ASSET TRACKING USING NODE MCU WITHOUT GPS MODULE

IOT

Muaz Ata Ur Rehman
muazthemaster@gmail.com

Asset Tracking using node mcu without GPS module

Description:

- In this project we will be using Node MCU esp8266 to track our location

Software:

- Arduino IDE

Components Required:

- Node mcu esp8266

CODE:

```
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include "ESP8266WiFi.h"
char ssid[] = "********";        // your network SSID name
char pass[] = "********";         // your network password
//Credentials for Google GeoLocation API...
const char* Host = "www.googleapis.com";
String thisPage = "/geolocation/v1/geolocate?key=";
String key = "your api key";
int status = WL_IDLE_STATUS;
String jsonString = "{\n";
double latitude    = 0.0;
double longitude   = 0.0;
double accuracy    = 0.0;
int more_text = 1;    // set to 1 for more debug output
void setup()   {
  Serial.begin(9600);
  Serial.println("Start");


 // Set WiFi to station mode and disconnect from an AP if it was previously connected
```

```
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");

// We start by connecting to a WiFi network
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(".");
}
void loop() {
  char bssid[6];
  DynamicJsonBuffer jsonBuffer;
  Serial.println("scan start");

// WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found...");
    if (more_text) {
```

```
    Serial.println("\"wifiAccessPoints\": [");
    for (int i = 0; i < n; ++i)
    {
      Serial.println("{");
      Serial.print("\"macAddress\" : \"");
      Serial.print(WiFi.BSSIDstr(i));
      Serial.println("\",");
      Serial.print("\"signalStrength\": ");
      Serial.println(WiFi.RSSI(i));
      if (i < n - 1)
      {
        Serial.println("},");
      }
      else
      {
        Serial.println("}");
      }
    }
    Serial.println("]");
    Serial.println("}");
  }
  Serial.println(" ");
 }
// now build the jsonString...
 jsonString = "{\n";
 jsonString += "\"homeMobileCountryCode\": 234,\n"; // this is a real UK MCC
 jsonString += "\"homeMobileNetworkCode\": 27,\n";  // and a real UK MNC
 jsonString += "\"radioType\": \"gsm\",\n";        // for gsm
 jsonString += "\"carrier\": \"Vodafone\",\n";     // associated with Vodafone
 jsonString += "\"wifiAccessPoints\": [\n";
```

```
for (int j = 0; j < n; ++j)
{
  jsonString += "{\n";
  jsonString += "\"macAddress\" : \"";
  jsonString += (WiFi.BSSIDstr(j));
  jsonString += "\",\n";
  jsonString += "\"signalStrength\": ";
  jsonString += WiFi.RSSI(j);
  jsonString += "\n";
  if (j < n - 1)
  {
    jsonString += "},\n";
  }
  else
  {
    jsonString += "}\n";
  }
}
jsonString += ("]\n");
jsonString += ("}\n");
//---------------------------------------------------------------- Serial.println("");
//Connect to the client and make the api call
WiFiClientSecure client;
Serial.print("Requesting URL: ");
Serial.println("https://" + (String)Host + thisPage + key);
Serial.println(" ");
if (client.connect(Host, 443)) {
  Serial.println("Connected");
  client.println("POST " + thisPage + key + " HTTP/1.1");
  client.println("Host: " + (String)Host);
```

```
  client.println("Connection: close");
  client.println("Content-Type: application/json");
  client.println("User-Agent: Arduino/1.0");
  client.print("Content-Length: ");
  client.println(jsonString.length());
  client.println();
  client.print(jsonString);
  delay(500);
 }
 //Read and parse all the lines of the reply from server
 while (client.available()) {
  String line = client.readStringUntil('\r');
  if (more_text) {
    Serial.print(line);
  }
  JsonObject& root = jsonBuffer.parseObject(line);
  if (root.success()) {
    latitude    = root["location"]["lat"];
    longitude   = root["location"]["lng"];
    accuracy   = root["accuracy"];
  }
 }
 Serial.println("closing connection");
 Serial.println();
 client.stop();
 Serial.print("Latitude = ");
 Serial.println(latitude, 6);
 Serial.print("Longitude = ");
 Serial.println(longitude, 6);
 Serial.print("Accuracy = ");
```

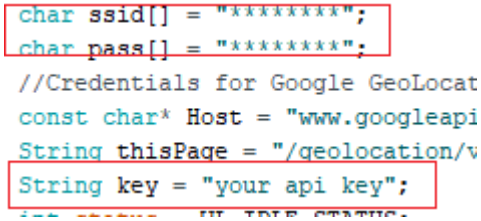
```
  Serial.println(accuracy);

}}
```

Procedure:

1. Now first thing you need to do get the google maps api from https://developers.google.com/maps/documentation/geolocation/get-api-key

2. After that open your code in Arduino IDE. Write the api key you got and the ssid of wifi and password.



3. Run the code and open up the serial monitor. In serial monitor you will see latitude and longitude. Copy and paste it in google maps and it will give you the location.