MINOR PROJECT IN PHYSICS

BRANCH

Electronics and Communication Engineering - C

PROJECT TITLE

DEVELOPMENT OF ANTI-SLEEP ALARM SYSTEM WITH GPS AND ESP8266 FOR PUBLIC SAFTEY

SUBMITTED BY

Navneeth Srinivas.S	310624106134
Neeraj.K	310624106135
K.K.Rakshith	310624106162

Easwari Engineering College
(Autonomous)
Ramapuram Campus
Chennai – 600 089
Tamil Nadu

30th March 2025

ANTI-SLEEP ALARM SYSYTEM – GPS AND ESP8266

1. Introduction

The development of a GPS-detectable anti-sleep alarm system using an ESP8266, GPS module (NEO-7M), an eye-blink sensor, and a buzzer is an innovative solution designed to enhance safety, particularly in automotive environments. Drowsiness-related accidents are a leading cause of fatalities worldwide, making real-time monitoring and alert mechanisms essential for preventing such incidents. This system efficiently detects early signs of fatigue and triggers alerts both locally and remotely via email.

The core component of this system is an infrared-based eye-blink sensor, which continuously monitors the user’s eye-blink patterns. If the driver closes their eyes for an extended period or exhibits abnormal blinking patterns, the ESP8266 microcontroller processes the data and determines whether the user is experiencing fatigue. Upon detection, a dual-response mechanism is triggered to enhance safety.

First, a buzzer is activated each time drowsiness is detected, providing an immediate alert to the driver. However, the system is designed to prevent false alarms. An email notification is sent only if the buzzer activates more than three times consecutively. This ensures that temporary distractions or momentary blinking do not trigger unnecessary alerts.

Once the threshold is reached, the GPS module (NEO-7M) retrieves the user’s real-time location coordinates. The ESP8266 WiFi module then sends an email notification containing the driver’s location to a predefined recipient using a dummy email account. This allows fleet managers, family members, or emergency responders to take quick action if necessary.

The ESP8266’s WiFi capabilities enable seamless integration with an SMTP server for automated email notifications. Additionally, the system can be extended to store data in a cloud database for further analysis or logging.

Designed for efficiency and reliability, the system includes low-power optimizations to ensure stable performance over long durations. Power management considerations, such as using a rechargeable battery or vehicle power source, ensure continuous operation without frequent maintenance.

By integrating real-time drowsiness detection, GPS-based tracking, a multi-stage buzzer alert, and email notifications via ESP8266, this system provides a comprehensive and effective solution to mitigate risks associated with driver fatigue. It significantly enhances safety and is applicable in various transportation and monitoring systems.

1.1. Problem Statement

Drowsiness-related accidents are a major cause of fatalities in road transportation and industrial operations. Drivers and machine operators often experience fatigue, leading to reduced alertness and slower reaction times, increasing the risk of accidents. Traditional methods of drowsiness detection rely on self-awareness or external monitoring, which may not always be effective.

To address this issue, a GPS-detectable anti-sleep alarm system using an ESP8266, GPS module (NEO-7M), an eye-blink sensor, and a buzzer is proposed. The system continuously monitors the driver’s eye-blink patterns using an infrared-based sensor. When abnormal blinking or prolonged eye closure is detected, the buzzer alerts the user. If the buzzer is activated more than three times, the system retrieves the driver’s real-time location via GPS and sends an email alert through ESP8266 Wi-Fi.

This solution enhances safety by integrating real-time monitoring, location tracking, and automated alert mechanisms, ensuring timely intervention and reducing accident risks.

1.2. Significance of the Project

The GPS-detectable anti-sleep alarm system is designed to prevent accidents caused by driver fatigue by integrating real-time drowsiness detection, GPS tracking, and automated email alerts. This system enhances safety, efficiency, and reliability in transportation and industrial applications.

The drowsiness detection system offers multiple benefits in enhancing driver safety and accident prevention. It detects early signs of drowsiness and alerts the driver through a buzzer, reducing the risk of fatigue-related accidents. To minimize false alarms, the system follows a multi-stage alert mechanism, where the buzzer is triggered upon initial detection, and only after three consecutive alerts is an email notification sent. Real-time GPS tracking ensures the driver’s exact location is shared for emergency intervention, enhancing response efficiency. Using ESP8266 Wi-Fi, the system enables remote monitoring by sending automated email alerts to predefined recipients. Designed with cost-effectiveness in mind, it utilizes affordable and widely available components, making it scalable for various applications, including public transport, logistics, and industrial safety. By integrating advanced monitoring and alert mechanisms, this system provides a reliable, efficient, and proactive solution for reducing drowsiness-related accidents, ultimately ensuring safer roads and workplaces.

1.3. Aim and Objective

Aim:

The aim of this project is to develop a GPS-detectable anti-sleep alarm system using an ESP8266, GPS module (NEO-7M), eye-blink sensor, and a buzzer to prevent accidents

caused by driver fatigue. The system monitors the user’s eye-blink patterns, triggers a buzzer alert, and sends an email notification with real-time location if drowsiness persists.
Objectives:
The drowsiness detection system is designed to enhance safety by utilizing an infrared-based eye-blink sensor to detect driver fatigue. When drowsiness is identified, the system activates a buzzer alert, ensuring immediate awareness. If the buzzer is triggered more than three times, an automated email alert containing the driver's GPS location is sent for emergency intervention. The integration of ESP8266 Wi-Fi enables real-time data transmission, allowing remote monitoring and quick responses. This system is developed to improve road and workplace safety by preventing drowsiness-related accidents, offering a proactive and efficient solution to minimize fatigue-induced risks.

1.4. Scope and Limitation

The drowsiness detection system is designed for real-time monitoring of driver alertness using an eye blink sensor to detect fatigue. When prolonged eye closure is identified, an immediate buzzer alert is triggered to help keep drivers awake and attentive. For emergency communication, the system sends an automated email with real-time GPS coordinates to predefined contacts, ensuring quick intervention. Accurate GPS tracking enhances location monitoring, making response times faster in critical situations. With its wide applicability, the system can be implemented in personal vehicles, commercial fleets, and public transport for improved road safety. Additionally, its cost-effective design, utilizing affordable and widely available electronic components, ensures easy implementation and scalability, making it an efficient solution for reducing drowsiness-related accidents.

Limitations of the Project:

The drowsiness detection system has certain limitations that may impact its performance in real-world conditions. The system's reliance on network connectivity means that a stable mobile network is required for sending email alerts, which may not always be available in remote areas. The accuracy of the eye blink sensor can be affected by lighting conditions, driver positioning, and sensor calibration, potentially leading to false positives or missed detections. Additionally, the system requires a continuous power supply for uninterrupted monitoring and communication, making power management crucial. GPS precision is also subject to environmental factors such as tall buildings and tunnels, which can interfere with location tracking. While some of these challenges are influenced by external conditions, power reliability can be improved by integrating a Battery Charging Module (TP4056) and a rechargeable lithium-ion battery, ensuring continuous operation even in cases of power failure.

2. Literature Review

2.1 Background of Study

Drowsiness-related accidents are a major cause of fatalities in transportation and industrial sectors. Studies indicate that fatigue impairs reaction time, decision-making, and alertness, increasing the risk of accidents. Traditional methods of fatigue detection rely on self-awareness or manual monitoring, which are often ineffective.

Recent advancements in IoT, sensors, and AI have enabled real-time drowsiness detection using eye-blink sensors, GPS tracking, and wireless communication. Research shows that infrared-based eye-blink sensors can effectively monitor blink rates and prolonged eye closure to detect fatigue. Integrating microcontrollers like ESP8266 with GPS and Wi-Fi-based alert systems ensures accurate location tracking and remote monitoring, making such solutions more reliable and scalable in accident prevention.

2.2 History of IR Based Alarms

Infrared (IR) sensors have been a crucial component in various applications since their development. The history of IR technology dates back to the 19th century, when Sir William Herschel discovered infrared radiation in 1800 while studying the temperature differences in light spectra. Over time, infrared sensing technology evolved, leading to its use in military, industrial, and medical applications. By the 20th century, IR sensors became widely used for motion detection, temperature sensing, and automation. The introduction of infrared photodiodes and LEDs in the 1960s and 1970s enabled more precise detection of IR radiation. These advancements led to the development of infrared-based eye-tracking and blink detection systems for fatigue monitoring.

