COM667 BSc (Hons) Computing Systems

FINAL REPORT

ON

Designing and Developing GPS IOT System for Tracking Pets

Student Name: Bibek Thapa

Student Number: B00792128

Project Supervisor Name: Amjad Alam

Date: 2021/2022

Abstract

The system is designed to track the location of pets with the help of Gps (global positioning system). GPS is a satellite-based navigation mechanism that gives location and timing information in all situations, everywhere on the Earth where there is a free line of sight to three or more GPS satellites. Military, municipal, and commercial customers across the globe are benefiting greatly from its capabilities. In order to modernise the global air traffic system, GPS serves as the foundation.

This project consists of technologies such as Internet of things (IOT), cloud-based technology and different programming language including script language which will also be able to overcome the security issues of user's data by removing third party involvement because all the control over data will be on the user's hand.

To create the prototype, the GPS/IoT system will combine the fixing of an electrical gadget and an IoT device. In a chosen integrated development environment, the proposed system will be constructed using two separate programming languages: the Arduino programming language and the Python programming language. The project has accomplished mentioned objectives successfully.

Acknowledgement

I would like to take this opportunity to convey my thanks to everyone who has assisted and encouraged me during the completion of this project and my stay at university. The lecturers and teaching staff at Ulster University deserve a heartfelt thank you for offering three years of great instruction, which culminated in the completion of this dissertation. It has been a pleasure to have had your ongoing support and education throughout my time at Ulster University.

I would like to express my gratitude to Amjad Alam, my project supervisor, for his assistance and supervision throughout the year of this project. I was really fortunate to have his knowledge and skill in the technologies involved, as well as his willingness to spend significant time and effort on the project, and his contribution was priceless.

Our course director, Anwar Haq, and Dr. Ekereuke Udoh, have contributed their time and effort to teach and assist my friends and me as we progressed through the course. As a final note, I'd like to offer my appreciation to my friends and family, who have been patient and supportive during this monumental project. I am grateful for your constant support throughout my undergraduate and professional endeavours.

I owe a debt of gratitude to everyone who has expressed their appreciation towards me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 CHAPTER 1 - INTRODUCTION

1.1 BACKGROUND

Lots of pets go missing every year around the world especially in UK. In United Kingdom itself, we see a lot of posters and pictures of missing or lost cats and dogs on the wall of sidewalks, shops, and other places as well (Smith, 2021). Even a simple googling can tell us the amount or percentage of missing pets. Although there are many Pets tracking system in the market, but many have different disadvantages in relation to battery life, GPS tracking efficiency, security issues, and cost. For the solution of above-mentioned problem, I wanted to design and develop a gps IoT system for pets that will track the location of pets and sending the location data to any private cloud like Azure, AWS, etc. On the other side, I will also build mobile application that can show the pets' owners the last location of pets in the map.

While designing this system, I will be exploring and doing research on using the most efficient micro-controller and sensors that be energy efficient and overcome the tracking and security issues. There are some similar solutions for this problem but those could be quite expensive also and moreover the GPS-tracking data are controlled by those companies, who sell the system which leads to security issues.

This project can overcome even security or privacy concern issues by implementing it with user's own cloud service. This project should also reduce the price to any other industrial system that exist in the market by making use of efficient and cheap microcontrollers and GPS and other sensors. The tracking software will be designed either in Nodejs or C# programming language.

1.2 PROJECT AIMS

The aim of this project is to design and develop GPS-IOT system to track the last location of pets, so the owner of pets could track it through the desktop application if it gets lost or

stolen. This system will entail the development of an IoT system as well as a web application that will provide information on the pets' previous locations.

The current chances of a lost pets finding its way back to its owner far away from its territory resembles the people of luck. According to McGrath, over 3 million cats and dogs are euthanized in American animal shelters each year, accounting for half of all animals admitted because their owners can't be found. Thousands of pets have gone missing because of natural disasters like Hurricane Katrina. Separation dramas can arise in the most insignificant of circumstances. According to Safe Animal, over 90% of all lost pets are never returned to their owners. Every week, over 60 pedigree dogs are stolen in the United Kingdom, and the number is increasing.

In my work, I will provide the knowledge I've gained, as well as the options and ideas that will allow for a practical assessment and execution.

To do so, I will break down my project into smaller objectives that, when combined with the development of the tasks, will result in the project's completion and implementation soon after my studies are through. I believe that neither the public nor the technological community has given this issue enough attention.

1.3 PROJECT OBJECTIVES

The main objectives of the project are:

- To improve gps tracking efficiency using energy efficient sensors and device.
- To overcome security or privacy concern issues by implementing it with user's own cloud service.
- To create user-friendly python programme and to get location in a map.
- To overcome expensive price of existing product by using most reliable and cheap microcontroller, sensor.

1.4 OUTLINE OF A DISSERTATION STRUCTURE

This is a synopsis of the chapters that comprise the whole dissertation.

Chapter 1 Introduction

This chapter offers a summary of the project, background, the objectives, and aims.

Chapter 2 Literature review

The present chapter is dedicated to the literature review, technological history, and any research conducted as part of the project. It serves as the foundation for this project, which includes a review of current programmes as well as new customer-applicable solutions. It would also concentrate on the software development life-cycle methodology selected and the tooling that would be used to implement it. Finally, it outlines the project's priorities and objectives, making the necessary conclusions.

Chapter 3 Requirements gathering and analysis

This chapter will address the main aspects of project management in this chapter, such as how to develop a project plan and assess risks, as well as strategies for building functional and non-functional requirements.

Chapter 4 Design

The following chapters include various system designs, such as User Interfaces (simple designs and interfaces of the application) and Use Case Diagrams.

Chapter 5 Implementation and testing

This section will provide a detail overview of the code fragments, and any specific challenges discovered, as well as solutions. This chapter describes the tactics and the programming language used during implementation. The priority list, requirement list, and operating principle of the Gps/Iot device will be used to guide the testing of the created system.

Chapter 6 Evaluation

This chapter evaluate the project's overall outcomes to see if all of the needed design and development work has been completed.

Chapter 7 Conclusion

This chapter will wrap up the final report by providing a critical assessment of the project's methodology and operations, as well as recommendations for future work.

2 CHAPTER 2 - LITERATURE REVIEW

In this chapter, a review of relevant research is conducted to thoroughly investigate the project's history. The review offers background knowledge for the project's scientific inquiry system. Various knowledge may be gathered from the review to meet the given objectives by selecting the best technique to include the features into the Pet Tracking System utilising IoT.

Location tracking and a mobile application are two components of the system that have been studied. These discussed aspects may aid in the generation of ideas and solutions for improving the suggested project. This chapter will go through some of the technologies and tools that were already explored.

2.1 WHAT IS GPS?

The Global Positioning System (GPS) is a navigation system that synchronises location, velocity, and time data for air, sea, and land travel by employing satellites, a receiver, and algorithms. The satellite system consists of 24 satellites in six Earth-centred orbital planes, each with four satellites, orbiting about 13,000 miles (20,000 kilometres) above Earth and travelling at a speed of 8,700 miles per hour (14,000 kilometres per hour). While just three satellites are required to create a position on the earth's surface, a fourth satellite is frequently utilised to verify the data from the other three. The fourth satellite also takes us into the third dimension, allowing us to determine a device's altitude.

The GPS system is made up of three parts, known as segments, that operate together to produce location data. The three segments of GPS are:

- Space segment: Satellites orbit the Earth, sending signals to users based on their geographic location and time of day.
- Control segment: The Earth-based monitor stations, master control stations, and ground aerial comprise the Control Segment. Tracking and operating satellites in space, as well as monitoring broadcasts, are examples of control activities. Monitoring stations can be found on practically every continent, including North and South America, Africa, Europe, Asia, and Australia.
- User segment: GPS receivers and transmitters, such as watches, cell phones, smart gadgets and telematic devices.

2.1.1 How does a gps system work?

Trilateration is a technology used by GPS. Trilateration receives data from satellites to calculate location, velocity, and elevation. It is frequently confused with triangulation, which is used to calculate angles rather than distances. Satellites circling the planet provide signals that a GPS device on or near the earth's surface reads and interprets. A GPS gadget must be able to read signals from at least four satellites to determine position. Each satellite in the network does two daily orbits around the planet, sending a unique signal, orbital characteristics, and time. A GPS gadget may read signals from six or more satellites at any given time.

When a satellite transmits a signal, it forms a circle with a radius equal to the distance between the GPS device and the satellite. When a second satellite is added, a second circle is created, and the location is narrowed down to one of two spots where the circles overlap.

Because the device is near the intersection of all three circles, the position of the gadget may eventually be identified with the help of a third satellite. However, because we live in a three-dimensional world, each satellite generates a sphere rather than a circle. Because three spheres overlap at two spots, the point closest to Earth is picked.

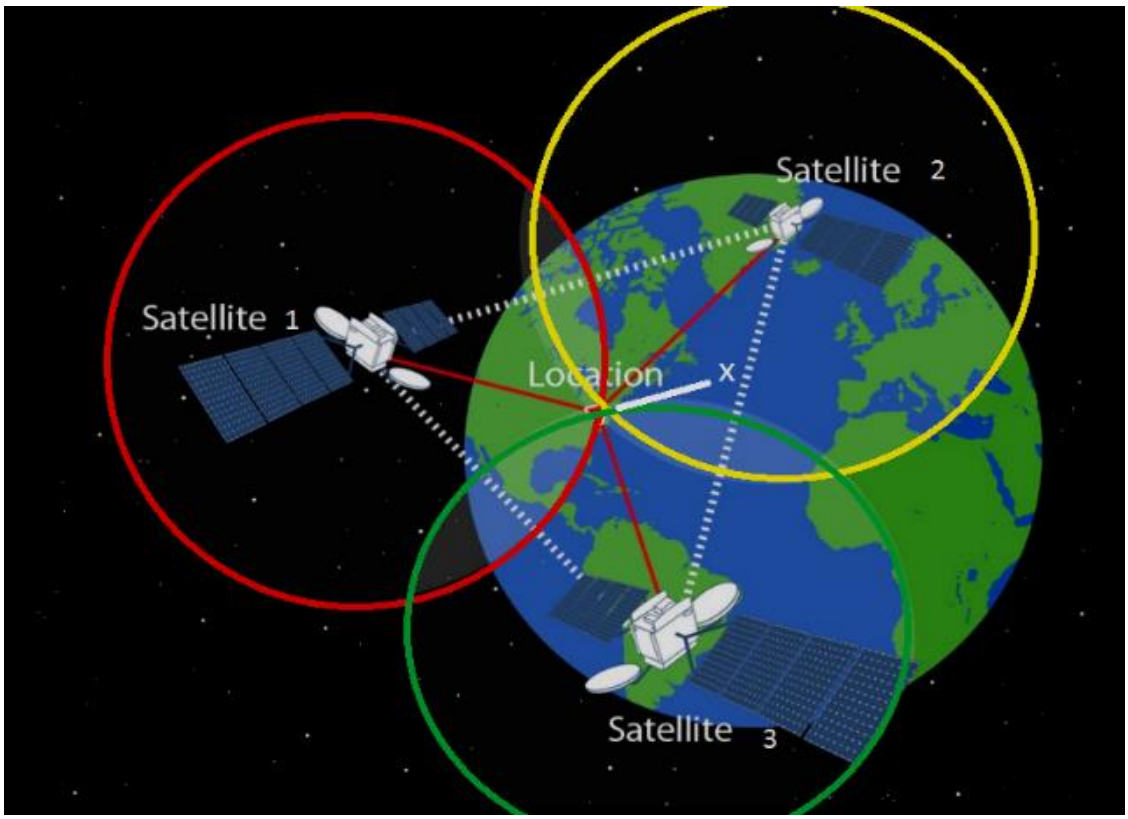


Figure 1: GPS System

2.2 EXISTING AVAILABLE SYSTEMS

To create an effective system for this project, previous work in the area of trackable pet and livestock system will be examined, analysed, and evaluated in this part. However, just the most frequent devices will be examined and divided into categories to showcase comparable tasks while still showing the requirement of the proposed solution, even if there are many devices accessible today. Key goals and suggested techniques for tackling the given issue make this proposed solution stand out.

2.2.1 Microchip based systems

A needle is inserted between the dogs' neck and shoulders and a biocompatible microchip measuring around 11mm long and 2mm wide is implanted under their skin. This microchip is based on Microchip Technologies' MicroID technology, namely the AN680 chip, which was created specifically for tracking. Pets are equipped with passive RFID tags, which transfer data to a reader and subsequently decode it into a 15-digit identification number that links to the pet's owner's personal information stored on a central computer network. Users must wait for someone to find their pet before they know where their pet is located, and there are significant hazards associated with implanting bio-hardware, so this method may not always yield the greatest outcomes. Because surgery should only be performed if absolutely required, it is preferable to look for an "external" remedy. As a result of pet microchipping, the following dangers have been identified:

- A little annoyance When the microchip is implanted under the skin of the pet.
- Migration is a possibility occasionally, the microchip will move away from the injection site, making it more difficult to identify and scan it.
- Numerous brands of microchips are available. They operate on one of three distinct frequencies in the United Kingdom, and not all veterinarians and animal shelters have universal scanners capable of reading all the various brands and frequencies.
- Health hazards that may exist. Although it is uncommon, soft tissue tumours might form at the injection site.

Figure 2: Pet's Microchip

Figure 3: Microchip inserted into pet's body

2.2.2 Bluetooth /QR code-based trackable system

In this area, there are several possibilities to choose from, but many of them do not necessarily meet the purpose of being trackable, as will be explored more below. Bluetooth collars that can be tracked are commonly accessible on the internet, but their functioning leaves much to be desired. Because Bluetooth 4.0, the most widely used standard in mobile devices at the time of writing (i.e., a class 2 device that generates a signal with a power of 2.5

mW as defined by the FCC), is only restricted to a range of roughly 10 metres, it is not suitable for long distances. Considering the manner of operation that these devices use, it is obvious that they are not suited for tracking collars since they need the user's pet to be within 10 metres of their mobile phone. This makes little to no sense since there is no active monitoring while the user's pet is within range, and when the user's pet is outside of range, an alarm is raised, which simply serves to warn the user that their pet has gone missing and provides no more information. The problem with QR/Bluetooth codes, that is similar to the one with microchips, in that the owner must wait until their pet is discovered by someone else before scanning the collar, assuming that the person who discovers the pet understands that the collar has a QR code on it.

Figure 4: Bluetooth Tracker

2.2.3 Gps/Gsm/Gsrm based system

Several similar solutions have been discovered; the first, developed by Farm Ranger SA, uses GPS and an internet connection to address the highlighted issue; however, as a backup option, it is also capable of sending an SMS to the user.

Following an article that appeared in Farmers Weekly, the second solution was identified. However, attempts to contact the product's owner or identify a website were useless at the

time of writing. Similarly, to this project, a third suggested solution was published in 2013 under the title "Design and Testing of a GPS/GSM Collar Prototype to Combat Cattle Rustling." It was identical to this project. While the study was confined to animals in Italy, it was found to be the strategy that is the most similar to the recommended solution since it makes use of GPS and GSM (2G) technology to track down cows. As a result of this that the technique proposed by this study remains relatively unique, as evidence of previous attempts based on GPS and GSM is difficult to come across.

This system implements a low-cost item tracking system that makes use of GPS and GPRS technology, among other things. It enables a user to observe the current and previous locations of a target item on a Google Map by connecting to the internet. It took advantage of the GPRS service, which resulted in a low-cost tracking solution for locating and monitoring an object's location and status. It is extremely beneficial in automobile theft scenarios and for adolescent drivers who are being watched and tracked by their parents, as well as for monitoring humans and animals.

