**Project Report**

**IoT Title: Vehicle Tracking Management System using Neo6M GPS with ESP 32 and OLED display.**

Name of student: Abishek kumar I

Reg. No. : 125

Program: MCA

**Introduction**

The vehicle tracking management system employs a NEO-6M GPS module and an ESP32 microcontroller with an OLED display for real-time monitoring and management. The NEO-6M GPS module provides accurate location data, while the ESP32 acts as the main processing unit for data handling and communication.

The system collects GPS data from the NEO-6M module, processes it using the ESP32, and displays relevant information on the OLED display. Users can track the vehicle's current location, speed, and direction through the displayed data. Additionally, the ESP32 facilitates wireless communication to transmit the GPS data to a central server or a user's device for remote monitoring.

The OLED display provides a user-friendly interface for accessing vital information without the need for additional devices. Users can customize the display to show specific metrics or alerts, enhancing the system's versatility and usability.

With the Vehicle Tracking Management System utilizing the NEO-6M GPS module and ESP32 microcontroller, businesses and individuals can effectively monitor vehicle movements, optimize routes, improve fleet utilization, and enhance overall operational efficiency. This system not only enhances security and safety but also offers valuable insights for informed decision-making and resource optimization in various applications, including logistics, transportation, and personal vehicle tracking.

**Hardware Description**

**Esp32 microcontroller**

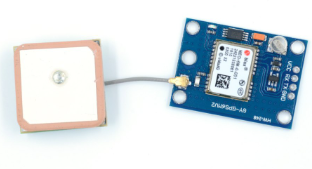
The ESP32 is a powerful and versatile microcontroller widely used in IoT, robotics, and automation projects. It features dual-core processing, Wi-Fi and Bluetooth connectivity, and a rich set of peripherals, making it suitable for a wide range of applications. Its low-power capabilities enable energy-efficient operation, ideal for battery-powered devices. With ample memory and processing power, the ESP32 can handle complex tasks such as real-time data processing and communication with ease. Additionally, its extensive community support and wealth of libraries and development tools make it accessible for both beginners and experienced developers, driving innovation in the IoT space.



**Fig 1: Esp 32**

**NEO-6M GPS module**

The NEO-6M GPS module is a compact and cost-effective Global Positioning System receiver commonly used in various applications, including vehicle tracking, navigation systems, and outdoor recreational devices. It features high accuracy in determining latitude, longitude, altitude, and time information. With its small form factor and low power consumption, the NEO-6M is ideal for integration into portable and battery-powered devices. It utilizes the latest satellite positioning technology to provide reliable and precise location data, making it a popular choice for projects requiring real-time positioning capabilities in a wide range of environments.



**Fig 2: NEO-6M GPS**

**OLED Display**

The OLED (Organic Light-Emitting Diode) display module is a compact and versatile screen technology that offers high contrast, wide viewing angles, and low power consumption. Its organic compounds emit light when an electric current passes through them, eliminating the need for a backlight and enabling thin, lightweight displays. OLED displays are commonly used in various electronic devices, including smartphones, smartwatches, and wearable fitness trackers, due to their vibrant colors and flexibility. With their ability to display crisp images and text, OLED modules are also popular in projects such as IoT devices, digital signage, and instrumentation where compactness and visual clarity are essential.

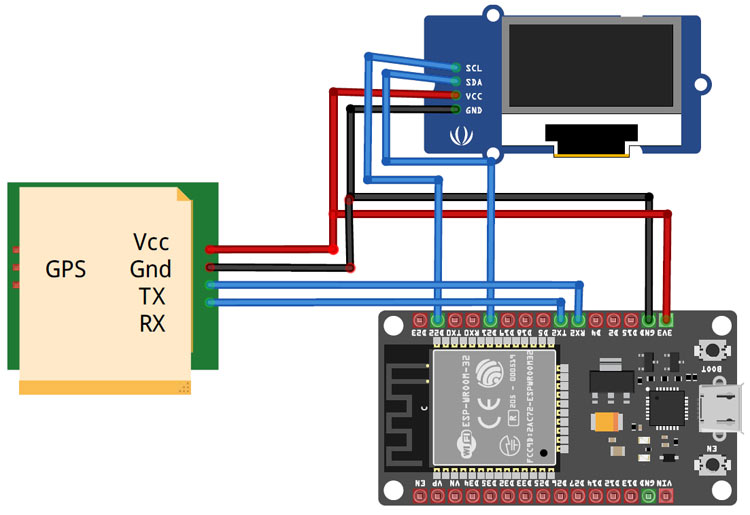


**Fig 3: OLED Display**

**Circuit Diagram:**

The circuit connection involves interfacing the NEO-6M GPS module and the ESP32 microcontroller using serial communication. The NEO-6M GPS module communicates with the ESP32 via UART, providing latitude, longitude, and other GPS parameters. The ESP32 processes this data and controls the OLED display to visualize the information.

