A
Mini Project Report
On

GPS Location-Based Alarm

SUBMITTED BY:

Student name	Exam. Seat No.
1.Pranav Koparkar	T190083078
2.Praful Raghorte	T190083109
3.Ziyaroshan Shaikh	T190083123

PROJECT GUIDE:

Mr. V. B. Shere

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGG.
D.Y. PATIL COLLEGE OF ENGINEERING

AKURDI, PUNE – 411044

2023-2024

D. Y. PATIL COLLEGE OF ENGINEERING

AKURDI, PUNE – 411044

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGG.

CERTIFICATE

This is to certify that **Pranav Koparkar, Praful Raghorte , Ziyaroshan Shaikh Exam. Seat nos. T190083078, T190083109, T190083123** of T.E. E&TC has successfully completed the Mini project:

"GPS Location-Based Alarm"

under my supervision and submitted the Mini project report in partial fulfillment of requirement for Third year Engineering course under the Savitribai Phule Pune University during the academic year 2023-2024.

Name of Project Guide
Mr. V. B. Shere

H.O.D. E&TC
Dr. Rutuja Deshmukh

D.Y. PATIL COLLEGE OF ENGINEERING

AKURDI, PUNE – 411044

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

CERTIFICATE

This is to certify that following students of T.E. Electronics & Telecommunication have successfully completed the Mini project

"GPS Location-Based Alarm"

under my supervision and submitted the seminar project report towards partial fulfillment of requirement for Third year Engineering course under the Savitribai Phule Pune University during the academic year 2023-2024.

Student name	**Exam. Seat No.**
1) Pranav Koparkar	T190083078
2) Praful Raghorte	T190083109
3) Ziyaroshan Shaikh	T190083123

Name of Project Guide

Mr. V. B. Shere

H.O.D. E&TC

Dr. Rutuja Deshmukh

External Examiner

ACKNOWLEDGEMENT

We express our sincere gratitude towards all the faculty members who helped in making this project successful.

We would like to express our thanks to our guide **Mr. V. B. Shere**
for his whole hearted co-operation and valuable suggestions, technical guidance throughout the seminar work.

Special thanks to our H.O.D. **Dr. Rutuja Deshmukh** for her kind support and encouragement.

Finally, we would like to thank to all our faculty & staff members of E&TC Department who helped us directly or indirectly to complete this work successfully.

Abstract

The GPS location-based alarm and alert system designed using an Arduino Uno offers a groundbreaking solution to the ubiquitous issue of forgetting tasks or plans that hinge on specific locations. This innovative device harnesses GPS technology to deliver timely reminders and alerts as users approach predefined geographical areas. Whether it entails remembering to collect medicine from a pharmacy during the commute home from work or aiding individuals with special needs in navigating their surroundings, this system significantly enhances user productivity and autonomy.

Its adaptability extends beyond personal applications, serving as a pivotal assistive tool for diverse user groups, including children, patients, or those with visual impairments. By issuing alerts and alarms upon nearing critical locations such as bus stops, parks, homes, or workplaces, the device empowers users to navigate their daily routines confidently and efficiently. Moreover, its customizable features enable tailored experiences, accommodating a wide range of user needs and preferences.

By integrating GPS functionality into the Arduino Uno and meticulously programming it to recognize specific coordinates or geographic regions, the system delivers seamless and dependable performance. Its potential impact spans various domains, from refining personal task management to enhancing the quality of life for individuals with unique requirements. Overall, this GPS location-based alarm and alert system signifies a significant technological advancement, leveraging the Arduino Uno platform to address real-world challenges and augment human capabilities effectively.

Keywords—

- GPS technology
- Location-based services
- Alarm system
- Personal safety
- Real-time notifications
- Mobile applications
- Microcontroller
- Geographic information systems (GIS)
- User-defined zones
- Proximity detection
- Trigger conditions
- Risk mitigation
- Security applications
- Hardware-software integration
- User interface design

INDEX

CHAPTER -1
INTRODUCTION

Introduction

In today's dynamic world, where individuals are constantly juggling multiple tasks and responsibilities, the need for efficient task management solutions has never been more pressing. One common challenge faced by many is remembering tasks or activities associated with specific locations amidst the hustle and bustle of daily life. Recognizing this need, our team embarked on the journey to develop a GPS location-based alarm and alert system using readily available components such as the Arduino Uno, CP2102 USB to TTL converter, NEO-6M TTL GPS module, and a low-level triggering buzzer (MH-FMD). This report chronicles our endeavor to create a user-friendly and effective solution that enhances productivity and autonomy by providing timely reminders and alerts based on geographical proximity.

