Internet Of Things

Title: WebSocket Bi-Directional communication | ESP32

Client → ESP32: Control LED
ESP32 → Client: Send sensor data (DHT11 temperature)

Objective

- Send commands from browser to ESP32 (toggle LED).
- Receive real-time sensor data (like temperature) from ESP32 and display it in the browser.
- Use a minimal HTML interface.

1. Requirements

- Hardware: ESP32, DHT11 sensor, LED, 330Ω resistor, jumper wires
- Libraries:
  - WiFi.h
  - WebSocketsServer.h
  - DHT.h

Flow Summary

```
Browser        <-- WebSocket -->        ESP32
 [Button]                          [LED Control]
 [Sensor Display]   <-- data push --   [Read sensor every 2s]
```

ESP32 Code with Explanation

Let’s build it line-by-line:

```
#include <WiFi.h>
#include <WebServer.h>
#include <WebSocketsServer.h>
```

```
#include <DHT.h>
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

WebServer server(80);
WebSocketsServer webSocket = WebSocketsServer(81); // Port 81 for WS

const int ledPin = 2;
```

HTML + JavaScript in ESP32

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
  <head><meta charset="utf-8"><title>ESP32 WebSocket</title></head>
  <body>
    <h2>LED Control</h2>
    <button onclick="send('ON')">ON</button>
    <button onclick="send('OFF')">OFF</button>

    <h3>Temperature: <span id="temp">--</span> °C</h3>

    <script>
      var ws = new WebSocket("ws://" + location.hostname + ":81/");
      ws.onmessage = function(evt) {
        document.getElementById("temp").innerText = evt.data;
      }
      function send(val) {
        ws.send(val);
      }
    </script>
  </body>
</html>
)rawliteral";
```

Setup Code

```
void setup() {
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  dht.begin();

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("WiFi connected: " + WiFi.localIP().toString());

  server.on("/", []() {
    server.send_P(200, "text/html", index_html);
  });
  server.begin();

  webSocket.begin();
  webSocket.onEvent(webSocketEvent);
}
```

WebSocket Event Handler

```
void webSocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t length)
{
  if (type == WStype_TEXT) {
    String cmd = (const char *)payload;
    if (cmd == "ON") digitalWrite(ledPin, HIGH);
    if (cmd == "OFF") digitalWrite(ledPin, LOW);}}
```

Main Loop

```
unsigned long lastTime = 0;

void loop() {
  server.handleClient();
  webSocket.loop();

  if (millis() - lastTime > 2000) {
    lastTime = millis();
    float temp = dht.readTemperature();
    if (!isnan(temp)) {
      String tempStr = String(temp);
      webSocket.broadcastTXT(tempStr);
    }
  }
}
```

Test

1. Upload to ESP32.
2. Connect to same Wi-Fi network.
3. Visit `http://<esp-ip>/`
4. Click buttons to toggle LED.
5. View real-time temperature updates.

Result

- Client sends `on` → LED turns ON → ESP32 replies "LED ON"
- ESP32 sends temperature data to browser every 2 seconds
