MAKERERE UNIVERSITY

COLLEGE OF ENGINEERING, DESIGN, ART AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

IMPLEMENTATION OF LOCALIZATION SOFTWARE USING UAVS IN REAL TIME

SSEMPIIRA EDWARD

17/U/10196/PS

Main Supervisor

Ms. Sheila Mugala

Co-Supervisor

Dr. Abubaker Matovu Waswa

A Report submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science in Telecommunication Engineering at Makerere University.

February, 2022

DECLARATION

I, Edward Ssempiira, declare that the report entitled **"IMPLEMENTATION OF LOCALIZATION SOFTWARE USING UAVS IN REAL TIME"** is my original and the results embodied in this report has not been submitted to any other university or institute for the award of any degree or diploma.

Signature:…………….

Date: 08/02/2022

EDWARD SSEMPIIRA

APPROVAL

The project report entitled, “IMPLEMENTATION OF LOCALIZATION SOFTWARE USING UAVS IN REAL TIME” is hereby approved by Makerere University, Department of Electrical and Computer Engineering as a creditable study of research topic and has been presented in a satisfactory manner to warrant its acceptance as prerequisite to the degree for which it was submitted.

Main Supervisor

Ms. Sheila Mugala Ndoboli

Signature: .... Date: 08/02/2022

Co- Supervisor

Dr. Abubaker Matovu Waswa

Signature: …………. Date ……

DEDICATION

I dedicate this piece of work to my parents, they have been source of inspiration, engine of courage and secret of my achievements since my childhood. I also dedicate it to my classmates for all the support.

ACKNOWLEDGEMENTS

First of all, I thank the Almighty God the most gracious and most merciful for the wisdom and knowledge, the health and protection and all that he has bestowed on me so that this report becomes a success.

Sincere gratitude to my beloved parents and guardians for the spiritual moral and financial support accorded to me throughout the entire course of my education.

I would like to take this opportunity to express my gratitude towards all the people who have been in various ways, helped in the successful completion of my project report.

I must convey my gratitude to Dr. Abubaker Matovu Waswa and Ms. Sheila Mugala for giving me the constant source of inspiration and help in preparing the project, correcting my work and providing encouragement throughout the project.

I would also like to thank my project partner, Job Barnett Wanzala, for his tireless efforts and team work exhibited and all my faculty members for steering me through the tough as well as easy phases of the project in a result oriented manner with concern attention.

ABSTRACT

Recently, localization in Global Positioning System (GPS) denied environments has witnessed an increase in interest, due to the potential wide range of using in different applications, such as Internet of Things (IoT) providing a solution for the absence of GPS signals inside buildings and tunnels. Different techniques have been used for performing this type of localization, such as sensors and wireless technologies. In this paper, localization and object tracking system is proposed based on WiFi transmission technique. It is done by deploying a WiFi access point with an Unmanned Aerial Vehicle to fly autonomously in such places to read the data of the tracked objects. This is to measure the received signal strength and distance between the WiFi access point and the object to allocate and track it efficiently. The test results show that the proposed system is working in an efficient way with low cost.

LIST OF FIGURES

Figure 2.1 shows the GPS III satellite…..…………………………………...6

Figure 2.2 shows degradation of GPS signal……………………………………9

Figure 2.3 shows ESP8266 Arduino board…………………………………….11

Figure 2.4 shows NodeMCU……...……………………………………………12

Figure 2.5 shows pin mapping from API documentation………………………13

Figure 2.6 shows how trilateration was used…………………………………...14

Figure 3.1shows the flowchart of the proposed algorithm……………………..19

Figure 3.2 shows the hardware connections……………………………………20

Figure 3.3 shows the possible location of the target node……….……………..21

Figure 3.4 shows the algorithm used for position estimation………………21

Figure 4.1 shows the Huawei P30 Lite smartphone MAC address …………..…22

Figure 4.2 shows the results obtained on a serial monitor……………………...23

Figure 4.3 shows the variation of distance between sensor and smartphone.….24

LIST OF TABLES

Table 4.1 The RSS measured results……………………………………..………25

LIST OF ACRONYMS AND ABBREVIATIONS

HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
WSN	Wireless Sensor Network
WIFI	Wireless Fidelity
API	Application Programming Interface
UAV	Unmanned Aerial Vehicle
NodeMCU	Node Microcontroller Unit
PC	Personal Computer
GPS	Global Positioning System
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
RISC	Reduced Instruction Set Computer JSON

JavaScript Object Notation

TCP Text Transfer Protocol IP

Internet Protocol

Table of Contents

DECLARATION ii

APPROVAL iii

DEDICATION iv

ACKNOWLEDGEMENTS v

ABSTRACT vi

LIST OF FIGURES vii

LIST OF TABLES viii

LIST OF ACRONYMS AND ABBREVIATIONS viii

CHAPTER ONE: INTRODUCTION 1

1.1 BACKGROUND 2

1.2 PROBLEM STATEMENT 2

1.3 Objective of the project 3

1.3.1 Main Objectives 3

1.3.2 Specific objective 3

1.4 SCOPE 3

1.4.1 Content scope 3

1.4.2 Technical scope 3

1.5 RELATED WORKS 3

CHAPTER TWO: LITERATURE REVIEW 4

2.1 Global Positioning System 4

2.1.1 How GPS works 6

2.1.2 GPS accuracy 7

2.1.3 Degradation of GPS Signal 7

2.2 UAVs 9
2.3 Localization 10
2.4 Radio Propagation.. 10
2.4.1 Path Loss and Fading 10
2.4.2 Indoor Propagation Models 10
2.5 Description of the Wi-Fi System Architecture and Program................................. 11
2.5.1 ESP8266 Arduino 11
2.5.2. NodeMCU.. 12
2.6. Position Estimation.. 13
2.7 Selection Metrics for RF-Based Location Identification...................................... 14
2.7.1 Scalability 14
2.7.2 Accuracy 15
2.7.3 Precision.. 15
2.7.4 Complexity.. 15
2.7.5 Robustness 16
2.7.6 Reliability 16
2.7.7 Energy Efficiency 16
2.7.8 Cost... 16
CHAPTER THREE: METHODOLOGY 18
3.1 Methods used to collect information.. 18
3.2 Tools Used... 18
3.3 SYSTEM DESIGN 18
3.3.1 Proposed algorithm... 18
3.4.1 Software requirement.. 19
3.4.1.1 Arduino... 19
3.4.1.2 Geolocation API key 19
3.4.3 Hardware Requirement.. 19
3.4.4 Distance estimation 20

3.4.5 Position estimation 21
CHAPTER FOUR: TESTING AND RESULTS 22
4.1 Testing environment 22
4.2 Results 23
4.4 Wi-Fi trilateration based on RSSI measurement 24
CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS 26
5.1 Conclusion 26
5.2 Challenges 26
5.3 Recommendation 26
REFERENCES: 27
APPENDIX: CODE 29

CHAPTER ONE: INTRODUCTION.

Man invented several methods and tools to identify their location a long time ago. Nowadays localization plays a very important role. Various location based services (LBS) has been developed using global positioning system (GPS) for outdoor environment. There are lots of applications where localization is used extensively such as navigation, map generation, complex activity recognition, patient identification, location and tracking in hospitals, child tracking, disaster management, monitoring firefighters, indoor and outdoor navigation for humans or mobile robots, inventory control in factories, anomaly detection, customer interest observation in supermarkets, visitors interest observation in exhibitions, and smart houses [1] [2] [3]. These applications of localization help to solve and improve a variety of real-life problems.

GPS has a high rate of accuracy for outdoor localization. But it is not viable to use GPS indoors or to use wireless sensor networks (WSNs) because it is expensive in terms of energy and cost. Also the signal is not available inside the building. Besides GPS, most of the other existing methods use infrastructure to estimate location both indoors and outdoors, so these methods require additional cost for the infrastructure. As infrastructure is stationary in terms of long range user mobility, it is not possible to identify the location of the smartphone accurately and sometimes they leave the service region. Some methods are adaptive and others need training each time there is a change in environment. Some of the approaches require additional setup time to start working. So to improve accuracy it takes time to recalibrate the system every time there is a change in environment.

Nowadays there is a huge growth in the number of Smartphones. Total shipments of Smartphones in 2011 were 491.4 million with annual growth 61.3% percent from 2010 [6]. Every Smartphone is equipped with various wireless adapters and offers a variety of useful sensors such as Accelerometer, Gyroscope, Orientation sensor, Magnetometer, Barometer, GPS, WiFi, and NFC. Use of a Smartphone based system eliminates cost of additional devices and sensors.

