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 codigo se compilo 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 pagina web.

```
#include <WiFi.h>

const uint16_t port = 8000; // 2^16 = 65,536 puertos disponibles para el servidor

WiFiServer server(port);  // se crea un objeto WiFiServer que crea un servidor para escuchar conexiones entrantes del cliente

const char ssid[] = "ULTRON 7186";     // Nombre de la red WiFi
const char passphrase[] = "x:L78467"; // Contrasela de la red WiFi

void setup() {
  Serial.begin(115200); // se inicia la comunicacion serial
  while(!Serial){delay(25);}
  set_up_WIFI();
  server.begin(); // comienza a escuchar conexiones entrantes del cliente
}

void loop() {
  // para escuchar las peticiones del cliente se crea un objeto de WiFiClient
  WiFiClient cliente = server.available(); // detecta al cliente que en este ejemplo es Node-Red
  if (cliente) {
    Serial.println("Se detecto un cliente en la conexión");
    while(cliente.connected()){
      if(cliente.available()){
        char c = cliente.read();
        Serial.println(c);
        if(c == 'a'){
          cliente.print("El servidor recibio un byte del cliente por TCP");
        }
      }
```

```
    }
  }
  //---------------------------------------
  if(WiFi.status() == WL_DISCONNECTED){
    Serial.println("Desconectado ");
    delay(5000);
  }
  //---------------------------------------
}

void set_up_WIFI(){

  Serial.print("Estableciendo conexión WiFi con ");
  Serial.print(ssid);

  WiFi.begin(ssid, passphrase);

  while(WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(50);
  }

  Serial.println("");
  Serial.println("Conexión exitosa con ESP32!!!");
  Serial.print("IP : ");
  Serial.println(WiFi.localIP());
  Serial.print("Mask : ");
  Serial.println(WiFi.subnetMask());
  Serial.print("GatewayIP : ");
  Serial.println(WiFi.gatewayIP()); // tipo de red
}
```

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/

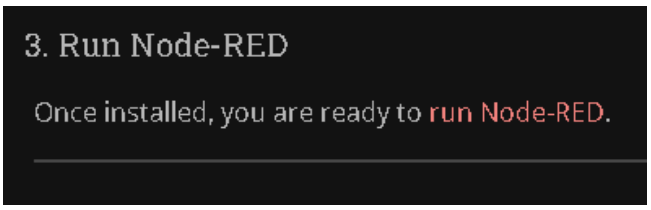

Figure 1: Al llegar al paso 3 puedes ejecutar mediante la termina node-red.

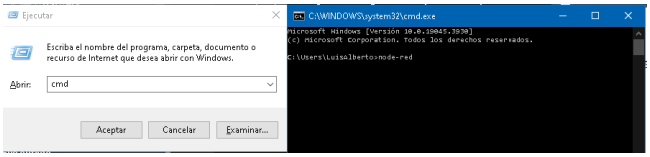

Figure 2: Abre símbolo de sistema y escribe cmd, despúes escribe node-red y da enter.

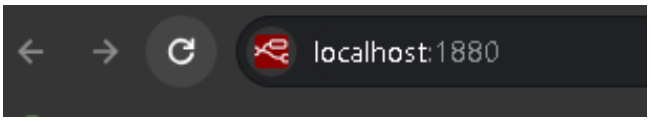

Figure 3: En el navegador Chrome o Firefox escribe localhost:1880

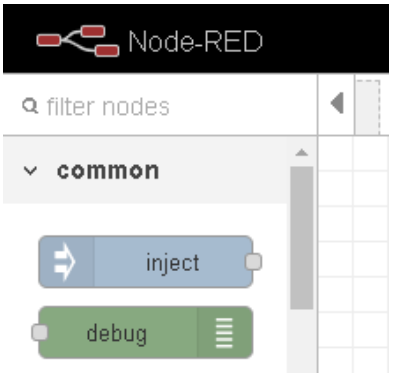

Figure 4: En la barra de nodos arrastra con el mouse dos nodos inject y un debug

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

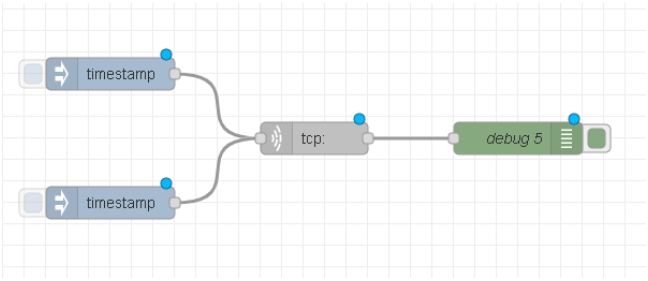

Figure 6: conecta la salida de los nodos y las entradas. La entrada de los nodos timestamp son botones que van a enviar un byte mediante el protocolo TCP