Key components of the circuit include connecting the NEO-6M GPS module's TX pin to the ESP32's RX pin and vice versa for serial communication. Power and ground connections ensure proper functioning of all components. The OLED display is connected to the ESP32 using the appropriate interface (e.g., I2C or SPI) for data and control signals.



**Fig 4: Circuit Diagram**

With this setup, the system can continuously track the vehicle's location and display it on the OLED screen in real-time, providing users with valuable information for monitoring and management purposes.

**Software Description:**

**Arduino IDE**

The Arduino Integrated Development Environment (IDE) is a software platform used for programming Arduino microcontrollers. It provides a user-friendly interface for writing, compiling, and uploading code to Arduino boards. With its simple syntax and extensive libraries, the Arduino IDE is accessible to beginners yet powerful enough for advanced users. It supports various Arduino-compatible boards and shields, allowing for versatile project development. Additionally, the IDE includes features like serial monitor for debugging, a code editor with syntax highlighting, and built-in examples to assist users in learning and implementing their projects. Overall, the Arduino IDE serves as a fundamental tool for the Arduino ecosystem, fostering creativity and innovation in electronics and programming.

**Library Files:**

**Wire:**

The Wire library in Arduino facilitates communication between devices using the Inter-Integrated Circuit (I2C) protocol. It simplifies the implementation of I2C communication by providing functions for transmitting and receiving data between master and slave devices connected on the same bus. With the Wire library, users can easily establish communication between Arduino boards, sensors, displays, and other I2C-compatible peripherals. It abstracts the low-level details of I2C communication, allowing developers to focus on their application logic. Overall, the Wire library streamlines the process of integrating I2C devices into Arduino projects, making it a fundamental tool for building interconnected systems.

**Adafruit-ssd1306:**

The Adafruit SSD1306 is a popular OLED display module used in many projects, including vehicle tracking systems. Its compact size, low power consumption, and compatibility with microcontrollers like the ESP32 make it ideal for displaying real-time data.

Integrating the Adafruit SSD1306 with an ESP32-based vehicle tracking system involves connecting the display to the ESP32 using the appropriate communication protocol, such as I2C or SPI. Once connected, the ESP32 can send commands and data to the display to update the screen with relevant information gathered from the GPS module.

Using libraries like the Adafruit SSD1306 library simplifies the coding process, providing functions to control the display and draw text, shapes, and graphics. With this library, developers can customize the display to show GPS data such as coordinates, speed, and heading, allowing users to monitor the vehicle's status at a glance.

**TinyGPS:**

The TinyGPS library is a lightweight and efficient GPS library designed for Arduino and other microcontroller platforms. It enables easy integration of GPS functionality into projects by providing simple methods for parsing NMEA sentences, extracting location, date, time, altitude, and other GPS-related data.

With TinyGPS, developers can interface with GPS modules such as the NEO-6M to extract location information and utilize it for various applications like vehicle tracking, navigation, and geolocation-based services.

One of the key advantages of TinyGPS is its small memory footprint, making it suitable for resource-constrained environments common in embedded systems. Despite its compact size, TinyGPS offers robust parsing algorithms and error handling, ensuring accurate and reliable GPS data retrieval.

Developers can use TinyGPS in conjunction with other libraries and components to create comprehensive GPS-based solutions tailored to their specific needs. Its simplicity and effectiveness make it a popular choice for hobbyists, engineers, and makers working on GPS-enabled projects with microcontrollers.