In order to solve existing problems such as improving accuracy, eliminating infrastructure, reducing cost and setup time, and adding mobility we worked on developing a system for localization. We developed a mathematical model for estimating location (distance and direction) of a smartphone indoors and outdoors using WiFi. We used our developed model to build a localization system for Smartphones (Android/iPhone). We also implemented an approach to identify Smartphone location using a mobile WIFI access point.

Our proposed approach has the following contributions:

• Established a new system to model the localization with RSSI

- Smartphone based system which is cost effective and easy to use. • Flying WIFI access point to estimate positions by trilateration

1.1 BACKGROUND.

Localization of mobile wireless devices carries critical importance for applications such as search and rescue, public safety, surveillance, and occupancy monitoring. In this paper, we study the problem of localizing WiFi-enabled mobile devices such as smartphones and tablets using the measurements captured by an unmanned aerial vehicle (UAV). We make use of the continuously broadcasted WiFi probe requests from mobile devices, capture them at different locations at a WiFi sniffer carried by a UAV, and subsequently estimate the user's location using random-forest based machine learning technique. More specifically, the geographical area of interest is partitioned into multiple zones, and based on the measured probe requests, we are interested to identify the zone where the WiFi device is located. Our experimental results show that the WiFi device can be detected in correct occupancy zone with a 81.8% accuracy.

It carries critical importance to localize victims in public safety, emergency, and search/rescue scenarios where victims may be trapped and/or unconscious, and have no proper wireless connectivity to call for aid [1], [2]. For example, during natural or manmade disasters the existing cellular infrastructure may be damaged [3], or, the victim may be outside the cellular coverage area which is common in popularly used hiking trails in many national parks [4].

Global Positioning Systems (GPS) are not an accurate form of indoor, tunnel and mountainous position tracking due to interference with their structures. A solution is needed that can accurately track devices. A portable device was built with an Arduino Uno and an Arduino WiFi Shield that contains a user interface and can be tracked anywhere using WiFi signal strengths.

1.2 PROBLEM STATEMENT.

Position estimation in GPS denied environments is a problem for UAVs. This is because the received GPS signals are too weak in areas like indoors venues, Tunnels, mining sites, Mountainous areas, forests.

The wide use of smart phones and other wireless devices makes it possible to locate and track them even without GPS

1.3 Objective of the project.

1.3.1 Main Objectives.

❒ To develop a localization software for smartphones using UAVs that processes the data in real-time without relying on GPS

1.3.2 Specific objective

- To locate different devices via WiFi sniffing
- To design a system that can estimate the distance of a device from an anchor node (UAV).
- To design an efficient and real-time autonomous localization algorithm based on the WiFi signal strength.

1.4 SCOPE.

1.4.1 Content scope.

The project was based on the development of localization software for unmanned UAVs, using NodeMcu, RISC microprocessor board and Geolocation API to Sniff WAPs and WiFi enabled devices for localization.

1.4.2 Technical scope.

Hardware such as computer with 4 GB RAM 500GB hard disk and with battery that does not go off immediately when power goes off, An android smart phone is also required, USB cable, resistor, jumpers, Esp8266 board, solder wire and the Arduino IDE was used to develop the code for scanning the WIFI devices to provide the randomized MAC addresses and their respective RSS and also in order to test the Arduino code (as well as write most of the code for the Arduino), the Arduino official software was installed. This allowed me to build, compile and run the Arduino code from the computer to the board itself (Esp8266 board).

1.5 RELATED WORKS

There are different techniques used in localization in the absence of GPS signals and some of them have been referenced in this section. In addition, several solutions are provided based on wireless technologies, such as Bluetooth, ZigBee, signals of cellular towers, WiFi and RFID. The related works can be categorized into two subsections.

Indoor localization can be defined as any system that provides the exact location of objects (a personal item) in a closed structure, such as supermarkets, hospitals, universities, airports and subways [1]. Indoor location system has become very popular in recent years. Therefore, many systems and technologies have been proposed to get the user and device localization based into Object. The object can be a mobile phone, keys with attached microcontrollers a keychain, a wallet with embedded microcontroller or microcontroller in a child pocket, among others [9]. One of the existing solutions for indoor localization was provided in [8] to discuss various indoor localization systems that are proposed in the literature and show challenges, such as accuracy of the localization systems. While the authors in [10] introduced an extensive study of many indoor localization techniques, such as the Angle of Arrival (AoA), Flight Time (ToF), Return Time of Flight (RTOF), the Received Signal Strength (RSS), Ultra-wide band (UWB), Radio Frequency Identification of Device (RFID), Bluetooth and Systems. The study looks at the localization and positioning of people users and their devices. For wireless indoor location, various wireless technologies are used. One of the existing research work based on Wi-Fi was indoor triangulation system [4]. This system uses its own Wi-Fi nodes to measure the location of several Wi-Fi devices within the indoor area. The solution provided is required for its own WiFi node infrastructure. Localization accuracy was defined by its location and its concentration. The next notable example of WiFi localization systems is based on several radio beacons [5]. These beacons may be access points for wireless LANs, fixed Bluetooth stations and GSM towers. They used all protocols, which resulted in the unique or semi-unique identification number for beacons. By using this identifier, the lab system detects the indoor position of the user. In [6], the applications of indoor localization have improved the sensing ability to detect the current state of mobile devices. The application was introduced for smartphones and it includes offline and online levels of fingerprinting. While in [10], the authors have created an application that describes the Wi-Fi trilateration method for internal positioning using Android based mobile devices. This was done by receiving a signal strength measurement group that improves localization accuracy. Finally, in [7] the authors presented the latest developed systems or solutions and their algorithms for wireless localization.

CHAPTER TWO: LITERATURE REVIEW

2.1 Global Positioning System

The GPS is a global navigation satellite system that geolocation information to GPS receivers [11]. GPS is a satellite-based radio navigation system; consequently, solid obstacles, such as mountain and buildings, weaken and can even block GPS signals. The GPS does not require any transmission data from users, and its operation is independent from telephonic or internet signal. Currently, the GPS is owned by the

United States’ government and operated by the United States Air Force. The United States’ government is responsible for creating and maintaining the system and makes it freely accessible to any GPS receiver [12].

The GPS calculates the receiver’s position using the time and position of the satellite system. The satellites use stable atomic clocks that are synchronized all satellites and with the ground receivers. Any offset from GPS time of ground receiver is corrected daily. Similarly, satellite locations are controlled with great accuracy. Each GPS satellite continuously transmits a radio signal to provide the current time and its position; therefore, GPS sensors can receive signals without issuing a request. The distance between a satellite and a GPS receiver is calculated based on the delay time of the transmitted and received signals due to the constant speed of radio waves. A GPS receiver updates the distance of multiple satellites and determines the accurate position of the receiver and its deviation from real time. At least four satellites must be used to compute four unknown quantities and solve the accuracy equation [13].

The GPS satellite system comprises 24–32 satellites in medium Earth orbit. Originally, the GPS had only 24 satellites. The orbital period is one-half a day (11 h and 58 min) so that the satellites pass over the same locations every day. The satellite orbits are carefully arranged so that at least six satellites are in line of sight from anywhere on the Earth’s surface [13].

Figure 2.1 GPS III Satellite. U.S. Government photo, GPS.gov Multimedia Library

2.1.1 How GPS works

A GPS tracking system uses the Global Navigation Satellite System (GNSS) network. This network incorporates a range of satellites that use microwave signals that are transmitted to GPS devices to give information on location, vehicle speed, time and direction. So, a GPS tracking system can potentially give both real-time and historic navigation data on any kind of journey. GPS provides special satellite signals, which are processed by a receiver. These GPS receivers not only track the exact location but can also compute velocity and time. The positions can even be computed in three-dimensional views with the help of four GPS satellite signals.

The control of the Positioning System consists of different tracking stations that are located across the globe. These monitoring stations help in tracking signals from the

GPS satellites that are continuously orbiting the earth. Space vehicles transmit microwave carrier signals. The users of Global Positioning Systems have GPS receivers that convert these satellite signals so that one can estimate the actual position, velocity and time.

2.1.2 GPS accuracy

Today's GPS receivers are extremely accurate, thanks to their parallel multi-channel design. Our receivers are quick to lock onto satellites when first turned on. They maintain a tracking lock in dense tree-cover or in urban settings with tall buildings. Certain atmospheric factors and other error sources can affect the accuracy of GPS receivers. Garmin GPS receivers are typically accurate to within 10 meters. Accuracy is even better on the water.

Some Garmin GPS receiver accuracy is improved with WAAS (Wide Area Augmentation System). This capability can improve accuracy to better than 3 meters, by providing corrections to the atmosphere. No additional equipment or fees are required to take advantage of WAAS satellites. Users can also get better accuracy with Differential GPS (DGPS), which corrects GPS distances to within an average of 1 to 3 meters. The U.S. Coast Guard operates the most common DGPS correction service, consisting of a network of towers that receive GPS signals and transmit a corrected signal by beacon transmitters. In order to get the corrected signal, users must have a differential beacon receiver and beacon antenna in addition to their GPS.

