**ESP-32 BLE Configuration**

**Overview**

“Low Energy Bluetooth Services Broadcasting “.

This project focuses on "Low Energy Bluetooth Services Broadcasting." We simulate some hardware sensors that captures temperature and humidity with some variables and created a personalized ESP32 server. The server broadcasts weather-like data, providing clients with the temperature and humidity details from the environment.

**Installation Instructions**

1. **Setting Up Arduino IDE**
   * Download and install the Arduino IDE .
   * Open Arduino IDE and configure settings as necessary for the esp32(refer below for esp32 configuration).
2. **Installing Libraries**
   * Use the Arduino Library Manager to install the following libraries:
     + BLEDevice
     + BLEServer
     + BLE2902
     + BLE2901
3. **Including Libraries**
   * Include all the libraries from sketch🡪Include library🡪Ble(under esp32 section).

**Configuration**

 **Include Libraries:**

* The code starts by including necessary libraries for Bluetooth Low Energy (BLE) functionality:

#include <BLEDevice.h>

#include <BLEServer.h>

#include <BLE2902.h>

#include <BLE2901.h>

 **Define UUIDs:**

* UUIDs (Universally Unique Identifiers) uniquely identify services.

#define Service\_id "00000002-0000-0000-FDFD-FDFDFDFDFDFD"

#define Temp\_Char\_id "1efd327d-7caa-4dad-90d8-d2b87dfc4882"

#define Humidity\_Char\_id "18ad1902-ac57-488c-b5fe-c6c8b4041860"

 **Global Variables:**

* Initialize global variables to simulate sensor data (temperature and humidity) and Creating necessary pointers for characteristics and advertising.

BLECharacteristic \*pTemp;

BLECharacteristic \*pHumidity;

BLEAdvertising \*padvertising;

int Temp = 0;

int Humidity = 0;

 **Setup Function ():**

* Initialization part is here when the device boots up:

void setup() {

Serial.begin(115200);

Serial.println("BLE SERVICE BEGIN");

pinMode(2, OUTPUT);

BLEDevice::init("ESP32 SERVICE");

* + Serial.begin(115200): Initialize serial communication for debugging.
  + Serial.println("BLE SERVICE BEGIN"): Print message to serial monitor.
  + pinMode(2, OUTPUT): Set pin 2 as an output (used as an indicator).
  + BLEDevice::init("ESP32 SERVICE"): Initialize BLE with a device name.

 **Create BLE Server and Service:**

* Establish a BLE server and create a custom service similar to the creation of pointers for the chracteristics and advertising:

BLEServer \*pserver = BLEDevice::createServer();

BLEService \*pservice = pserver->createService(Service\_id);

 **Create BLE Characteristics:**

* Define and configure BLE characteristics (temperature and humidity) with read and notify operations as mentioned:

pTemp = pservice->createCharacteristic(Temp\_Char\_id,

BLECharacteristic::PROPERTY\_READ |

BLECharacteristic::PROPERTY\_NOTIFY);

pHumidity = pservice->createCharacteristic(Humidity\_Char\_id,

BLECharacteristic::PROPERTY\_READ |

BLECharacteristic::PROPERTY\_NOTIFY);

 **Set Initial Values and Descriptors:**

* Initialize characteristic values and add descriptors:

pTemp->setValue((uint8\_t\*)&Temp, sizeof(float));

pHumidity->setValue((uint8\_t\*)&Humidity, sizeof(float));

pTemp->addDescriptor(new BLE2902());

pTemp->addDescriptor(new BLE2901());

pHumidity->addDescriptor(new BLE2902());

pHumidity->addDescriptor(new BLE2901());

* + setValue(): Set initial sensor data values.
  + addDescriptor(): Add descriptors for client and user descriptions.

 **Start BLE Service and Advertising:**

* Start the BLE service and advertising:

pservice->start();

padvertising = BLEDevice::getAdvertising();

padvertising->addServiceUUID(Service\_id);

padvertising->start();

* + pservice->start(): Start the BLE service.
  + padvertising->start(): Start BLE advertising with the specified service UUID.

 **Indicate Setup Completion:**

* Use an LED or indicator to signal that setup is complete:

digitalWrite(2, HIGH);

 **Loop Function (loop()):**

* Continuously update sensor data and notify clients:

void loop() {

pTemp->setValue(Temp);

pTemp->notify();

Temp++;

delay(1000);

pHumidity->setValue(Humidity);

pHumidity->notify();

Humidity++;

delay(1000);

padvertising->start();

}

**Screenshots**