In the 1990s and 2000s, IR sensors became an essential part of driver assistance systems, medical devices, and consumer electronics. With the rise of IoT and AI, modern infrared-based eye-blink sensors are now used in driver monitoring systems, virtual reality (VR), and assistive technologies. These sensors work by detecting changes in infrared reflections from the eyes, providing a non-intrusive, accurate method for drowsiness detection.

Today, IR-based eye-blink sensors play a key role in fatigue detection systems, ensuring enhanced safety in automotive, industrial, and healthcare sectors.

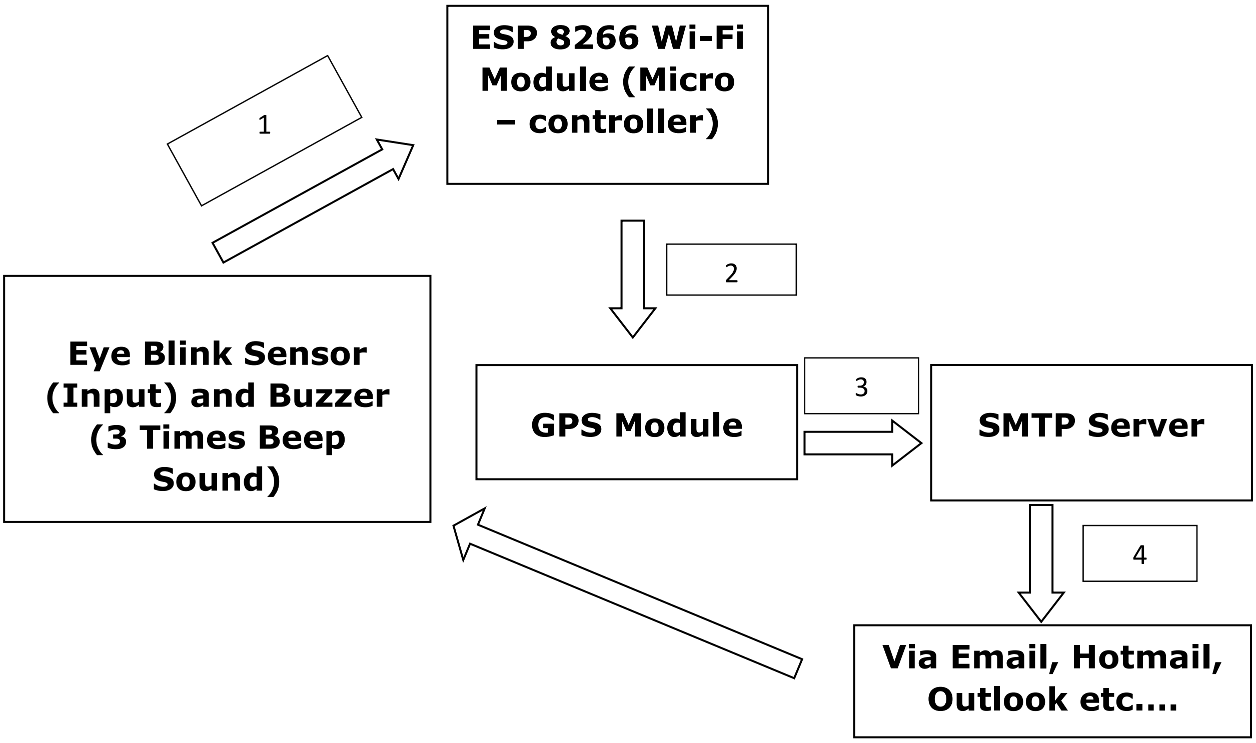
2.3 Market Growth Phase

The market for drowsiness detection systems using IR-based sensors, GPS tracking, and IoT technology is experiencing rapid growth. With rising concerns over road safety, governments and automotive manufacturers are integrating driver monitoring systems (DMS) into vehicles. According to market reports, the global driver fatigue detection market is expected to grow at a CAGR of over 10% from 2024 to 2030.

The demand for IR-based eye-blink sensors is increasing in automotive, healthcare, and industrial sectors due to advancements in AI, IoT, and smart sensor technology. The adoption of real-time fatigue detection systems is driving innovations, making them essential for accident prevention and workplace safety worldwide.

3. Methodology with Component Description

The Anti Sleep Alarm System consists of three different processes (with modules); ESP 8266 Wi-Fi Module, Eye Blink Sensor, Buzzer, and GPS Module these are represented in the block diagram as shown below:



3.1 List of Material Used

3.4.1 Equipment Used

- Bread Board
- Soldering iron
- Cutter
- Wire Stripper
- De-soldering pump
- USB connector
- Male to Male and Female to Male Jumper Wires

3.4.2 Component Used

- Buzzer
- ESP 8266 Wi-Fi Module (Microcontroller)
- GPS Module NEO 7M
- 3.7 Volt Battery
- Witty fox – Eye Blink sensor

3.2 ESP 8266 Wi-Fi Module

The ESP8266 is a highly versatile and cost-effective Wi-Fi-enabled microcontroller, making it a popular choice for IoT applications, smart home automation, wireless sensor networks, and embedded systems. It is powered by a 32-bit Tensilica L106 processor, running at up to 80 MHz (overclockable to 160 MHz), providing efficient processing for a wide range of applications.

Despite its compact size, the ESP8266 offers multiple GPIOs with PWM support, ADC (Analog-to-Digital Converter), and communication interfaces like I²C, SPI, and UART, ensuring seamless integration with sensors, actuators, and peripherals. Supporting 802.11 b/g/n Wi-Fi, it can function as an Access Point (AP), Station (STA), or both (AP+STA mode), making it ideal for wireless networking and edge computing applications.

The ESP8266 includes 64 KB of instruction RAM, 96 KB of data RAM, and external Flash memory support up to 16 MB, allowing ample storage for firmware, OTA updates, and data logging. It is programmable using the Arduino IDE, MicroPython, Lua, or Espressif's ESP-IDF, supporting development in C, C++, and Python, which enhances its flexibility for rapid prototyping and industrial applications.

With built-in TCP/IP, HTTPS, and MQTT support, the ESP8266 can directly connect to cloud platforms like AWS IoT, Google Firebase, and Blynk, making it an excellent choice for remote monitoring, smart metering, real-time automation, and AI-driven IoT solutions. Additionally, its power-saving modes (deep sleep, light sleep, and modem sleep) optimize energy consumption, making it ideal for battery-powered applications such as wearable tech and remote sensors.

Thanks to its affordability, strong developer community, robust networking capabilities, and ease of use, the ESP8266 continues to be a preferred microcontroller for DIY electronics, professional embedded systems, industrial automation, smart cities, and energy-efficient IoT solutions.

The ESP8266 also features Secure Sockets Layer (SSL) support, enabling encrypted communication for enhanced cybersecurity in IoT applications. Its low latency, real-time processing, and OTA update capabilities make it ideal for automated control systems, robotics, and machine-to-machine (M2M) communication.