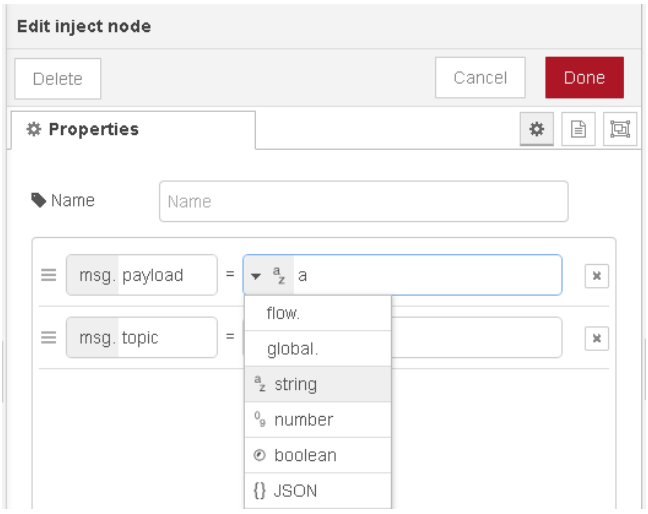

Figure 7: Da doble click en el primer nodo inject y cambia el dato a string. Realiza el mismo procedimiento pero al segundo agrega una b

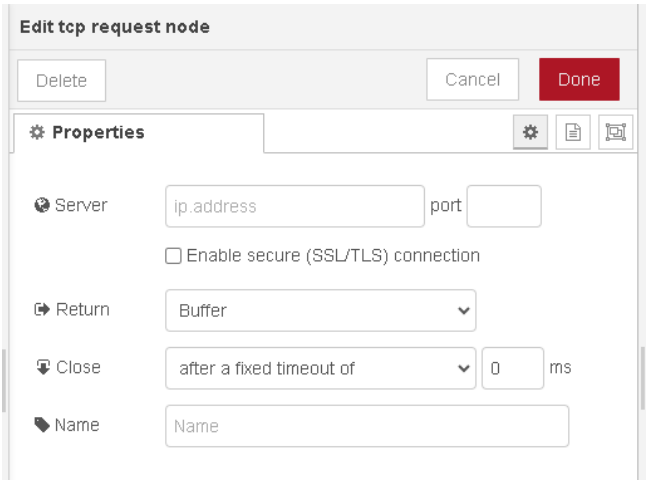

Figure 8: Da doble click en el nodo tcp request, para la comunicación vía wifi necesitamos la ip de la esp32 y el puerto por el que se realizara la transmisión vía protocolo TCP

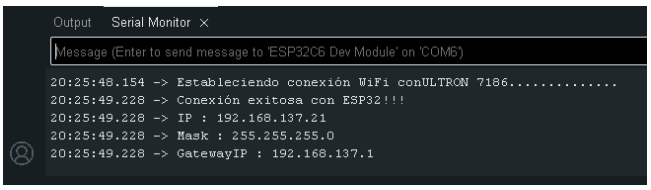

Figure 9: Programa la esp32 con el codigo de arriba, y en el monitor serial del IDE de arduino podras ver la ip que tiene la esp32

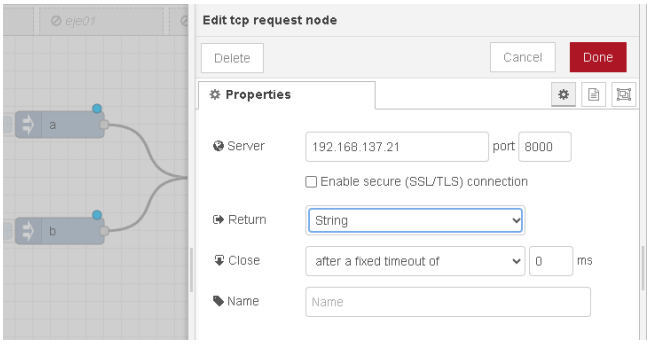

Figure 10: Una vez puesto la ip y el puerto se configura para regresar un byte

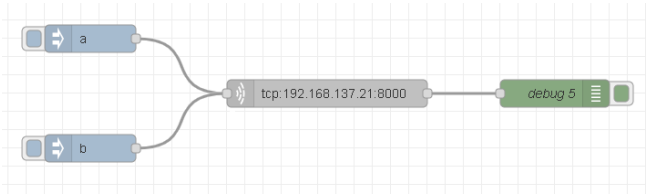

Figure 11: Si todo se realizo correctamente deberías ver algo parecido, el nodo debug no se le cambia nada