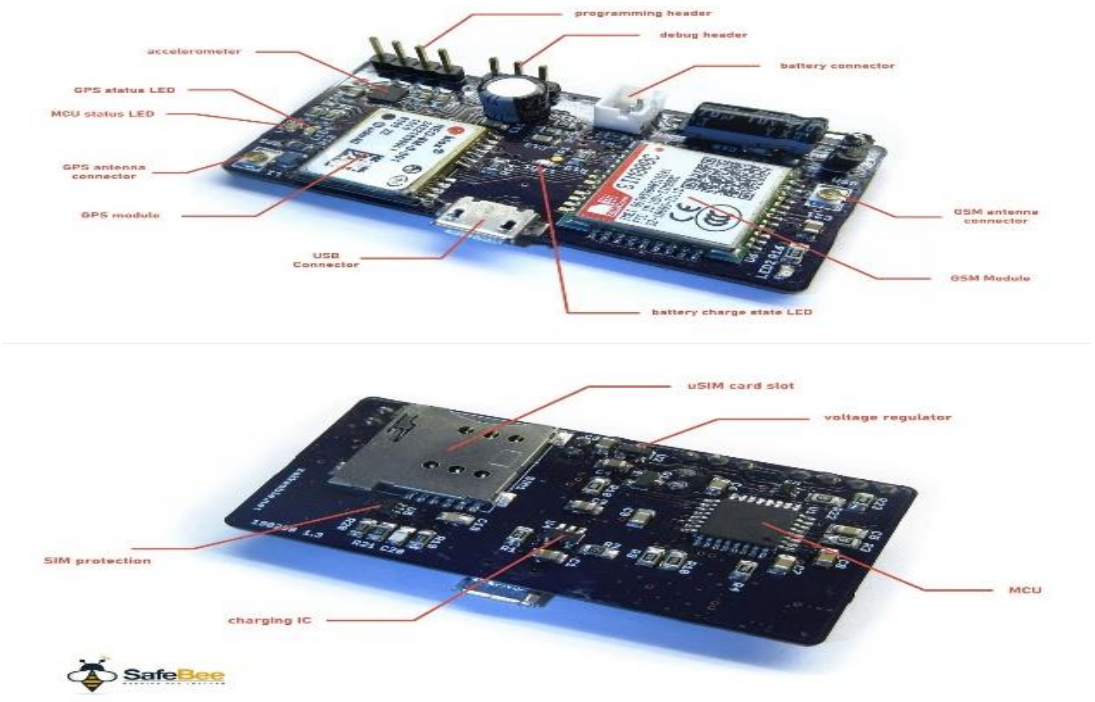


Figure 5:Gps module

The availability of a significant variety of GPS-tracking systems for pets has been reported by numerous sources, even though some are only available in certain countries, need a monthly charge or subscription fee, and do not offer users with cloud storage of GPS data; others are rather expensive. Considering that these things are designed for the general population, there hasn't been many studies available on them, and as a consequence, when seen in the context of this report, the "data" supplied may seem to be lacking in technical rigour and accuracy.

2.3 AVAILABLE TECHNICAL COMPONENTS

Different approaches are taken in the design, development, and implementation of tracking systems. Each approach has its own set of features that ensure a smooth operation of the system. They use a variety of hardware components and databases to store their user's or client's data and information, each employing a different approach to data and information security protection. Here are the most used resources and components to build similar system in the past:

2.3.1 Microcontroller

A microcontroller is a small integrated circuit that is used to manage a specific activity in an embedded system. It is meant to be small and inexpensive. An example of a microcontroller is one that contains a processor, memory, and input/output (I/O) peripherals all on a single piece of silicon. Microcontrollers can be found in automobiles, robotics, office machinery, medical equipment, mobile wireless transceivers, vending machines, and household appliances, among other items. They are sometimes referred to as embedded controllers or microcontroller units. It is simply a little PC with no sophisticated front-end operating system meant to operate certain aspects of a bigger component (OS).

2.3.1.1 How does microcontroller work?

A microcontroller is a small computer that is embedded into a system and is responsible for controlling a single function of a device. By utilising its core CPU, it interprets data received from its input/output peripherals and displays it. It is the data memory of the microcontroller that stores the temporary information that the microcontroller receives. It is here that the processor accesses it and employs instructions contained in its programme memory to decode and apply the incoming data. It then makes use of its I/O peripherals to interface with and perform the required action on the computer.

A microcontroller uses its central processing unit (CPU) to process data received through its input ports and to generate output through its output pins. A synchronous sequential logic circuit is used to execute or drive the logic circuit.

As per digital circuits theory, a sequential circuit is a logical circuit whose output not only depends on present inputs but also on past input history. And in synchronous sequential circuits, the state of the device changes only at discrete timing decided by a clock signal. The main advantage of a synchronous system is its simplicity when compared to an asynchronous system. Microcontrollers are found in a diverse range of systems and gadgets. Devices frequently make use of several microcontrollers, each of which performs a specific purpose in conjunction with the others.

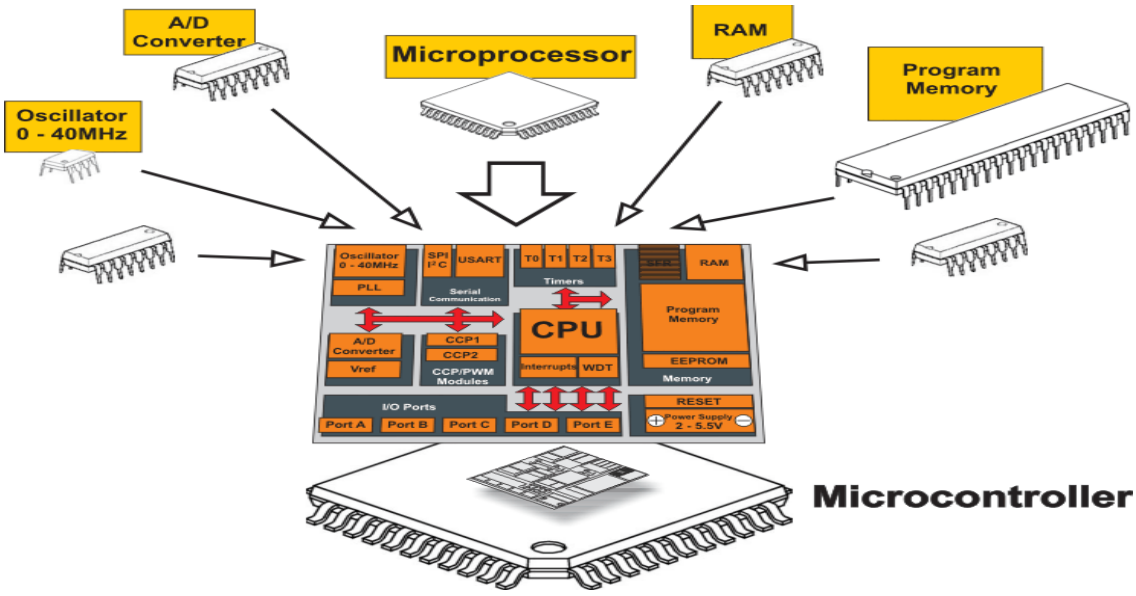


Figure 6:Architecture of Microcontroller

2.3.2 Raspberry Pi

The Raspberry Pi is a low-cost, credit-card-sized computer that connects to a computer display or television and may be controlled with a conventional keyboard and mouse (not included). In short, it is a competent tiny gadget that allows individuals of all ages to explore computing and learn how to create in programming languages such as Scratch and Python. In addition to performing everything a desktop computer can do, like accessing the internet and watching high-definition video, it can also create spreadsheets, perform word processing tasks, and play games. To top it all off, with its capacity to communicate with other devices in its immediate vicinity, the Raspberry Pi has been utilised in a diverse range of digital maker projects, ranging from music machines to parent detectors to weather stations and tweeting birdhouses equipped with infrared cameras. Kids all across the globe should be able to learn to programme and grasp how computers operate with the Raspberry Pi, and we want to see this happen with the Raspberry Pi.

The Raspberry Pi Foundation was established to provide a low-cost computer that can do the same for today's schoolkids, inspired by the 1980s 8-bit computers that converted a generation of Britons into developers. First came the Raspberry Pi Model B, with 256MB of RAM and a 26-pin GPIO extension connector, in 2012 A family of ARM-based devices, comprising a full-fledged computer, a microcontroller CPU, and a series of modules that may be combined to construct more complicated devices, has evolved over the following decade.

2.3.2.1 How does Raspberry Pi work ?

There are various sorts of models available on the market, each with its own design, size, and operating principle to consider.

All Raspberry Pi devices run Linux as their operating system. When a computer's hardware and software are not talking to each other, Linux is the go-to operating system. Python, a high-level general-purpose programming language, is used with Raspberry Pi to create graphical user interface (GUI) apps and websites. When using Raspberry Pi, you don't need to

be an expert in Linux or Python in order to get started on a project. In reality, the product's goal is to make learning about the system and language fun.

The Pi Zero and Pi Zero W models are the most basic models available, with Wi-Fi and Bluetooth capabilities provided by the Zero W model and not by the Zero model (which lacks these capabilities). The fundamental model gives the user the chance to study computer programming and to explore the Internet of Things (IoT) through projects that are designed to keep the learner interested throughout the course. It is a network of interconnected computing devices, mechanical and digital machines, and other items that have the potential to transport data via a network that is called the Internet of Things (IoT). Projects involving the Raspberry Pi Zero include word clocks, environmental monitors (temperature, humidity, and so on), broadcast speakers, informative displays, drones, vintage games, and selfiebots.

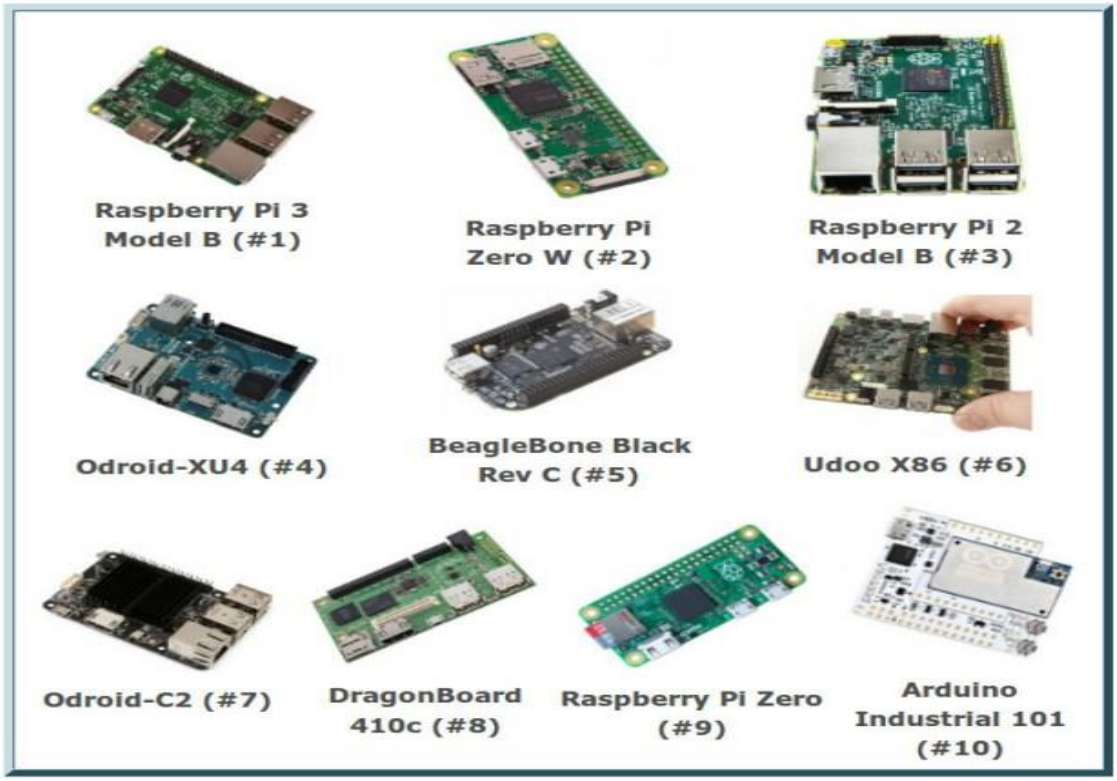


Figure 7: several types of raspberry pi

2.4 AVAILABLE SOFTWARE/INTEGRATED DEVELOPMENT ENVIRONMENT(IDE)

To put the project's principles into action, the appropriate tools must be employed. There is various development environment available to develop the code for Android, Ios and Web based application. Here are some popular software programmes that may aid in accomplishing goals.

2.4.1 Visual studio code

A Microsoft-developed editor, that provides syntax highlighting, debugging, and GIT support, among other features. It is possible to install plugins made by other users, which makes it quite versatile. In addition, the application may be customised in terms of themes and keyboard shortcuts, and it is fully free and open-source software. It supports a vast variety of languages and offers capabilities such as code restructuring, deep semantic code interpretation and navigation, and a graphical user interface. Although it is a lean editor, Visual Studio Code includes a debugger that allows programmers to go through their code to find and repair issues.

Figure 8: Visual studio code

2.4.2 Arduino IDE

The Arduino IDE is an open-source programme that enables users to develop and upload code in real-time to a work environment. Since the written code will be sent to the cloud, it is widely employed by those in need of an additional layer of redundancy. The Arduino IDE is fully compatible with any software board based on the Arduino platform. The programme may be simply installed on any operating system, including Linux, Mac OS X, or Windows. The majority of its components are built in JavaScript, which allows for smooth compilation and modification of the finished product. Even though the tool's primary focus is code writing, it also includes several useful features. In the case of the Arduino IDE, for example, users may exchange critical project information with other stakeholders inside their organisation. Internal layouts and schematic alterations can be made by the user at any time if necessary. For individuals who want assistance throughout the installation procedure, comprehensive guidelines are provided.

The primary features of Arduino IDE include its ability to act as both an on-premises and online editor, direct sketching, board module possibilities, and integrated libraries.

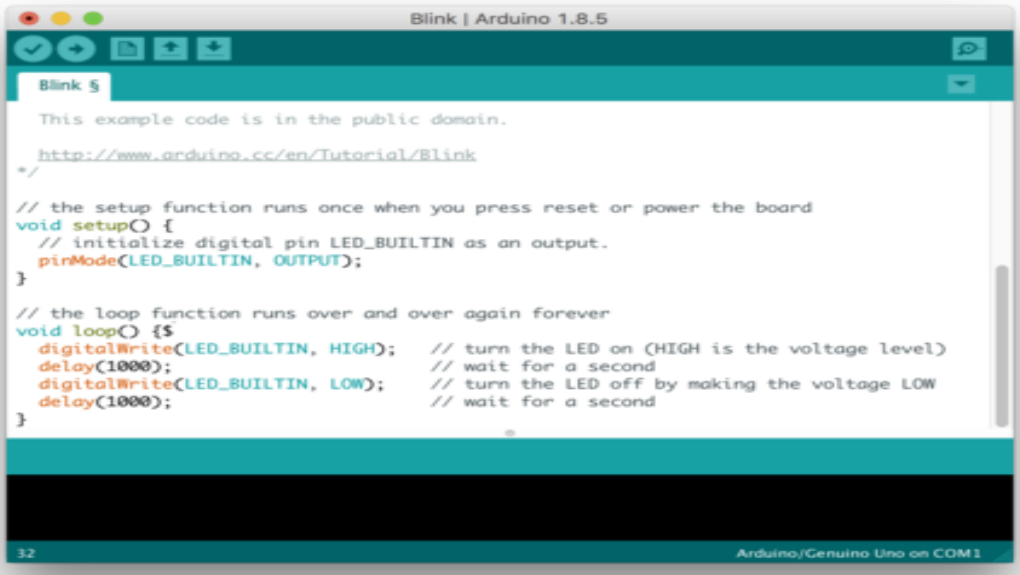


Figure 9: Arduino integrated environment (IDE)

2.4.3 IntelliJ IDEA

IntelliJ IDEA is an Integrated Development Environment (IDE) for Java virtual machine (JVM) programming languages that is optimised for developer productivity. It automates mundane and repetitive chores for you using intelligent code completion, static code analysis, and refactoring's, allowing you to focus on the more pleasant aspects of software development.

The primary reason it is recognised as one of the greatest Java programming tools is because of its help features, which make it simple to use and result in well-designed projects. Additionally, it has improved error checking capabilities that enable faster and easier error checking. Despite being intended primarily for Java development, IntelliJ IDEA understands and provides smart coding assistance for a variety of additional programming languages, including Groovy, Kotlin, Scala, JavaScript, TypeScript, and SQL. The IDE may construct a virtual map of your project by indexing the source code from the start. It can identify mistakes on the fly, offer code completion variations with exact context awareness, do refactoring, and more using the information from the virtual map.

