**Low Energy Bluetooth Service Broadcasting using ESP32 Devkit V1**

**By Nirvan Tamhane**

**Table of Contents**

1. **Task Details**
2. **Hardware Components Used**
3. **Tools Used**
4. **Microcontroller Interfacing**
5. **Schematic**
6. **Complete Hardware Assembly**
7. **Firmware**
8. **Project Overview Video**
9. **Issues Faced and Resolution**
10. **Outputs**
11. **nRF Connect Application Interface**
12. **References**

**Task Details**

**Services to be Broadcasted:**

* Temperature Measurement
* Humidity

**Details about the Service to Broadcast Over Bluetooth:**

* **Service UUID:** 00000002-0000-0000-FDFD-FDFDFDFDFDFD
* **Characteristics:**
  + Temperature Measurement: Standard BLE characteristic - GATT Char UUID: 0x2A1C
  + Humidity: Standard BLE characteristic - GATT Char UUID: 0x2A6F
* **Supported Operations:** Both characteristics support read and notify.

**Hardware Components Used**

1. Espressif System's ESP32 Devkit V1 Development Board
2. DHT11 Temperature and Humidity Sensor
3. 10K Ohm Resistor (Pull-up)
4. USB to MicroUSB Cable
5. Breadboard
6. Jumpers

**Tools Used**

**Arduino IDE**

Arduino Integrated Development Environment or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

* For more details, refer to: [Arduino IDE](https://www.arduino.cc/en/software)

**Fritzing**

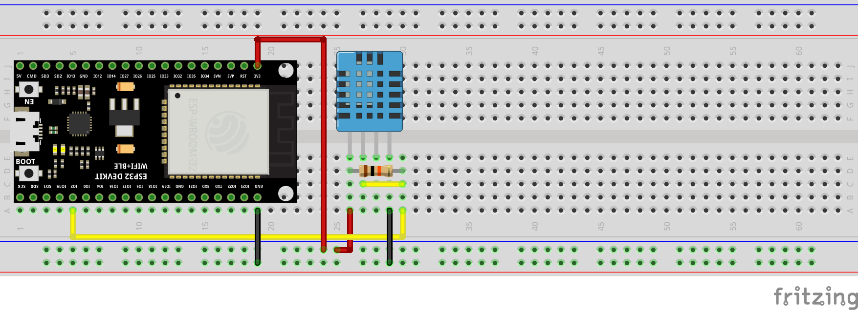
Fritzing is an open-source hardware initiative that makes electronics accessible as creative material for anyone. We offer a software tool, a community website, and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional PCBs.

* For more details, refer to: [Fritzing](https://fritzing.org/)