2.1.3 Degradation of GPS Signal

In an era of ever increasing wireless RF congestion, GNSS systems are becoming more at risk of signal degradation due to interference. GPS uses signals typically 20 dB below the ambient noise floor and it has only limited interference mitigation. Thus, there is a need to characterize GPS signal degradation and quantify the effects of interference sources. GPS signal deterioration typically occurs by signal masking caused by natural (e.g. foliage) and man-made (e.g. buildings) obstructions, ionospheric scintillation, Doppler shift, multipath, jamming, spurious satellite transmissions, and antenna effects. The impact of anyone of the above can result in partial to total loss of tracking and possible tracking errors, depending on the severity of the effect and the receiver tracking characteristics. Tracking errors, especially if undetected by the receiver firmware, can result in large position errors. Partial loss of tracking results in geometry degradation, which in turn affects position accuracy.

This paper provides an overview of degradation phenomena affecting GPS satellite navigation signals.

There are a number of reasons that lead to the loss of received GPS signal strength and these include the following;

- Signal multi path: this occurs when the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches the receiver. This increases the t ravel time of the signal, thereby causing errors.

- Orbital errors: also known as ephemeris errors, these are inaccuracies of the satellite's reported location. The gravitational and non-gravitational forces perturb the motion of the GPS satellites, causing the orbits to deviate from a Keplerian ellipse in the inertial space.

- Number of satellites visible: the more satellites a GPS receiver can "see," the better the accuracy. Buildings, terrain, electronic interference, or sometimes even dense foliage can block signal reception, causing position errors or possibly no position reading at all. GPS units typically will not work in indoors, underwater or underground.

The figure below shows the GPS signal propagation.

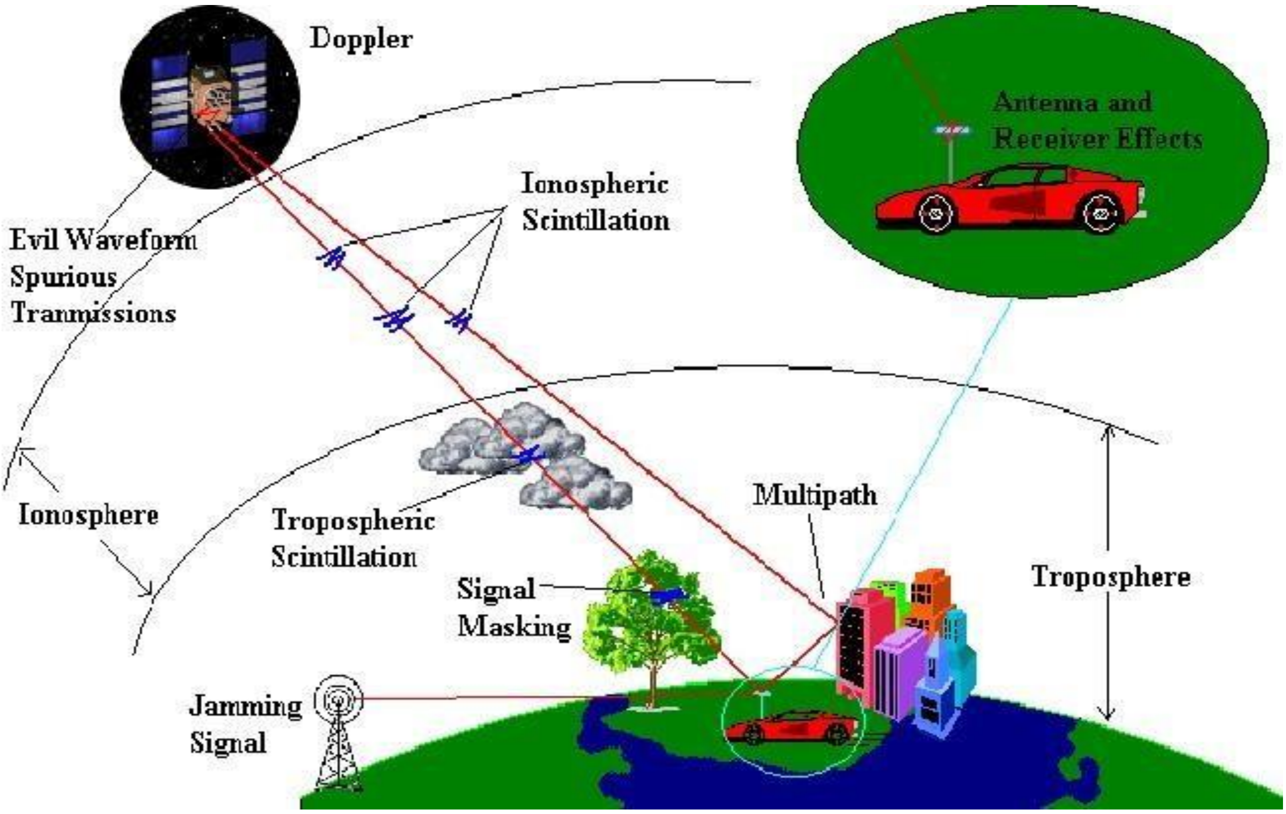


Figure 2.2 Degradation of GPS signal: semanticsscholar.org

2.2 UAVs

Unmanned aerial vehicles (UAVs) have been the subject of immense interest in recent years and have developed into a mature technology applied in areas such as defense, search and rescue, agriculture, manufacturing, and environmental surveillance. Without any required alterations to the existing infrastructure, for example, deployment stations on the wall or guiding lines on the floor, UAVs are capable of covering flexible wider areas in the field. However, this advantage comes at a price. To utilize this flexible resource efficiently, there is a need to establish a coordination and monitoring system for the UAV or fleet of UAVs to determine their environment-based route and schedule in a safe, collision-free, and a time-efficient manner.

While the most common models of UAVs can be identified as quadcopters and the hexa-copters, the most common types of UAVs are multi-rotors, fixed-wing, flapping wing, and hybrid systems, where multi-rotor system is the most popular type of UAVs because it is used for versatile applications and the number of rotors can be in the range of 1 to 12.

2.3 Localization

Localization is the process of finding the physical location of a target in accordance with some real or virtual coordinate system. Localization is an important task when direct measurement of the target location is not available. Generally, system performance with localization is evaluated based on the accuracy of the estimated location information at a given time.

Localization process involves distance estimation and position estimation.

2.4 Radio Propagation

In order to maximize accuracy in location estimation, it is prudent to take note of various factors affecting the propagation of signals in the area that the UAV is deployed. The major propagation factors are path loss and fading.

2.4.1 Path Loss and Fading

Path loss is the reduction in power density of an electromagnetic wave as it propagates through space and is a major component in the analysis and design of the link budget of a telecommunication system. Network planners model the path loss as a function of the path loss exponent which is dependent on the type of environment being served (urban, rural, dense urban, sub- urban, open, forest, sea etc.). Causes of path loss include;

- Free space loss: the energy of any signal reduces when it travels larger distances in space with no obstacles nearby to cause reflection or diffraction.
- Absorption: when the radio signal passes through medium like plants, hills, it gets absorbed.
- Diffraction: this occurs when an obstruction unexpectedly appears in the path. The signal bends around the object and losses occur.
- Multipath: the signal follows a large number of paths from the transmitter to the receiver. During the journey from the transmitter to the receiver, the signal will be reflected and will reach the receiver via a number of different paths. These signals may combine constructively or destructively depending on the relative phases of the signals. Signals that cancel each other or combine destructively lead to multipath loss and fading.

2.4.2 Indoor Propagation Models

Indoor propagation models express transmitted signal strengths as functions of distances between a pair of transceiver and receiver. In a real indoor environment, these models may not be ideal due to many factors (e.g. obstacles, signals interference, and multipath effect) that impact the localization accuracy. The indoor propagation models are closely dependent on several parameters that capture the dissipation effect of signal energy in amplitude, phase, and time in a varying environment.

Free Space Path Loss Model: Equation (2.1) is the simplified free space model that links an RSSI measurement with the estimated distance. This propagation model shows the decrease of RSSI as the distance increases logarithmically Ri,k[dBm] = 10nklog(dk)+Ck where nk is the path loss exponent and the constant Ck captures RSSI measurements

at a reference distance.

2.5 Description of the Wi-Fi System Architecture and Program

This is the technology used to connect computers, tablets, smartphones and other devices to internet. It uses radio waves to transmit data from a wireless router to your WI-FI enabled devices like televisions, smartphones, tablets and computers.