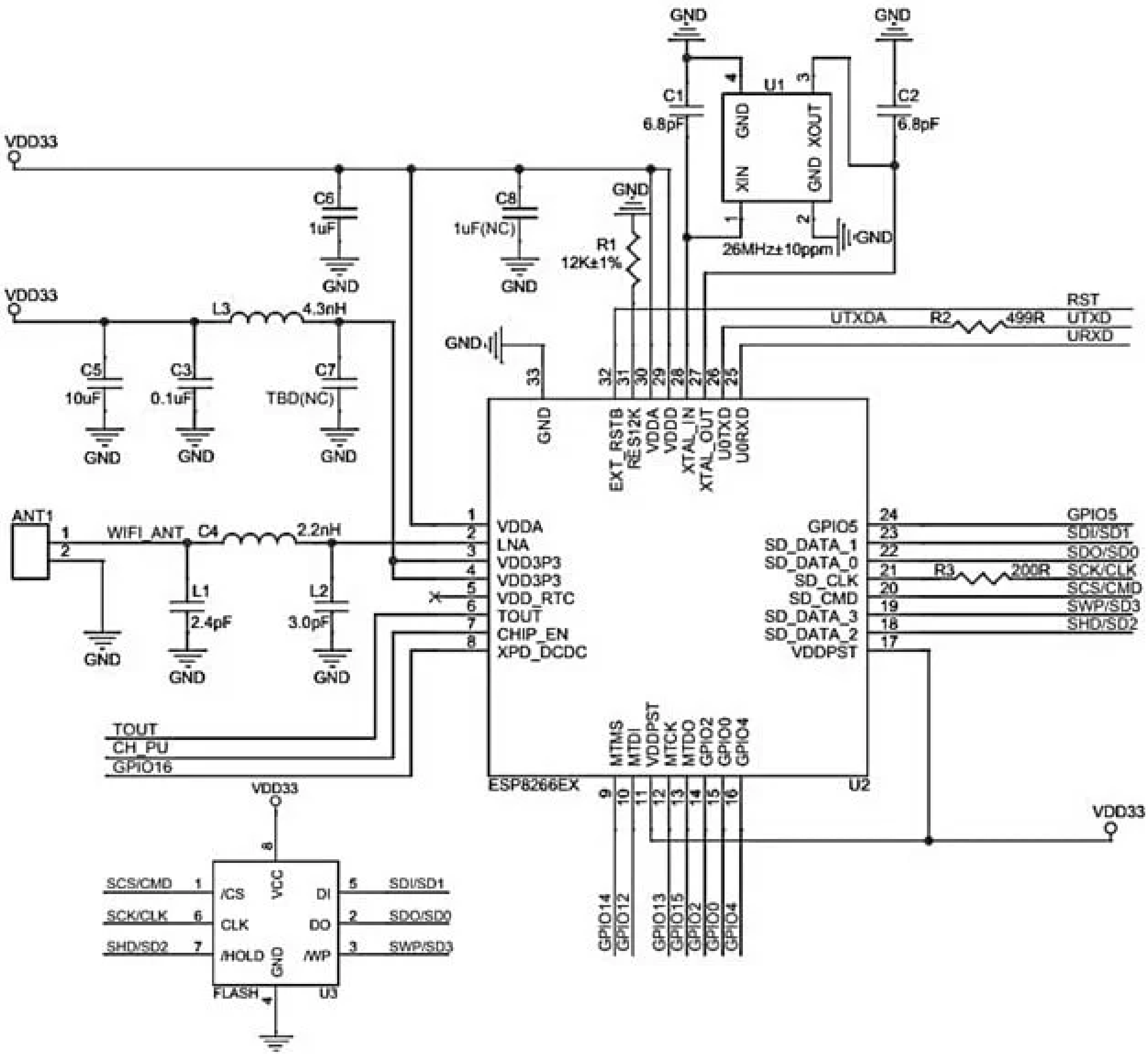


Fig. 1. Tensilica Xtensa L106 processor

3.1.1 Tensilica Xtensa L106 processor

The Tensilica Xtensa L106 is a 32-bit RISC microprocessor developed by Cadence and used in the ESP8266 microcontroller by Espressif Systems. It is designed for low power consumption and high efficiency, making it an ideal choice for embedded systems and IoT applications. Operating at a default clock speed of 80 MHz, it can be overclocked to 160 MHz for higher performance. The processor integrates SRAM for fast data processing and supports external Flash memory for program storage. One of its key advantages is its customizable instruction set, allowing optimized processing for various applications. It features built-in WiFi support, enabling seamless wireless communication, and supports multiple peripheral interfaces, including I²C, SPI, UART, PWM, and ADC, making it highly versatile for connecting external devices. With deep sleep and power-saving modes, the Xtensa L106 is well-suited for battery-powered and energy-efficient applications. Its combination of high processing power, compact size, and integrated networking makes it widely used in smart home devices, automation, remote monitoring, and IoT-based wireless applications.

3.1.2 Pin Description and Diagram of ESP 8266

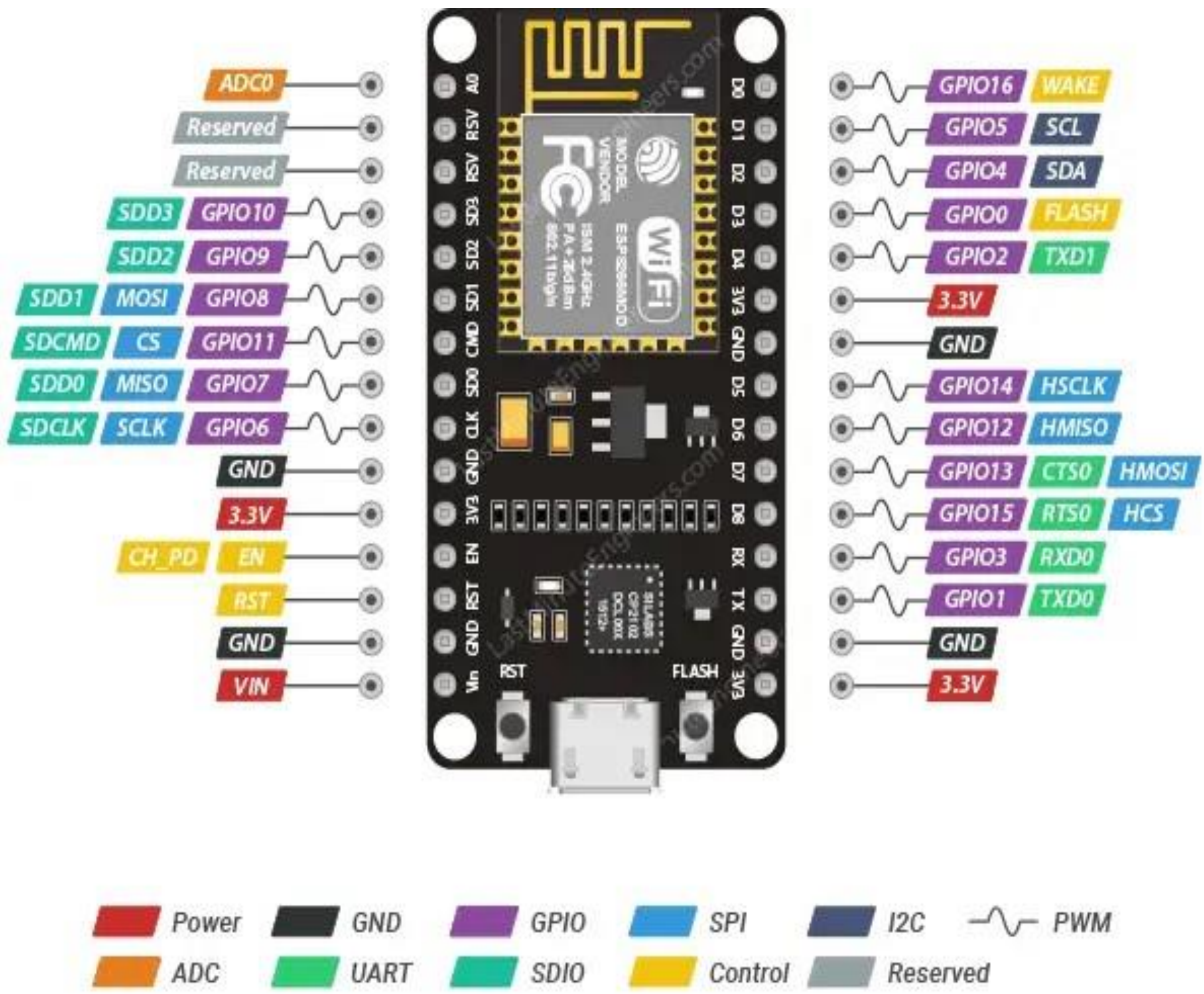


Fig. 2. Pin Diagram of ESP8266

The ESP8266 microcontroller, specifically the ESP-12E/F module used in NodeMCU, has 17 GPIO pins, along with power and communication pins. Below is a detailed description of its pins:

1. Power Pins:

- VCC (3.3V): Powers the ESP8266 (Requires 3.3V, *not 5V*).
- GND: Ground pin.

2. GPIO (General Purpose Input/Output) Pins:

- GPIO0 to GPIO16: Usable for digital input/output. Some have special functions.
- GPIO0 & GPIO2: Used for boot mode selection.
- GPIO16: Can be used for deep sleep wake-up.