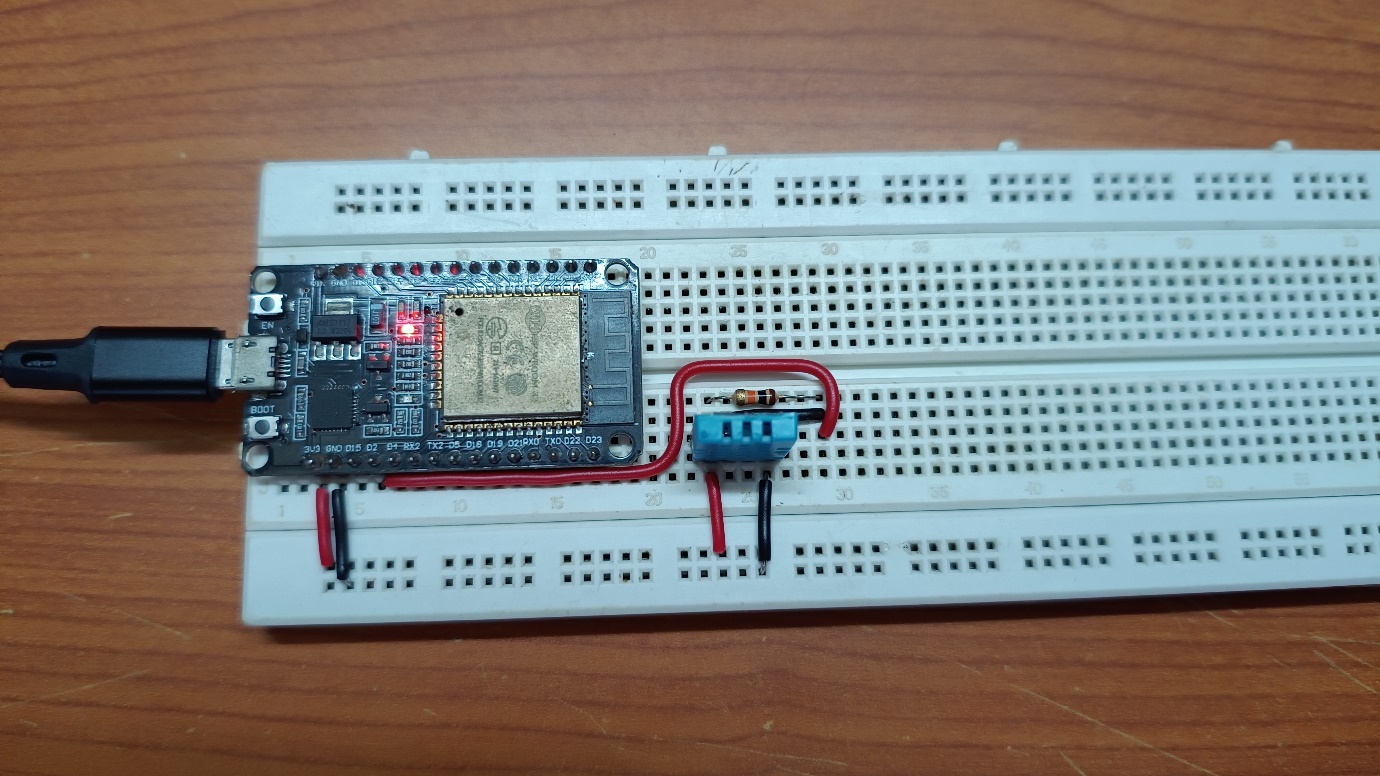
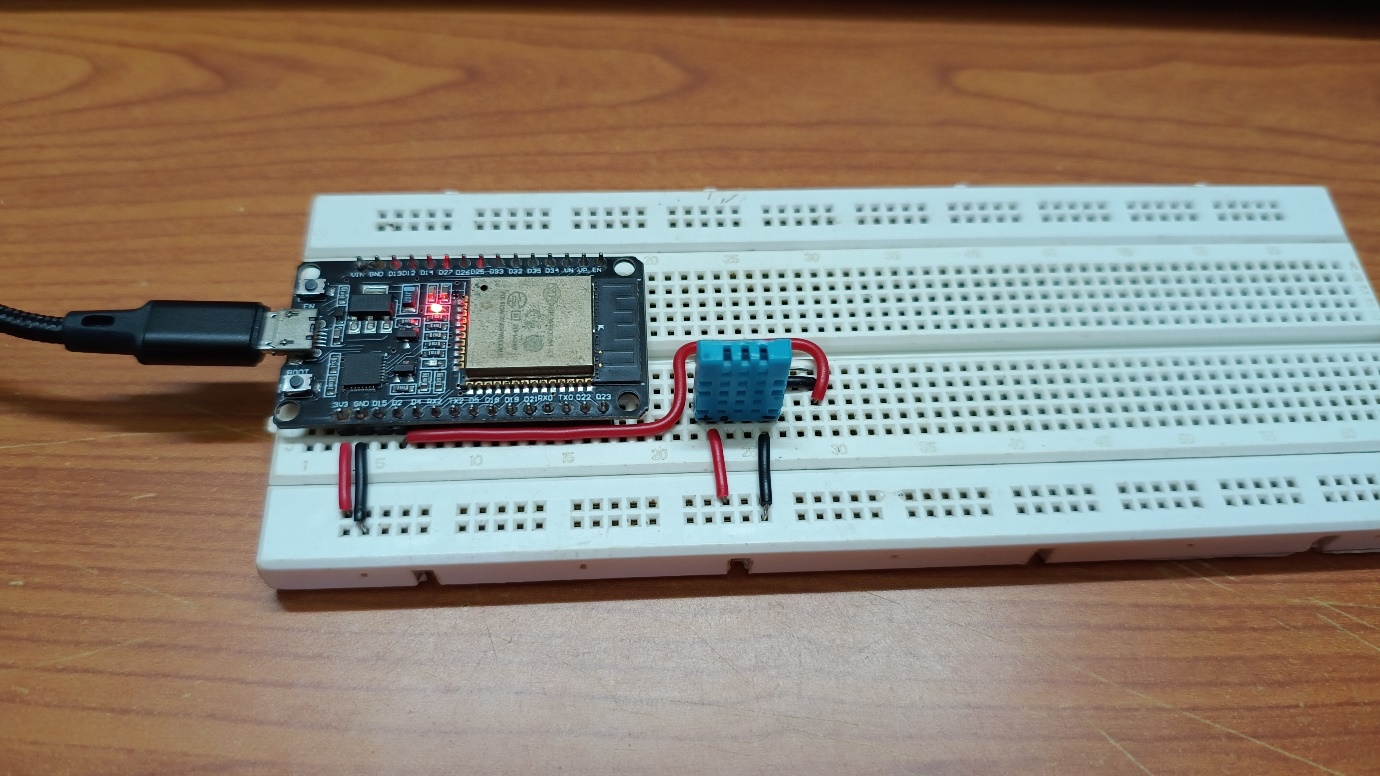
**Microcontroller Interfacing**

