

Arduino y ESP32
IOT

Primera parte WiFi y Node-Red

TOPICOS PREVIOS SENSORISTICA

ARDUINO Y TECNOLOGÍAS INALÁMBRICAS PARA IOT

Tutor: L. A. Ángeles-Hurtado

QUERÉTARO, ENERO 2024

0.1 Conexión por DHCP

- Paso 1: configurar el SSID y password
- Paso 2: config ip, default gateway (ip router) y mask
- la configuración depende si se hace por DHCP o statica
- los dispositivos de red caseros trabajan con la DHCP
- Los recursos para modo estación están en la clase WiFiSTAClass

El siguiente código se compiló en el IDE de Arduino. Para que en conjunto con Node-Red se comuniquen por medio del protocolo TCP y se puedan visualizar los datos en una página web.

```
1  #include <WiFi.h>
2
3  const uint16_t port = 8000; // 2^16 = 65,536 puertos disponibles para el servidor
4
5  WiFiServer server(port); // se crea un objeto WiFiServer que crea un servidor para
   escuchar conexiones entrantes del cliente
6
7  const char ssid[] = "ULTRON 7186"; // Nombre de la red WiFi
8  const char passphrase[] = "x:L78467"; // Contraseña de la red WiFi
9
10 void setup() {
11     Serial.begin(115200); // se inicia la comunicacion serial
12     while(!Serial){delay(25);}
13     set_up_WIFI();
14     server.begin(); // comienza a escuchar conexiones entrantes del cliente
15 }
16
17 void loop() {
18     // para escuchar las peticiones del cliente se crea un objeto de WiFiClient
19     WiFiClient cliente = server.available(); // detecta al cliente que en este ejemplo es
   Node-Red
20     if (cliente) {
21         Serial.println("Se detecto un cliente en la conexión");
22         while(cliente.connected()){
23             if(cliente.available()){
24                 char c = cliente.read();
25                 Serial.println(c);
26                 if(c == 'a'){
27                     cliente.print("El servidor recibio un byte del cliente por TCP");
28                 }
29             }
29         }
```

```
30     }
31 }
32 //-----
33 if(WiFi.status() == WL_DISCONNECTED){
34     Serial.println("Desconectado ");
35     delay(5000);
36 }
37 //-----
38 }
39
40 void set_up_WIFI(){
41
42     Serial.print("Estableciendo conexión WiFi con ");
43     Serial.print(ssid);
44
45     WiFi.begin(ssid, passphrase);
46
47     while(WiFi.status() != WL_CONNECTED){
48         Serial.print(".");
49         delay(50);
50     }
51
52     Serial.println("");
53     Serial.println("Conexión exitosa con ESP32!!!");
54     Serial.print("IP : ");
55     Serial.println(WiFi.localIP());
56     Serial.print("Mask : ");
57     Serial.println(WiFi.subnetMask());
58     Serial.print("GatewayIP : ");
59     Serial.println(WiFi.gatewayIP()); // tipo de red
60 }
```

0.2 Instalación de NODE-RED

Ingresa a la siguiente pagina: <https://nodered.org/docs/getting-started/windows> sigue los tres pasos para instalar node-red. Son más que suficientes para ejecutarlo en tu computadora. Para mayor información sobre el funcionamiento de node-red revisa en: <https://nodered.org/docs/user-guide/editor/>

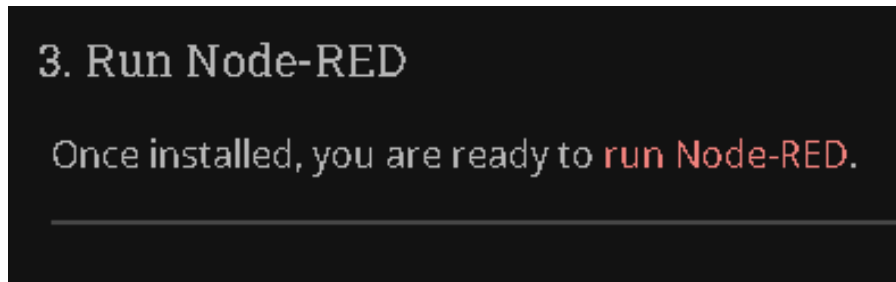


Figure 1: Al llegar al paso 3 puedes ejecutar mediante la terminal node-red.