In this system, a microcontroller is used and the Arduino. These components are adopted as they are easy in programming and use. In addition, they are cheap comparing to other controllers and easy in implementation. Finally, the design of the system interface is performed, in which it becomes simple with the lowest cost possible.

2.5.1 ESP8266 Arduino

The ESP8266 Arduino compatible module is a cheap Wi-Fi chip with full TCP / IP capabilities. This small module allows microcontrollers to connect to the Wi-Fi network and establish simple TCP / IP connections using Hayes style commands [14].

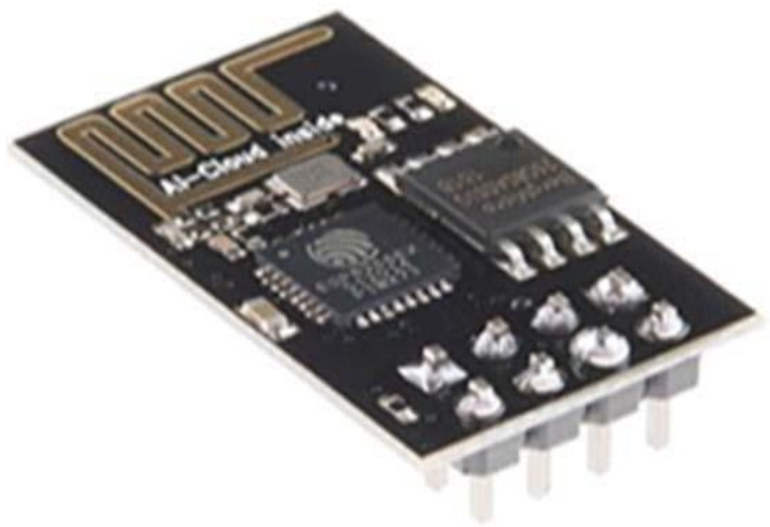


Figure 2.3 ESP8266

In this paper, the ESP8266 Arduino is used to connect the server (PC) and the access point to detect the Wi-Fi signal. The Arduino is chosen instead of Wi-Fi shield because it is cheaper than Wi-Fi shield, easy in implementation, and easy in programming. ESP8266 is built in with node MCU, so there is no need to buy two boards for interconnection and we avoid connections problems.

2.5.2. NodeMCU

NodeMCU is a firmware on ESP8266 and it is basically a System on Chip (SoC). In addition, the NodeMCU is an IoT open source platform. By default, the word "MCU node" refers to the firmware rather than development kits [14]. NodeMCU is smaller than Arduino board, and also it has a built in ESP8266, thus there is no need to connect external ESP8266 for easy implementation as shown in Figure 2.3 and Figure 2.4

Figure 2.4 NodeMCU

The NodeMCU software includes a series monitor, which allows text data exchange with the board. The NodeMCU provides access to the GPIO (General Purpose Input/Output) and access to development. Figure 2.5; shows the pin mapping from the API documentation [14].

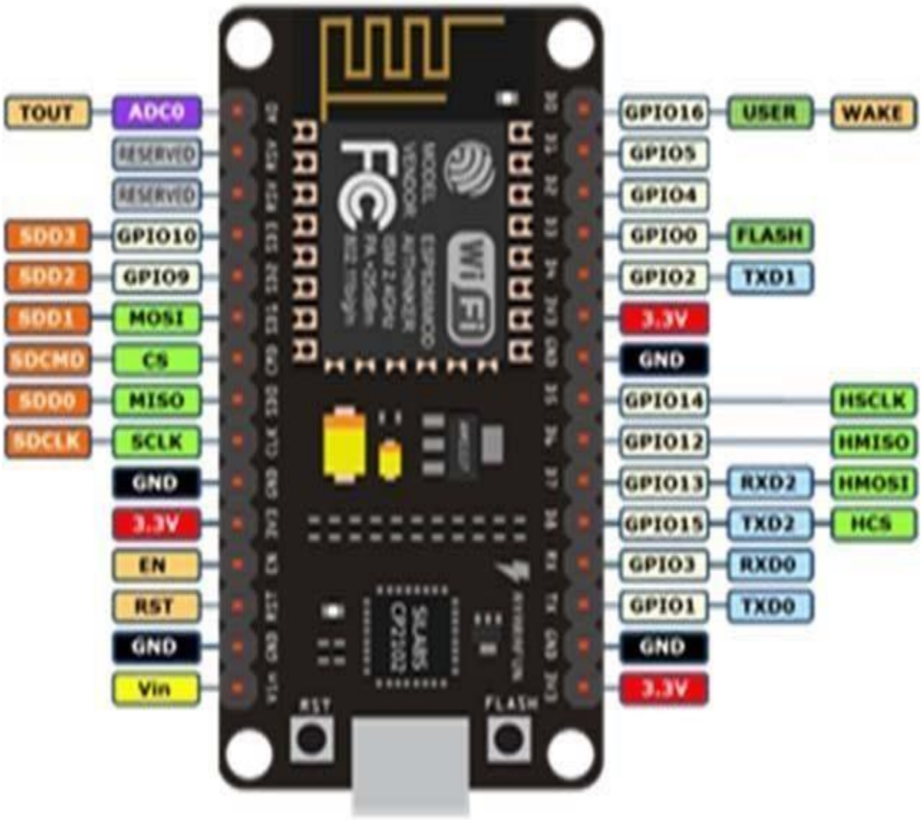


Figure 2.5 Pin mapping

2.6. Position Estimation

The above method provides the estimated distances between the anchor or reference node(s) and the target node(s). With these distances, other techniques can be applied to estimate the position of the target. Communication between anchor nodes and target also helps determine this geometric information about their relative placement e.g. distance between the two nodes or the angles of a triangle formed by the anchor nodes. This information is then further used to determine node location. When distance is used as a primary means to determine node location, this is termed as lateration, and when angle information is used for localization, it is known as angulation. For node localization in a plane, precise distance measurements from at least three anchor nodes are required and we use trilateration for position estimation of a node. Intersection of three circles around the three target node gives a single point as position of the node as is shown in the figure below.

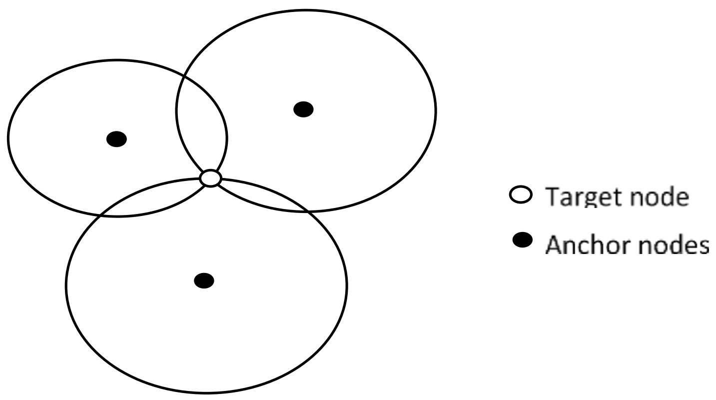


Figure 2.6 Trilateration algorithm

2.7 Selection Metrics for RF-Based Location Identification

In this section, we shall focus on the performance of RF-based location identification technologies using a common generic metric of scalability, accuracy, complexity, robustness, reliability, energy efficiency, cost, and throughput. The metrics are used for the empirical evaluation of each indoor technology with a view to establish the viability of each RF-based technologies.

2.7.1 Scalability

Scalability describes the adaptive ability of a location identification system to allow many instances of the technology to operate independently without interfering with each other, while at the same time the system stability is still maintained. Scalability is a critical factor that guarantees the overall performance of location identification systems since the number of cooperating small-scale, low-powered nodes with limited sensing and computational capability may often need to be replaced or increased. However, location identification efficiency degrades with an increase in distance between the nodes and the receiver, which usually require most techniques used to be modified. To modify the system to make it more scalable, recent location identification techniques scale the target on two axes i.e. the density and the geography.

2.7.2 Accuracy

Accuracy gives the degree of deviation of the estimated position from the actual position, i.e. it is the closest calculated position to the true position without a prior knowledge of the target known location often termed location identification error. Location identification error refers to the measurement that gives the average Euclidean distance between the estimated location and the true location. Thus, since obtaining accurate information of a target location is the primary objective of any location identification system, accuracy can be improved through joint detection and estimation of the sensor nodes.

2.7.3 Precision

Precision is an indication of the repeatability of a measurement. Precision is a sensitivity-based performance metric that determines the reliability of a location identification system in presence of changing environmental conditions. Precision is often used interchangeably as resolution in much location identification literature. However, precision and accuracy are closely related, as precision relates to the deviation of the distance error, whilst accuracy relates to the mean of the distance error. Therefore, an accurate location identification system might not necessarily be precise, but a precise location identification system will have high accuracy.