3. Communication Pins:

- UART (Serial Communication) Pins:

  TX (GPIO1): UART Transmit (TXD).

  RX (GPIO3): UART Receive (RXD).

- I²C (Inter-Integrated Circuit) Pins:

  GPIO4 (SDA) and GPIO5 (SCL) are commonly used for I²C communication.

- SPI (Serial Peripheral Interface) Pins:

  MOSI (GPIO13), MISO (GPIO12), SCLK (GPIO14), CS (GPIO15) are used for SPI communication.

4. PWM (Pulse Width Modulation) Pins:

- Any GPIO can be used for PWM (except GPIO16).

5. Analog Input Pin:

- A0 (ADC0): 10-bit ADC for reading analog signals (0–1V input range).

6. Reset & Control Pins:

- EN (CH_PD): Chip Enable, must be pulled HIGH for normal operation.
- RST: Resets the module when pulled LOW.
- WAKE (GPIO16): Used to wake up the ESP8266 from deep sleep.

3.3 Eye Blink Sensor

The Witty Fox Eye Blink Sensor is an infrared (IR)-based module that detects eye blinks using an IR LED and a photodiode. The core principle is infrared light reflection, where the amount of reflected light varies based on whether the eye is open or closed. The photodiode captures the reflected signal, and an operational amplifier (Op-Amp) or comparator (such as LM358) processes it, converting it into a digital output (HIGH or LOW). This sensor is lightweight, non-intrusive, and cost-effective, making it ideal for various applications, including driver drowsiness detection, assistive technology, gaming interfaces, and human-computer interaction (HCI). The module is designed for easy integration with microcontrollers such as Arduino, Raspberry Pi, and ESP32, which can then process the output to trigger alarms, activate devices, or interface with other systems.

Fig. 3. a. Eye Blink Sensor

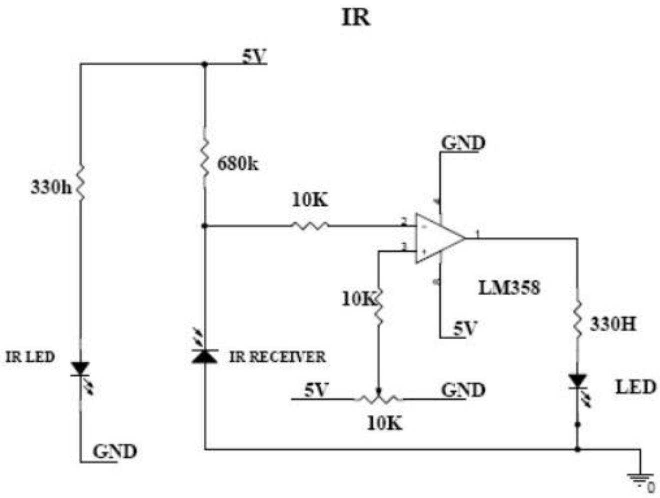


Fig. 3. b. Eye Blink Sensor PCB Structure

Components of the Eye Blink Sensor

The sensor consists of the following main components:

(a) Infrared (IR) Transmitter and Receiver: The eye blink sensor consists of an infrared (IR) LED as a transmitter and a photodiode as a receiver. The IR LED emits infrared light towards the eye, and the photodiode detects the reflected light. When the eye is open, most of the infrared light is absorbed by the eye, and only a small amount is reflected, resulting in a weak signal received by the photodiode. However, when the eye is closed, the eyelid reflects a higher amount of infrared light, increasing the signal detected by the photodiode. This change in reflection helps in determining eye blinks and detecting drowsiness.

(b) Operational Amplifier (Op-Amp) / Comparator (LM358): The weak signal received by the photodiode is amplified using an operational amplifier (Op-Amp) or a comparator such as the LM358. The comparator then processes the amplified signal and determines whether the eye is open or closed based on the intensity of the reflected infrared light.

(c) Potentiometer (VR1): The blue circular component in the image is a variable resistor, also known as a potentiometer. It is used to adjust the sensitivity of the IR sensor, allowing it to adapt to different lighting conditions and variations in skin types for more accurate detection.

(d) Indicator LEDs: The module features two LEDs: a power LED that indicates when the sensor is powered and an output LED that lights up when an eye blink is detected.

(e) Output Pins

- VCC (Power Supply, 3.3V – 5V)
- GND (Ground)
- OUT (Digital Output, High/Low signal depending on eye state)

Working Principle of the Eye Blink Sensor

1. Infrared Transmission & Reflection: The IR LED emits infrared light towards the eye. When the eye is open, most of the IR light is absorbed, and little reflection reaches the photodiode. When the eye is closed, the eyelid reflects more IR light back to the photodiode.
2. Signal Processing with Comparator (LM358): The photodiode generates a small electrical signal based on the reflected IR light. This signal is fed into the LM358 comparator, which compares it with a reference voltage. The output is HIGH (1) when the eye is closed and LOW (0) when the eye is open.
3. Microcontroller Integration: The digital output (HIGH/LOW) can be connected to a microcontroller (Arduino, Raspberry Pi, ESP32, etc.). The microcontroller can then trigger an alarm, send a notification, or activate a device when frequent eye closure is detected (for drowsiness detection).

3.4 Buzzer

Fig. 4. a. Buzzer

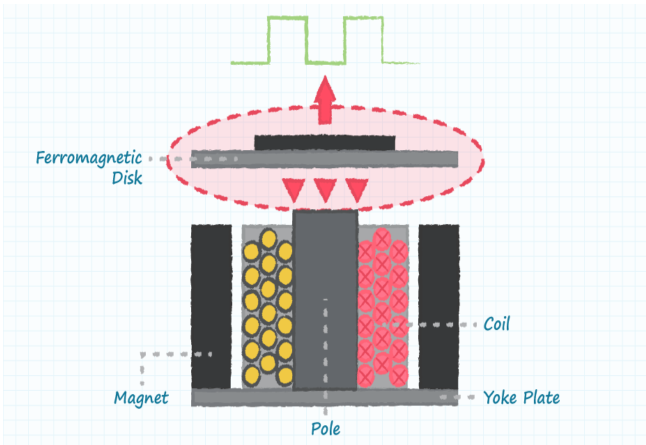


Fig. 4. b. Buzzer Working

A buzzer is an electroacoustic device that converts electrical energy into sound using vibrations. It is commonly used in alarms, timers, and various electronic devices for sound notifications.

Buzzers operate on the principle of piezoelectricity or electromagnetism.

Types of Buzzers

Buzzers are primarily categorized into two types:

1. Active Buzzer
2. Passive Buzzer

Active Buzzer:

An active buzzer has a built-in oscillator circuit that generates sound when a DC voltage is applied. Its working principle is based on a piezoelectric material or magnetic coil that vibrates upon receiving electrical energy. The internal circuit produces a constant frequency, typically ranging from 2 kHz to 4 kHz, allowing it to emit a single-tone sound without requiring an external control signal. Using an active buzzer is simple—connecting it to a power source, such as 5V or 12V, and providing a HIGH (ON) or LOW (OFF) signal via a microcontroller like ESP8266, Arduino Uno, or Nano enables operation. Unlike passive buzzers, no PWM or frequency adjustment is needed. Active buzzers are commonly used in alarm systems, notification devices, and household appliances like microwaves and washing machines. They offer several advantages, including easy interfacing with microcontrollers, fixed and consistent sound output, and no need for external circuitry. However, a key limitation is the lack of sound customization, making them suitable only for basic alert functions.

Passive Buzzer: A passive buzzer lacks an internal oscillator and requires an external signal, typically a PWM (Pulse Width Modulation) signal, to generate sound. Its working principle is based on a piezoelectric disk that vibrates when an alternating electrical signal is applied. The frequency of the PWM signal determines the tone or pitch of the sound, allowing for a range of tones between 500 Hz and 5 kHz. To use a passive buzzer, it must be connected to a microcontroller like ESP8266 or Arduino Uno or Nano, where functions like tone () in Arduino can generate the desired sound. This makes it ideal for creating melodies, alarms, and sound effects. Passive buzzers are commonly used in musical devices, toys, games, and multitone alarm systems. They offer advantages such as the ability to produce a wide range of sounds, support for custom sound effects and melodies, and lower power consumption. However, they require external circuits for signal generation and can be complex to implement without a microcontroller.