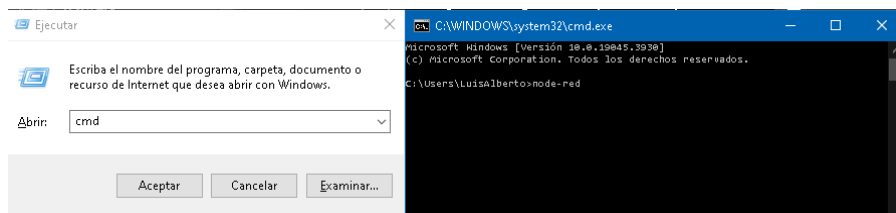


Figure 2: Abre símbolo de sistema y escribe cmd, después escribe node-red y da enter.

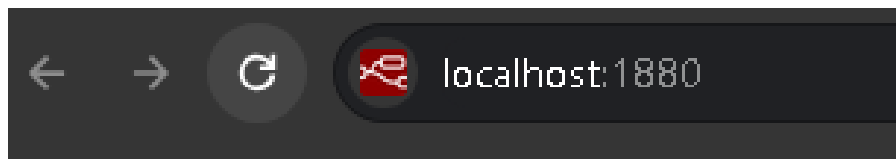


Figure 3: En el navegador Chrome o Firefox escribe localhost:1880

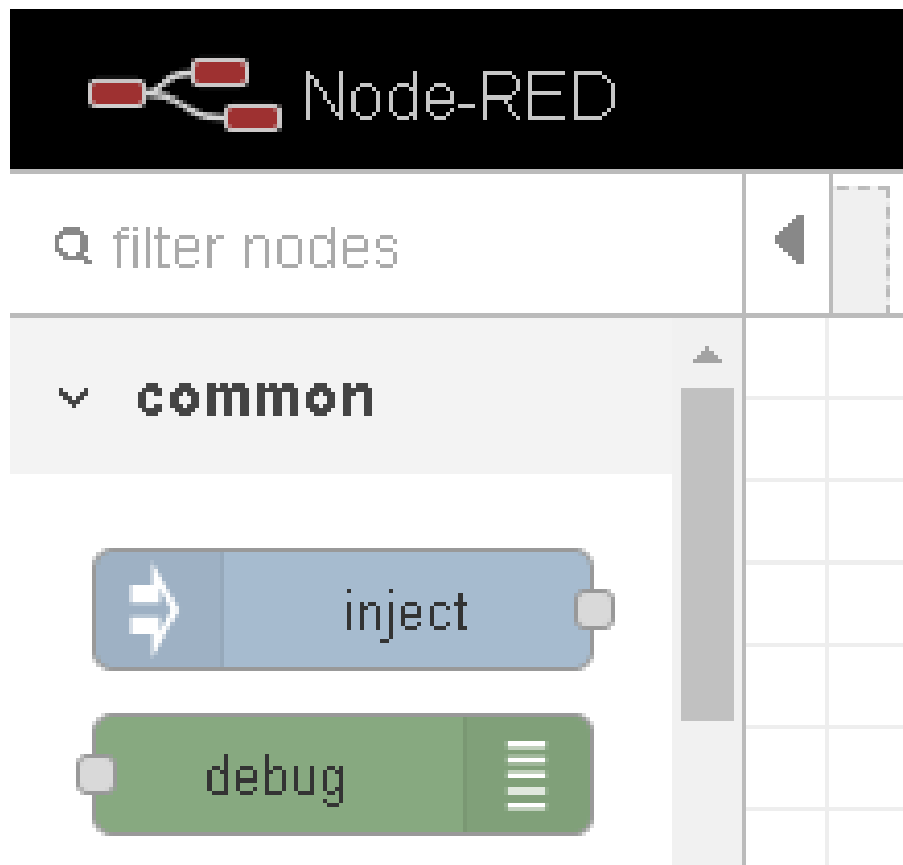


Figure 4: En la barra de nodos arrastra con el mouse dos nodos inject y un debug



Figure 5: Busca en el conjunto de nodos network y arrastra al área de trabajo el nodo tcp request