2.7.4 Complexity

The complexity of location identification systems can be described as the degree of hardware, software and operating factors applied to the overall system. Complexity is often more difficult to support when there is no power. It is, however, less of a problem when the algorithm used for location identification runs on a server with a power supply. However, it can be challenging when the platform running the algorithm is mobile where strong processing power and durable power supply can affect the reliability of the system. Thus, location rate estimates the delay between the mobile target reaching the new location and the time of estimation of the new location.

2.7.5 Robustness

Location identification systems often face the challenge of one or more failing components within the network. In addition, factors such as the complexity of the indoor environments and other physical effects often do present the location identification system with incomplete data to localize a target, which affects the system performance. Thus, robustness describes the ability of location identification systems to continue to provide reliable and accurate measurements even when conditions are adverse, e.g. in the presence of interfering sources or many targets colocated.

2.7.6 Reliability

Reliability is a crucial performance metric as it measures the percentage of measurements for which the accuracy meets some threshold. This is because a system might sometimes give an accurate measurement and at other times give no measurement or an inaccurate one. Thus, reliability indicates the level of dependability of the overall system.

2.7.7 Energy Efficiency

Location identification applications often do require reasonable longevity, and since most sensor nodes are likely to be battery-powered. Charging or recharging of these batteries for the nodes is not usually feasible in many applications, thus an aggressive energy management policy is crucial for these battery-powered nodes. Moreover, energy efficiency in sensor-based applications is affected by environmental factors resulting from collision, overhearing and idle listening and control packet overhead. Therefore, energy efficiency in most indoor applications is considered one of the current design challenges and a critical performance metric.

2.7.8 Cost

Cost divides between the various hardware infrastructures (beacons, servers, tags) used in a location identification system, which can be evaluated in several diverse ways such as roll-out, operating/maintenance, and space cost. Roll-out costs involve the cost required in either installing the new hardware infrastructures or extending the existing infrastructure. Operating or maintenance costs are cost involved in maintaining the optimal functionality of a location identification system, thereby reducing the overall complexity of the system. Space cost relates to the number and size of the hardware infrastructure. Thus, overall cost trade-offs will depend on how many hardware infrastructures are deployed.

CHAPTER THREE: METHODOLOGY.

The project specification above is clear and defined, and so we were capable of building a system that can perform these actions. However, in order to build this system, we first ascertain what was required to complete these tasks. In this chapter, we covered what tools and methods we used to reach our objectives. Furthermore, we also identified and justified any risks for the project.

3.1 Methods used to collect information.

a) consultation and online search

The information use d to implement the project was obtained through reading different online papers about localization and UAVs performances to come up with a viable solution for an effective localization in GPS denied environments.

Consulting both the main and co-supervisors was also a helpful method in having a working project.

3.2 Tools Used

The software tool used was Arduino Integrated Development Environment (IDE). Arduino IDE is a cross-platform application that is written in the Java programming language. The Arduino IDE supports the languages C and C++. It is used to write and upload programs to Arduino compatible boards, but also with other vendor boards. In this project, Arduino IDE 1.8.19 version was used for programming the board because of its vast libraries that make it easy to visualize and gain the data.

The hardware tool used is ESP8266 NodeMCU, which is an open-source firmware and development kit that helps you to prototype or build Internet of Things (IoT). MCU stands for MicroController Unit which really means it is a computer on a single chip. The ESP8266 WiFi module is a powerful, yet cost-effective, surfacemountable WiFi module with an embedded ESP8266 System on Chip (SoC). The module has a wireless transceiver operating in an unlicensed frequency range of 2400-2484 MHz in the IEEE 802.11 b/g/n standard, with support for TCP/IP communication protocol stack and WiFi security including WAP3.

3.3 SYSTEM DESIGN.

3.3.1 Proposed algorithm

The operation of the proposed algorithm is built on estimating the distance and direction of the device. It starts by initialization where the NodeMCU connects to internet and then starts scanning for available WIFI devices or sniffing to identify them through RSSI and MAC addresses. Then through using the Geolocation API access we can get the location of our anchor node (NodeMCU + UAV) and then further steps are taken to estimate the unknown location of the WIFI device through trilateration as shown below.

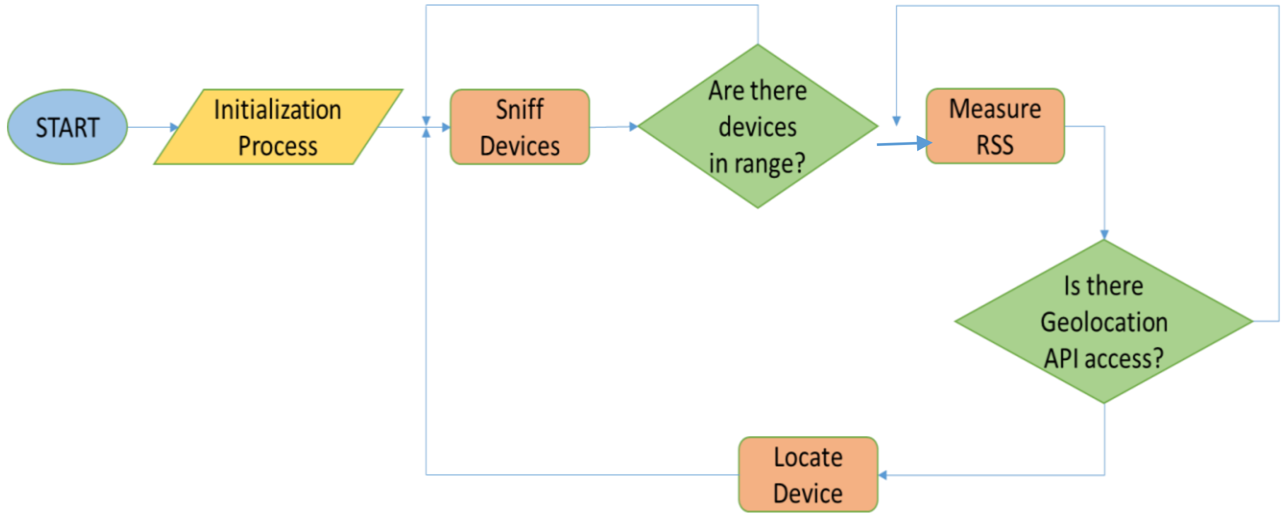


Figure 3.1 Proposed algorithm

3.4.1 Software requirement.

3.4.1.1 Arduino

The software is designed with the use of Arduino Integrated Development Kit Environment that allows writing, compiling and uploading code to the board. The program or sketch is written using C programming language or C compiler.

3.4.1.2 Geolocation API key

The Geolocation API returns the location and accuracy radius of the access point (NodeMCU + UAV) based on information about the WIFI node (NodeMCU).

Communication is done over HTTP using POST, both request and response are formatted as JSON and the content type of both is **application/json.**

3.4.3 Hardware Requirement

The process began by soldering the resistor, OLED display, and the NodeMCU onto the Veroboard. The NodeMCU was then connected to the 5V power supply of the laptop and upload the code.