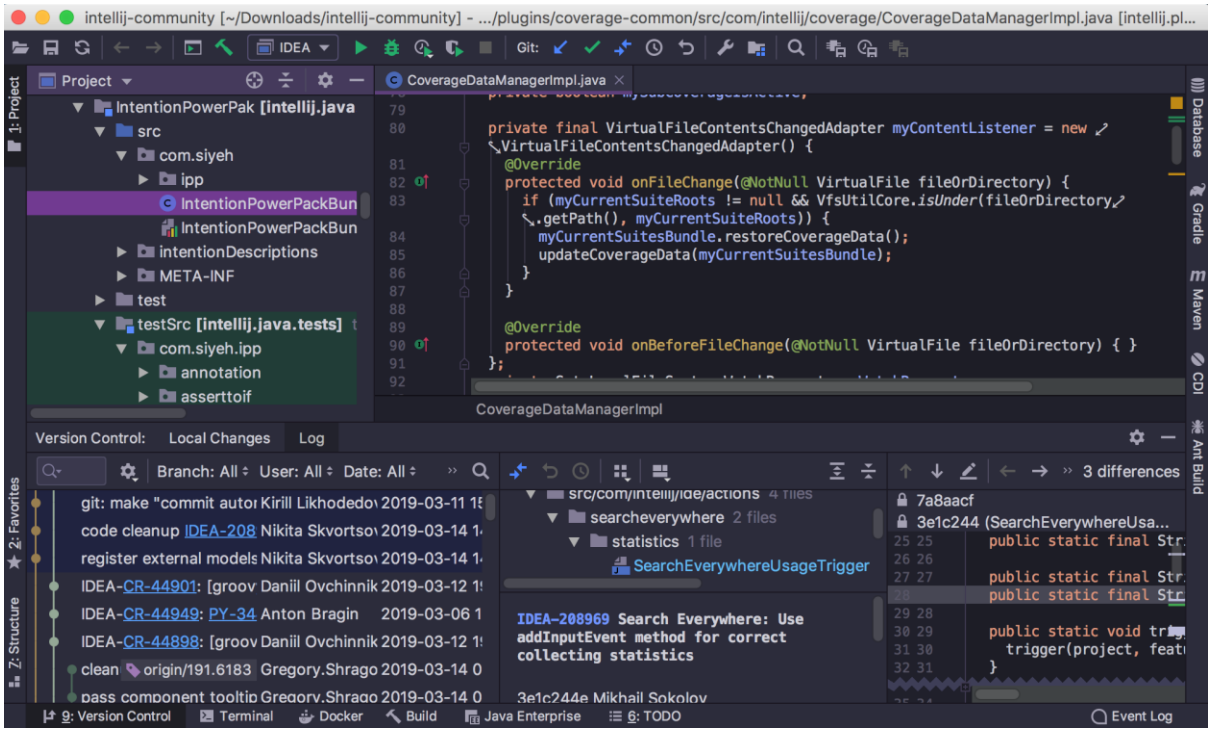


Figure 10: IntelliJ IDEA

2.5 CHOSEN DEVELOPMENT FRAMEWORK

After doing in-depth research on major features and specification in existing and most popular integrated development environment Arduino IDE for coding in microcontroller (esp32) and visual studio code will be used in this project to create python code that creates map to get the location.

2.6 SOFTWARE DEVELOPMENT LIFECYCLE(SDLC)

The Software Development Life Cycle (SDLC) is a framework that describes the procedures required to complete each phase of software development. It details the process of developing, deploying, and maintaining the programme.

The SDLC encapsulates the whole development lifecycle i.e. everything involved in the process of designing, developing, testing, and delivering a Software Product.

2.6.1 The Waterfall model

The waterfall model, also known as the linear-sequential life cycle model, is simple to understand and use. It is also widely used. This paradigm necessitates the completion of each step before moving on to the next one. It is most effective in projects when there is little likelihood of requirements changing after work has begun, as the waterfall technique makes such changes expensive. At the conclusion of each phase this model, a review is conducted to establish whether the project is on schedule or not.

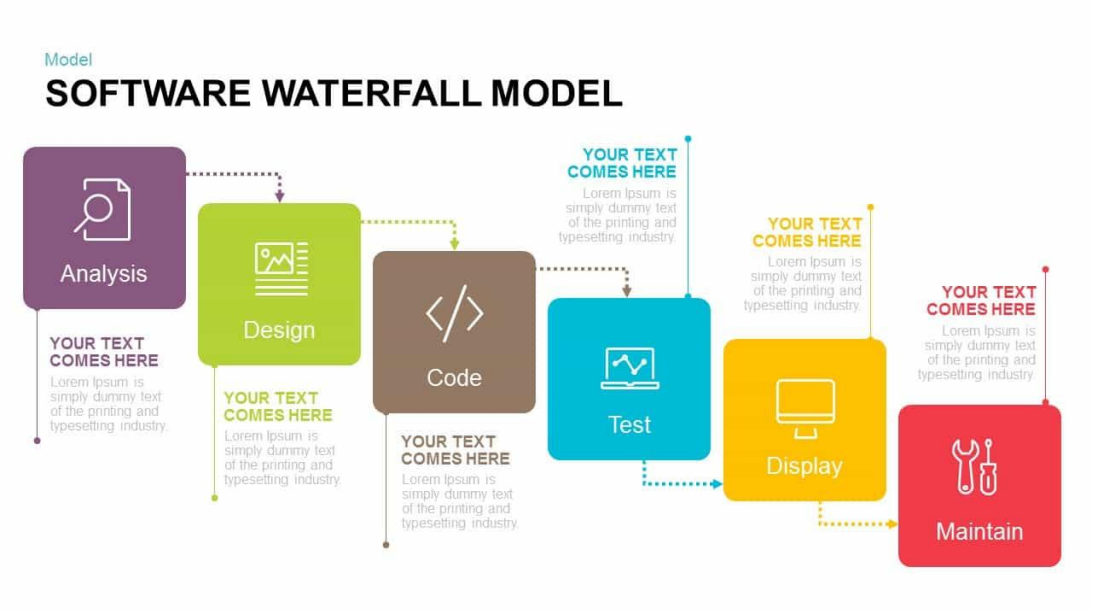


Figure 11: Waterfall model

2.6.1.1 Advantages of waterfall model

- Simple and straightforward in its comprehension and application.
- Each phase is handled and completed sequentially.
- It works best for smaller projects with well-defined criteria.
- Stages are well defined.
- Tasks are simple to organise.
- Both the process and the outcomes are thoroughly recorded.

2.6.1.2 Disadvantage of waterfall model

- Not a good way to start a project that is very complex or has many different parts.
- Poor model for long and ongoing projects.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

2.6.2 Iterative model

The iterative process model is the implementation of the software development life cycle in which the initial development is started based on the initial requirements and more features are added to the base software product with the ongoing iterations until the final system is created. The underlying principle of this approach is to build a system incrementally (i.e., iteratively) (incremental).

The following diagram depicts the Iterative and Incremental paradigm in action.

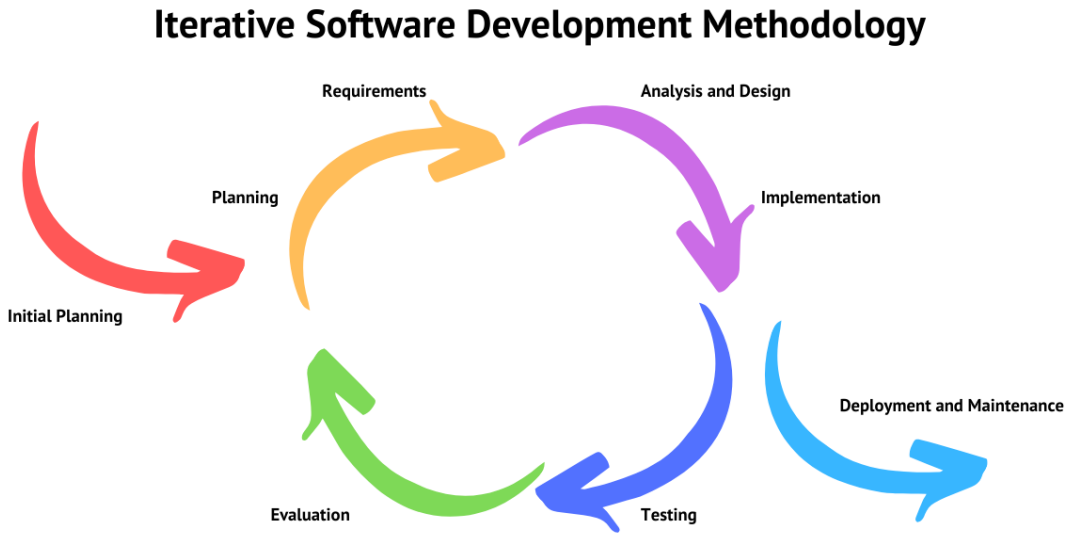


Figure 12: Iterative model

2.6.2.1 Advantage of iterative model

- Faster and more efficient software development is possible with this methodology.
- This model is very flexible. As new functionality can be added to it at any time of development.
- This model is a lot cheaper than the other process models because it costs less to change requirements in this model than in the other process models.
- The problems and flaws in the system can be found before they become big problems.

- Takes a smaller development team than other process models, which require a bigger team.

2.6.2.2 Disadvantage of iterative model

- Since all needs are not acquired upfront, system architectural issues might arise.
- The entire system must be defined to define increments.
- The level of management complexity is more.
- A project's development is heavily dependent on the risk assessment phase.

2.6.3 Spiral model

Iterations are conducted in accordance with the spiral model stages. The model's loops correspond to the stages of the SDLC process, with the innermost loop representing requirement collection and analysis and the outermost loop representing planning, risk analysis, development, and assessment. The next cycle is design, followed by implementation and finally testing.

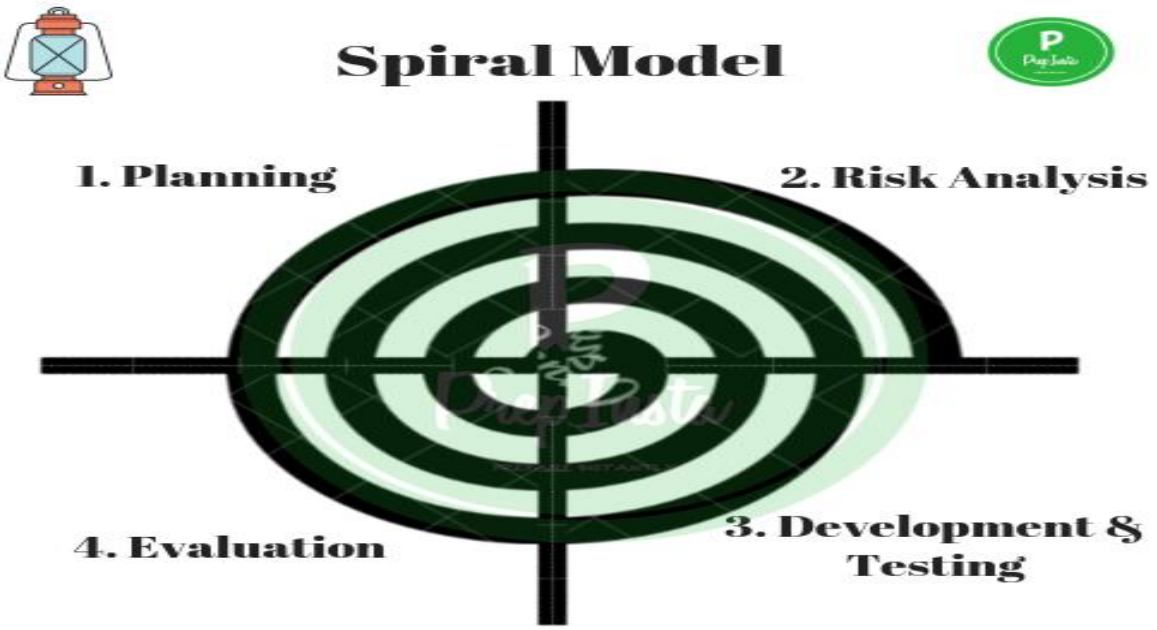


Figure 13: Spiral model

2.6.3.1 Advantage of spiral model

- Additional capabilities or modifications are possible at a later stage.
- Because the prototype is built in tiny pieces, cost estimating becomes simple.
- It's easy to add new features quickly with Spiral development, and they're introduced sequentially.
- Ideal for large-scale initiatives with a high stake.

2.6.3.2 Disadvantage of spiral model

- Using this approach may be expensive.
- Risk of missing a deadline or budget.
- Performing a risk analysis requires the use of highly specialised knowledge.

2.6.4 Agile model

Project management practises that encourage a collaborative, iterative, and incremental approach to project management is known as agile methodologies or agile project management methods.

A product's development is broken down into smaller, incremental steps under the Agile methodology. It is not created in its entirety in a single step. In terms of features, each new build is a step forward. The following build is a continuation of the prior one. According to the agile philosophy, each project should be approached uniquely, and current methodologies should be customised to the project's specific objectives. Agile divides jobs into time boxes (short time limits) to ensure that specified features for a release are delivered on schedule.

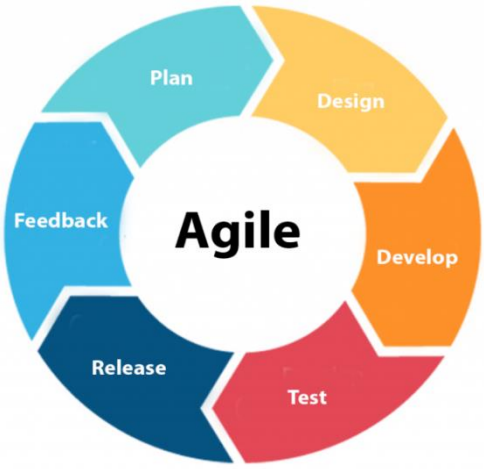


Figure 14: Agile methodology

2.6.4.1 Advantage of agile model

- Satisfaction of the customer by delivering valuable software in a timely and consistent manner.
- Consistent focus on technical proficiency and smart design.
- Issues and flaws are easier to notice and rectify quickly.
- Even at the final phases of development, changes to the criteria are allowed.

2.6.4.2 Disadvantage of agile model

- Since the features constantly change, there is a possibility that the project will continue indefinitely.
- It's hard to evaluate the amount of time and resources needed for complicated tasks.
- There isn't enough attention paid to the important parts of designing and documenting things.

2.7 CHOSEN SOFTWARE DEVELOPMENT LIFE CYCLE

After weighing the benefits and drawbacks of each software development lifecycle described in this report, it was determined that the optimal strategy for this project would be a modified simple agile Kanban board model. With this method, it is possible to maintain a systematic emphasis on phases while also allowing for changes in requirements to be implemented with little interruption to development. Appendix A provide the figure of Kanban board.

The biggest disadvantage of this strategy is determining the amount of time and resources required for a complex work, and there is also the risk that the project may continue eternally due to its continually changing characteristics.

2.8 CLOUD DATABASE STORAGE OPTIONS

Cloud storage is accessible in three configurations: private, public, and hybrid. Using cloud computing means you get IT resources when you need them, and you pay for them as you go. This is called pay-per-use pricing. In place of having to buy and own your own physical servers, you can get computing power, storage space, and databases from a cloud service provider when you need them. You don't have to buy, own, and maintain servers and data centres.

In the cloud, organisations and individuals may use computer resources including virtual machines, databases, processors, memory and services on demand. Pay-as-you-go cloud computing is also available. It's common for newer cloud services to improve on previous ones. The pay-as-you-go approach, for example, charges for resources only as they are used. You don't have to pay if you don't utilise any resources in cloud computing. Similar to the water or electricity metre, you have a metre to measure your monthly use. As a result, a certain cost is charged to cover your use. Some of the most popular cloud computing service providers are given below.

2.8.1 Amazon Web Service (AWS):

AWS (Amazon Web Services) is Amazon's extensive and developing cloud computing platform. It consists of a combination of infrastructure as a service (IaaS), platform as a service (PaaS), and packaged software as a service (SaaS) solution. AWS services may provide an organisation with resources like as computational capacity, database storage, and content delivery.

AWS provides a variety of tools and solutions for businesses and software developers that may be utilised in data centres located in more than 190 countries. AWS services are available to organisations of all sizes, including government agencies, educational institutions, NGOs, and for-profit businesses.

Amazon web services (AWS) offers three main types of cloud storage which are:

1. **Object storage:** Cloud-based applications often use object storage's enormous scalability and metadata capabilities. Object storage systems such as Amazon Simple Storage Service (Amazon S3) and Amazon Glacier are perfect for developing contemporary applications from the ground up that need scalability and flexibility. They may also be used to import existing data stores for analytics, backup, or archiving.

2. **File storage:** Numerous programmes need access to shared files, necessitating the implementation of a filing system. Typically, this kind of storage is managed by a Network Attached Storage (NAS) server. Elastic File System (Amazon EFS) is a file storage system that is suited for big content repositories, development environments, media stores, and user home directories.

3. **Block storage:** Other corporate applications, such as databases or enterprise resource planning systems, often need the use of dedicated, low latency storage for each server. The term "direct-attached storage" (DAS) or "cargo area network" is often used to describe this (CAN). Amazon Elastic Block Store (Amazon EBS) and Amazon EC2 Instance Storage are examples of block-based cloud storage systems.