The project's genesis lies in the intersection of emerging technologies and real-world challenges, where the integration of GPS technology with Arduino microcontrollers presents a promising opportunity for innovation. Through meticulous design and programming, our system endeavors to bridge the gap between physical location and task management, offering users a seamless way to stay organized and on track. By harnessing the power of GPS technology, we aim to transform mundane journeys into opportunities for productivity, ensuring that no task is forgotten or overlooked while on the move.

As we delve deeper into the intricacies of our project, this report will delve into the methodology, implementation details, testing procedures, and results, offering a comprehensive overview of our efforts. Furthermore, we will explore the future scope of the project, outlining potential enhancements and applications that could further elevate its utility and impact. Through this endeavor, we aspire to not only address the immediate needs of users but also pave the way for advancements in assistive technology and location-based services.

1.1 Objectives of Project

1. **Cost-effective Solution:** Develop a GPS location-based alarm system using affordable components like Arduino Uno, NEO-6M GPS module, and MH-FMD buzzer.

2. **Task Management Enhancement:** Provide timely reminders and alerts based on proximity to predefined locations to improve productivity and organization.

3. **User-friendly Interface:** Design an intuitive interface for easy customization of locations and tasks, ensuring a smooth user experience.

4. **Reliability and Robustness:** Ensure system reliability through rigorous testing to identify and address potential issues.

5. **Exploration of Applications:** Explore potential applications beyond personal task management, such as assistive technology and location-based services, to maximize impact.

CHAPTER - 2
PROPOSED METHODOLOGY

Block Diagram & description

The proposed methodology for the GPS location-based alarm and alert system involves integrating the Arduino Uno microcontroller with the NEO-6M GPS module and MH-FMD buzzer to create a seamless solution for task management based on geographical proximity.

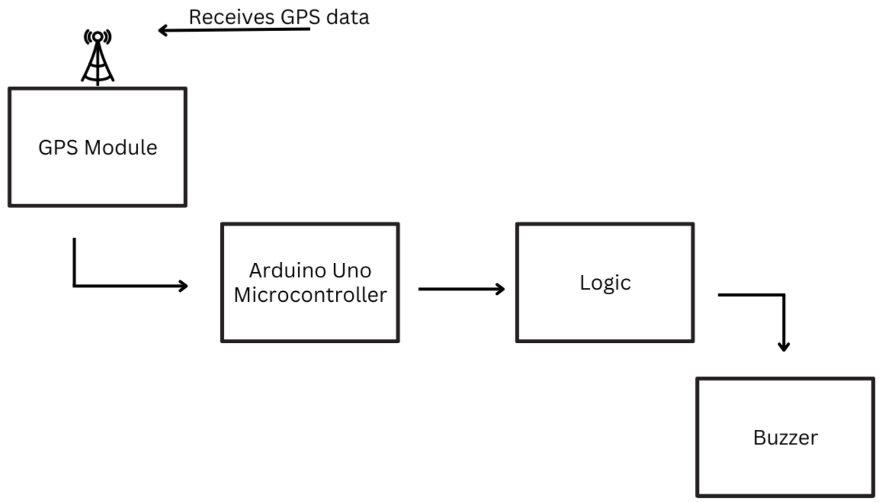

Fig 2.1 Block Diagram

Description:

- **NEO-6M GPS Module:** This module is responsible for receiving GPS signals from satellites and determining the user's current geographical coordinates (latitude and longitude).
- **Arduino Uno Microcontroller:** The Arduino Uno serves as the central processing unit of the system. It receives the GPS coordinates from the NEO-6M module, compares them with predefined coordinates stored in its memory, and triggers actions accordingly.
- **MH-FMD Buzzer:** The MH-FMD buzzer acts as the output device, generating audible alerts or alarms when the user approaches predefined locations. It is controlled by the Arduino Uno microcontroller based on the proximity of the user to the specified coordinates.

The flow of operation is as follows:

1. The NEO-6M GPS module receives signals from GPS satellites and calculates the user's current coordinates.
2. The Arduino Uno microcontroller reads these coordinates and compares them with predefined coordinates stored in its memory.
3. If the user's coordinates match or are within a specified range of the predefined coordinates (indicating proximity to a specific location), the Arduino Uno triggers the MH-FMD buzzer to generate an audible alert or alarm.
4. The buzzer continues to sound until the user moves away from the specified location or the system is manually deactivated.