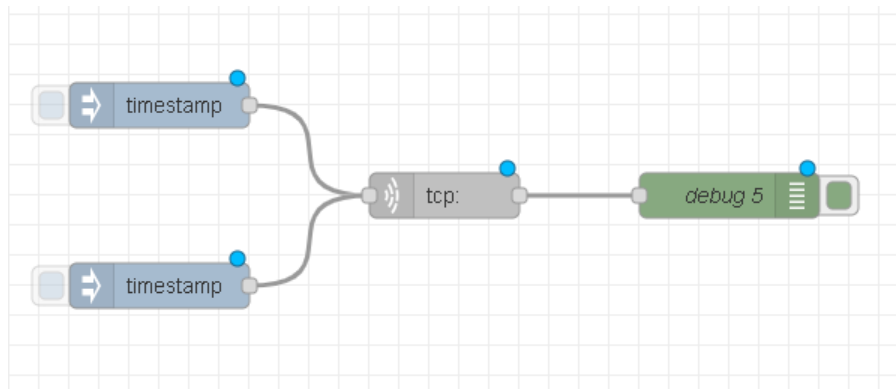


Figure 6: conecta la salida de los nodos y las entradas. La entrada de los nodos timestamp son botones que van a enviar un byte mediante el protocolo TCP

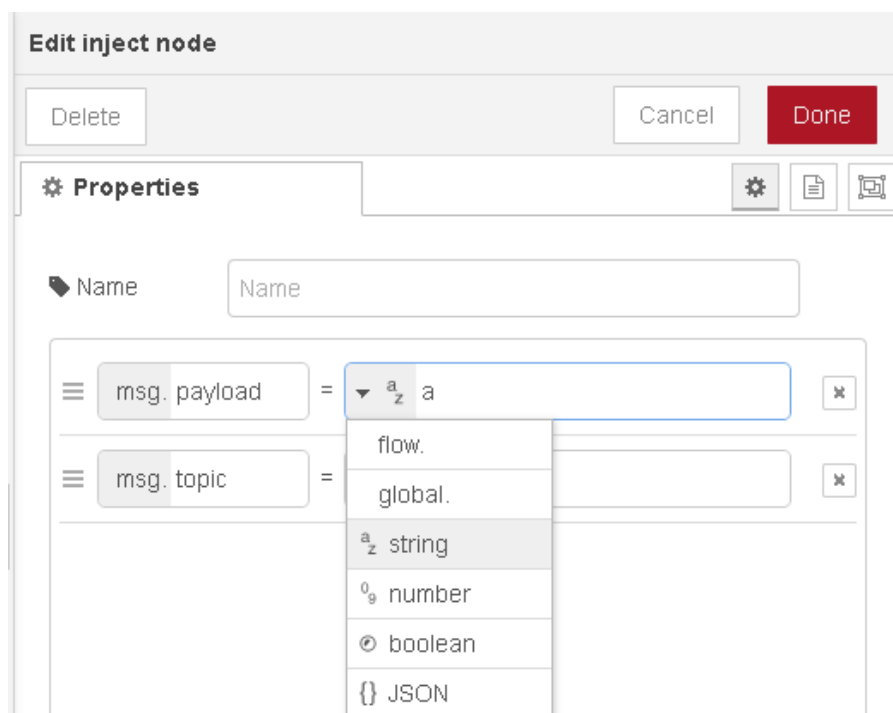


Figure 7: Da doble click en el primer nodo inject y cambia el dato a string. Realiza el mismo procedimiento pero al segundo agrega una b