Note –

A technical information about SMTP Servers: An SMTP (Simple Mail Transfer Protocol) server is a specialized mail server responsible for sending, receiving, and relaying emails across the internet. Acting as a bridge between email clients and recipient mail servers, it ensures proper email transmission following standardized rules defined by RFC 5321. When an email is sent, the SMTP server processes it by verifying the sender, looking up the recipient’s domain’s Mail Exchange (MX) records, and relaying it to the destination server. If the recipient's server is unavailable, the SMTP server queues the email for later delivery. It uses specific commands such as HELO/EHLO (initiating conversation), MAIL FROM (identifying sender), RCPT TO (specifying recipient), DATA (transmitting the email body), and QUIT (closing the connection). SMTP servers operate over ports 25, 465, 587, and 2525, with Port 587 being the preferred option for secure email submission using STARTTLS encryption. To enhance security, modern SMTP servers require SMTP Authentication (SMTP

AUTH) and use TLS or SSL encryption to prevent spam and unauthorized access. SMTP servers can be categorized into outgoing, relay, and inbound servers, handling email transmission at different levels. Free SMTP services, such as those from Gmail, Yahoo, and Outlook, have sending limits, whereas paid SMTP services like SendGrid, Mailgun, and Amazon SES allow bulk email sending with advanced analytics. SMTP servers provide reliable email delivery, support for marketing campaigns, and error handling through response codes like 421 (service unavailable), 450 (mailbox issues), and 550 (email blocked due to spam filtering). Understanding SMTP servers is crucial for efficient, secure, and scalable email communication in both personal and business applications.

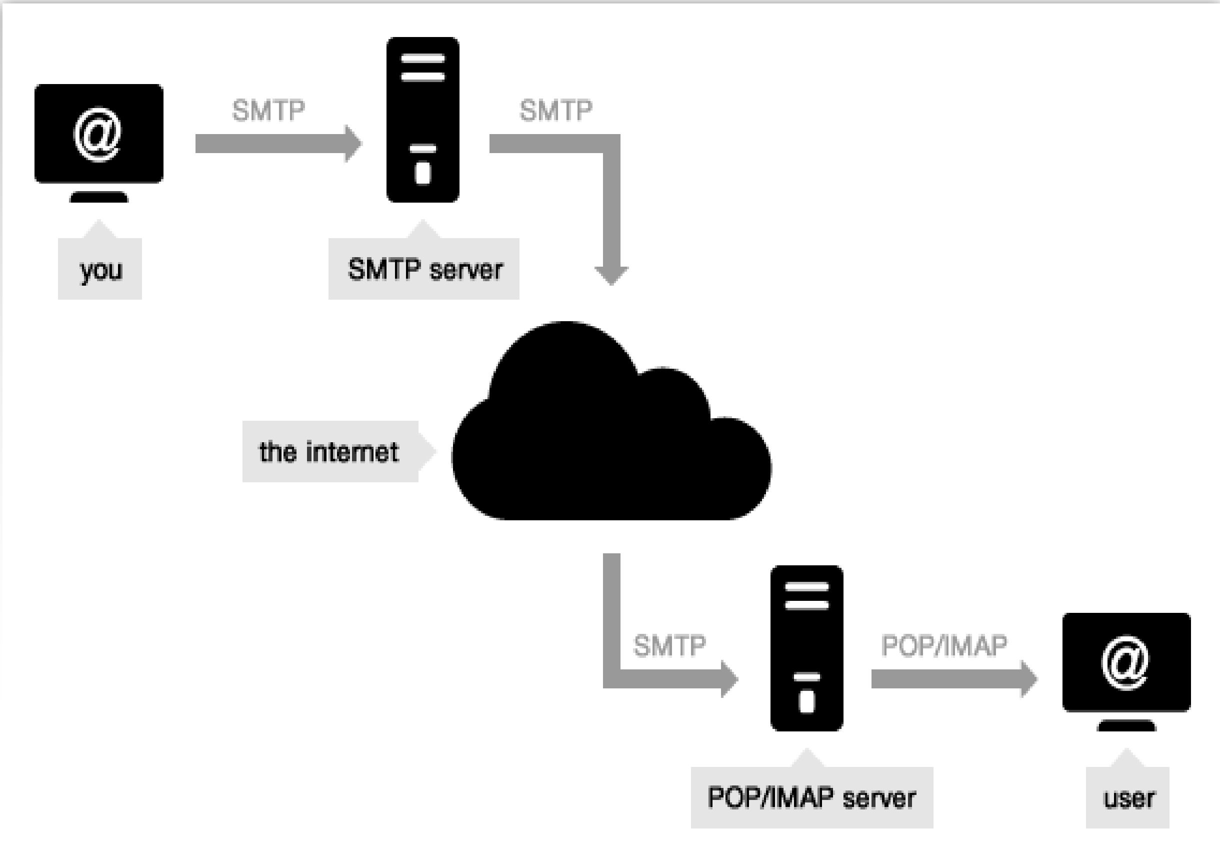


Fig. 5. SMTP Server

In my project, SMTP can be used to send real-time email alerts when drowsiness is detected using the eye blink sensor, ensuring immediate notification to family members, fleet managers, or emergency contacts. The ESP8266, connected via Wi-Fi, can trigger an SMTP request to send an email with details like detection time, driver's condition, and GPS location formatted as a Google Maps link, allowing real-time tracking. If multiple drowsiness events occur within a short period, SMTP can escalate alerts to emergency services, enhancing road safety. Implementing SMTP with ESP8266 involves using an SMTP server (e.g., Gmail's smtp.gmail.com, port 587), configuring authentication, and sending secure, encrypted emails using the ESP-Mail-Client library in Arduino. This approach enhances driver safety by enabling automated, location-tracked, and instant notifications, ensuring rapid response in critical situations, making the system more reliable and effective, especially in Wi-Fi-enabled environments such as company vehicles or smart transportation systems

