

Tecnicatura Superior en Telecomunicaciones

Materia: Práctica sensores y actuadores

Profesor: C. GONZALO VERA

Profesor: JORGE E. MORALES

Tema: Práctica Semana 4

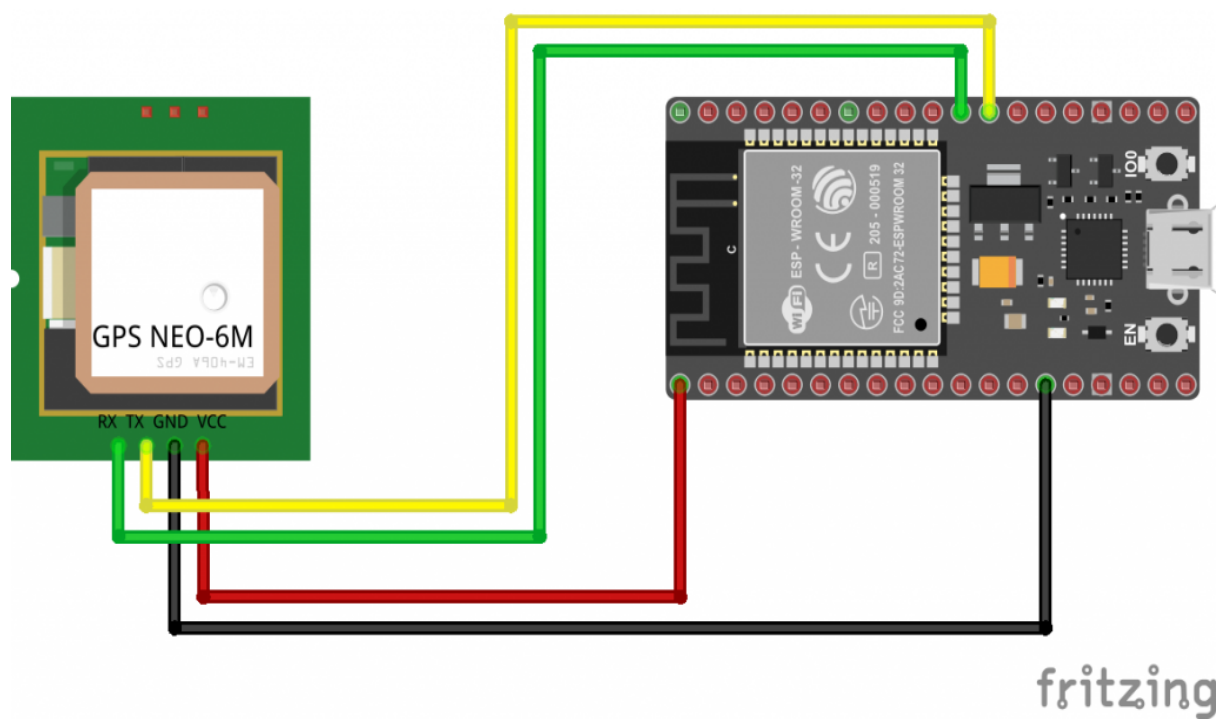
Ciclo lectivo: 2022

Alumnos : Grupo 6

- Guzmán, Lilén <https://github.com/lilenguzman01>
- López, Maximiliano <https://github.com/Maxilopez28>
- Moyano, Emilio <https://github.com/TerraWolf>
- Muguruza, Sergio <https://github.com/sergiomuguruza>
- Gonzalez, Mario <https://github.com/mariogonzalezispc>
- Ripoli, Enrique <https://github.com/enriqueripoli>

Ejercicio 1-f

Implementación de un GPS y conexión a esp32 por SoftwareSerial.



Ejemplo de código a usar

```
practica-de-la-4ta-semana-grupo-6 - Ejercicio-1fino

1  /*
2   * Materia actuadores y sensores
3   * Grupo 6
4   * Alumno : Mario Gonzalez
5   * Ejercicio 1-F
6   */
7
8  #include <WiFi.h>
9  #include <PubSubClient.h>
10 #include <SoftwareSerial.h>
11 #include <TinyGPS++.h>
12
13 const char *ssid = "TP-LINK_B33E";           // red de wifi a la que me conecto
14 const char *password = "50868155";          // password de la red de wifi
15 const char *mqtt_server = "mgalarmasserver1.ddns.net"; // dns del server mosquitto (MQTT)
16 unsigned int mqtt_port = 1883;              // socket port del server MQTT Mosquitto
17 const char *Topico = "/grupo6/gps/";        // topico para publicar los datos en el server
18 const unsigned int writeInterval = 25000;    // tiempo de demora para publicar
19 static const int RXPin = 16, TXPin = 17;     // pines del serial por soft
20
21 WiFiClient esp32_Client;                    // creacion de objeto wifi client
22 PubSubClient client(esp32_Client);          // creacion de objeto pubsubclient
23 TinyGPSPlus gps;                            // creacion de objeto del gps
24 SoftwareSerial ss(RXPin, TXPin);             // declaracion de los pines del software serial
25
26 void setup()
27 {
28   Serial.begin(9600);                        // puerto serial nativo 9600
29   ss.begin(9600);                            // puerto virtual serial 9600
30   WiFi.begin(ssid, password);               // conecto al wifi del lugar (micasa)
31
32   while (WiFi.status() != WL_CONNECTED)
33   {
34     // inicio conexion
35     delay(1500); // demora para volver a intentar la conexion
36   }
37   client.setServer(mqtt_server, mqtt_port); // estableco conexion al server mqtt
38   client.setCallback(callback);             // inicio el callback de server mqtt y espero datos
39 }
```

practica-de-la-4ta-semana-grupo-6 - Ejercicio-1fino

```
1  /*
2  * Materia actuadores y sensores
3  * Grupo 6
4  * Alumno : Mario Gonzalez
5  * Ejercicio 1-F
6  */
7
8  void loop()
9  {
10     if (!client.connected()) // si la conexion esta negada reconecto
11         reconnect();
12     client.loop();
13     while (ss.available() > 0) // verifico si llego datos por el serial por soft
14         if (gps.encode(ss.read())) // si llego leo el dato
15             displayInfo();
16 }
17
18 void displayInfo()
19 { // rutina de captura del dato por GPS
20     if (gps.location.isValid())
21     {
22         double latitude = (gps.location.lat()); // verifico que el dato sea valido del objeto gps
23         double longitude = (gps.location.lng()); // desdoble la info a cada variable y desprecio velocidad
24         char mqtt_payload[50] = ""; // desdoble la info a cada variable y desprecio velocidad
25         snprintf(mqtt_payload, 50, "m1=%1f;%1f", latitude, longitude); // genero el arreglo para alojar los datos
26         client.publish(Topico, mqtt_payload); // armo el payload a enviar
27         delay(writeInterval); // publico en el broker el dato del gps
28     } // espero este tiempo para hacer otra publicacion
29 }
30
31 void callback(char *topic, byte *payload, unsigned int length)
32 {
33     // aca va el codigo del callback para hacer operaciones remotas
34     // en este caso no se usa
35 }
36
37 void reconnect()
38 {
39     while (!client.connected())
40     {
41         delay(5000); // demora para reintentar conectarse
42     }
43 }
44 }
```

Implementación de API/REST para la visualización de los datos del gps en el mapa

Fuente: [tutorial GPS AWS](#)

Manos a la obra 6: API

1. Siguiendo los servicios enumerados en el diagrama, el siguiente paso será crear la API REST que traerá la información GPS al FrontEnd.