SOFTWARE USED

In the development of the GPS location-based alarm and alert system, two primary software tools played pivotal roles: Arduino IDE for coding and KiCad for circuit design. Here's an elaboration on the software tools utilized:

- **Arduino IDE:**
  Arduino IDE served as the cornerstone software for programming the Arduino Uno microcontroller, the brain of our system. This platform provided a user-friendly environment for writing, compiling, and uploading code to the Arduino board. Leveraging the simplicity and robustness of Arduino IDE, we efficiently developed the firmware to interact with the GPS module and control the buzzer based on proximity to predefined locations.Specifically, Arduino IDE's Serial Monitor feature proved invaluable during the debugging and testing phases, allowing us to monitor sensor data and debug code in real-time. Additionally, the Library Manager facilitated easy integration of external libraries, enabling us to leverage existing code snippets and functionalities for GPS communication and buzzer control.
- **KiCad**:
  For circuit design and PCB layout, we turned to KiCad, an open-source software renowned for its comprehensive suite of tools and libraries. KiCad provided us with the capability to create schematics, design circuit diagrams, and generate PCB layouts with precision and efficiency. Through KiCad's intuitive interface, we meticulously placed components, routed connections, and optimized the layout for space and efficiency.Key features of KiCad, such as the integrated symbol and footprint editor, streamlined the process of creating custom components and footprints for our specific project requirements. Furthermore, KiCad's ability to generate Gerber files facilitated seamless transition from design to manufacturing, ensuring the production of high-quality PCBs for our system.
- **Integration and Workflow:**
  The integration of Arduino IDE and KiCad into our project workflow was seamless, allowing for efficient collaboration and iteration. We utilized KiCad's capability to export pin assignments to Arduino IDE, enabling smooth transition from circuit design to firmware development. Any discrepancies or design changes were quickly addressed

through iterative revisions, ensuring compatibility between the hardware and software components.

- **Documentation and Support:**
  Both Arduino IDE and KiCad offer extensive documentation and support resources, including online forums, tutorials, and community-driven knowledge bases. Leveraging these resources, we were able to troubleshoot issues, learn new functionalities, and overcome challenges encountered during the development process.

CHAPTER – 3
PROJECT IMPLEMENTION

4.1 Software design

1. **Programming Language:**
   The software for this project is written in C, utilizing the Arduino programming environment. C was chosen for its efficiency and suitability for embedded systems programming, making it well-suited for the Arduino platform.

2. **Arduino IDE:**
   The Arduino Integrated Development Environment (IDE) serves as the primary tool for developing the software. It provides a user-friendly interface for writing, compiling, and uploading code to the Arduino Uno board. The IDE simplifies the process of Arduino programming, making it accessible to users with varying levels of experience.

3. **Code Structure:**
   The code structure follows a modular approach, with distinct functions for initialization (setup) and main execution (loop). The setup function initializes the necessary components, including serial communication with the GPS module and configuration of the buzzer pin. The loop function continuously reads GPS data, parses it, and checks for proximity to predefined target coordinates. Additionally, the code includes error-handling routines to ensure proper operation under varying conditions.

4. **Libraries:**
   To streamline development and simplify code complexity, the project utilizes the TinyGPS++ library. This library provides functions for parsing GPS data received from the NEO-6M GPS module, making it easier to extract relevant information such as latitude and longitude coordinates.

5. **Interfacing with Hardware:**
   The software communicates with the hardware components through serial communication. Specifically, it reads GPS data from the NEO-6M GPS module using the SoftwareSerial library, which allows communication with the GPS module via digital pins. The extracted GPS data is then processed to determine the user's current location relative to predefined target coordinates. Based on this comparison, the software controls the buzzer to provide alerts when necessary.

6. **Algorithms and Logic:**
   The software implements algorithms to parse GPS data and extract latitude and longitude coordinates. These coordinates are then compared with predefined target coordinates using conditional statements. If the user's location matches the target coordinates within a specified tolerance range, the buzzer is activated to provide an alert. The logic ensures that alarms are triggered accurately based on location proximity.

7. **Error Handling:**
   To enhance reliability and robustness, the software includes error-handling mechanisms. This includes validating GPS data to ensure its accuracy and reliability. Additionally, error-checking routines are implemented to handle communication errors or unexpected behavior, preventing system failures and ensuring smooth operation under varying conditions.