2.8.1.1 Key feature of Amazon cloud storage:

- There are no upfront costs associated with this strategy; it is a pay-as-you-go arrangement.
- Access on a global scale: You may access all of your data from anywhere in the globe just by connecting to the internet.
- Storage capacity may be expanded or lowered in response to changes in the amount of the data.
- Low-cost data storage that maintains a high level of durability and availability
- Numerous options for backing up/archiving data in the event of a catastrophe recovery.

2.8.2 IBM Cloud

IBM Cloud Object Storage is a service provided by IBM that allows users to store and retrieve unstructured data in the cloud. On-premises deployments, as part of IBM Cloud Platform services, or in a hybrid configuration are all possible for the object storage service. The service may store any form of item, making it suitable for a variety of applications such as data archiving and backup, online and mobile apps, and scalable, permanent storage for analytics. Rest-based application programming interfaces (APIs) are used to interact with Cloud Object Storage. With built-in high-speed file transfer capabilities, cross-region offers, and integrated services, IBM Cloud Object Storage enables exponential data expansion and cloud-native applications. Data transmission is simplified with the IBM Aspera high-speed data transfer option, and variable storage class tiers assist minimise expenses while satisfying data access requirements.

2.8.2.1 Key feature of IBM cloud storage:

- Durability and resilience are quite high.
- Take control of your encryption keys. With IBM Cloud Identity and Access Management Services, you can create role-based policies.
- IBM Cloud Object Storage may be accessed quickly using IBM SQL Query. ETL (Extraction, Transformation, and Loading) is a simple way to access your data.
- Maintain electronic records and safeguard data against erasure or change.
- IBM Cloud Hyper Protect Crypto Services integration provides increased data protection, encryption choices, and granular control and authority.

2.8.3 Microsoft azure storage

Microsoft's Azure Storage platform is a cloud storage technology designed for current data storage situations. Azure Storage provides highly accessible, massively scalable, robust, and

secure cloud storage for a wide range of data items. Azure Storage data items are available via HTTP or HTTPS using a REST API from anywhere on the globe. Additionally, Azure Storage provides client libraries for developers using.NET, Java, Python, JavaScript, C++, and Go to construct apps or services. Developers and IT workers may develop scripts for data management and configuration activities using Azure PowerShell and Azure CLI. The Azure portal and Azure Storage Explorer offer graphical user interfaces for working with Azure Storage.

2.8.3.1 Key features of Azure storage

- Durable and highly available: Redundancy protects your data even if your hardware fails temporarily. In the case of an unforeseen interruption, this data is highly accessible.
- Secure: The service encrypts any data written to Azure storage accounts. Azure Storage allows you to restrict who gets access to your data.
- Scalable: Modern applications have high data storage and processing demands; therefore, Azure Storage was built to scale exponentially.

2.8.4 Google cloud storage

Google Cloud is a web-based file storage service that enables you to store and retrieve files utilising the Google Cloud Platform infrastructure. It enables global object storage that is scalable and very durable. It's one of the world's major public cloud storage systems, along with Amazon, Microsoft, IBM, and Oracle (opens in new tab). It provides unified object storage to businesses and developers alike. Google Cloud is a cloud computing platform that is geared at developers and businesses. It supports a variety of storage classes, enabling you to choose the price plan that is optimal for your data. These tiers are Standard, Nearline, Cold line, and Archive, and are defined by the frequency with which you need access to your data, as well as the speed and durability of the storage required.

Google One is a consumer cloud storage service that includes Google Drive, Gmail, and Google Photos, although it is distinct from Google Cloud, which is a commercial cloud storage service.

2.8.4.1 Key feature of google cloud storage

- Because of its low latency and long endurance, it is a good choice (up to 99.9999 percent)
- It offers limitless storage with no need for a minimum item size.
- It provides consumers with a dependable and safe method of storing their objects.

2.8.4.2 Key feature of google consumer cloud storage

- It allows Importing data in tables
- Reference data from other spreadsheets
- It Embed spreadsheets on websites.
- Allows Customization spreadsheets with script editor

2.9 CHOSEN CLOUD STORAGE OPTION

After analysing the most popular cloud storage provider in terms of their price and their features and the service they provide Google consumer cloud service also known as Google one, where it provides storage for different types of media/photo/files/spreadsheets is going to be used in this project. From google one, google sheets will be used in this project to receive and store data from the gps device. Google one, provides 15GB of free space, If the 15GB of free storage space isn't enough, there are a variety of paid storage options available. When compared to other cloud storage alternatives, Google One is a priced choice. If you're already a frequent user of Google goods, this is a fantastic service.

2.10 COMPONENTS AND SOFTWARE TO BE USED IN THIS PROJECT

This section will give an in-depth discussion of the many possibilities accessible in terms of the components considered necessary; the only thing that will be needed is to present alternatives as well as the justification for the choice of the already available components. The following are the basic components that must be present:

2.10.1 Hardware to be used in this project:

- Microcontroller (ESP32)
- GPS tracking device
- Accelerometer for tracking the movement also.

2.10.2 Software to be used in this project:

- Arduino IDE for doing the coding on Microcontroller.
- Python programming language for displaying the location on desktop application
- Libraries linked to sensors.
- Google consumer cloud storage (google sheets) for database storage.

2.11 LITERATURE REVIEW SUMMARY

After doing in-depth research on hardware and software to be related in this project the modified agile model has been chosen to keep track of the software development in systematic way. Chosen environment to develop the code and chosen database option to store data are clearly mentioned right above this section along with chosen hardware.

3 CHAPTER 3-REQUIREMENT GATHERING AND ANALYSIS

The requirements collection and finalisation, as well as an in-depth assessment of project risks, are all included in this chapter.

The term "software requirement" refers to a specification of a specific feature or function of a given computer system. They may alternatively be defined as "descriptions of the services a software system must deliver and the limitations under which they must function."

3.1 REQUIREMENT PRIORITISATION STRATEGY.

I have chosen Moscow method. It is a Popular method to prioritize requirements which helps to understand the amount of efforts and resources that will be required for each project element.

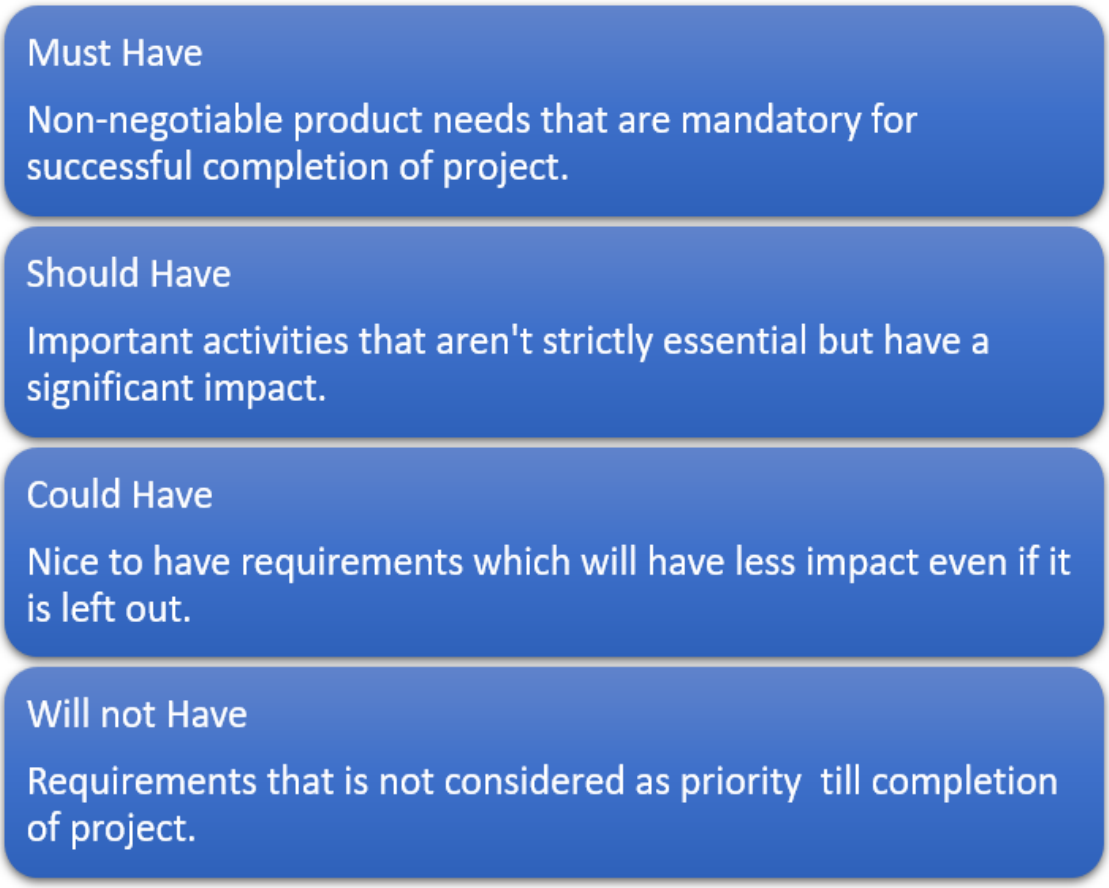


Figure 16: Moscow requirement prioritisation strategy

MoSCoW analysis enables teams to choose which needs or activities to prioritise. MoSCoW was created in 1994 by Oracle employee Dai Clegg and is the most often used prioritising strategy in agile projects. While the MoSCoW technique is often used to prioritise client needs on projects, it may also be employed in non-project situations. For instance, to prioritise business-as-usual activities inside an organisation or to prioritise personal tasks personal projects at home.

3.2 PRIORITIZED REQUIREMENT LIST

Effective requirement prioritisation is critical to the success of any project. It guarantees that the project prioritises the most critical parts first and that everyone knows and agrees on what those elements are. A well-organized set of requirements will also guarantee that engineers, programmers, and database analysts build the most crucial components of a project in accordance with business objectives. The following tables contains the list of prioritized requirements:

Title	Priority
Microcontroller and GPS device	Must have
Code to display location on map	Must have
Cloud based	Should have
Accuracy	Should have
Time response	could have
Protecting case for Microcontroller and GPS IOT device	Won't have

Table 1: Prioritized list

3.3 REQUIREMENTS

Following significant research into the project's requirements, a comprehensive list of requirements was compiled. The requirement of this project defines how the GPS/IOT system works and what it should do to accomplish this project.

3.3.1 Functional Requirements

The functional requirements for this project define what a system "should" perform; they often define a behaviour or function and represent the GPS/IoT system's mission. The following table 2 covers the functional specifications for this project.

Functional requirements of GPS/IOT system
• The device should connect to GPS for obtaining its own geographical location. (Latitude, Longitude, Altitude, Date and time).
• The device then should send geographical location based on latitude and longitude to google consumer cloud (google sheets).
• A python programming code to display the location on map.

Table 2: Gps/Iot functional requirements

3.3.2 Non-Functional Requirements

System qualities such as reliability, performance, maintainability, scalability, and usability are defined by non-functional requirements. Below table contains non-functional requirements of this project.

Non-functional requirements of GPS/IOT system
• The Gps/Iot device should be portable.
• The Gps/Iot device should be user friendly.
• The Gps/Iot device should use less electrical energy.
• The stored data in google sheets should not crash.
• The coding to display location can be python or c#.

Table 3: Gps/Iot non-functional requirement

3.4 RISK ANALYSIS

In Simple terms, risk analysis is the process of detecting potential threats to the project's success. A risk is nothing more than a possible issue that might arise throughout the software development lifecycle. Any identified risks must be controlled to prevent the project from being harmed, corrupted, or completely lost.

Risk analysis comprises identifying all possible hazards associated with the project and developing remedies if a risk becomes evident. Risk table containing risk description, severity, probability and impact for this project is provided below:

Risk descriptions	Severity	Probability (1-5)	Impact
Most have requirements being neglected	Critical	1	Catastrophic
Secondary requirements being neglected	Medium	1	Serious
Modification of requirements	Medium	1	Serious
Deficient planning	Critical	1	Catastrophic
Bugs and errors in coding	Medium	4	Tolerable
Hardware/software failure	Critical	1	Catastrophic
Medical issues	Critical	3	Serious
Lack of communication with supervisor	Medium	2	Tolerable
References and plagiarism in final reports	critical	1	Serious

Table 4: Risk analysis table

3.5 RISK MITIGATION STRATEGY

Developing a risk mitigation plan is an important step in the risk management lifecycle. Following the identification of risks and the assessment of the possibility of their occurrence, as well as the potential effect they may have, the following table outlines how they should be dealt with in this project.

Risk description	Mitigations
Most have requirements being neglected	Start collecting needy hardware and software as early as possible.
Secondary requirements being neglected	Get help from supervisor and mention it in final report.
Modification of requirements	request supervisor's advice throughout implementation to verify requirements are established appropriately and completed.
Deficient planning	Handling project complexity and deadline pressures with the help of Gantt charts and lots of GitHub comments
Bugs and errors in coding	Check in GitHub or in a stack overflow or get help from supervisor.
Hardware/software failure	Back up the implemented data in a device to a cloud storage or in case of hardware failure replace it with a new one.
Medical issues	Inform to supervisor or module instructor.
Lack of communication with supervisor	Try to schedule meeting with the supervisor on a weekly basis.
References and plagiarism in final reports	Double check the report in detail and use the appropriate refencing

Table 5: risk mitigation strategy

3.6 PROJECT TIME MANAGEMENT ANALYSIS

For the better time management to complete the project in given time period the Gantt chart has been created where the date and time that could a task use in the project is specified so that any breakdown work could be accomplish in time. The appendix B provide the Gannt chart. The time for each task which is separated and shown in Gantt chart is calculated with PERT estimation technique.

3.7 CONCLUSION

Following the specification of requirements, the design stage might commence. It is critical to keep in mind that, despite the fact that these criteria have been carefully considered, they may change throughout the course of the project's development. Due to the agile model's adaptability, this is not a significant issue, but all required modifications or amendments will be actively monitored and handled during the development process.

4 CHAPTER4 - DESIGN

The major goal of this phase is to turn the requirements from the previous phase into a comprehensive system design that is both practicable and reliable. As a result, accuracy and completeness in all designs are critical to the success of a project. It is possible that poor design will result in requirements not being satisfied.

Everything in this chapter is designed in accordance with the specifications provided in chapter 3. It contains block diagram of GPS/Iot device, actual diagram of device with detail information of used hardware and flow chart of Gps/Iot functioning system.

4.1 BLOCK DIAGRAM

The Global Positioning System (GPS) operates on the premise of satellite communication. The GPS module is depicted in the On-Board Block Diagram. After receiving the signal from the satellite, it communicates with the ESP32-WROOM-32 microcontroller, which subsequently executes the command. Then, via the internet, this microcontroller sends a signal to the GPS transmitter. All of the components are wired together using jumper wires to the appropriate source of power through a breadboard or circuit board.

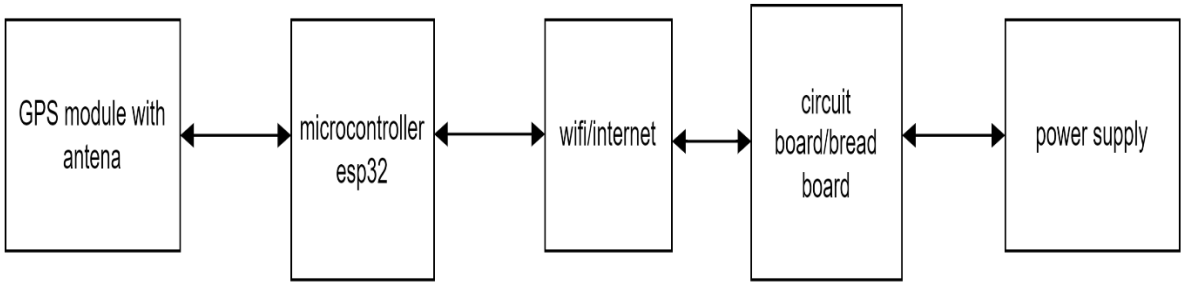


Figure 17: on board block diagram of gps/iot system

4.2 SYSTEM FLOW DIAGRAM

The following system flow chart depicts the working principle of this system where each block represents its work from start to end.