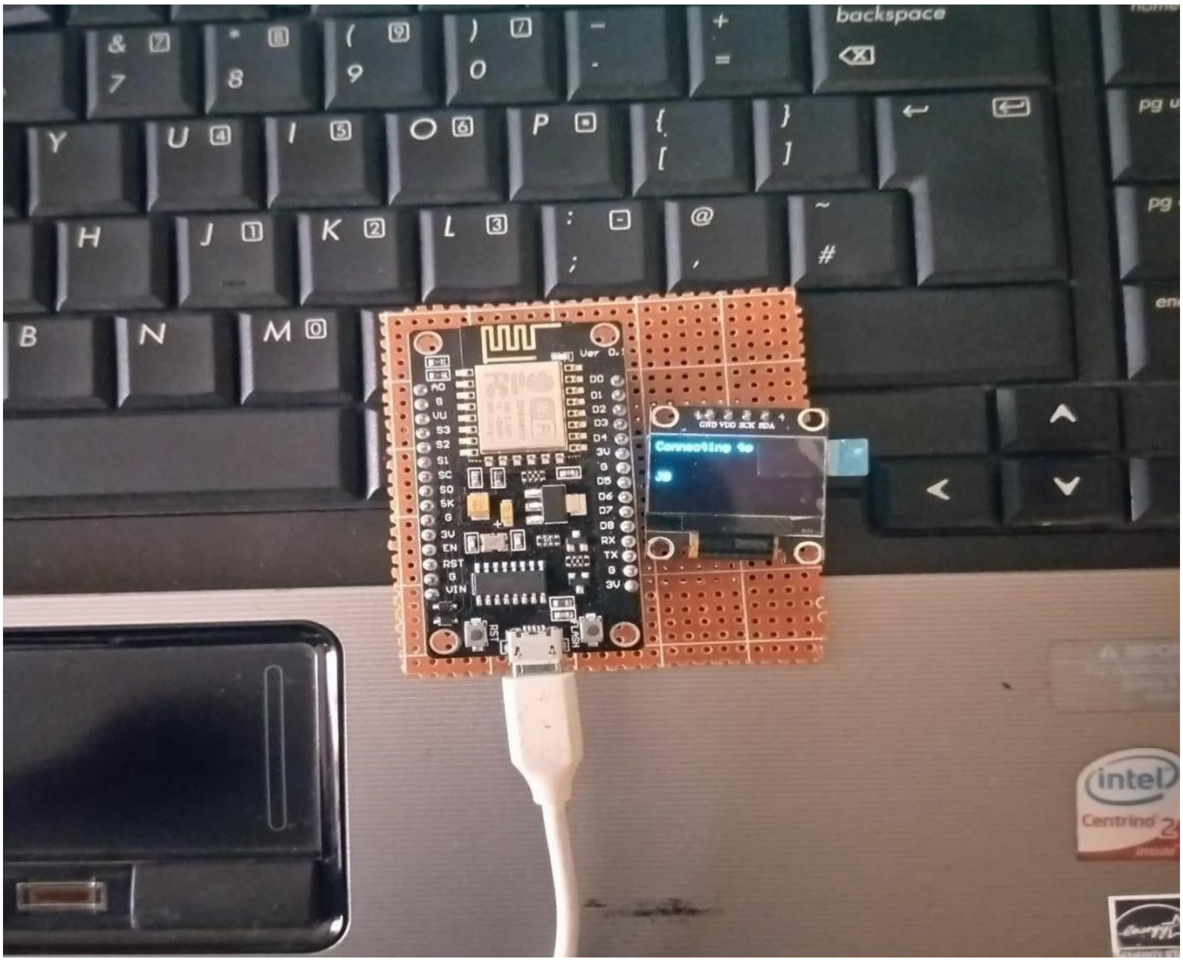


Figure 3.2 Hardware

3.4.4 Distance estimation

Measure RSSI at target node from anchor node(s). In order to estimate the distance between the target and each reference node, a relationship between the received signal power and distance is used. We apply the Log Normal Shadowing mode [16]

$$RSSI(d) = -10n\log10(d)-C$$

n = path loss exponent, C = environment constant

For indoor environment, n can be from 1.5 to 5 (obstacles raise the value)

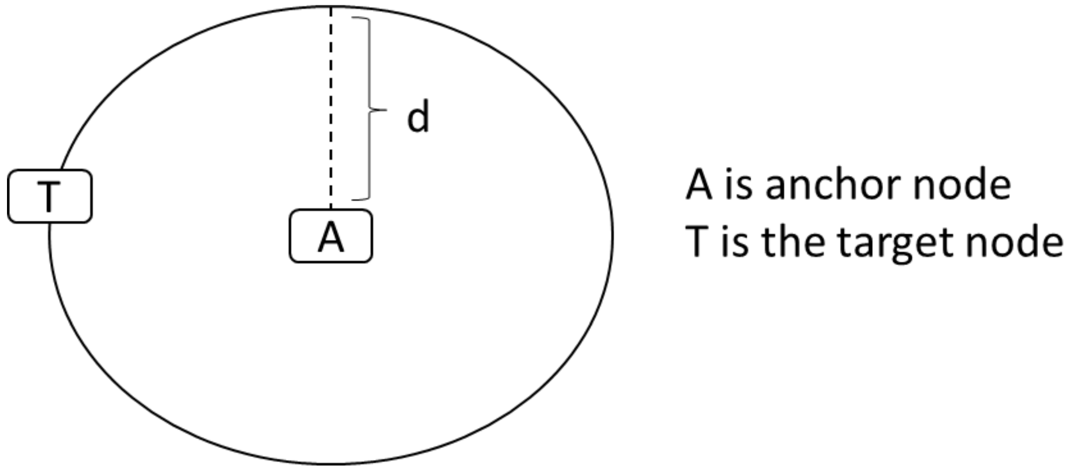


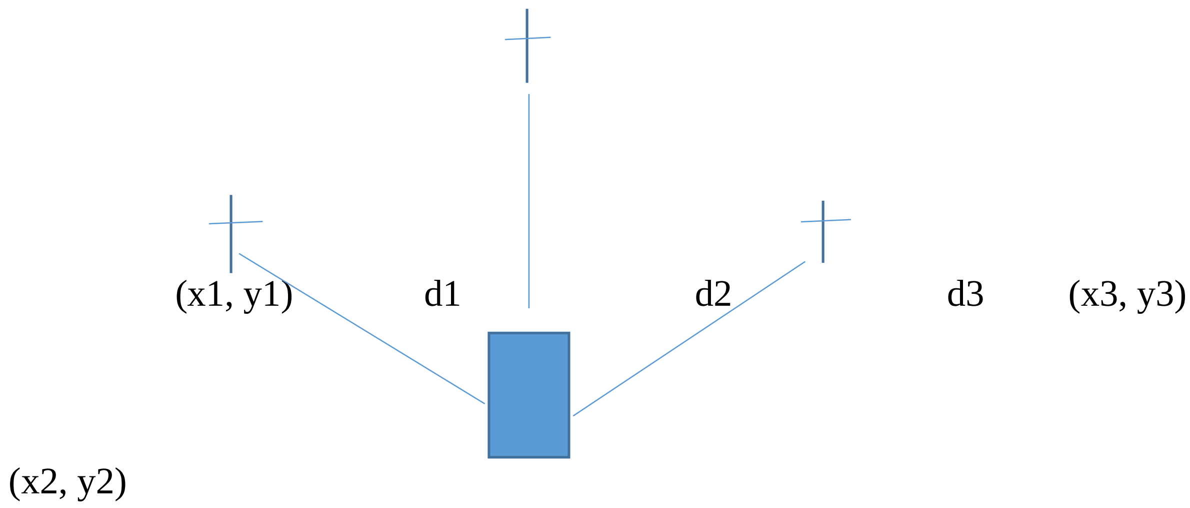
Figure 3.3 Distance estimation

The circle represents the possible locations of the smartphone as the target node.

3.4.5 Position estimation

We considered a single UAV with a NodeMCU to act as an access point flying at three different positions which are known (provided by Geolocation API). At each location of the UAV, the received signal strength is recorded to determine the distance between the two nodes.

For the three different locations of the UAV, three different distances for the same target node are recorded and the unknown position (x,y) is determined by trilateration.



Smartphone

(x, y)

Figure 3.4 Position estimation

CHAPTER FOUR: TESTING AND RESULTS

4.1 Testing environment

In this project, the test environment was selected to be a simple room which had different WIFI access points, hotspots and WIFI enabled smartphones. This was done by connecting the NodeMCU.

For my case study, I used Huawei P30 Lite smartphone as it could display its real MAC address as shown below.

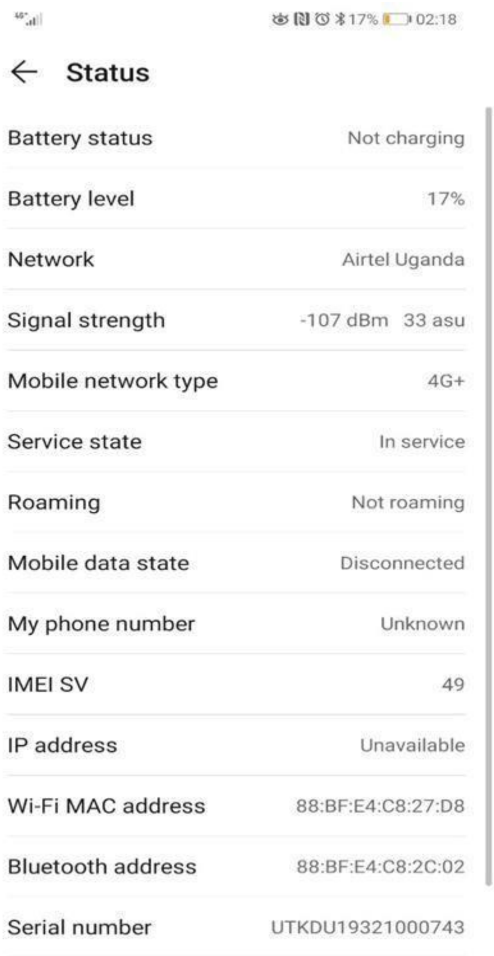


Figure 4.1 Smartphone MAC address

4.2 Results

During testing, I found out that the NodeMCU scanned all the access points, hotspots and the WIFI enabled smartphones by displaying their MAC addresses and the RSSI. Furthermore, it was observed that most of the smartphones provided random MAC addresses.