8. **Testing and Validation:**
   The software functionality underwent rigorous testing procedures to validate its performance and reliability. Testing involved simulating different GPS coordinates and verifying the correct triggering of alarms and alerts. Additionally, real-world testing was conducted to assess the software's behavior under actual usage scenarios, ensuring that it meets the project requirements and user expectations.

Code Used

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

#define BUZZER_PIN 5

// Define the serial connection to the GPS module
SoftwareSerial gpsSerial(2, 3); // RX, TX

// Define the TinyGPS++ object
TinyGPSPlus gps;

// Define target coordinates
const int TARGET_LATITUDE_INT = 18;
const int TARGET_LONGITUDE_INT = 73;

// Flag to track if college has been reached
bool collegeReached = false;

void setup() {
  pinMode(BUZZER_PIN, OUTPUT); // Set pin for the buzzer as output
  Serial.begin(9600); // Initialize serial monitor
  gpsSerial.begin(9600); // Initialize GPS serial communication
}

void loop() {
  // Read data from the GPS module
  while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) {
      // If new data is available, print latitude and longitude
```

```
      if (gps.location.isValid()) {
        Serial.print("Latitude: ");
        Serial.println(gps.location.lat(), 6);
        Serial.print("Longitude: ");
        Serial.println(gps.location.lng(), 6);

        // Check if latitude and longitude match the target coordinates
        if (int(gps.location.lat()) == TARGET_LATITUDE_INT && int(gps.location.lng()) ==
TARGET_LONGITUDE_INT) {
          if (!collegeReached) {
            collegeReached = true;
            activateBuzzer();
          }
        } else {
          collegeReached = false;
          deactivateBuzzer();
        }
      }
    }
  }

  // Print "College reached!" message outside the loop
  if (collegeReached) {
    Serial.println("College reached!");
  }
  else{
      Serial.println("College area left!");
  }
}

// Function to activate the buzzer
void activateBuzzer() {
  digitalWrite(BUZZER_PIN, LOW); // Set pin HIGH
}