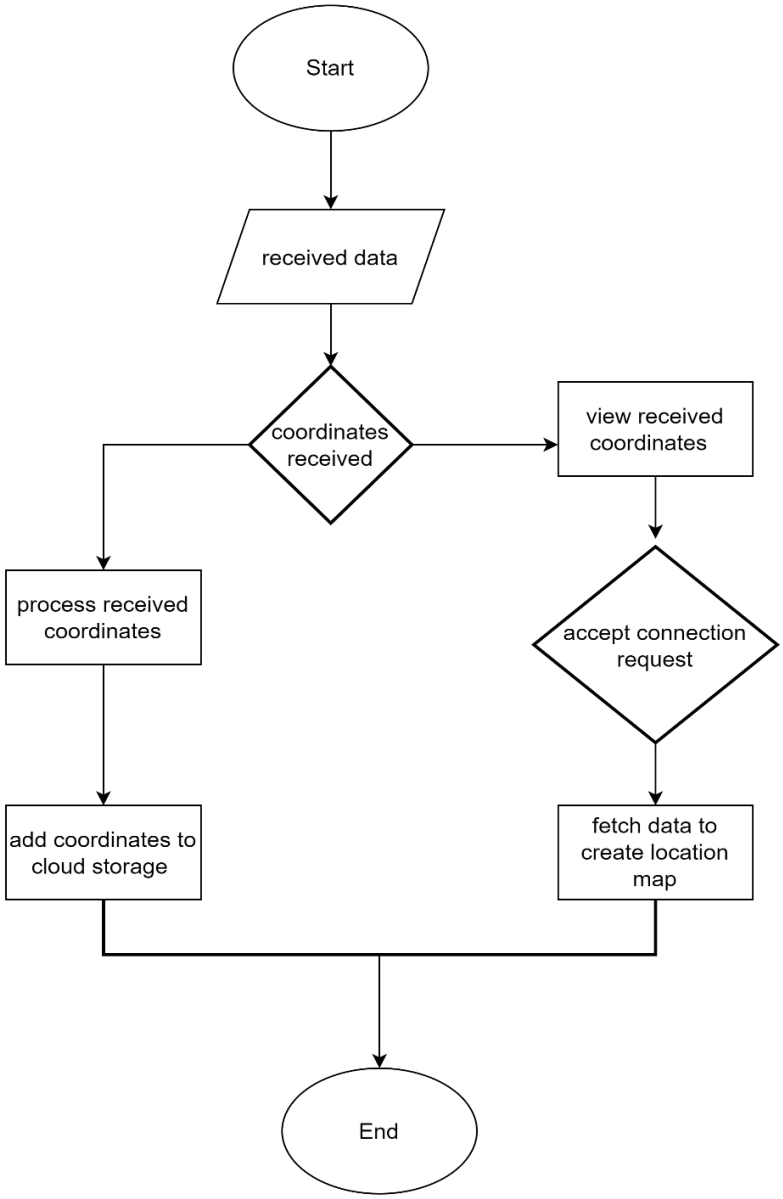


Figure 18: System flow diagram

4.3 DETAIL DESIGN OF USED COMPONENTS IN THIS PROJECT

The complete design of actual component that are used in this project are:

4.3.1 GT-U7 GPS Module with EEPROM And Active Antenna

This is an open-smart small GPS module that uses the GT-U7 GPS receiver. A USB serial interface is also included in the module, which may be accessed through a micro-USB connection or directly through its 0.1" pitch serial TTL header (supplied header pins require soldering).

The module will automatically produce GPS data in the standard National Marine Electronics Association (NMEA) format right out of the box, making it compatible with a wide range of devices and applications without the need for configuration. The module also includes a detachable antenna that attaches to the module via a U.FL pigtail cable, allowing the antenna to be placed in the best location possible. U.FL Connections are small surface-mount RF coaxial connectors. These high-frequency connectors are appropriate for a wide range of small devices that require a compact connector solution. They have a 50-ohm impedance.

Figure 19: Gps module

Figure 20: Gps module with pin layout

GND: This is the ground pin that will be connected with the ground of the ESP32 board.

TX: This is the transmission pin used for serial communication.

RX: This is the receiver pin used for serial communication.

VCC: This is the VCC pin used to power up the GPS module. Connect it with the 3.3V of the ESP32 board

4.3.1.1 Product technical specification

These are the main technical specification of the gps module which comes with the product that will be used in this project.

Product code: HCMODU0172
Antenna dimensions: 15.3 x 15.3 x 6.8mm
Operating frequency: L1 (1575.42 +/- 10MHz)
Operating voltage: 3.3 - 5.2V
Operating current: normal mode is 50mA, power-saving mode is 30mA
Communication interface: TTL serial port, micro-USB interface (can be connected to a computer to debug)
Serial port baud rate: 9600bps
Communication format: 8N1
Interface logic voltage: 3.3V or 5V
External antenna interface: IPX.
Module dimensions: 27.5 x 26.5 x 4mm

Table 6: Gps module specification

4.3.1.2 Product key features

- Even in difficult to locate environments, such as restricted urban skies or dense jungle environments, the GT-U7 GPS receiver module can provide high-precision locating in these situations.
- The GT-U7 module's high sensitivity, low power consumption, compactness, and exceptionally high tracking sensitivity have allowed it to significantly increase the coverage of its placement.

- Using a USB port, you may easily connect it to your computer. Because the host computer provides the serial port capability, an additional serial module is not required to transfer data over the IPX interface active antenna.

4.3.2 ESP32 WROOM 32D

ESP32 WROOM 32D is a powerful generic wi-fi plus Bluetooth modules that target a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks.

The chip that has been implanted is intended to be scalable and adaptable. There are two CPU cores that can be operated independently, and the clock frequency of the CPU may be adjusted from 80 MHz to 240 MHz. As an additional feature, the chip has a low-power coprocessor that may be used in place of the CPU to save power while performing activities that do not need a lot of computational power, such as peripheral monitoring. The ESP32 incorporates a diverse variety of peripherals, including capacitive touch sensors, Hall sensors, an SD card interface, and an Ethernet interface.

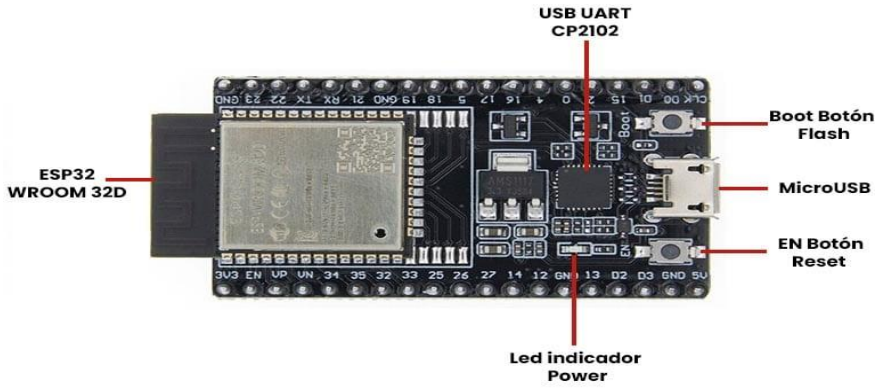


Figure 21:ESP32-WROOM 32D

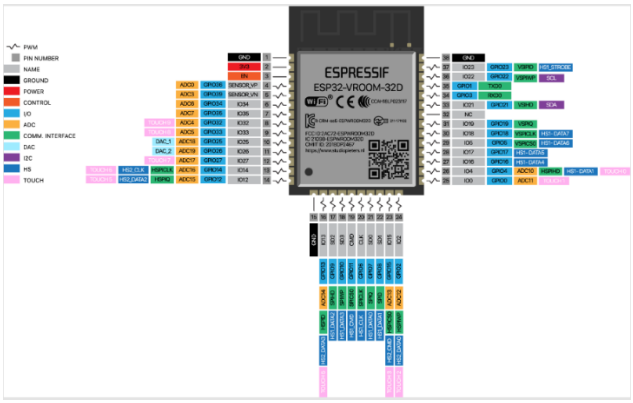


Figure 22: Pin layout of ESP32 WROOM 32D

4.3.2.1 Specification of ESP32 WROOM 32D

The detail features and specification of chosen ESP32 WROOM 32D for this project is given below in understandable form

- Processors:
  - CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 Dhrystone|DMIPS
  - Ultra-low power (ULP) co-processor
- Memory: 520 KiB SRAM
- Wireless connectivity:
  - Wi-Fi: IEEE 802.11|802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR and BLE
- Peripheral interfaces:
  - 12-bit successive approximation ADC|SAR ADC up to 18 channels
  - 2 × 8-bit digital-to-analog converter|DACs
  - 10 × touch sensors (capacitive sensing GPIOs)
  - Temperature sensor
  - SDIO/SPI slave controller
  - Ethernet MAC interface with dedicated DMA and Precision Time Protocol|IEEE 1588 Precision Time Protocol support
  - CAN bus 2.0
  - Infrared remote controller (TX/RX, up to 8 channels)

  - Motor Pulse-width modulation|PWM
  - LED Pulse-width modulation|PWM (up to 16 channels)
  - Hall effect sensor
  - Ultra-low power analog pre-amplifier
- Security:
  - IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WLAN Authentication and Privacy Infrastructure|WAPI
  - Secure boot
  - Flash encryption
  - 1024-bit OTP, up to 768-bit for customers
  - Cryptographic hardware acceleration: Advanced Encryption Standard|AES, SHA-2, RSA (cryptosystem)|RSA, elliptic curve cryptography (ECC), random number generator (RNG)
- Power management:
  - Internal low-dropout regulator
  - Individual power domain for RTC
  - 5uA deep sleep current

4.3.3 Bread board

A breadboard is a rectangular plastic board with several small holes. These holes make it simple to insert electrical components to prototype (create and test an early version of) an electronic circuit, such as this one with a battery, switch, resistor, and LED (light-emitting diode). Because the connections are not permanent, it is simple to delete a component if you make a mistake, or just start again and develop a new project. This makes breadboards ideal for novices who are new to electronics.

Modern breadboards are composed of plastic and come in a variety of forms, sizes, and colours. While bigger and smaller sizes are available, "full-size," "half-size," and "micro"

breadboards are the most frequent. Most breadboards also include tabs and notches on the edges that allow you to connect numerous boards. A single half-sized breadboard, on the other hand, is plenty for many beginner-level projects.

In this project the full-size bread board has been used. At the time of purchase the small size bread board was unavailable so the full-size bread board has been used. If the small size bread board is used it will be easily portable and can be easily fitted to make dog collar tag.

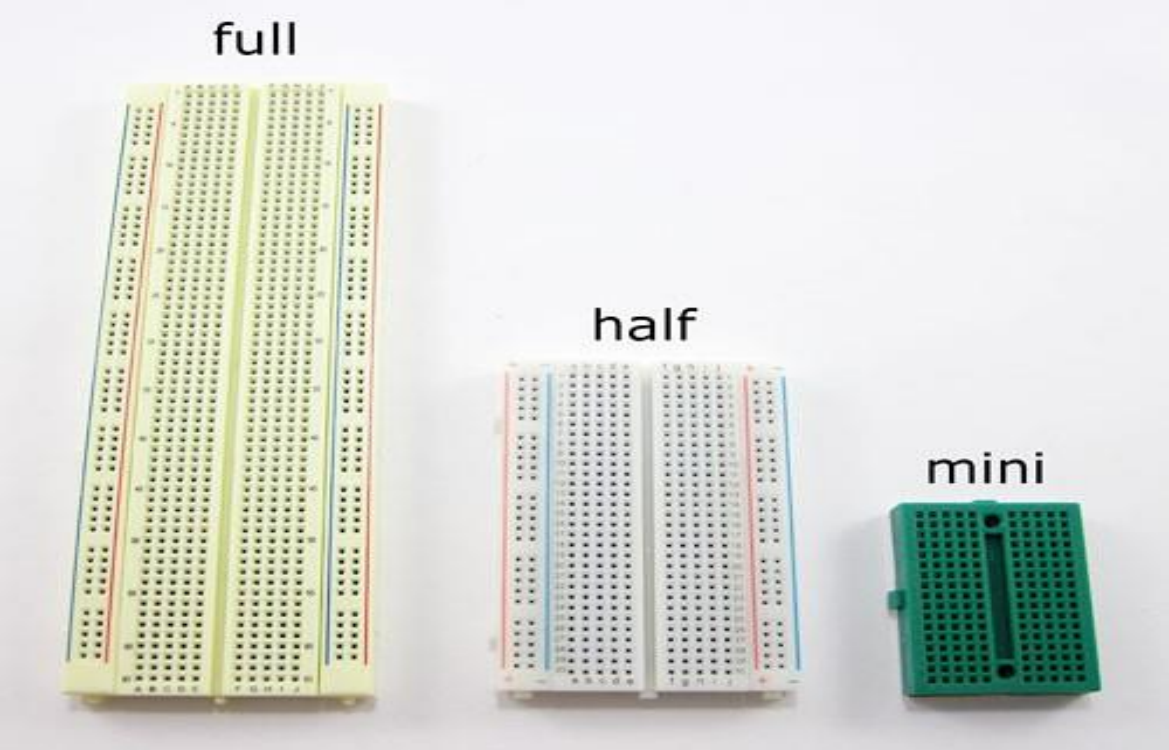


Figure 23 : Bread board

4.3.4 Jumper wire

Jumper wires are simple cables with connector pins at either end that may be used to connect two places without soldering. Jumper wires are commonly used with breadboards and other prototyping tools to allow for quick circuit changes as needed.

Although jumper wires are available in several colours, the colours themselves do not indicate anything. This means that a red jumper wire is the same as a black jumper wire in terms of functionality. However, you may utilise the colours to your advantage in order to identify between different sorts of connections, such as ground and power.

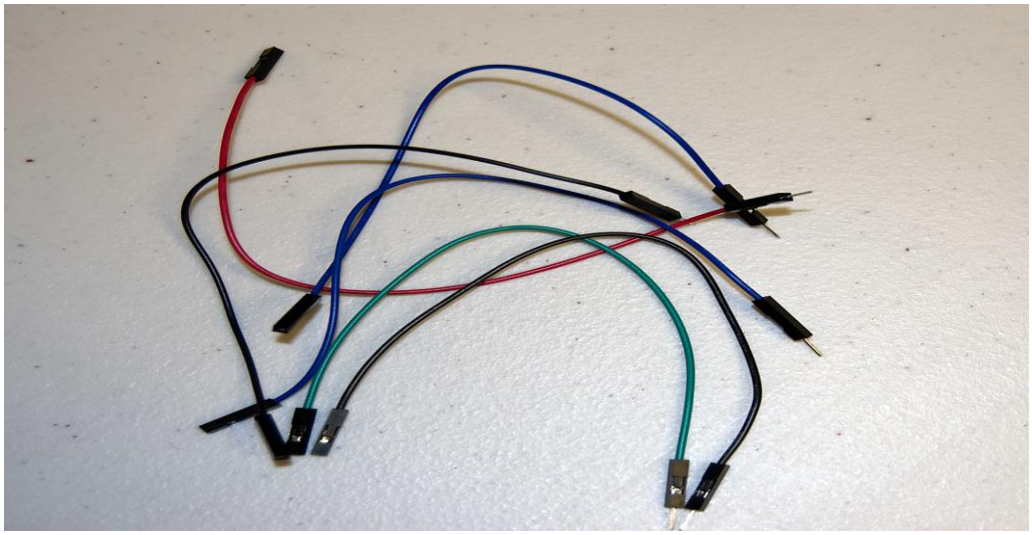


Figure 24: jumper wire

4.4 DESIGN/PROTYPE OF THE SYSTEM

In this project the design of the Gps/Iot contains all of the above-mentioned parts in this chapter. The following prototype has been designed and wired with jumper wire along with ESP32 WROOM 32D, GPS module, and Bread board also a 9-volt battery. A separate picture of each device is also added to understand the devices that are actually used in this project.
The cost of the hardware to set up the prototype was around £16.47 which is cheaper than the existing device price and for the data storage user don't need to pay extra because it will be using google cloud service for free.
The actual design will be the future work with smaller integrated device that can be easily fit in the collar tag of the pets.