3.5 GPS Module NEO 7M

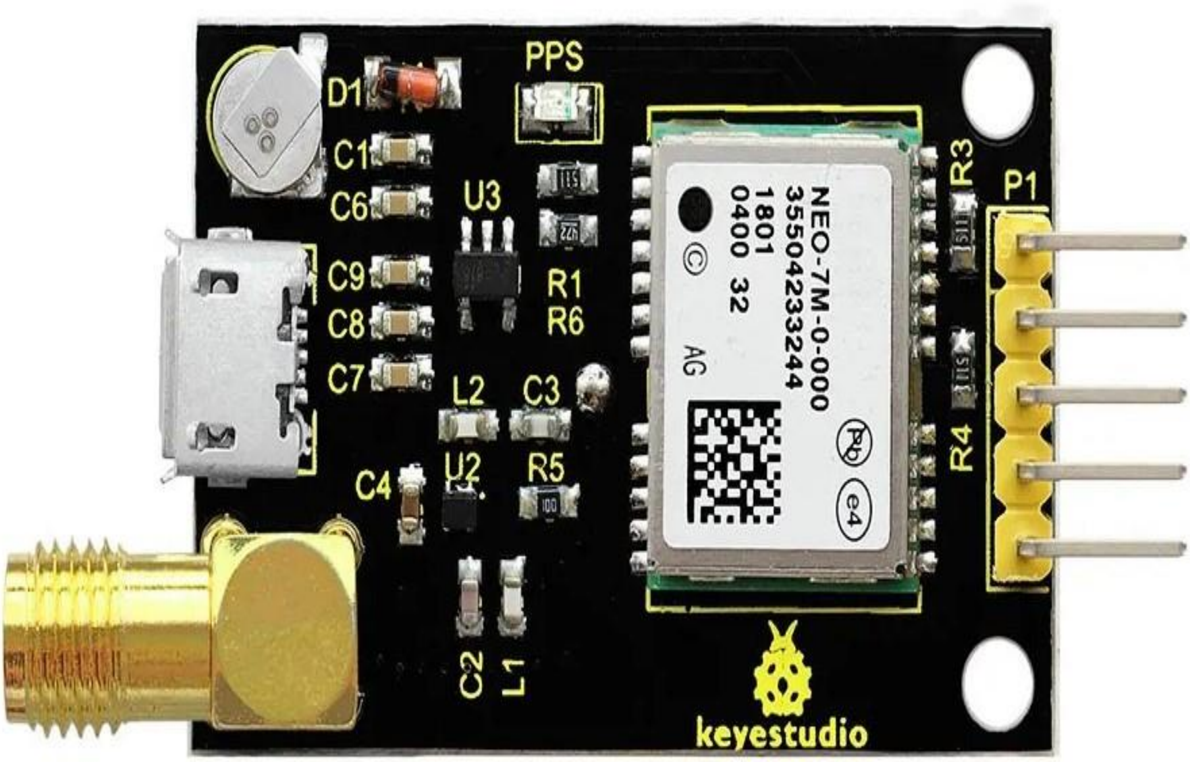


Fig. 6. a. GPS Module NEO 7M

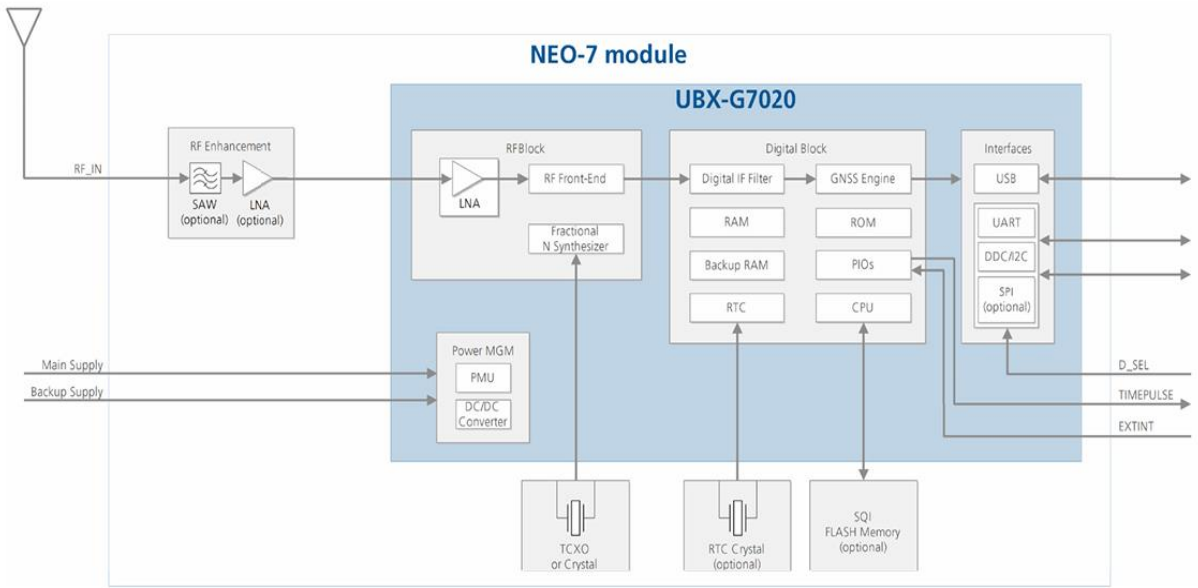


Fig. 6. b. Schematic Diagram of GPS Module NEO 7M

The Neo-7M is a high-performance GPS module from u-blox, designed for applications requiring precise positioning and navigation. It supports GPS, GLONASS, and SBAS (WAAS, EGNOS, MSAS, GAGAN) systems for enhanced accuracy.

Pin	Description
VCC	Power supply (3.3V - 5V)
GND	Ground (0V reference)
TX (Transmit)	Serial data output (connect to MCU RX pin)
RX (Receive)	Serial data input (connect to MCU TX pin)
PPS (Pulse Per Second)	Time pulse output for high-accuracy timing applications
RST (Reset)	Active LOW reset pin (optional)
V_BCKP (Backup Power)	Optional backup battery for RTC and faster cold starts
SDA/SCL (I²C - Optional)	Some variants support I²C for alternative communication

Tabular Column. 1

- Operating Voltage: 3.3V - 5V (Most modules have a built-in voltage regulator)
- Current Consumption: ~45mA (Active Mode)
- Baud Rate (Default): 9600 bps (Can be changed via U-Center software)
- Antenna: Needs an active or passive GPS antenna for better reception
- Fix Time:

  Cold Start: ~27 sec; Warm Start: ~1 sec

- Accuracy:

  Positioning: ~2.5 meters; Timing (PPS Output): ~30 ns

- Protocols: Supports NMEA, UBX (u-blox proprietary), RTCM

4. Circuit Connection (With Tabular form), Output screen and Code

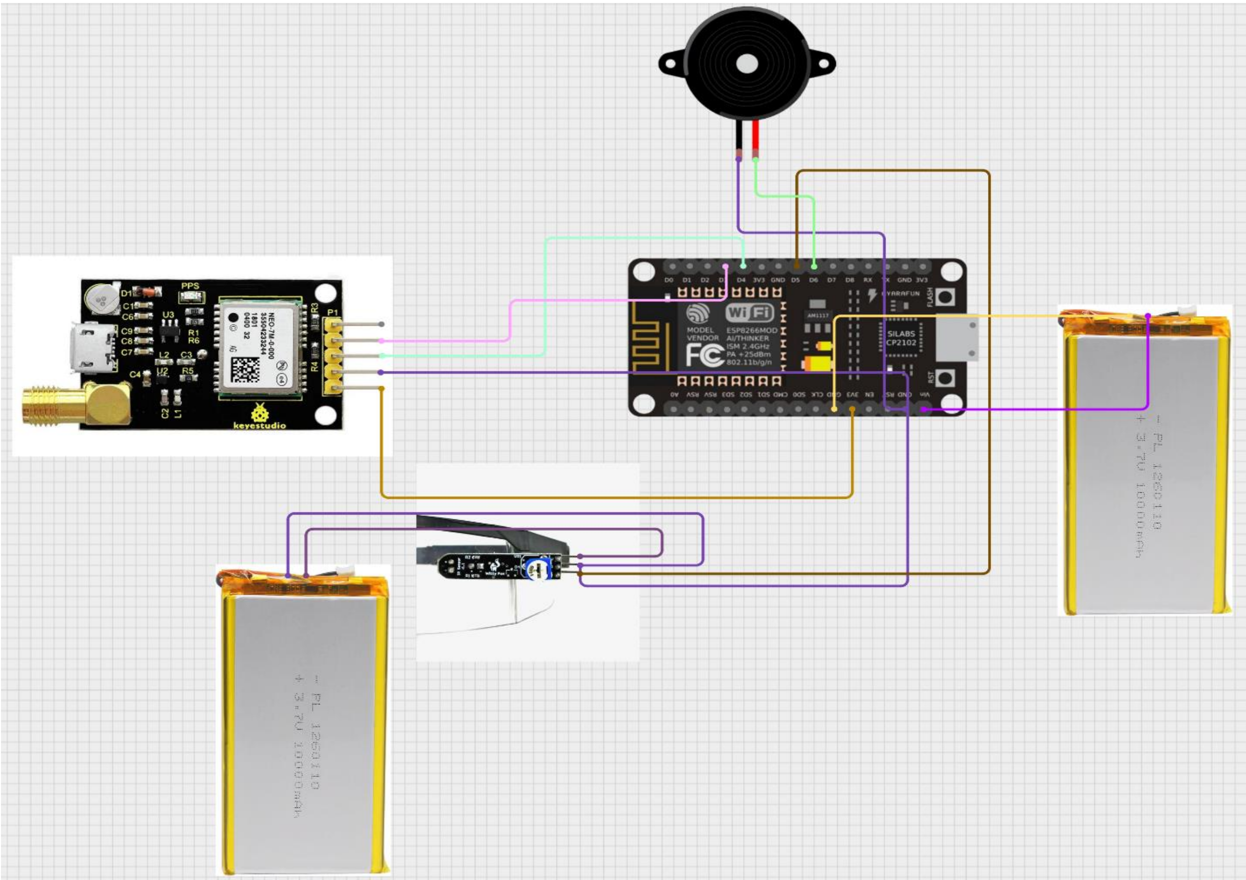


Fig. 7. a. Circuit Diagram

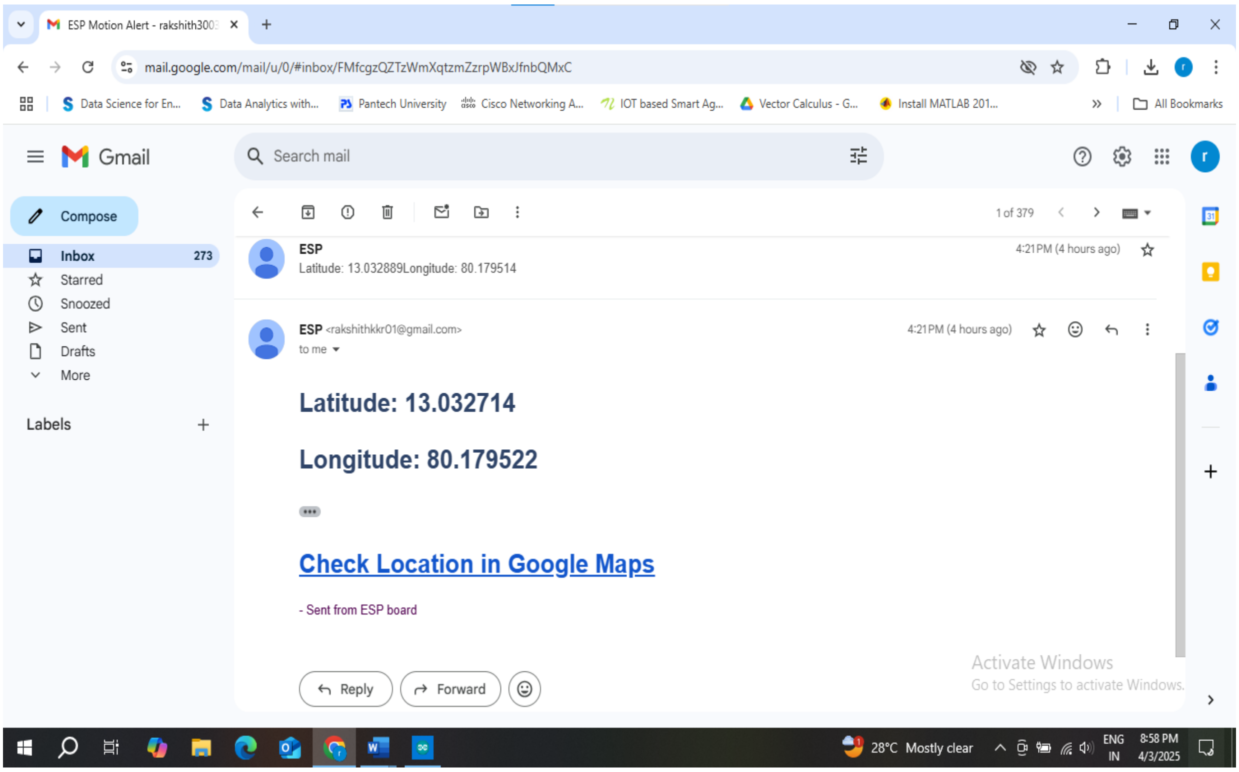


Fig. 7. b. Output Screen

Component 1	Pin	Connected to	Pin
Eye Blink Sensor	VCC	ESP 8266	3.7 V
Eye Blink Sensor	GND	ESP 8266	GND
Eye Blink Sensor	Data	ESP 8266	GPIO D5
Buzzer	+ve	ESP 8266	GPIO D6
Buzzer	-ve	ESP 8266	GND
GPS Module	TX	ESP 8266	GPIO D4
GPS Module	RX	ESP 8266	GPIO D3
GPS Module	VCC	ESP 8266	3.3 V
GPS Module	GND	ESP 8266	GND
3.7V Battery	+ve	ESP 8266	VIN
3.7V Battery	-ve	ESP 8266	GND

Tabular Column. 2

4.2 Code (Remember to install the libraries before compiling the code)

```
#include <Arduino.h>

#include <ESP_Mail_Client.h>

#include <TinyGPSPlus.h>

TinyGPSPlus gps;

#if defined(ESP32)

  #include <WiFi.h>

  #define RXD2 16

  #define TXD2 17

  HardwareSerial neo7m(2);

  #define eye_blink_sensor 5

  #define BUZZER_PIN 26  // Define the buzzer pin

#elif defined(ESP8266)

  #include <ESP8266WiFi.h>

  #include <SoftwareSerial.h>

  const int rxPin = D4, txPin = D3;

  SoftwareSerial neo7m(rxPin, txPin);

  #define eye_blink_sensor D5

  #define BUZZER_PIN D6

#endif


#define AUTHOR_EMAIL "rakshithkkr01@gmail.com"

#define AUTHOR_PASSWORD "diekvjbxtwtdkgij"

#define RECIPIENT_EMAIL "rakshith300306@gmail.com"

#define SMTP_HOST "smtp.gmail.com"

#define SMTP_PORT 465


#define WIFI_SSID "SSID"

#define WIFI_PASSWORD "Password"


SMTPSession smtp;
```

```
void smtpCallback(SMTP_Status status);

int drowsinessCount = 0;

void setup() {
  Serial.begin(115200);
  pinMode(eye_blink_sensor, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);

  #if defined(ESP32)
    neo7m.begin(9600, SERIAL_8N1, RXD2, TXD2);
  #elif defined(ESP8266)
    neo7m.begin(9600);
  #endif

  Serial.println();
  Serial.print("Connecting to AP");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(200);
  }
  Serial.println("\nWiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  int val = digitalRead(eye_blink_sensor);
```

```
  if (val == HIGH) {
    Serial.println("Drowsiness detected!!!");

    // Activate buzzer for 1 second
    digitalWrite(BUZZER_PIN, HIGH);
    delay(1000);
    digitalWrite(BUZZER_PIN, LOW);

    drowsinessCount++;

    if (drowsinessCount > 3) {
      send_email_alert();
      drowsinessCount = 0;
    }
  }
  delay(1000);
}

void send_email_alert() {
  boolean newData = false;

  for (unsigned long start = millis(); millis() - start < 10000;) {
    while (neo7m.available()) {
      if (gps.encode(neo7m.read())) {
        newData = true;
        break;
      }
    }
  }
  if (!newData || !gps.location.isValid()) {
```

```
  Serial.println("No Valid GPS Data Found.");
  return;
}

newData = false;
String latitude = String(gps.location.lat(), 6);
String longitude = String(gps.location.lng(), 6);
Serial.println("Latitude: " + latitude);
Serial.println("Longitude: " + longitude);
smtp.debug(1);
smtp.callback(smtpCallback);

ESP_Mail_Session session;
session.server.host_name = SMTP_HOST;
session.server.port = SMTP_PORT;
session.login.email = AUTHOR_EMAIL;
session.login.password = AUTHOR_PASSWORD;
session.login.user_domain = "";


SMTP_Message message;
message.sender.name = "ESP";
message.sender.email = AUTHOR_EMAIL;
message.subject = "ESP Motion Alert";
message.addRecipient("rakshith sooriya", RECIPIENT_EMAIL);

String htmlMsg = "<div style=\"color:#2f4468;\">";
htmlMsg += "<h1>Latitude: " + latitude + "</h1>";
htmlMsg += "<h1>Longitude: " + longitude + "</h1>";
```

```
  htmlMsg += "<h1><a href=\"http://maps.google.com/maps?q=loc:" + latitude + "," + longitude + "\">Check Location in Google Maps</a></h1>";
  htmlMsg += "<p>- Sent from ESP board</p>";
  htmlMsg += "</div>";

  message.html.content = htmlMsg.c_str();  // Set HTML content
  message.text.charSet = "us-ascii";
  message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;

  if (!smtp.connect(&session)) {
    Serial.println("Error: Failed to connect to SMTP server.");
    return;
  }

  if (!MailClient.sendMail(&smtp, &message)) {
    Serial.println("Error sending email: " + smtp.errorReason());
  } else {
    Serial.println("Email sent successfully.");
  }
}

void smtpCallback(SMTP_Status status) {
  Serial.println(status.info());
  if (status.success()) {
    Serial.println("----------------");
    ESP_MAIL_PRINTF("Message sent success: %d\n", status.completedCount());
    ESP_MAIL_PRINTF("Message sent failed: %d\n", status.failedCount());
    Serial.println("----------------\n");

    struct tm dt;
```

```
    for (size_t i = 0; i < smtp.sendingResult.size(); i++) {
      SMTP_Result result = smtp.sendingResult.getItem(i);
      time_t ts = (time_t)result.timestamp;
      localtime_r(&ts, &dt);

      ESP_MAIL_PRINTF("Message No: %d\n", i + 1);
      ESP_MAIL_PRINTF("Status: %s\n", result.completed ? "success" : "failed");
      ESP_MAIL_PRINTF("Date/Time: %d/%d/%d %d:%d:%d\n", dt.tm_year + 1900, dt.tm_mon + 1, dt.tm_mday, dt.tm_hour, dt.tm_min, dt.tm_sec);
      ESP_MAIL_PRINTF("Recipient: %s\n", result.recipients);
      ESP_MAIL_PRINTF("Subject: %s\n", result.subject);
    }
    Serial.println("----------------\n");
  }
}
```

