LAB REPORT

-LAB FINAL

# IRE 212 : IoT Architecture and Technologies Sessional

|  |  |
| --- | --- |
| PREPARED BY  Mobashira Mehajabin Arpita  Rupu Rani Ghosh  Mehrin Farzana  ID: 2101008, 2101010, 2101013  Session: 2021-2022  Date: 09/12/2024 | SUPERVISED BY  Suman Saha  Assistant Professor  & Chairman  Department of IRE, BDU |

|  |  |
| --- | --- |
| logo1 | BANGABANDHU SHEIKH MUJIBUR RAHMAN DIGITAL UNIVERSITY  (BDU) |

**Lab Report-Lab Final**

**Experiment Title: Use GPS and GSM modules to track the location of a vehicle and send alerts if unauthorized movement is detected.**

**Components and Libraries:**

Components Required for this Project are:

* Arduino UNO: The microcontroller board used for interfacing and data processing.
* GPS Module (e.g., NEO-6M): Receives signals from satellites and provides NMEA data.
* ESP32: For processing data and sending alerts
* TinyGPS++ Library: Parses and extracts meaningful data from raw GPS data.
* USB Cable: For powering the Arduino and monitoring serial output.
* Connecting Wires: To establish a connection between the Arduino and GPS module.
* Computer with Arduino IDE: Used for programming and viewing the output.

**Circuit Diagram:**

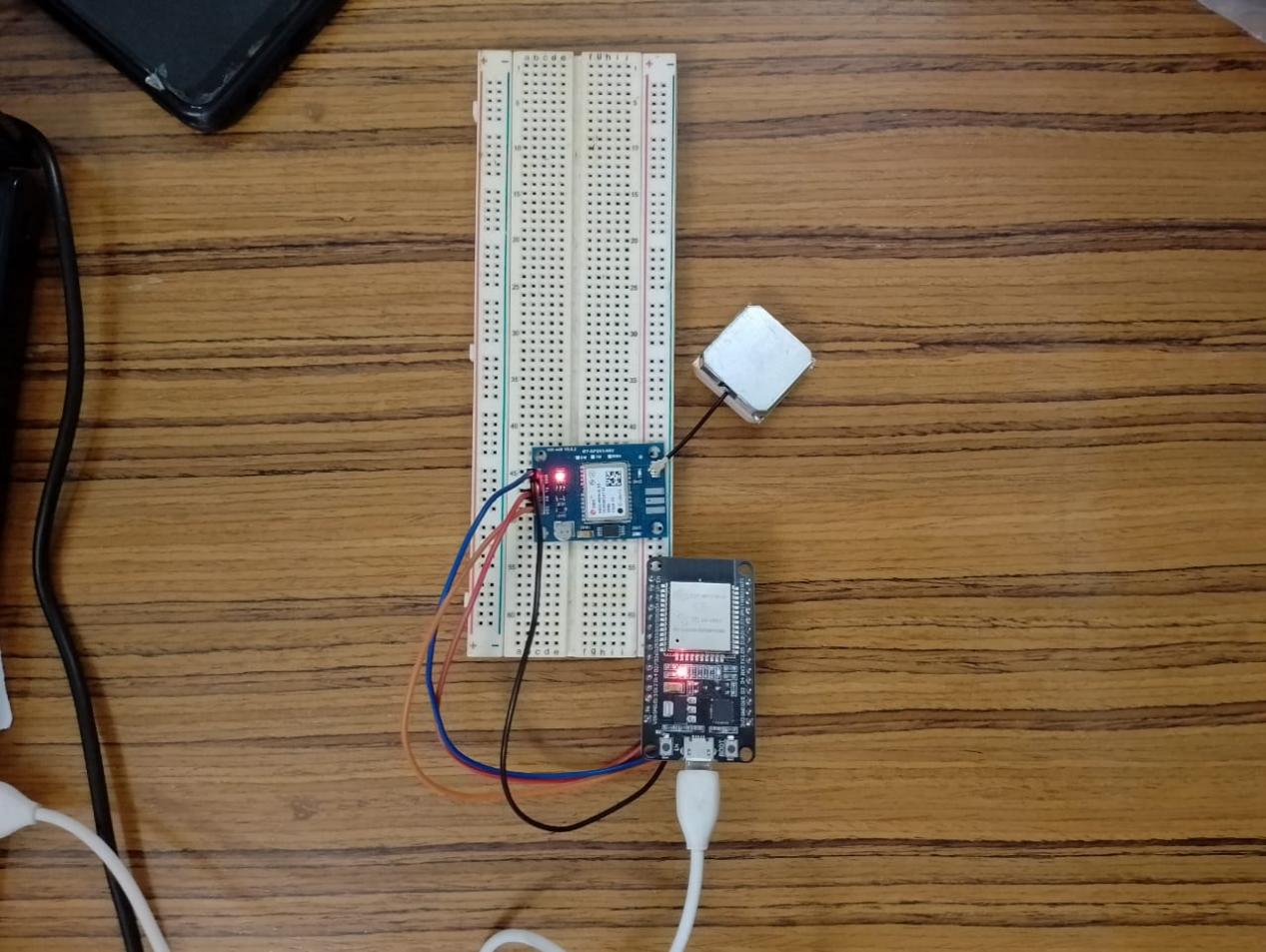


Fig-1:GPS Module Interfacing With Arduino UNO

**Code:**

#define BLYNK\_TEMPLATE\_ID "TMPL6Zu-0oBiD"