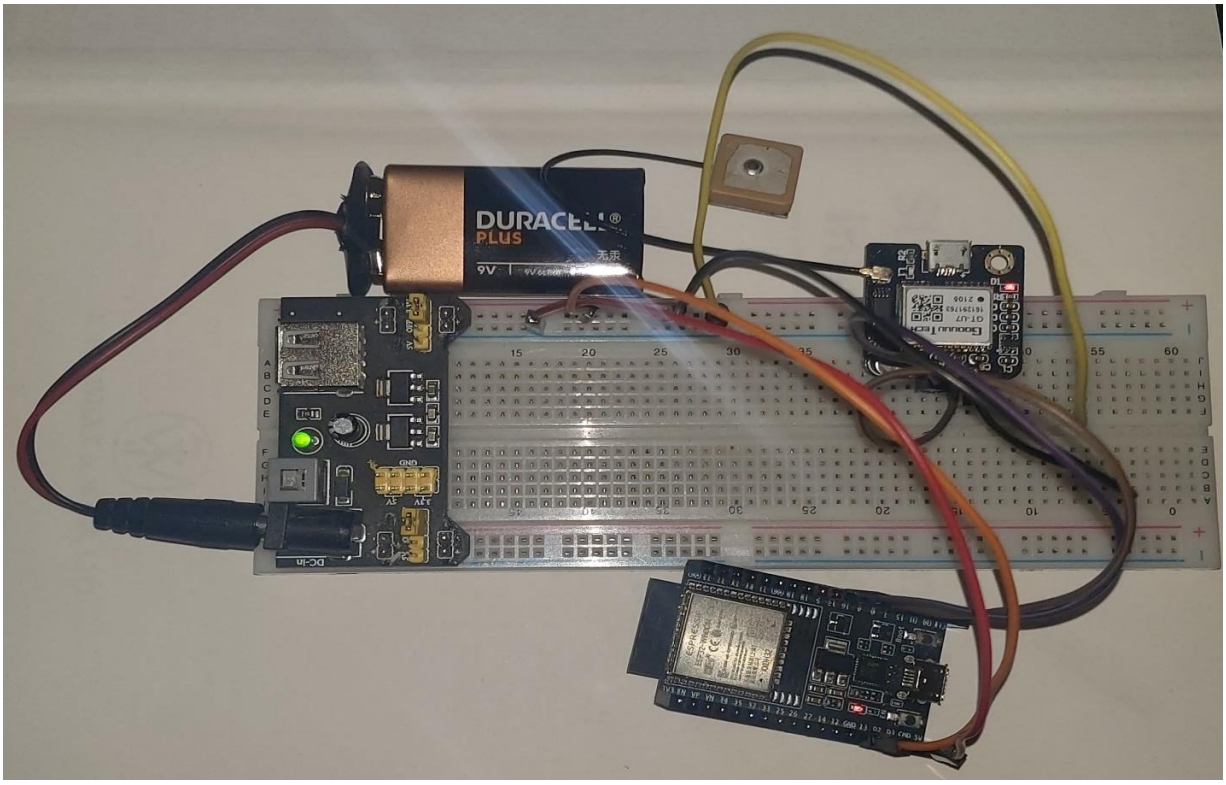


Figure 25: Prototype of Gps/Iot device connected with jumper wire in a bread board with a battery

Figure 27:ESP32 WROOM 32D

Figure 26: 9-volt battery

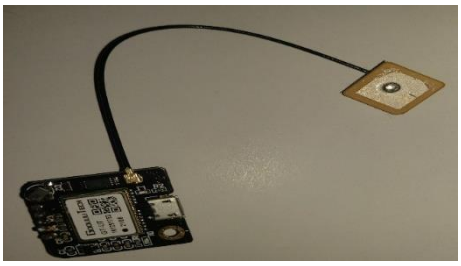


Figure 28: Gps module with sensors

5 CHAPTER 5 - IMPLEMENTATION

This chapter contains the development language used for coding into ESP32 device and coding used for creating map for the location also the testing part will be described after the completion of successful implementation.

5.1 CHOSEN DEVELOPMENT LANGUAGE

5.1.1 Arduino programming language

Arduino programming language is chosen for the coding purpose in the microcontroller (ESP32). The Arduino code is written in C++, with the addition of certain unique methods and functions that we'll go through later. C++ is a computer language that is easy to understand.

When you develop a "sketch" (an Arduino code file), it is analysed and compiled into machine language. Functions, values (variables and constants), and structure are the three primary aspects of the Arduino programming language.

The detail explanation of working principle of the code which is used to code in ESP32 microcontroller along with screenshot of code is given below

```
..
#include <WiFi.h>
#include <HTTPClient.h>
//------------------------------

//-----------------------------------------------
#include <TinyGPS++.h>
TinyGPSPlus gps;

#define RXD2 17
#define TXD2 16
HardwareSerial neogps(1);
//-----------------------------------------------
```

Figure 29: Installing libraries

First and foremost, we'll need to add libraries for connecting to a WiFi network and making HTTP requests in order to retrieve data from the GPS device. These three directives were used to install the libraries: #include <WiFi.h>, #include <HTTPClient.h>, and #include <TinyGPS++.h>. We may use #includeGSM.h> to install libraries that allow us to connect to the internet as well as send/receive text sms if we require the Gsm module.

#define RXD2 and #define TXD2 are used to name constant values 17 and 16, respectively, which are ESP32 port numbers in use.

The GPS module will interact with the Arduino using the Hardwareserial neogps(1); command.

```
//
//ENTER_GOOGLE_DEPLOYMENT_ID
const char * ssid = "BT-M6AF7G";
const char * password = "nPifRyk9DLNuUp";
String GOOGLE_SCRIPT_ID = "AKfycbyY2tWjKq0qgsBevivxEH1KCr2XzOBvPNPOWmMMLVazBypNN7i9XCzaBG_HXuEfxaDK";
//----------------------------------------------------------------------
const int sendInterval = 2000;
```

Figure 30: block of code for the credentials

After installation of necessary libraries, we need to give credentials of Wi-Fi to be used and google script id where the data will be stored. const char * ssid is to store the constant value which is a character(name of the wifi) and const char * password is to store the value which is also a character(password of wifi}. String GOOGLE_SCRIPT_ID is used to store the value of generated google script id and is string value. const int sendInterval is specifying the integer value which is time in millisecond (2000 millisecond=2 second) and used here to send the analog data from ESP32 to Gps module in given time interval. The process to get the google id script will be discuss later in this chapter.

```
/*****************************************************************************************
 *  setup function starts
 *****************************************************************************************/
 void setup() {
  //-------------------------------------------
  Serial.begin(115200);
  //start serial communication with Serial Monitor
  //-------------------------------------------
  //start serial communication with Neo6mGPS
  neogps.begin(9600, SERIAL_8N1, RXD2, TXD2);
  delay(10);
  //-------------------------------------------
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("OK");
  //-------------------------------------------
```

Figure 31: void function block

After specifying the needed credentials in the previous block of code the setup function() is created in the above figure 26 where Serial.begin(115200); is used to communicate with serial monitor of Arduino Ide where the baud rate number of monitor is 115200. neopgps.begin is specifying the requirement of gps device in order to communicate with gps device. WiFi.mode(WIFI_STA); is used to set the wifi station mode in ESP32 so that it connects to the access point. Then we call the begin method on the WiFi object, passing as arguments the SSID (network name) and password variables specified early. This will start the connection to the network. The Serial.print will print out the given arguments in the serial monitor of Arduino Ide. Then the while loop has been introduced and it will run until the connection is established effectively,to do so the status method on the wifi object has been called and then it waits for the result to match the enum value != WL_CONNECTED. Between each iteration, we introduce a small delay, to avoid a constant poll. Then the Serial.println will print the passing argument “OK”.

```
void loop() {

  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (neogps.available())
    {
      if (gps.encode(neogps.read()))
        {newData = true;}
    }
  }

  //If newData is true
  if(newData == true)
  {
    newData = false;
    Serial.println(gps.satellites.value());
    print_speed();
  }
  else
  {
    Serial.println("No new data is received.");
  }
```

Figure 32: loop to get data from gps to microcontroller

In this block of code Boolean data type specify the initial state as false. Then for loop start and run to receive gps coordinates and store in given (start) variable in unsigned long data type. unsigned long variables are extended size variables for number storage. Then while loop starts with the condition of (neogps.available()). After the condition of while loop the IF statement starts with the condition (gps.encode(neogps,read())). When If statement is true that is if(newData == true) it will print the value of Serial.println . and if the statement is false then again it prints the value of Serial.println of else statement that is Serial.println(“no new data is received”) in the serial monitor of Arduino Ide

```
 *  if_else statement
 ********************************************************************************************/
void print_speed()
{
  if (gps.location.isValid() == 1)
  {
    //String gps_speed = String(gps.speed.kmph());
    Serial.println(gps.location.lat(),6);
    Serial.println(gps.location.lng(),6);
    Serial.println(gps.speed.kmph());
    Serial.println(gps.satellites.value());
    Serial.println(gps.altitude.meters(), 0);

    String param;
    param  = "latitude="+String(gps.location.lat(), 6);
    param += "&longitude="+String(gps.location.lng(), 6);
    param += "&speed="+String(gps.speed.kmph());
    param += "&satellites="+String(gps.satellites.value());
    param += "&altitude="+String(gps.altitude.meters());
    param += "&gps_time="+String(gps.time.value());
    param += "&gps_date="+String(gps.date.value());

    Serial.println(param);
    write_to_google_sheet(param);
  }
  else
  {
    Serial.println("No any valid GPS data.");
```

Figure 33: if else statement to print coordinates in serial monitor

In this block of code if else statement is used inside the print_speed function where if the condition if (gps.location.is valid()==1) then it will print received coordinates from gps device to serial monitor of Arduino ide. Here string variable is assigned to store coordinates in long string format in the given name “param” then Serial.println(param); will print stored data in google sheet. If the IF statement is not satisfied the Else statement will print out the value of Serial.println of Else statement which is “No any valid GPS data.”. Figure 31 and figure 32 are the block of code to communicate with Arduino serial monitor, ESP32 along with gps device

```
/*****************************************************************************************
 *  loop function starts
 *****************************************************************************************/
void write_to_google_sheet(String params) {
   HTTPClient http;
   String url="https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?"+params;
   //Serial.print(url);
    Serial.println("Postring GPS data to Google Sheet");
    //----------------------------------------------------------------------
    //starts posting data to google sheet
    http.begin(url.c_str());
    http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    int httpCode = http.GET();
    Serial.print("HTTP Status Code: ");
    Serial.println(httpCode);
    //----------------------------------------------------------------------
    //getting response from google sheet
    String payload;
    if (httpCode > 0) {
        payload = http.getString();
        Serial.println("Payload: "+payload);
    }
    //----------------------------------------------------------------------
    http.end();
}
```

Figure 34: http get request

In this block of code the write_to_google_sheet(String params) function is created where in string url the url of script.google plus the GOOGLE_SCRIPT_ID is given. The value for the GOOGLE_SCRIPT_ID is assigned early with the wifi credentials can be seen in figure 29. The begin method calls http object which has url.c_str() in the value. c_str returns a const char* that points to a null-terminated string (i.e., a C-style string). It is useful when you want to pass the "contents" of an string to a function that expects to work with a C-style string then http.setFollowRedirects methods has been used in order to get data in google sheets . The setFollowRedirects is the method of Java HttpURLConnection class. This method is used to set HTTP redirects (requests with response code 3xx). It returns true by default. Integer withvariable name httpCode is assigned with http.GET(); in order to get the http get request. And then string with variable name payload; assigned and then to conform that the response from google sheet is coming if statement is assigned with (httpCode > 0) where payload is defined with http.getString(); and then finally print payload with Serial.println. and at last the code is end with http.end();}

5.1.2 Python programming language

Python programming language is used to create map in order to show the location of gps coordinates which is stored in google sheet. Python is a high-level, general-purpose programming language. Its design philosophy emphasises code readability through the use of extensive indentation. Its language elements and object-oriented approach are intended to assist programmers in writing clear, logical code for small and large-scale projects.

Figure 35: importing libraries

At first, we need to install libraries to successfully run the code in python, here the import statement is importing pygsheets, pandas, numpy, and gmplot from tokenize. pygsheets in python is a simple, straightforward Python module for using the Google Sheets API v4 to access Google spreadsheets and pandas is used for data manipulation and analysis similarly, numpy is used to add support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on arrays whereas gmplot allows to plot data in google maps.

Apikey is generated while deploying google sheets in google cloud and defined here in apikey. Then gc function is defined with pygsheets.authorize(service _account_file='location of.jsonfile') to authorize python .json file. And sh is defined with gc.open_by_key(,value'). The value of key is obtained from google spread sheet URL.

```
#select the first sheet
size=20  #how many line of data you are fetching
wks = sh[0]
lat=[]
long=[]
latavg=0.0
longavg=0.0
for i in range (2, size+2):
    if (wks.cell((i, 2)).value!=""):
        lat.append(float(wks.cell((i, 2)).value))
        long.append(float(wks.cell((i, 3)).value))
        print(lat[i-2]," ", long[i-2])
        latavg= latavg+lat[i-2]
        longavg= longavg+long[i-2]

latavg=latavg/size
longavg=longavg/size
print(latavg,"    ", longavg)
```

Figure 36: defining size to picked from the google sheet

Here size =20 stands for the data of google sheets that will be used to create and plot in map. Similarly, wks = sh[0] will open the work sheet of spreadsheets. Then latitude and longitude is define with lat=[], long=[] respectively. Latitude and longitude average is also defined before the loop. Then for loop is introduced with given range to print the value. After that if statement is introduced to return the work sheet cell value. The append method is used to add a single item into the existing lat and long(list). Then in print (“value”) will print the defined lat and long value from the worksheet cell. For the average value of latitude and longitude latavg = latavg/size and longavg=longavg/size is defined. Then print (“value”) will print the average value of latitude and longitude.

Below Figure 36 is about plotting the fetched latitude and longitude in the google map and is performed with gmplot. The scatter here is used to create the relation by connecting dots with its given size and the colour representation then, .draw will draw the map and stored it in a given path ("value"). Then finally when all the statements are fulfilled the print will print its value

```
gmap1 = gmplot.GoogleMapPlotter(latavg, longavg, 14, apikey=apikey)
gmap1.scatter(lat, long, '#00008B', size = 30, marker = False )

gmap1.plot(lat, long, 'FF0000', edge_width = 2.5)
gmap1.draw( "C:\\Users\\Bibek\\Desktop\\gpsproject\\map1.html" )


print("Finish fetching the data from Bibek's_google_sheet")
```

Figure 37: Map plotting

5.1.3 Java script code

A java script code is also used in this project to receive the gps coordinates data from ESP32 to google spreadsheet in this project. In order to obtain the google script id the script is deployed in the script editor of google sheet. Once successfully deployed then google sheet provides the script id.

While coding Arduino programming language, google script id has been used which is generated by java script in the google sheet spread editor.

```
var ss = SpreadsheetApp.openById('1Ezehmls7bmUd3TzxKvzF5tFmd4W4fKh9vTEbR5jWj74');
var sheet = ss.getSheetByName('Sheet1');


//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
function doGet(e){
  //----------------------------------------------------------------------------------
  //write_google_sheet() function in esp32 sketch, is send data to this code block
  //----------------------------------------------------------------------------------
  //get gps data from ESP32
  if (e.parameter == 'undefined') {
    return ContentService.createTextOutput("Received data is undefined");
  }
  //----------------------------------------------------------------------------------
  var dateTime = new Date();
  latitude    = e.parameters.latitude;
  longitude   = e.parameters.longitude;
  speed       = e.parameters.speed;
  satellites  = e.parameters.satellites;
  altitude    = e.parameters.altitude;
  gps_time    = e.parameters.gps_time;
  gps_date    = e.parameters.gps_date;
  //Logger.log('latitude=' + latitude);
  //----------------------------------------------------------------------------------
  var nextRow = sheet.getLastRow() + 1;
  sheet.getRange("A" + nextRow).setValue(dateTime);
  sheet.getRange("B" + nextRow).setValue(latitude);
  sheet.getRange("C" + nextRow).setValue(longitude);
  sheet.getRange("D" + nextRow).setValue(speed);
  sheet.getRange("E" + nextRow).setValue(satellites);
  sheet.getRange("F" + nextRow).setValue(altitude);
  sheet.getRange("G" + nextRow).setValue(gps_time);
  sheet.getRange("H" + nextRow).setValue(gps_date);
  //----------------------------------------------------------------------------------

  //returns response back to ESP32
  return ContentService.createTextOutput("Status Updated in Google Sheet");
  //----------------------------------------------------------------------------------
}
```

Figure 38:java script code

In the above figure 37 of java script code, ss variable name is assigned with globally defined var where the value is the id of google spread sheet and taken from URl= https://docs.google.com/spreadsheets/d/1Ezehmls7bmUd3TzxKvzF5tFmd4W4fKh9vTEbR5jWj74/edit#gid=0. Similarly, sheet variable name is assigned with var where the value is the name of google_sheet and its given with ss.getsheetByName.

Then function start with the name doGet along with parameter e doGet(e) after that if statement is introduced with parameter e.parameter ==’undefined’ which will return the value ‘Received data is undefined’ to the ESP32. After the IF statement, if the value assigned with e.parameter and the object in the nextRow variable is defined/fulfilled the spreadsheet will return response back to the ESP32 with the ’ Status updated in Google Sheet’ message and can be seen in the serial monitor.