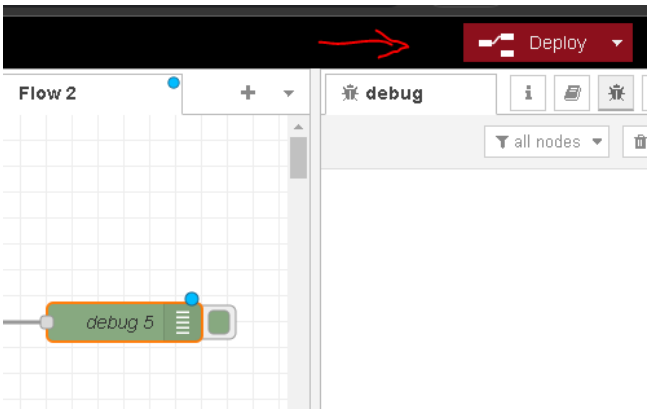

Figure 12: Para guardar los cambios da click en la boton Deploy

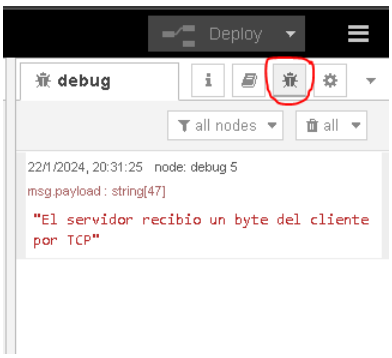

Figure 13: Al presionar el primer nodo inyect con el string a, en la terminal debug podríamos visualizar que ya tenemos comunicación entre el servidor y el cliente mediante node-red y la esp32

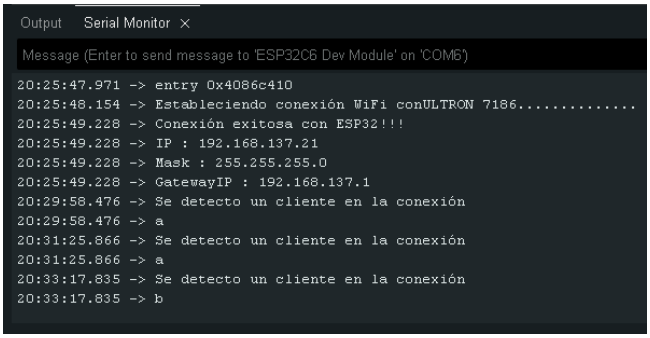

Figure 14: En el monitor serial del IDE arduino se puede ver el caracter a y b

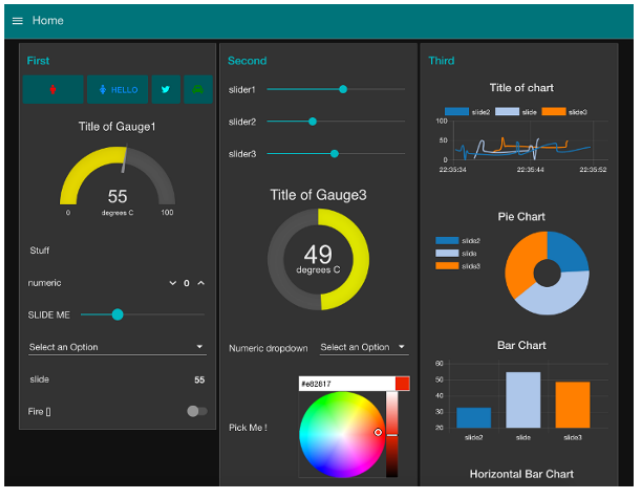

Figure 15: Para la segunda etapa se creara un panel para visualizar los datos obtenidos por la esp32 por el puerto serial y wifi

El siguiente codigo esta en formato json el cual se puede exportar en node-red y describe todos los pasos que se realizaron en las figuras de arriba.

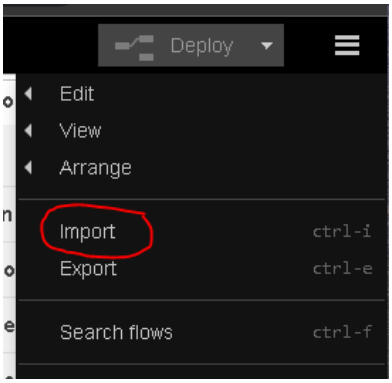

Figure 16: Selecciona importar en la nueva ventana selecciona en un nuevo flow y después busca el formato el archivo a importar en formato json

```
[
    {
        "id": "03da84c4eac89217",
        "type": "tab",
        "label": "Flow 2",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "546e4e1ed99b72cb",
        "type": "inject",
        "z": "03da84c4eac89217",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "a",
```

```
        "payloadType": "str",
        "x": 130,
        "y": 120,
        "wires": [
            [
                "2f2b07ef8dcb82c1"
            ]
        ]
    },
    {
        "id": "c1c7bfa79a07038a",
        "type": "inject",
        "z": "03da84c4eac89217",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "b",
        "payloadType": "str",
        "x": 130,
        "y": 180,
        "wires": [
            [
                "2f2b07ef8dcb82c1"
            ]
        ]
    },
    {
        "id": "b2abfe25e6f320f4",
        "type": "debug",
        "z": "03da84c4eac89217",
```

```
        "name": "debug 5",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 720,
        "y": 160,
        "wires": []
    },
    {
        "id": "2f2b07ef8dcb82c1",
        "type": "tcp request",
        "z": "03da84c4eac89217",
        "name": "",
        "server": "192.168.137.21",
        "port": "8000",
        "out": "time",
        "ret": "string",
        "splitc": "0",
        "newline": "",
        "trim": false,
        "tls": "",
        "x": 450,
        "y": 160,
        "wires": [
            [
                "b2abfe25e6f320f4"
            ]
        ]
    }
]
```