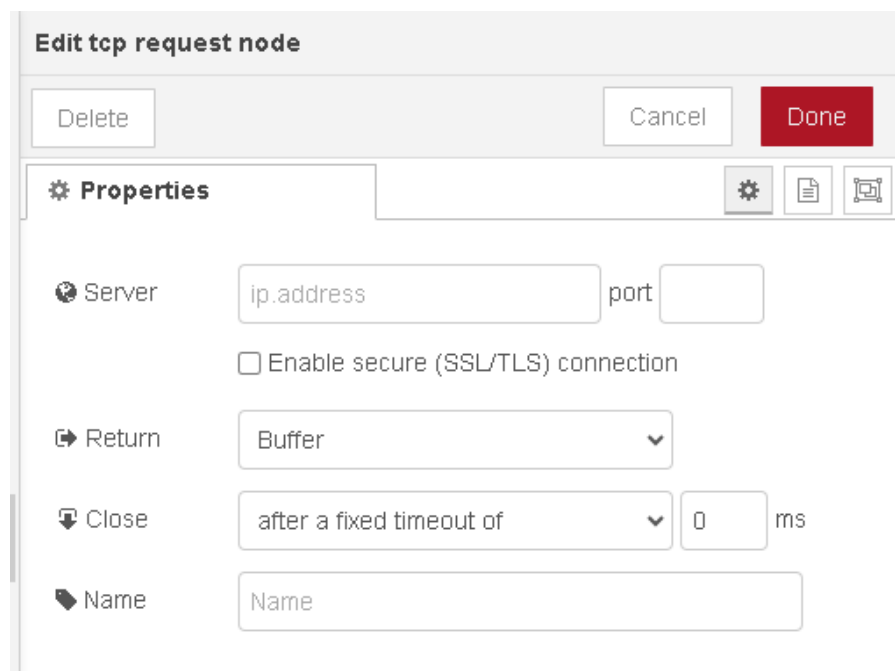


Figure 8: Da doble click en el nodo tcp request, para la comunicación vía wifi necesitamos la ip de la esp32 y el puerto por el que se realizara la transmisión vía protocolo TCP



Figure 9: Programa la esp32 con el código de arriba, y en el monitor serial del IDE de Arduino podrás ver la IP que tiene la esp32

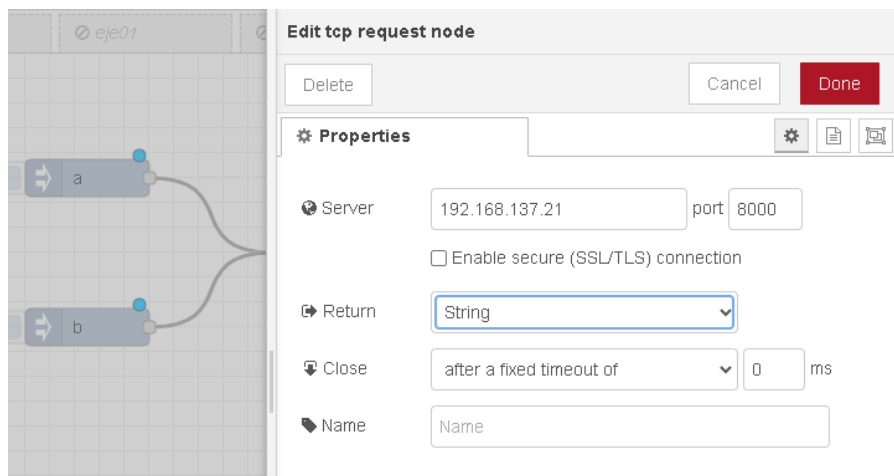


Figure 10: Una vez puesto la ip y el puerto se configura para regresar un byte

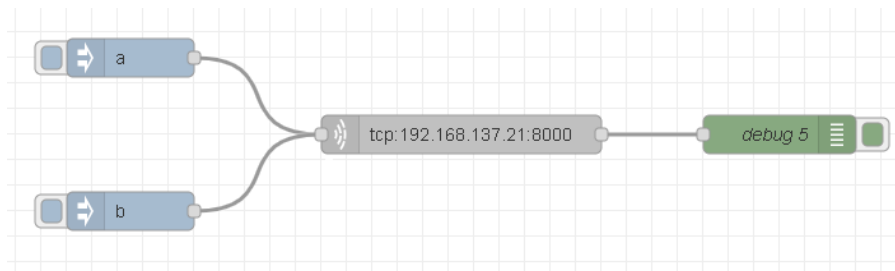


Figure 11: Si todo se realizó correctamente deberías ver algo parecido, el nodo debug no se le cambia nada