5.2 TESTING

This is the last phase in the software development lifecycle (SDLC). The goal of Testing is to find device flaws and see if the Gps/Iot works as expected based on the requirements supplied during the Requirement Gathering and Analysis step. The testing will be performed to see if the implemented code in microcontroller is working as expected or not also the coding to create map is perfectly working or not.

The following method of Gps/Iot system were tested:

- implementation of code to ESP 32
- testing the google sheet data
- python programme testing
- Plotted map testing

5.2.1 Implementation of code to ESP 32

At first, the code is implemented in the microcontroller through USB with the help of Arduino IDE before that the code is compiled and there is no error.

Figure 39:Comipling Arduino code

Once compiling done then the code has been uploaded in the microcontroller with the selected port number in Arduino Ide.

The figure 40 and figure 41 show the code has been successfully uploaded in the microcontroller and its ready receive the data to google sheet from the gps from the designed protype.

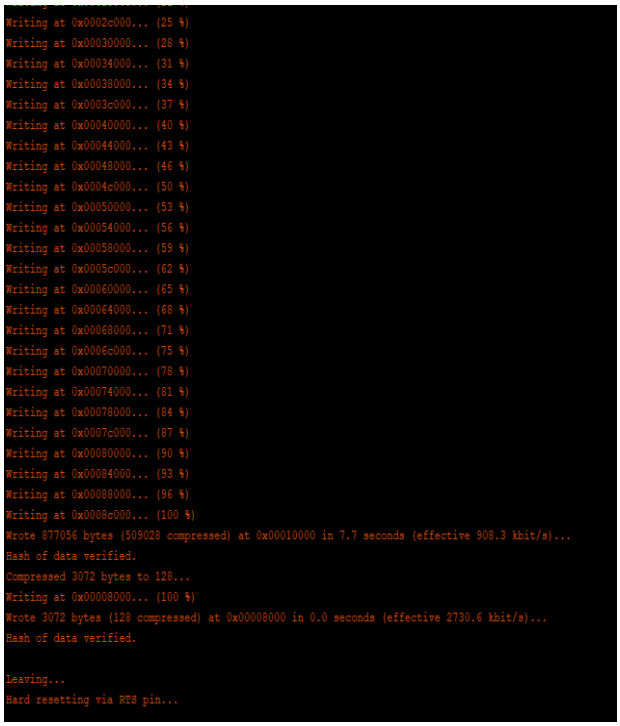


Figure 41: uploading successful 1

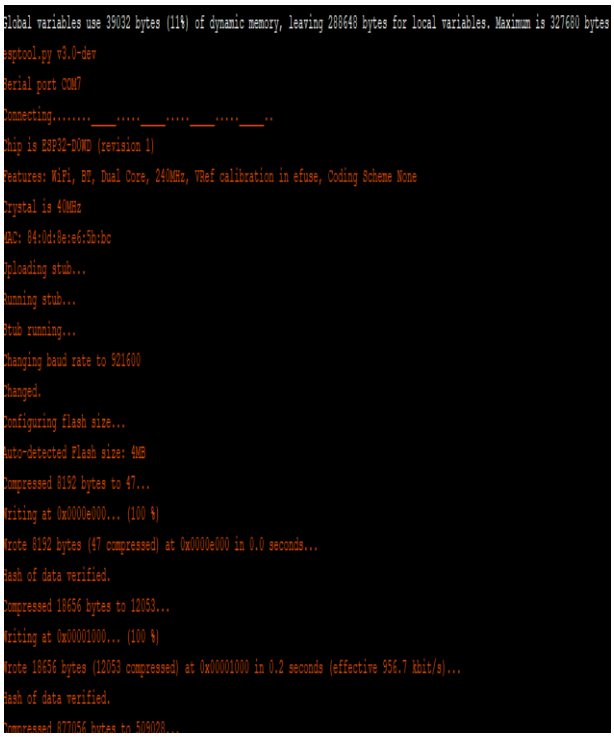
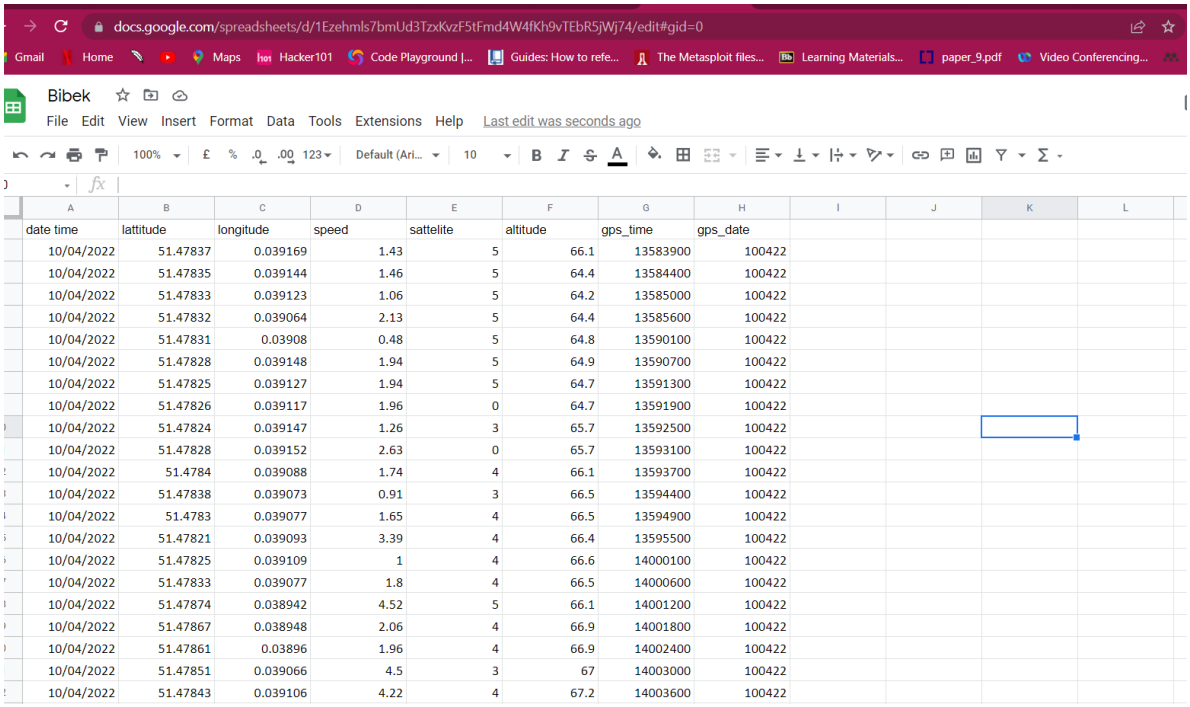


Figure 40: uploading successful

5.2.2 Testing the google spread sheet data

Then the next test step is to test google sheet whether it is receiving data from the microcontroller or not. The google sheet used for the testing purpose was already created while implementing the programme in ESP32.

In order to get coordinates gps needs to connect with minimum 3 satellites. It took 14 minutes to get connected with satellites possible reason could be the poor internet connection. Below figure show the data stored in google spread sheet.



Bibek

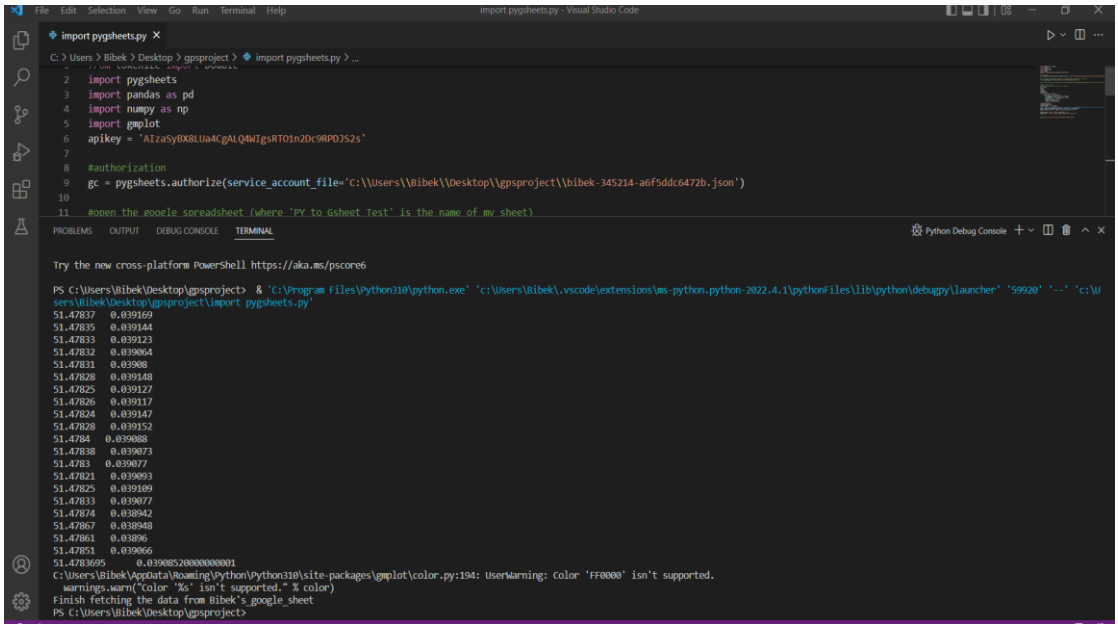
File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

date time	lattitude	longitude	speed	sattelite	altitude	gps_time	gps_date
10/04/2022	51.47837	0.039169	1.43	5	66.1	13583900	100422
10/04/2022	51.47835	0.039144	1.46	5	64.4	13584400	100422
10/04/2022	51.47833	0.039123	1.06	5	64.2	13585000	100422
10/04/2022	51.47832	0.039064	2.13	5	64.4	13585600	100422
10/04/2022	51.47831	0.03908	0.48	5	64.8	13590100	100422
10/04/2022	51.47828	0.039148	1.94	5	64.9	13590700	100422
10/04/2022	51.47825	0.039127	1.94	5	64.7	13591300	100422
10/04/2022	51.47826	0.039117	1.96	0	64.7	13591900	100422
10/04/2022	51.47824	0.039147	1.26	3	65.7	13592500	100422
10/04/2022	51.47828	0.039152	2.63	0	65.7	13593100	100422
10/04/2022	51.4784	0.039088	1.74	4	66.1	13593700	100422
10/04/2022	51.47838	0.039073	0.91	3	66.5	13594400	100422
10/04/2022	51.4783	0.039077	1.65	4	66.5	13594900	100422
10/04/2022	51.47821	0.039093	3.39	4	66.4	13595500	100422
10/04/2022	51.47825	0.039109	1	4	66.6	14000100	100422
10/04/2022	51.47833	0.039077	1.8	4	66.5	14000600	100422
10/04/2022	51.47874	0.038942	4.52	5	66.1	14001200	100422
10/04/2022	51.47867	0.038948	2.06	4	66.9	14001800	100422
10/04/2022	51.47861	0.03896	1.96	4	66.9	14002400	100422
10/04/2022	51.47851	0.039066	4.5	3	67	14003000	100422
10/04/2022	51.47843	0.039106	4.22	4	67.2	14003600	100422

Figure 42: stored data in google spread sheet

5.2.3 Python programming testing

Once the data is received then the python program is tested which will then fetch the data of google spread sheet and plot the coordinates in google map.



```
import pygsheets
import pandas as pd
import numpy as np
import gmplot
apikey = 'AIzaSyBX8LUa4CgALQ4WIgsRTO1n2Dc9RPDJS2s'

#authorization
gc = pygsheets.authorize(service_account_file='C:\\Users\\Bibek\\Desktop\\gpsproject\\bibek-345214-a6f5ddc6472b.json')

#open the google spreadsheet (where 'PY to Gsheet Test' is the name of my sheet)
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Bibek\Desktop\gpsproject> & 'C:\Program Files\Python310\python.exe' 'c:\Users\Bibek\.vscode\extensions\ms-python.python-2022.4.1\pythonFiles\lib\python\debugpy\launcher' '59920' '--' 'c:\Users\Bibek\Desktop\gpsproject\import pygsheets.py'
51.47837   0.039169
51.47835   0.039144
51.47833   0.039123
51.47832   0.039064
51.47831   0.03908
51.47828   0.039148
51.47825   0.039127
51.47826   0.039117
51.47824   0.039147
51.47828   0.039152
51.4784    0.039088
51.47838   0.039073
51.4783    0.039077
51.47821   0.039093
51.47825   0.039109
51.47833   0.039077
51.47874   0.038942
51.47867   0.038948
51.47861   0.03896
51.47851   0.039066
51.4783695         0.039085200000000001
C:\Users\Bibek\AppData\Roaming\Python\Python310\site-packages\gmplot\color.py:194: UserWarning: Color 'FF0000' isn't supported.
  warnings.warn("Color '%s' isn't supported." % color)
Finish fetching the data from Bibek's_google_sheet
PS C:\Users\Bibek\Desktop\gpsproject>
```

Figure 43: running python coding to fetch the data of google sheet

As we can see the code is successfully run and fetched the data to plot map. Now in the next step map will be tested.

5.2.4 Plotted map testing

After running the python programming it will then create a map on the basis of fetched coordinates in html file in is placed in specified folder

Figure 44 location on map

As we can see in the above figure, we got the plotted location of gps coordinates in map. We can fetch the data stored in google spread sheet as much as we want according to that it plots the data in map

With this last step of testing confirms that the system has met all the requirement and given the positive result to accomplish this project.

6 CHAPTER6 - EVALUATION

The evaluation and results of the implemented Gps/Iot system are documented in this chapter.

6.1 REFLECTION AND EVALUATION ON TECHNOLOGIES USED

6.1.1 Arduino Ide

The developer had not much experience with Arduino ide prior to the beginning of this project. That experience became vital in the time of implementation of code to the ESP32 microcontroller. The selection of port, board manager was difficult to set up bot the huge online community of Arduino helped the developer to set the device properly.

6.1.2 Google cloud platform

The google cloud platform was complete new for the developer. The developer was constantly in contact with online community to create Api key of google sheet and for the deployment of java script to get the google script id because developer has knowledge with Microsoft Azure, to be fully sure the developer confirmed by asking in online community.

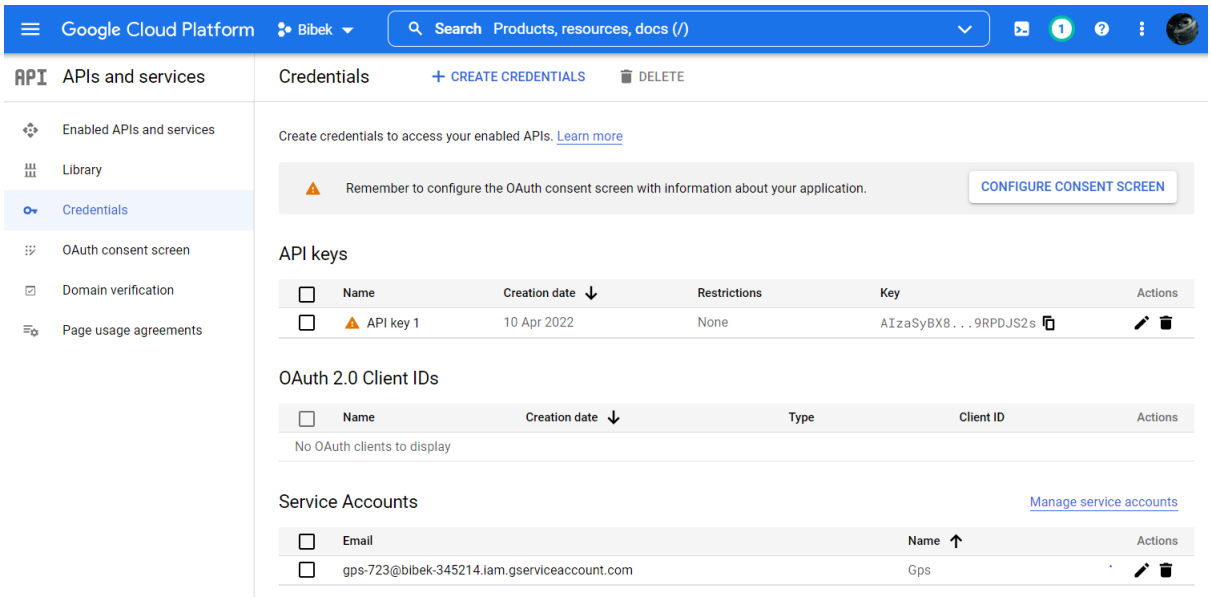


Figure 45 creation of api key on google cloud platform

6.2 EVALUATION AND RESULT OF REQUIREMENT LIST

For the implemented system to be fully accomplished, the requirement list focused on functional requirement must be met. The functional requirement of this project was made on three main steps of the system that define what actually the designed system should perform these all three main requirements were implemented and tested successfully.