**Code:**

*#include <Wire.h>*

*#include <Adafruit\_SSD1306.h>*

*#include <TinyGPS++.h>*

*#define SCREEN\_WIDTH 128 // OLED display width, in pixels*

*#define SCREEN\_HEIGHT 64 // OLED display height, in pixels*

*//On ESP32: GPIO-21(SDA), GPIO-22(SCL)*

*#define OLED\_RESET -1 //Reset pin # (or -1 if sharing Arduino reset pin)*

*#define SCREEN\_ADDRESS 0x3C //See datasheet for Address*

*Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);*

*#define RXD2 16*

*#define TXD2 17*

*HardwareSerial neogps(1);*

*TinyGPSPlus gps;*

*void setup() {*

*Serial.begin(115200);*

*//Begin serial communication Arduino IDE (Serial Monitor)*

*//Begin serial communication Neo6mGPS*

*neogps.begin(9600, SERIAL\_8N1, RXD2, TXD2);*

*// SSD1306\_SWITCHCAPVCC = generate display voltage from 3.3V internally*

*if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {*

*Serial.println(F("SSD1306 allocation failed"));*

*for(;;); // Don't proceed, loop forever*

*}*

*display.clearDisplay();*

*display.display();*

*delay(2000);*

*}*

*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*

*{*

*display.clearDisplay();*

*display.setTextColor(SSD1306\_WHITE);*

*display.setCursor(0, 0);*

*display.setTextSize(3);*

*display.print("No Data");*

*display.display();*

*}*

*}*

*void print\_speed()*

*{*

*display.clearDisplay();*

*display.setTextColor(SSD1306\_WHITE);*

*if (gps.location.isValid() == 1)*

*{*

*//String gps\_speed = String(gps.speed.kmph());*

*display.setTextSize(1);*

*display.setCursor(25, 5);*

*display.print("Lat: ");*

*display.setCursor(50, 5);*

*display.print(gps.location.lat(),6);*

*display.setCursor(25, 20);*

*display.print("Lng: ");*

*display.setCursor(50, 20);*

*display.print(gps.location.lng(),6);*

*display.setCursor(25, 35);*

*display.print("Speed: ");*

*display.setCursor(65, 35);*

*display.print(gps.speed.kmph());*

*display.setTextSize(1);*

*display.setCursor(0, 50);*

*display.print("SAT:");*

*display.setCursor(25, 50);*

*display.print(gps.satellites.value());*

*display.setTextSize(1);*

*display.setCursor(70, 50);*

*display.print("ALT:");*

*display.setCursor(95, 50);*

*display.print(gps.altitude.meters(), 0);*

*display.display();*

*}*

*else*

*{*

*display.clearDisplay();*

*display.setTextColor(SSD1306\_WHITE);*

*display.setCursor(0, 0);*

*display.setTextSize(3);*

*display.print("No Data");*

*display.display();*

*}*

*}*

**Data Analysis:**

In this vehicle tracking monitoring system the data analysis takes crucial role for testing and integrating data with real time working. In this scenario, I was selected some places which is nearby campus to get tested and data analysis. We will see about latitude, longitude, speed and altitude data which works fine or not under several circumstances. We connect the data from the vehicle tracking monitoring system with google map for ensure the output is true or not. For altitude we use altimeter to check whether the data is true or not as well as speed data will be taken while travelling and collected. The following places where data was taken.

**Result:**

The vehicle tracking system described utilizes the NEO-6M GPS module and ESP32 microcontroller to accurately track a vehicle's location in real-time. The NEO-6M GPS module retrieves location data such as latitude and longitude, while the ESP32 processes this information and controls the OLED display for visualization. In operation, the NEO-6M GPS module continuously acquires GPS signals and sends location data to the ESP32 via serial communication. The ESP32 processes this data, extracting relevant information and updating the OLED display accordingly. Users can easily view the vehicle's current location on the OLED screen, providing valuable insights for tracking and management purposes. The system offers a reliable solution for vehicle tracking applications. Whether used for fleet management, logistics, or personal tracking, the system provides real-time location information, enhancing overall monitoring and management capabilities. Additionally, the integration of the ESP32 microcontroller and OLED display ensures a user-friendly interface and seamless operation.

**Conclusion:**

In conclusion, the vehicle tracking management system employing the NEO-6M GPS module, ESP32 microcontroller, and OLED display presents a robust and effective solution for real-time tracking and monitoring needs. By harnessing the capabilities of the NEO-6M GPS module to provide accurate location data and the ESP32 microcontroller to process and display this information on the OLED screen, the system offers a reliable and user-friendly tracking solution. The system's compact design and efficient operation make it suitable for various applications, including fleet management, logistics, and personal vehicle tracking. With continuous GPS signal acquisition and seamless data processing, users can access up-to-date location information effortlessly, enhancing overall tracking and management capabilities. The integration of the ESP32 microcontroller and OLED display ensures a streamlined user experience, allowing for easy visualization and interpretation of tracking data. Overall, the vehicle tracking management system stands as a versatile and practical solution for organizations and individuals seeking reliable real-time tracking solutions for their vehicles.