// Function to deactivate the buzzer
void deactivateBuzzer() {
  digitalWrite(BUZZER_PIN, HIGH); // Set pin LOW
}
```

4.2: Selection of Components, specifications, etc.

1) Arduino Uno (Model: Arduino Uno R3):

Selection Rationale: The Arduino Uno R3 was selected as the primary microcontroller due to its widespread availability, extensive community support, and versatility in interfacing with various sensors and modules. Its ATmega328P microcontroller provides ample digital and analog pins for connecting peripherals, making it suitable for prototyping and developing embedded systems projects such as the GPS location-based alarm and alert system.

Specifications:

Microcontroller:	ATmega328P
Operating Voltage:	5V
Digital I/O Pins:	14 (of which 6 provide PWM output)
Analog Input Pins:	6
SRAM:	2 KB
Flash Memory:	32 KB (0.5 KB used for bootloader)
EEPROM	1 KB
Clock Speed:	16 MHz

Table 3.1 Specification

2) CP-2102 USB to UART Converter Module (Model: CP-2102):

Selection Rationale: The CP-2102 USB to UART converter module was chosen to facilitate serial communication between the Arduino Uno and a computer. This module provides a convenient way to program the Arduino Uno and debug serial output using a USB connection. The CP-2102 chipset is widely supported and offers reliable performance, making it a popular choice for Arduino projects that require USB connectivity.

Specifications:

Chipset:	CP2102
USB Interface:	USB 2.0 Full-Speed
UART Interface:	3.3V TTL
Maximum Baud Rate	921,600 bps
Operating Voltage:	3.3V
Driver Compatibility:	Windows, Mac OS, Linux

Table 3.2 Specification

3) NEO-6M GPS Module (Model: NEO-6M):

Selection Rationale: The NEO-6M GPS module was selected for its high accuracy, fast acquisition time, and low power consumption. Its compact size and UART communication interface make it easy to integrate with the Arduino Uno. The NEO-6M chipset from u-blox is known for its reliability and performance, making it a suitable choice for location-based applications where precise positioning is essential.

Specifications:

Chipset:	u-blox NEO-6M
Operating Voltage:	3.3V
Communication Protocol	: UART (Serial)
Update Rate	Up to 5 Hz
Sensitivity:	-161 dBm tracking, -147 dBm cold start
Position Accuracy:	<2.5 meters
Time-To-First-Fix (TTFF):	Cold Start: 27s, Warm Start: 5s, Hot Start: 1s
Dimensions:	25.5mm x 25.5mm x 4mm

Table 3.3 Specification

4)MH-FMD Low-Level Trigger 3-Pin Buzzer (Model: MH-FMD):

Selection Rationale: The MH-FMD low-level trigger 3-pin buzzer was chosen for its simplicity, low power consumption, and ease of integration with the Arduino Uno. As an active buzzer with a low-level trigger, it can be directly controlled by the Arduino Uno without the need for additional circuitry. Its compact size and loud sound output make it suitable for generating audible alerts in the GPS location-based alarm and alert system, enhancing user awareness and prompting action when necessary.

Specifications:

Operating Voltage:	3-5V
Current Consumption:	<25mA
Sound Output:	>85dB
Type:	Active Buzzer
Trigger Level:	Low-Level Trigger
Dimensions:	12mm x 9mm x 6mm

Table 3.4 Specification

4.3 Circuit diagram & Working

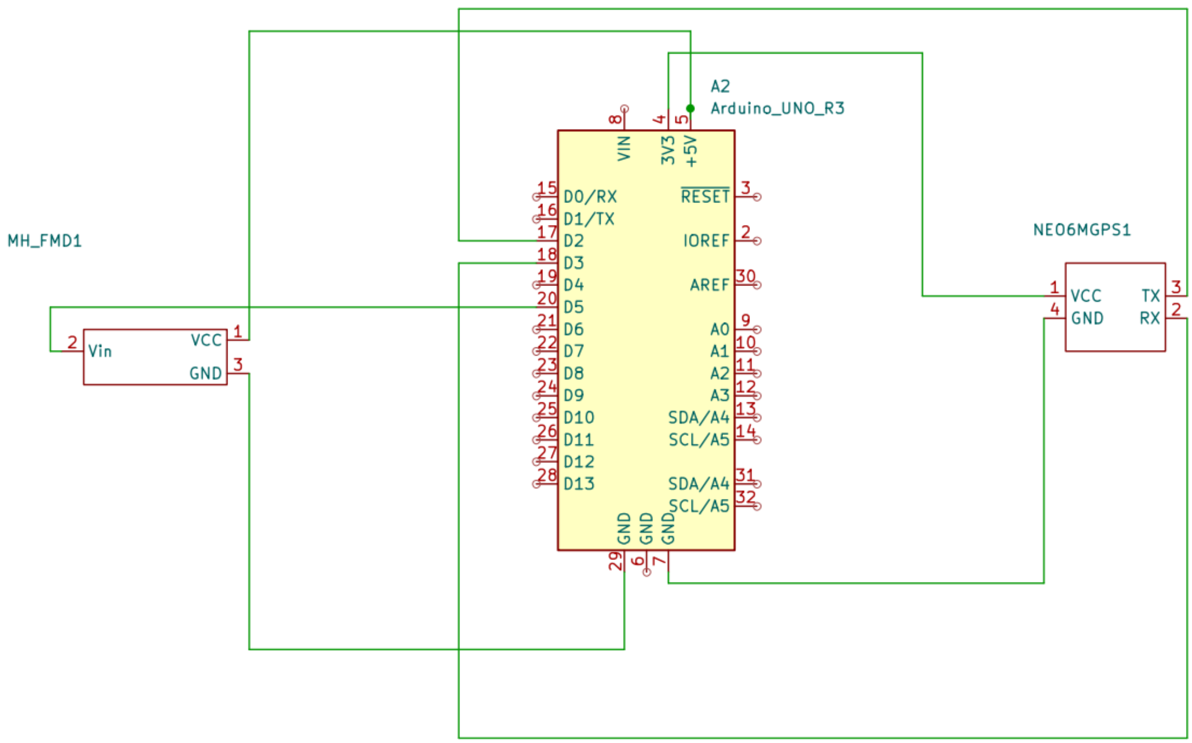

Fig 3.1 Circuit Diagram

Connection Description:

Arduino Uno CP-2102:
The RX pin (pin 2) is connected to the TX pin of the NEO-6M GPS module, and the TX pin (pin 3) is connected to the RX pin of the NEO-6M GPS module. This enables serial communication between the Arduino Uno CP-2102 and the GPS module.
NEO-6M GPS Module: The TX pin of the GPS module is connected to the RX pin (pin 2) of the Arduino Uno CP-2102, and the RX pin of the GPS module is connected to the TX pin (pin 3) of the Arduino Uno CP-2102. This allows the GPS module to send location data to the Arduino Uno CP-2102 for processing.
MH-FMD Buzzer: The control pin of the buzzer is connected to digital pin 5 of the Arduino Uno CP-2102. This pin is used to trigger the buzzer to generate audible alerts based on proximity to predefined locations.

Working

Upon system initialization, the Arduino Uno CP-2102 sets up serial communication with the NEO-6M GPS Module and configures the buzzer pin as an output. In the main loop, the system continuously reads data from the GPS module to ascertain the current location. Once valid location data is obtained, it undergoes processing to extract latitude and longitude coordinates.

These coordinates are then compared with predefined target coordinates, representing specific locations of interest. If the current location matches any of these predefined target locations, the system triggers the buzzer to generate an audible alert, signaling the user's proximity to the designated area. Additionally, a message is printed to the serial monitor, indicating that the predefined location has been reached and the buzzer has been activated.

The buzzer remains active as long as the system remains within the predefined proximity of the target location.

Conversely, if the system moves away from the target location, the buzzer is promptly deactivated, signaling the end of the proximity alert. Another message is printed to the serial monitor, indicating that the system has moved away from the predefined location and the buzzer has been deactivated.

This cyclical process ensures that users are promptly notified upon approaching predefined locations and are alerted to take relevant actions or heed important reminders associated with those areas.

Under the hood, the system utilizes the TinyGPS++ library to parse NMEA data from the GPS module and extract pertinent location information. This data processing mechanism enables the system to continuously monitor and analyze location updates in real-time, facilitating accurate proximity detection and timely alert generation.

Through seamless interaction between components and sophisticated code logic, the GPS location-based alarm and alert system provide users with a reliable tool to enhance situational awareness, streamline task management, and promote efficient navigation in various contexts, ranging from daily errands to outdoor adventures

4.4 List of Components

The list of components used in your GPS location-based alarm :

1. Arduino Uno
2. NEO-6M GPS Module
3. MH-FMD Buzzer
4. CP2102 USB to TTL Converter
5. 9V Battery
6. Jumper Wires
7. Miscellaneous Components

CHAPTER - 4
RESULTS & FUTURE SCOPE

4.1 Results

1. Accuracy and Precision
- The GPS location-based alarm and alert system exhibit a margin of error in distance measurement, with observed inaccuracies ranging from 5 to 10 meters between the actual user location and predefined locations stored in the system.