6.3 REFLECTION ON AIMS AND OBJECTIVE

The aim of this project was to design and develop the Gps/Iot system to track the location of pets apart from that for the database it should use google cloud platform so the user can handle their data where no third party were involved so that cost of management of data would save. Where it becomes secure that no other third party could access the user data and track the daily activity of user and their pets. In this system Api key is used to make map application secure. An API key, or application programming interface key, is a code that computer programmes use to communicate with one other. The API, or application programming interface, is then used by the software or application to identify its user, developer, or calling programme to a website.

In order to achieve main objectives, the following activities were performed:

1. To overcome security or privacy concern issues by implementing it with user's own cloud service:

Developer used existing google account in order use google sheet and to get the Api key developer used Google cloud platform dashboard. The figure 45 of the google cloud dashboard is shown in this chapter.

2.To create user-friendly python programme and to get location in a map:

Python programme is developed and implemented in chapter 5 and got the positive results while testing the programme.

3. To overcome expensive price of existing product by using most reliable and cheap microcontroller, sensor.

The cost of the hardware is comparably cheaper than most of the existing system the total price of the hardware is discussed in chapter 4, 4.4 in design part.

7 CHAPTER 7 - CONCLUSION

This chapter concludes the reports and offers a perspective on the project's success as well as developer intentions for future programme developments.

7.1 FUTURE ENHANCEMENTS THAT MIGHT BE MADE

Although the completed system satisfies the functional requirements outlined in Chapter 3, the developer may continue working on this project. The following modification might be implemented and is contemplated in the future:

1. **GSM module:**

The main reason not use Gsm module in this project is because it was bit costly than the Gps module which is used in this project where it has Wi-Fi and Bluetooth. Because of its Wi-Fi covering limitation, device will lose signal outside of the Wi-Fi coverage. In the case of Gsm module, it can use 3G/4G internet enable sim card where we can easily connect to that sim card, and it also helps to send receive SMS text notification on specified user number.

2.Integrated smaller device

The current designed prototype system is just for the demo purposes but in the future the developer might build all the device integrated in single device or could purchase the device and then implement the code. So that it is portable and can be fitted in the collar tag for the pets.

3.Mobile app

A fully functioning mobile application could be built by the developer in the future so where user credentials could be used to log in into the application for that the backend and frontend could be developed in the future.

4.Project reflection and conclusion

When starting this project developer had moderate experience with Iot devices and the development framework for Arduino programming languages was new so the couple of technical term was hard to configure but with the help online community developer was able to overcome the problem.

But developer was used to with visual basic code studio and there was not any problem with framework while running code. The developer is happy with overall work done in this project.

Overall implementation and testing were successfully accomplished which shows this project is successful.

REFERENCES

ThatMutt.com. (2020). *Microchips for Dogs Pros & Cons | Good Idea? ThatMutt.com*. [online] Available at: https://www.thatmutt.com/2020/05/06/microchips-for-dogs/ [Accessed 1 Apr. 2022].

Akash Peshin (2018). *What Is The Range Of Bluetooth And How Can It Be Extended?* [online] Science ABC. Available at: https://www.scienceabc.com/innovation/what-is-the-range-of-bluetooth-and-how-can-it-be-extended.html.

Tractive Blog. (2021). *GPS vs Bluetooth Tracker: What's The Difference, And What's Best For Cats And Dogs?* [online] Available at: https://tractive.com/blog/en/tech/pet-tracking-device-comparison-bluetooth-vs-gps-tracker [Accessed 1 Apr. 2022].

Digital Matter. (n.d.). *Livestock Tracking Devices & GPS Trackers*. [online] Available at: https://www.digitalmatter.com/applications/livestock-tracking/ [Accessed 1 Apr. 2022].

Metacept. (n.d.). *Disadvantages of Bluetooth Based Tracking Systems*. [online] Available at: https://metacept.com/disadvantages-of-bluetooth-based-tracking-systems/ [Accessed 2 Apr. 2022].

Lutkevich, B. (2019). *What is microcontroller? - Definition from WhatIs.com*. [online] IoT Agenda. Available at: https://internetofthingsagenda.techtarget.com/definition/microcontroller.

electroSome. (2020). *What is a Microcontroller ? How does it work ?* [online] Available at: https://electrosome.com/microcontroller/.

Raspberry Pi (2013). *What is a Raspberry Pi?* [online] Raspberry Pi. Available at: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/.

www.sparkfun.com. (n.d.). *Building a GPS System - SparkFun Electronics*. [online] Available at: https://www.sparkfun.com/gps.

Tutorialspoint.com. (2019). *SDLC - Waterfall Model*. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.

Rana, K. (2020). *Iterative Model - Features, Advantages & Disadvantages*. [online] ArtOfTesting. Available at: https://artoftesting.com/iterative-model#What_is_Iterative_Model [Accessed 12 Apr. 2022].

GeeksforGeeks. (2020). *Agile Methodology Advantages and Disadvantages*. [online] Available at: https://www.geeksforgeeks.org/agile-methodology-advantages-and-disadvantages/.

k21academy.com. (n.d.). *AWS Storage: Overview, Types & Benefits | K21Academy*. [online] Available at: https://k21academy.com/amazon-web-services/aws-solutions-architect/aws-storage-overview-types-benefits/ [Accessed 12 Apr. 2022].

www.ibm.com. (n.d.). *IBM Cloud Object Storage - Overview*. [online] Available at: https://www.ibm.com/uk-en/cloud/object-storage [Accessed 6 Apr. 2022].

A Cloud Guru. (2021). *What is Google Cloud Platform (GCP)?* [online] Available at: https://acloudguru.com/blog/engineering/what-is-google-cloud-platform-gcp.

PlatformIO Community. (2021). *SW_CPU_RESET on esp32 (Boot Loop)*. [online] Available at: https://community.platformio.org/t/sw-cpu-reset-on-esp32-boot-loop/22067 [Accessed 8 Apr. 2022].

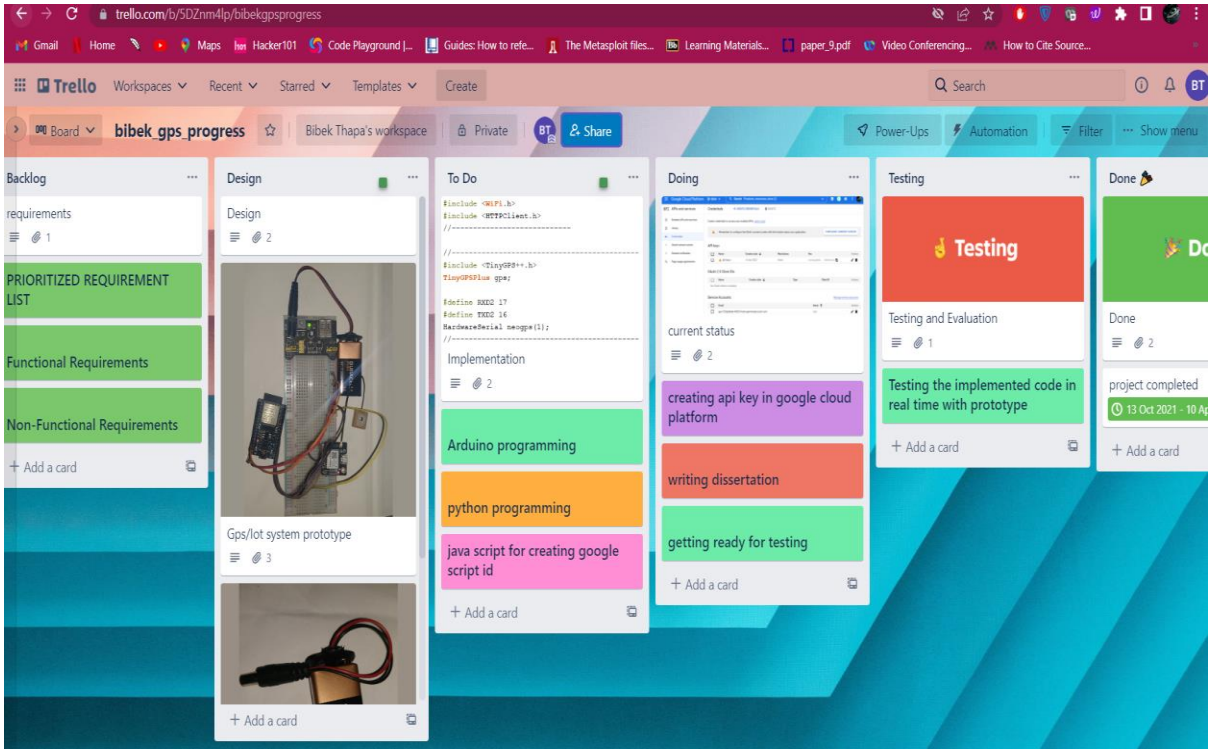
Buehring, S. (2021). *MoSCoW Prioritisation | MoSCoW Analysis | MoSCoW Method | MoSCoW Project Management | MoSCoW Technique | MoSCoW Agile | MoSCoW Priority*. [online] Knowledge Train. Available at: https://www.knowledgetrain.co.uk/agile/agile-project-management/agile-project-management-course/moscow-prioritisation.

Random Nerd Tutorials. (2019). *ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials*. [online] Available at: https://randomnerdtutorials.com/esp32-pinout-reference-gpios/.

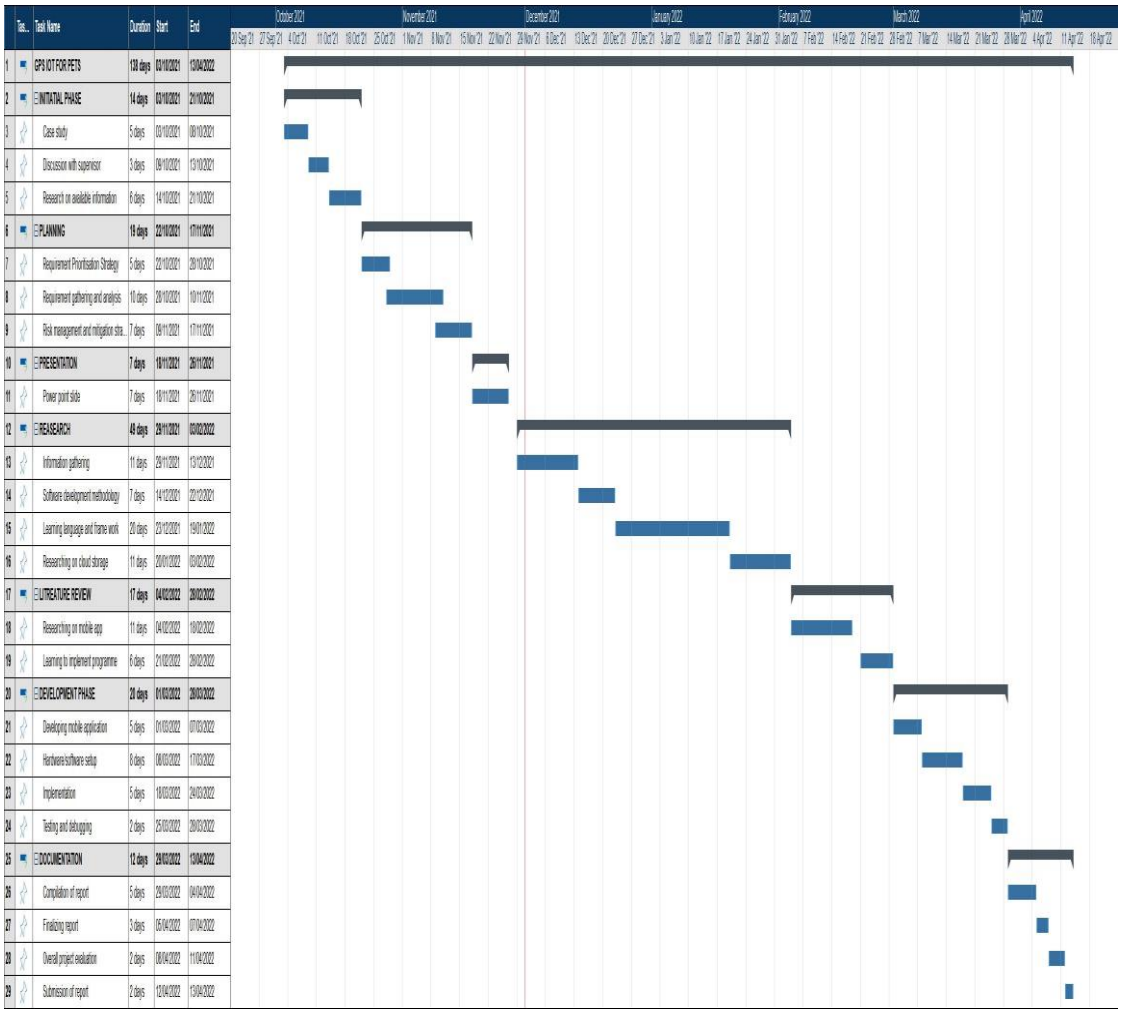
Science Buddies. (n.d.). *How to Use a Breadboard*. [online] Available at: https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard#:~:text=A%20breadboard%20is%20a%20rectangular.

Stack Overflow. (n.d.). *c++ - What is use of c_str function?* [online] Available at: https://stackoverflow.com/questions/7416445/what-is-use-of-c-str-function [Accessed 12 Apr. 2022].

APPENDIX A


trello.com/b/5DZnm4lp/bibekgpsprogress
Trello
Workspaces
Recent
Starred
Templates
Create
Search
Board
bibek_gps_progress
Bibek Thapa's workspace
Private
Share
Power-Ups
Automation
Filter
Show menu
Backlog
requirements
PRIORITIZED REQUIREMENT LIST
Functional Requirements
Non-Functional Requirements
Add a card
Design
Design
Gps/lot system prototype
Add a card
To Do
Implementation
Arduino programming
python programming
java script for creating google script id
Add a card
Doing
current status
creating api key in google cloud platform
writing dissertation
getting ready for testing
Add a card
Testing
Testing
Testing and Evaluation
Testing the implemented code in real time with prototype
Add a card
Done
Done
project completed
Add a card

APPENDIX B



	Tas...	Task Name	Duration	Start	End
1		GPS IOT FOR PETS	133 days	03/10/2021	13/04/2022
2		INITIATAL PHASE	14 days	03/10/2021	21/10/2021
3		Case study	5 days	03/10/2021	08/10/2021
4		Discussion with supervisor	3 days	08/10/2021	13/10/2021
5		Research on available information	8 days	14/10/2021	21/10/2021
6		PLANNING	19 days	22/10/2021	17/11/2021
7		Requirement Prioritization Strategy	5 days	22/10/2021	28/10/2021
8		Requirement gathering and analysis	10 days	28/10/2021	10/11/2021
9		Risk management and mitigation stra...	7 days	08/11/2021	17/11/2021
10		PRESENTATION	7 days	18/11/2021	26/11/2021
11		Power point slide	7 days	18/11/2021	26/11/2021
12		REASEARCH	43 days	29/11/2021	03/02/2022
13		Information gathering	11 days	29/11/2021	13/12/2021
14		Software development methodology	7 days	14/12/2021	22/12/2021
15		Learning language and frame work	20 days	23/12/2021	19/01/2022
16		Researching on cloud storage	11 days	20/01/2022	03/02/2022
17		LITREATURE REVIEW	17 days	04/02/2022	28/02/2022
18		Researching on mobile app	11 days	04/02/2022	18/02/2022
19		Learning to implement programme	6 days	21/02/2022	28/02/2022
20		DEVELOPMENT PHASE	28 days	01/03/2022	28/03/2022
21		Developing mobile application	5 days	01/03/2022	07/03/2022
22		Hardware/software setup	8 days	08/03/2022	17/03/2022
23		Implementation	5 days	18/03/2022	24/03/2022
24		Testing and debugging	2 days	25/03/2022	28/03/2022
25		DOCUMENTATION	12 days	29/03/2022	13/04/2022
26		Compilation of report	5 days	29/03/2022	04/04/2022
27		Finalizing report	3 days	05/04/2022	07/04/2022
28		Overall project evaluation	2 days	08/04/2022	11/04/2022
29		Submission of report	2 days	12/04/2022	13/04/2022