electron. Syst. Mag. 2004, 19, 5–18.
[34] Vermaak, J.; Godsill, S.J.; Perez, P. Monte carlo filtering for multi target tracking and data association. IEEE Aerosp. Electron. Syst. Mag. 2005, 41, 309–332.
[35] Bar-Shalom, Y. Multitarget-Multisensor Tracking: Advanced Applications; Artech House:Norwood, MA, USA, 1990; Volume 1, p. 391.
[36] Stone, L.D.; Streit, R.L.; Corwin, T.L.; Bell, K.L. Bayesian Multiple Target Tracking; Artech House: Boston, MA, USA, 2013.
[37] "Arduino Uno Rev3", Store.arduino.cc, 2017. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Accessed: 13- Nov- 2017].
[38] ESP8266 Arduino Core Documentation. Ivan Grokhotkov, 2017, pp. 5-6.
[39] "Arduino - Environment", Arduino.cc, 2017. [Online]. Available: https://www.arduino.cc/en/Guide/Environment. [Accessed: 13- Nov- 2017].
[40] L. Revolvy, ""App Inventor for Android" on Revolvy.com", Revolvy.com, 2017. [Online]. Available: https://www.revolvy.com/main/index.php?s=App%20Inventor%20for%20 Android &item_type=topic. [Accessed: 13- Nov- 2017].