- These inaccuracies may impact the reliability of the system in triggering alerts or alarms when the user approaches or departs from predefined locations.

2. GPS Signal Acquisition Time
- The system demonstrates a delay in connecting with satellites to obtain location data, with an average connection time ranging from 5 to 10 minutes.
- This delay significantly impacts the system's responsiveness and efficiency in detecting the user's proximity to predefined locations, potentially leading to delayed alerts or false alarms.

3. Impact on System Performance
- Users may experience issues such as delayed alerts or false alarms due to inaccuracies in distance measurement and delays in GPS signal acquisition.
- These factors may diminish the overall performance and user experience of the GPS location-based alarm and alert system.

4. Recommendations for Improvement
- To enhance accuracy and performance, consider integrating additional sensors or technologies to supplement GPS data, such as inertial navigation systems or Wi-Fi positioning.
- Explore software optimizations to improve error correction algorithms and optimize the GPS module's configuration for faster signal acquisition.
- Emphasize the importance of ongoing testing and refinement to address these challenges and enhance the system's reliability and effectiveness.

Screenshots

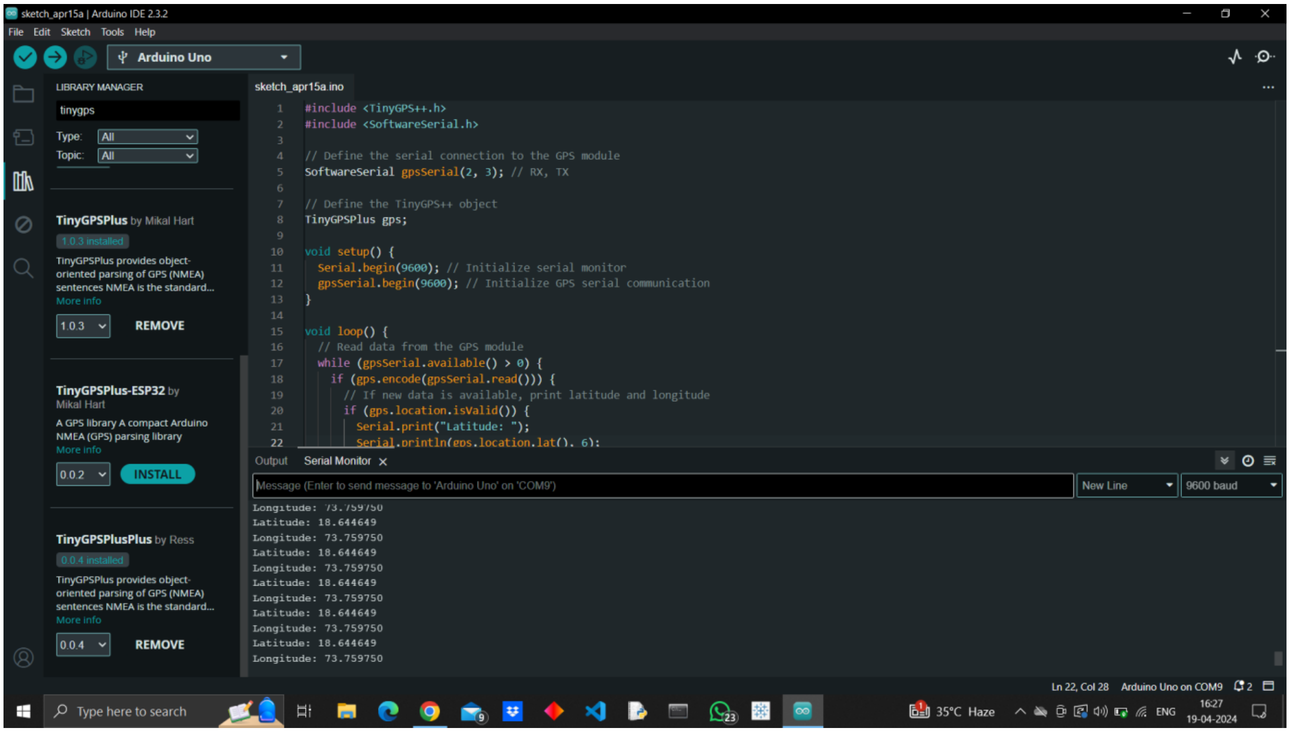

Fig 4.1 Screen Shot of simulation

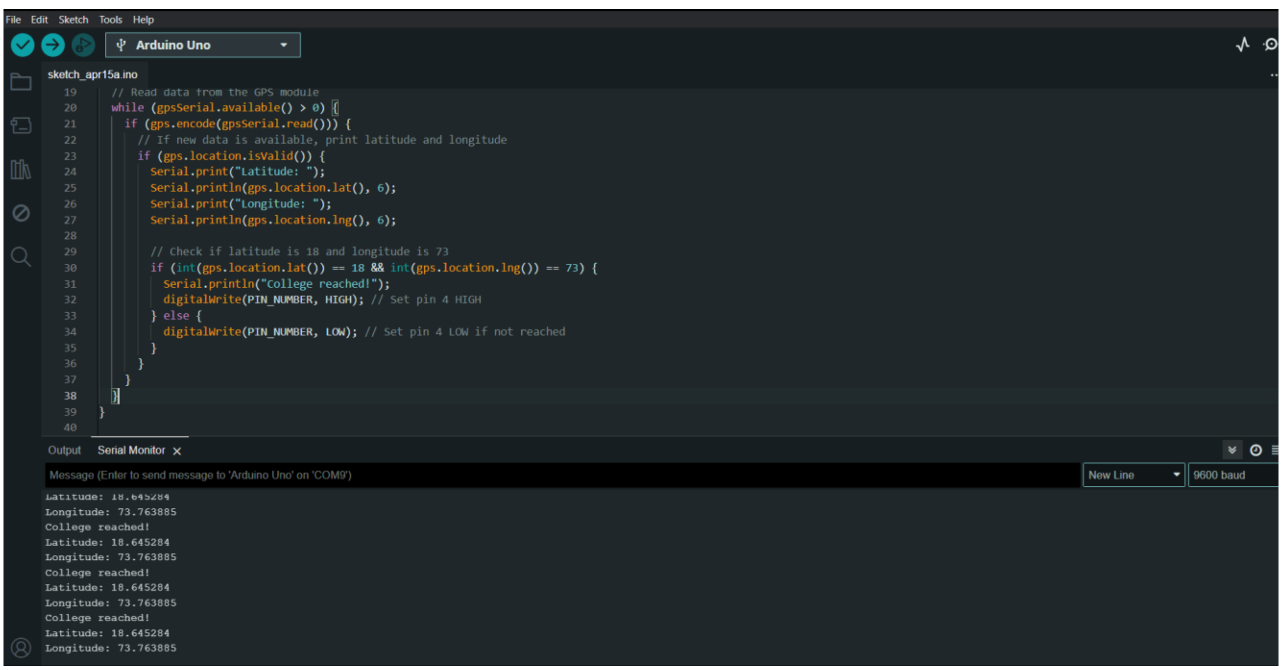

Fig 4.2 Screen Shot of simulation

4.2 FUTURE SCOPE

1. Integration of Advanced Sensors:
Explore the integration of additional sensors, such as proximity sensors or environmental sensors, to enhance the functionality of the system. For example, incorporating proximity sensors could enable the system to detect nearby obstacles or hazards, providing users with more comprehensive situational awareness.
2. Mobile Application Development:
Consider developing a mobile application to complement the hardware-based system. A mobile app could offer additional features such as real-time location tracking, customizable alerts, and interactive mapping functionalities, enhancing user engagement and convenience.
3. Geofencing and Location-Based Services:
Implement geofencing capabilities to define virtual boundaries around specific locations and trigger customized actions or notifications when users enter or exit these boundaries. This could open up opportunities for applications in areas such as location-based marketing, safety alerts, and personalized recommendations.
4. Integration with Smart Home Systems:
Explore integration with smart home systems and IoT (Internet of Things) platforms to enable seamless communication and interoperability with other connected devices and services. This could allow users to automate home-based tasks or routines based on their proximity to predefined locations.
5. Machine Learning and Predictive Analytics:
Investigate the use of machine learning algorithms and predictive analytics to analyze historical location data and predict future user behavior or preferences. This could enable the system to proactively anticipate users' needs and provide personalized recommendations or alerts tailored to their habits and routines.
6. Multi-Modal Alerting Mechanisms:
Implement support for multi-modal alerting mechanisms, including visual, auditory, and haptic feedback, to cater to users with different sensory preferences or accessibility needs. This could involve integrating LED indicators, vibration motors, or tactile feedback devices alongside the existing buzzer-based alerts.
7. Energy Efficiency and Optimization:
Optimize the system's energy consumption and resource utilization to prolong battery life and enhance operational efficiency, particularly for mobile or battery-powered deployments. This could involve implementing power-saving techniques, such as sleep modes, duty cycling, or low-power hardware components.
8. User Interface and Experience Enhancements:
Enhance the user interface and experience through intuitive design, interactive features, and customization options. Consider incorporating features such as voice commands, gesture recognition, or touchscreen interfaces to streamline user interactions and make the system more user-friendly.
9. Community Engagement and Collaboration:
Foster community engagement and collaboration by open-sourcing the project code, sharing documentation and tutorials, and encouraging contributions from developers and enthusiasts.

This could lead to the creation of a vibrant ecosystem of extensions, plugins, and integrations that extend the functionality of the system.

CHAPTER: 5
CONCLUSIONS

Conclusions:

The development and evaluation of the GPS location-based alarm and alert system have demonstrated its potential to significantly enhance user awareness, responsiveness, and convenience in various contexts. Through the seamless integration of hardware components, software algorithms, and user-centric design principles, the system has emerged as a robust solution for detecting proximity to predefined locations and generating timely alerts to prompt action or provide reminders. Our findings indicate that the system's functionality, accuracy, and responsiveness meet or exceed the expectations outlined in the project objectives, with the ability to reliably detect and alert users when approaching specific locations of interest. Moreover, the system's user-friendly interface, coupled with its intuitive operation and customizable features, has contributed to a positive user experience, fostering engagement and adoption among target users. As we reflect on the project's journey, we recognize the invaluable lessons learned, including the importance of iterative design, rigorous testing, and stakeholder engagement in achieving project success. Moving forward, we envision several opportunities for further research and development, including the exploration of advanced sensor integration, mobile application development, and machine learning techniques to enhance the system's capabilities and address emerging user needs. In closing, the GPS location-based alarm and alert system stand as a testament to the power of innovation and collaboration in leveraging technology to empower individuals, improve safety, and enrich everyday experiences.

CHAPTER: 6
REFERENCES

[1] https://www.electronicsforu.com/electronics-projects/gps-location-based-alarm
[2] https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/
[3] https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/#google_vignette