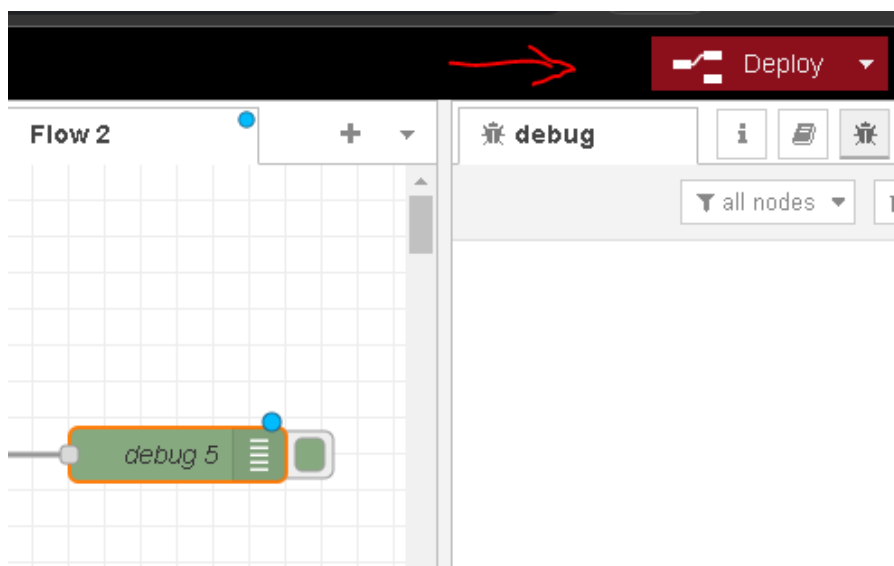


Figure 12: Para guardar los cambios da click en la boton Deploy

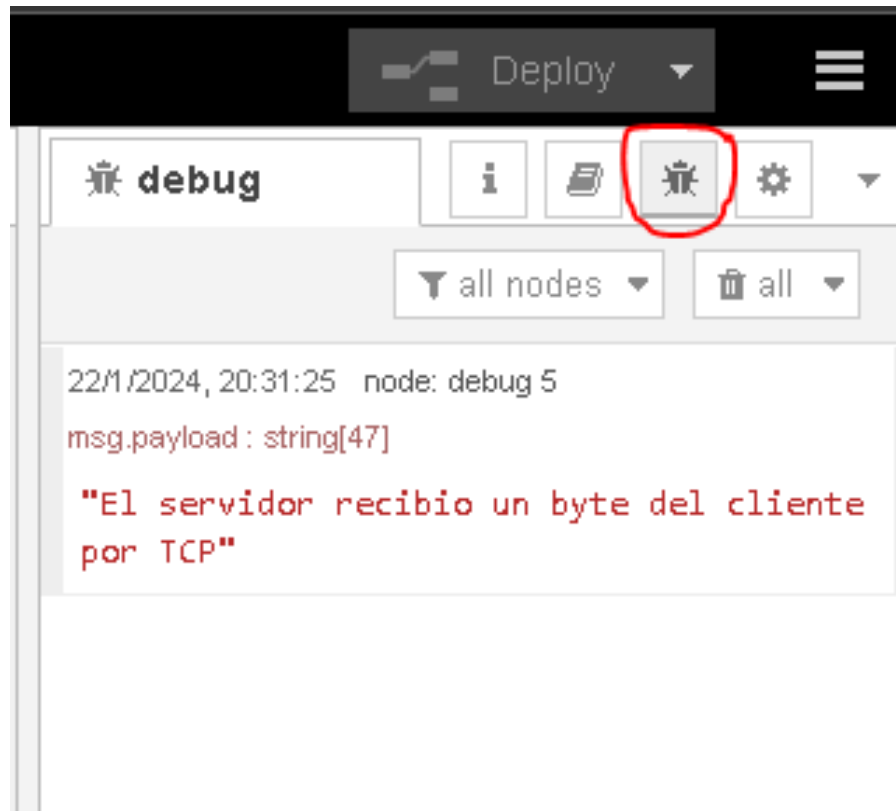


Figure 13: Al presionar el primer nodo inject con el string a, en la terminal debug podríamos visualizar que ya tenemos comunicación entre el servidor y el cliente mediante node-red y la esp32