#define BLYNK\_TEMPLATE\_NAME "iot motion detection"

#define BLYNK\_AUTH\_TOKEN "wnAoDCRF4wGjJXwnLEXVdqrIO2hf7CwX"

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <TinyGPS++.h>

// Blynk credentials

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Rupu's dimension";

char pass[] = "zaqwsxcde";

// PIR Sensor Pin

#define PIR\_SENSOR 13

// GPS Module Pins

#define GPS\_RX\_PIN 16

#define GPS\_TX\_PIN 17

// Create objects for GPS and Timer

TinyGPSPlus gps;

HardwareSerial gpsSerial(1); // Use Serial1 for GPS module

BlynkTimer timer;

// Function to send theft alert

void notifyOnTheft() {

  int isTheftAlert = digitalRead(PIR\_SENSOR);

  if (isTheftAlert == 1) {

    Serial.println("Unauthorized Movement Detected!");

    Blynk.logEvent("theft\_alert", "Unauthorized Movement Detected!"); // Changed to "theft\_alert"

    // Log the current GPS location

    if (gps.location.isValid()) {

      Serial.println("Sending GPS Location with Theft Alert...");

      Serial.print("Latitude: ");

      Serial.println(gps.location.lat(), 6);

      Serial.print("Longitude: ");

      Serial.println(gps.location.lng(), 6);

      Blynk.virtualWrite(V0, gps.location.lat(), gps.location.lng()); // Send to Blynk virtual pin

    } else {

      Serial.println("GPS Location: Invalid");

    }

  }

}

void setup() {

  // Initialize Serial for Debugging

  Serial.begin(115200);

  Serial.println("Initializing...");

  // Initialize PIR Sensor

  pinMode(PIR\_SENSOR, INPUT);

  // Initialize GPS Module

  gpsSerial.begin(9600, SERIAL\_8N1, GPS\_RX\_PIN, GPS\_TX\_PIN);

  Serial.println("GPS Module Initialized");

  // Connect to Wi-Fi

  Serial.println("Connecting to Wi-Fi...");

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    Serial.println("Connecting...");

  }

  Serial.println("Wi-Fi Connected");

  // Initialize Blynk

  Blynk.begin(auth, ssid, pass);

  // Set PIR sensor check interval

  timer.setInterval(5000L, notifyOnTheft);

}

void loop() {

  // Handle GPS Data

  while (gpsSerial.available() > 0) {

    gps.encode(gpsSerial.read());

    if (gps.location.isUpdated()) {

      Serial.println("GPS Location Updated:");

      Serial.print("Latitude: ");

      Serial.println(gps.location.lat(), 6);

      Serial.print("Longitude: ");

      Serial.println(gps.location.lng(), 6);

    }

  }

  // Run Blynk and Timer

  Blynk.run();

  timer.run();

}

**Output:**

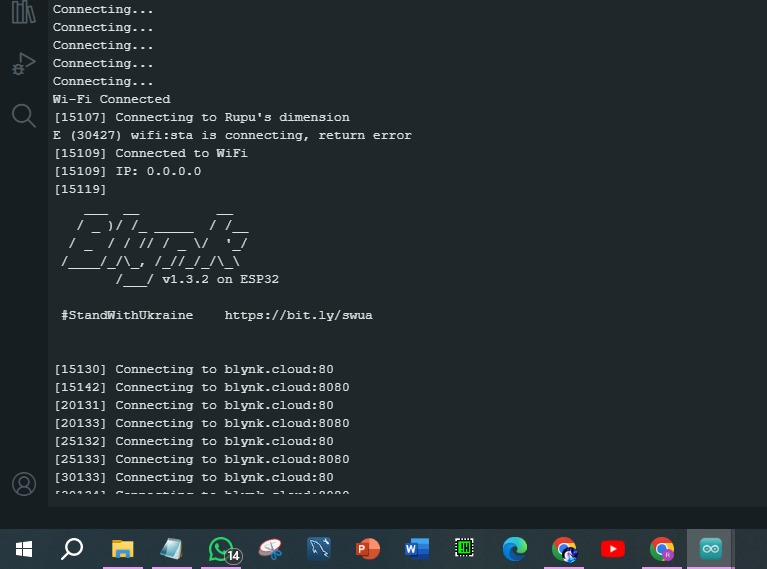
****

Fig-2: Serial Monitor reading of GPS Module Interfacing