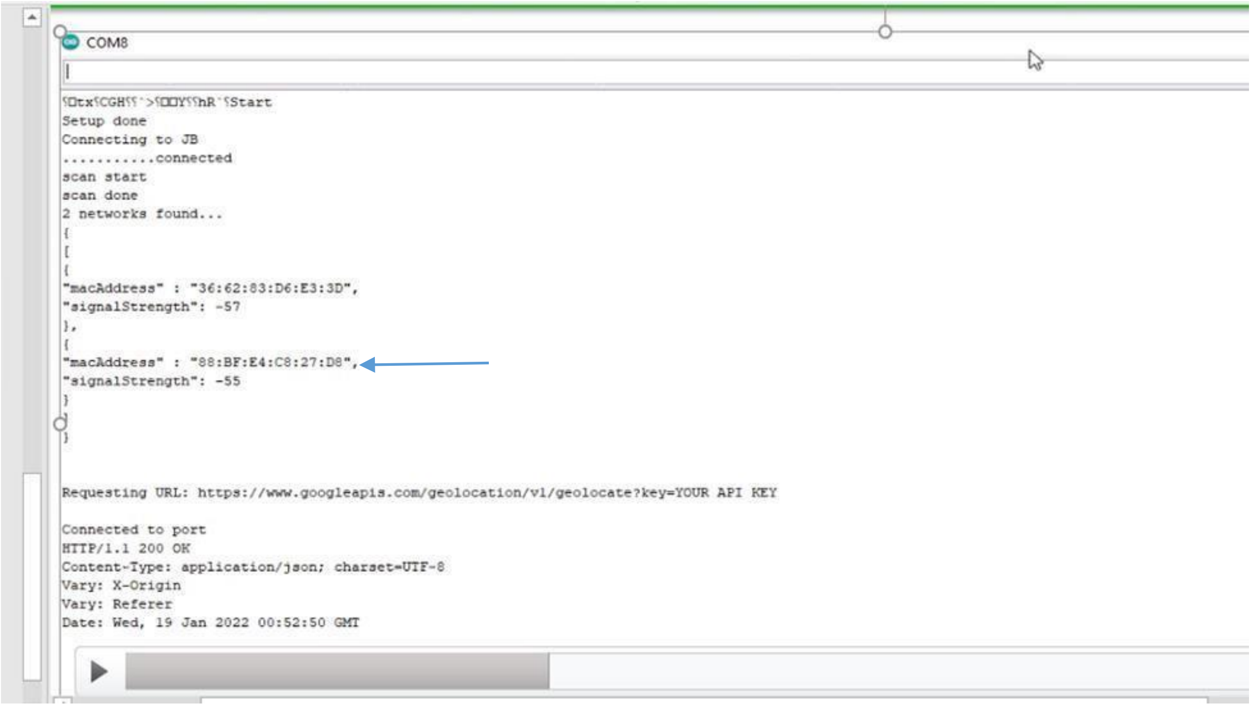


Figure 4.2 Serial monitor

As the distance between the sensors and the smartphone increased, the WiFi-signal strengths were observed to decay in a logarithmic manner. This behavior is presented in Figure 4.3 below.

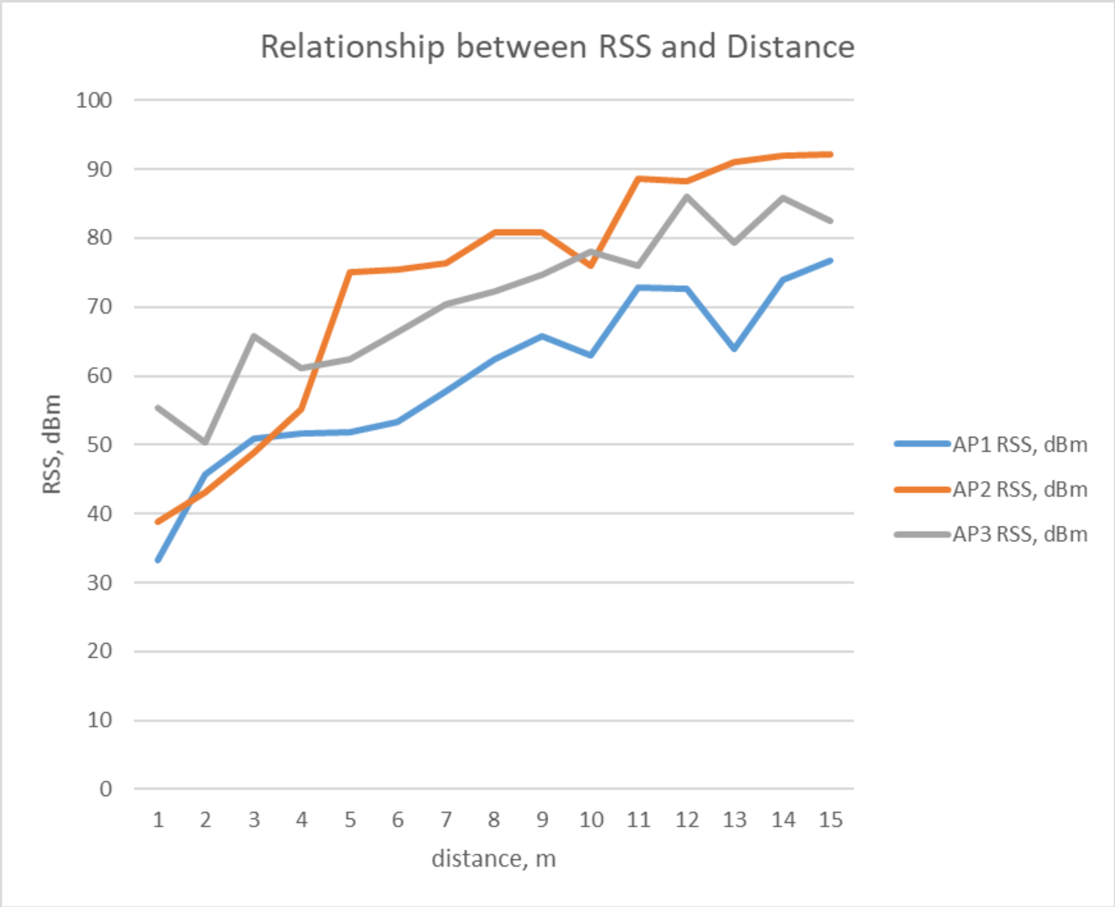


Figure 4.3 Decay of signal strength when the distance from the source is incremented.
The yellow line represents the strongest signal strength recorded, the red line represents the weakest signal strength recorded and the blue line represents the average signal strength recorded.

4.4 Wi-Fi trilateration based on RSSI measurement

The signal strength levels were measured by distance of three access points defined by different locations of the NodeMCU to estimate the unknown position of the smartphone. These measurements were made in 15 points at the 1-meter interval for each access point using developed NodeMCU. The access point RSS levels are displayed in the Table 4.1 below.

Table 0.1 The RSS measured results

Distance, m	AP1 RSS, dBm	AP2 RSS, dBm	AP3 RSS, dBm
1	33.3	38.8	55.3
2	45.7	43.1	50.3
3	50.9	48.9	65.7
4	51.7	55.2	61.2
5	51.8	75.1	62.5
6	53.4	75.5	66.4
7	57.8	76.4	70.5
8	62.4	80.8	72.3
9	65.7	80.8	74.7
10	62.9	76	78
11	72.9	88.6	76
12	72.7	88.2	86
13	63.9	91	79.3
14	74	91.9	85.8
15	76.7	92.1	82.5

CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

In this paper, the Wi-Fi signal strength-based object detection and tracking system in GPS denied environments was proposed. The proposed system used the WiFi signal strength read from different devices to evaluate the position of the required object. The location is assigned using Geolocation API access and trilateration of the flying WIFI access point. The obtained results showed the accurate performance of the proposed system in terms of allocating objects.

5.2 Challenges

Main obstacle to designing an algorithm for smart phones is the limitations of the phones themselves. One major limitation for phones is battery life. This is in fact the single largest obstacle to designing more complex algorithms, as we cannot insist that the user constantly be scanning Wi-Fi networks in order to localize, as that would be an enormous strain on battery consumption. One aspect of phones that we could not account for at all was dealing with the wide variety of phones and corresponding hardware, putting limits on how much we could trust the data given to us.

5.3 Recommendation

Our future work includes deploying two to three UAVs to gather larger sets of data to evaluate the scalability and improve on accuracy of our proposed approach by relating different positions obtained by different UAVs for one target node.

REFERENCES:

[1] A. Kumbhar, F. Koohifar, I. Guvenc, and B. Mueller, “A Survey on Legacy and Emerging Technologies for Public Safety Communications,” IEEE Commun. Surveys Tuts., 2016.