5. Implementation, Testing and Evaluation

5.1 Implementation

After assembling and testing the different modules—ESP32, Neo-7M GPS module, eye blink sensor, buzzer, and voltage sources—each component was verified for proper functionality. The ESP32 was programmed to detect drowsiness through the eye blink sensor, triggering an alert via the buzzer if abnormal blinking patterns were detected. After three consecutive alerts, the system automatically captured the GPS coordinates and sent an email notification using an SMTP server.

The Neo-7M GPS module was tested to ensure real-time location tracking, while the Wi-Fi connectivity of ESP32 was configured to send alerts over the internet reliably. Once all the modules were confirmed to work correctly, they were securely enclosed in a compact and durable plastic casing to protect against environmental factors and ensure easy installation within a vehicle. The enclosure was designed to provide ventilation and easy access to ports for maintenance or future upgrades. The final prototype was mounted in a convenient location within the vehicle, ensuring optimal sensor placement for accurate monitoring and efficient operation.

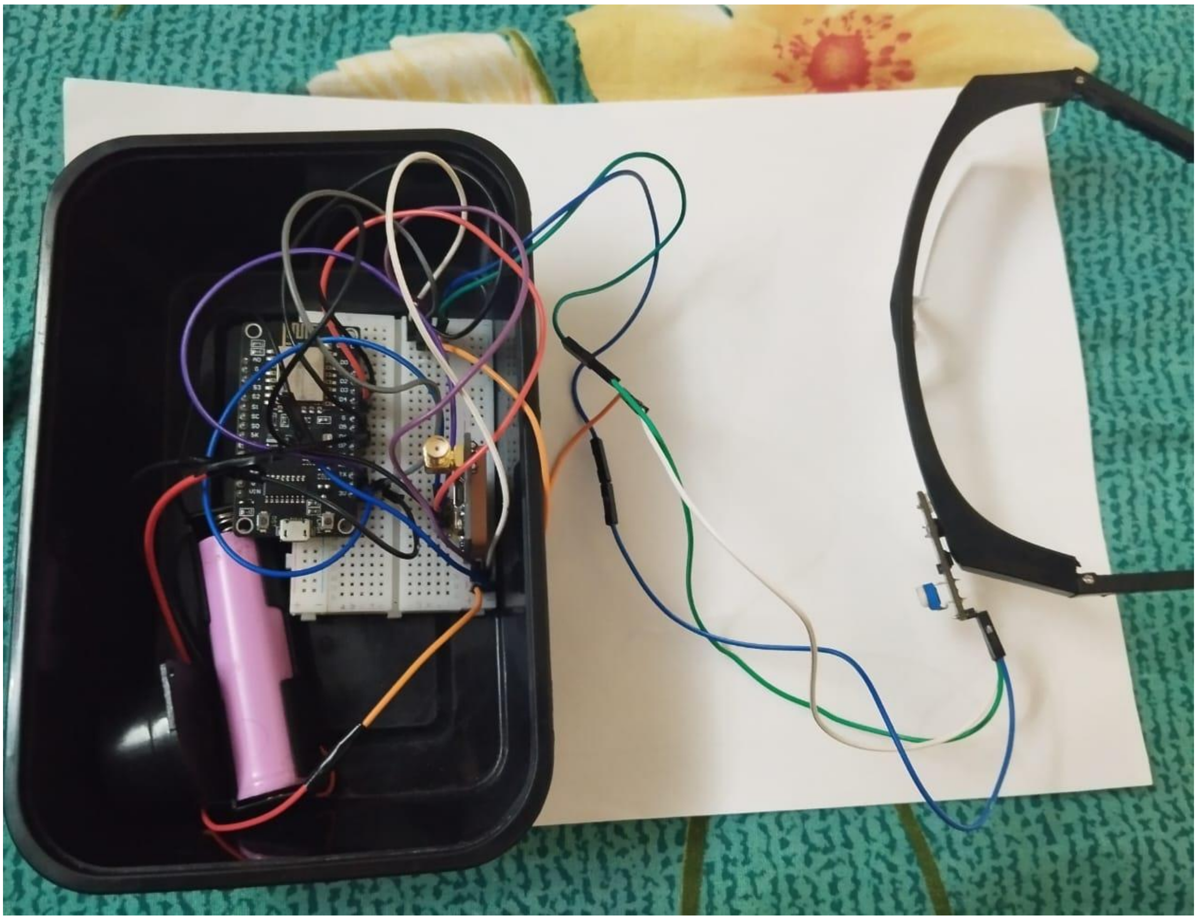


Fig. 8. Picture of Developed Anti-Sleep Alarm System

5.2 Testing and Result

The system was tested by simulating drowsiness using the eye blink sensor. When abnormal blinking patterns were detected, the buzzer activated, alerting the user. After three consecutive detections, the ESP32 successfully retrieved GPS coordinates from the Neo-7M module and sent an email alert via SMTP. Multiple trials confirmed accurate GPS location tracking and reliable email delivery. The Wi-Fi-based alert system functioned without delays, ensuring real-time notifications. The final prototype demonstrated effective drowsiness detection, prompt alerts, and seamless integration into a vehicle environment.

6. Conclusion

The implemented drowsiness detection system successfully integrates an ESP32, an eye blink sensor, a buzzer, and a GPS module to enhance driver safety. The system accurately detects drowsiness through eye blink monitoring and provides an immediate buzzer alert. After three consecutive detections, it sends an email notification with GPS coordinates via WiFi. Testing demonstrated its reliability in real-time scenarios, ensuring accurate detection and quick alerts. The compact and cost-effective design allows easy installation in vehicles. This system offers an effective solution for preventing fatigue-related accidents, contributing to road safety and driver awareness. Future improvements may include AI-based enhancements.

7. References

[1] **ESP32 Microcontroller:** Espressif Systems (2023). *ESP32 Series Datasheet.* Retrieved from Espressif Documentation.

[2] **Neo-7M GPS Module:** u-blox (2018). *NEO-7M GPS Receiver Datasheet.* Retrieved from u-blox Official Site.

[3] **Eye Blink Sensor (IR-Based):** Robokits India (2022). *IR Eye Blink Sensor Module Datasheet.* Available at Robokits.

[4] **Buzzer Module:** SparkFun (2021). *Active and Passive Buzzer Guide.* Retrieved from SparkFun Datasheets.

[5] **Drowsiness Detection with ESP32 & GPS Tracking:** IoT Projects Hub (2022). *ESP32 Sleep Alert System for Drivers.* Retrieved from IoT Design Pro.

[6] **GPS-Based Tracking Systems with ESP32:** Random Nerd Tutorials (2023). *ESP32 GPS Tracker Using Neo-7M Module.* Available at Random Nerd Tutorials.

[7] **WiFi-Based Notification Systems using ESP32:** Rui Santos (2023). *How to Send Email Alerts with ESP32.* Retrieved from Random Nerd Tutorials.

[8] **Power Optimization for Battery-Based ESP32 Projects:** Espressif (2023). *Low Power Modes in ESP32 for IoT Applications.* Available at Espressif Technical Documents.