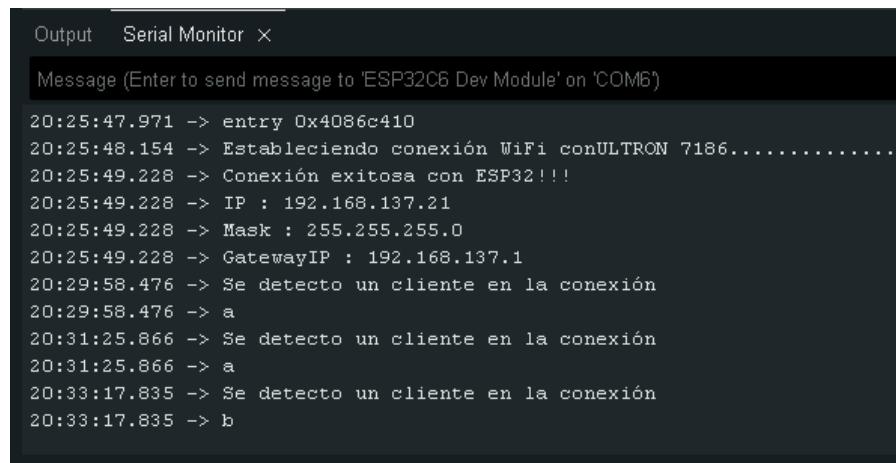


Figure 14: En el monitor serial del IDE arduino se puede ver el caracter a y b

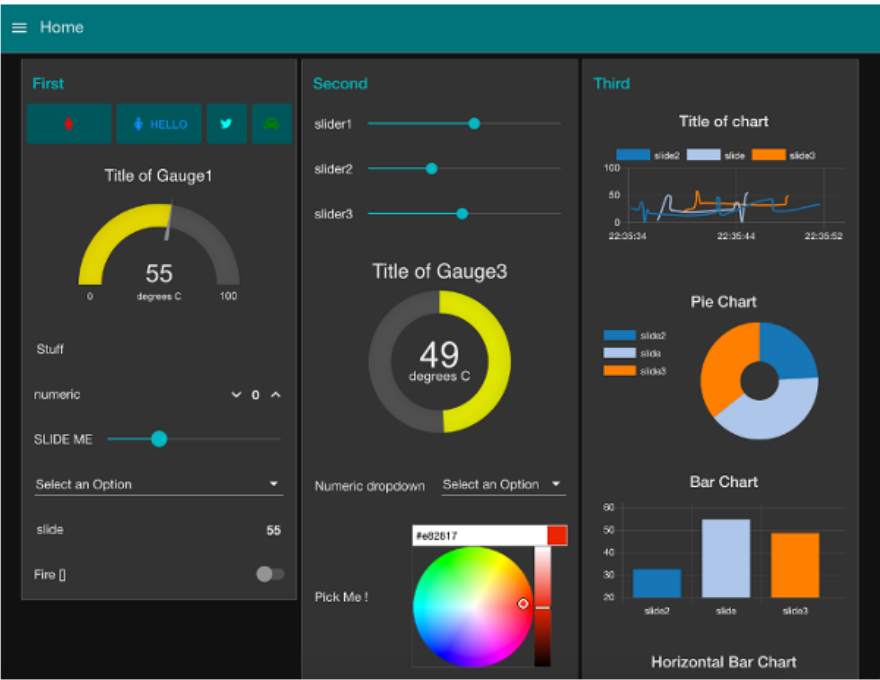


Figure 15: Para la segunda etapa se creara un panel para visualizar los datos obtenidos por la esp32 por el puerto serial y wifi