[2] A. Birk, B. Wiggerich, H. B??low, M. Pfingsthorn, and S. Schwertfeger, “Safety, Security, and Rescue Missions with an Unmanned Aerial Vehicle (UAV),” Journal of Intelligent & Robotic Systems, vol. 64, no. 1, pp. 57–76, 2011.

[3] E. Vattapparamban, I. Guvenc, A. Yurekli, K. Akkaya, and S. Uluagac, “Drones for Smart Cities: Issues in Cybersecurity, Privacy, and Public Safety,” in Proc. IEEE Int. Wireless Commun. Mobile Comp. Conf. (IWCMC), Paphos, Cyprus, Sep. 2016.

[4] S. Waharte and N. Trigoni, “Supporting search and rescue operations with UAVs,” in Proc. IEEE Int. Conf. Emerging Security Tech.(EST), 2010, pp. 142–147.

[5] P. Rudol and P. Doherty, “Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery,” in Proc. IEEE Aerospace Conf., Mar. 2008, pp. 1–8.

[6] P. Doherty and P. Rudol, “A UAV search and rescue scenario with human body detection and geolocalization,” in Australasian Joint Conf. Artificial Intelligence. Springer, Dec. 2007, pp. 1–13.

[7] M. Carpin, S. Rosati, M. E. Khan, and B. Rimoldi, “UAVs using Bayesian Optimization to Locate WiFi Devices,” Preprint arXiv:1510.03592, 2015.

[8] Marcus Utter, “Indoor Positioning using Ultra-wideband Technology”, Examensarbete 30 hp, December 2015.

[9] Fr ´ed´ eric Evennou and Franc¸oisMarx, “Advanced Integration of Wi-Fi and Inertial Navigation Systems for Indoor Mobile Positioning”, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 86706, Pages 1–11, DOI 10.1155/ASP/2006/86706.

[10] Faheem Zafari, Athanasios Gkelias, Kin K. Leung, “A Survey of Indoor

[11]Localization Systems and Technologies", arXiv:1709.01015v1 [cs.NI] 4 Sep 2017 Daly, P. Navstar GPS and GLONASS: Global satellite navigation systems. *Acta Astronaut.* **1991**, *25*, 399–406. [Google Scholar] [CrossRef]

[12] Alkan, R.M.; Karaman, H.; Sahin, M. GPS, GALILEO and GLONASS satellite navigation systems & GPS modernization. In Proceedings of the 2nd International Conference on Recent Advances in Space Technologies (RAST 2005), Istanbul, Turkey, 9–11 June 2005. [Google Scholar]

[13] https://en.wikipedia.org/wiki/Global_Positioning_System (accessed on 21 September 2019).

[14] Suraj Tonage , Sandhya Yemul , Rajendra Jare , Veena Patki, "IoT based home automation system using NodeMCU ESP8266 module", International Journal of Advance Research and Development, Volume3, Issue3, 2018.

[15] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih. Indoor location position based on bluetooth signal strength. In 2015 2nd International Conference on Information Science and Control Engineering, pages 769–773.

[16] S. Barai, D. Biswas, and B. Sau. Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique. In 2017 IEEE Conference on Antenna Measurements Applications (CAMA), pages 170–173, December 2017.

APPENDIX: CODE

```
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include "ESP8266WiFi.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);  char
myssid[] = "JB";         // your network SSID (name) char
mypass[] = "piper#123";          // your network password

//Credentials for Google GeoLocation API... const char*
Host = "www.googleapis.com"; String
thisPage = "/geolocation/v1/geolocate?key=";
String key = "AIzaSyBmw85FxJO_LS45CC4vVHkI0CbfH1s4Txw";

int status = WL_IDLE_STATUS;
String jsonString = "{\n";

double latitude    = 0.0; double longitude   = 0.0;
double accuracy    = 0.0; int more_text = 1; // set to
1 for more debug output

void setup()   {
Serial.begin(9600);
```

```
  Serial.println("Start");
   display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();

  //Display Text   display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 10);
display.println("Welcome!");
display.display();


  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
WiFi.disconnect();   delay(100);
  Serial.println("Setup done");
  // We start by connecting to a WiFi network
  Serial.print("Connecting to ");
Serial.println(myssid);
display.clearDisplay();
display.setCursor(0, 0);
display.println("Connecting to");
display.setCursor(0, 30);
display.println(myssid);   display.display();
  WiFi.begin(myssid, mypass);

  while (WiFi.status() != WL_CONNECTED) {
delay(500);     Serial.print(".");
  }
  Serial.println("connected");


}

void loop() {
char bssid[6];
  DynamicJsonDocument doc(1024);
```

```
  Serial.println("scan start");
display.clearDisplay();      display.setCursor(0,
10);   display.println("Scanning....");


  display.display();
  // WiFi.scanNetworks will return the number of networks found
int n = WiFi.scanNetworks();   Serial.println("scan done");   if (n ==
0)
    Serial.println("no networks found");   else
  {
    Serial.print(n);
    Serial.println(" networks found...");


    if (more_text) {
      // Print out the formatted json...
      Serial.println("{");
      //Serial.println("\\");  // this is a real UK MCC
      //Serial.println("\\");   // and a real UK MNC
      //Serial.println("\\");          // for gsm
      //Serial.println("\\");       // associated with Vodafone
      Serial.println("[");


      for (int i = 0; i < n; ++i)
      {
        Serial.println("{");
        Serial.print("\"macAddress\" : \"");
        Serial.print(WiFi.BSSIDstr(i));
        Serial.println("\",");
        Serial.print("\"signalStrength\":                  ");
Serial.println(WiFi.RSSI(i));
display.clearDisplay();            display.setCursor(0,
0);         display.print("MC:"); display.print(WiFi.BSSIDstr(i));
display.setCursor(0, 30);         display.print("SS:");
display.println(WiFi.RSSI(i));         display.display();
delay(1000);         if (i < n - 1)
```

```
        {
          Serial.println("},");
        }         else
        {
          Serial.println("}");
        }
      }
      Serial.println("]");
      Serial.println("}");
    }
    Serial.println(" ");
  }
  // now build the jsonString...
  jsonString = "{\n";    jsonString += "\"homeMobileCountryCode\": 234,\n"; //
this is a real UK MCC
jsonString += "\"homeMobileNetworkCode\": 27,\n";  // and a real UK MNC   jsonString
+= "\"radioType\": \"gsm\",\n";         // for gsm    jsonString += "\"carrier\":
\"Vodafone\",\n";      // associated with Vodafone
jsonString += "\"wifiAccessPoints\": [\n";   for (int j = 0; j < n; ++j)
  {     jsonString +=
"{\n";      jsonString +=
"\"macAddress\" : \"";
jsonString += (WiFi.BSSIDstr(j));
jsonString += "\",\n";     jsonString +=
"\"signalStrength\": ";     jsonString +=
WiFi.RSSI(j);     jsonString += "\n";
if (j < n - 1)      {       jsonString +=
"},\n";
    }     else
    {       jsonString +=
"}\n";
    }   }   jsonString +=
("]\n");   jsonString +=
("}\n");
  //-------------------------------------------------------------------
```

```
  Serial.println("");

  WiFiClientSecure client;

  //Connect to the client and make the api call
  Serial.print("Requesting URL: ");
  Serial.println("https://" + (String)Host + thisPage + "YOUR API KEY");
Serial.println(" ");   delay(500);   client.setInsecure();   if
(client.connect(Host, 443)) {     Serial.println("Connected to port");
client.println("POST " + thisPage + key + " HTTP/1.1");
client.println("Host: " + (String)Host);
client.println("Connection: close");     client.println("Content-Type:
application/json");     client.println("User-Agent: Arduino/1.0");
client.print("Content-Length: ");
client.println(jsonString.length());     client.println();     client.print(jsonString);
delay(500);
  } else
{
  Serial.println("not Connected to port");   delay(10000);
}

  //Read and parse all the lines of the reply from server   while
(client.available()) {     String line =
client.readStringUntil('\r');     if (more_text) {
Serial.print(line);       display.clearDisplay();
display.setCursor(0, 0);        display.print(line);
display.display();        delay(2000);
    }

      JsonObject root = doc.as<JsonObject>();

//      if (!root.success()) {
    {       latitude    =
root["location"]["lat"];       longitude   =
```

```
root["location"]["lng"];       accuracy   =
root["accuracy"];

    }
  }
delay(2000);
  Serial.println("closing connection");
Serial.println();   client.stop();

  Serial.print("Latitude = ");
  Serial.println(latitude, 6);
  Serial.print("Longitude = ");
  Serial.println(longitude, 6);
  Serial.print("Accuracy = ");
  Serial.println(accuracy);

}
```