* Used OneWire protocol to interface the microcontroller to the DHT11 sensor.
* Used BLE protocol for Connectivity with the nRF Connect Mobile Application.

**Schematic**



**Complete Hardware Assembly**

****

**Firmware**

**Link:** [**Firmware .ino File**](https://github.com/Nirvan007/ESP32_BLE/blob/main/Firmware/Nineti_Assignment_ESP32_BLE/Nineti_Assignment_ESP32_BLE.ino)

**Project Overview Video**

**Link:** [**Project Overview Video**](https://drive.google.com/file/d/1l5og78Y-2ZODacyJ4YPPP19z49ZOXAzD/view?usp=sharing)

**Issues Faced and Resolution**

**ESP32 Not Advertising After Disconnection**

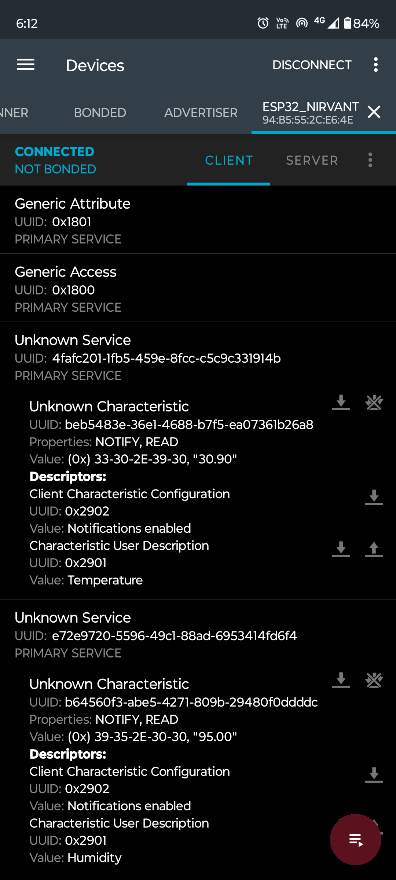
**Approach 1:**

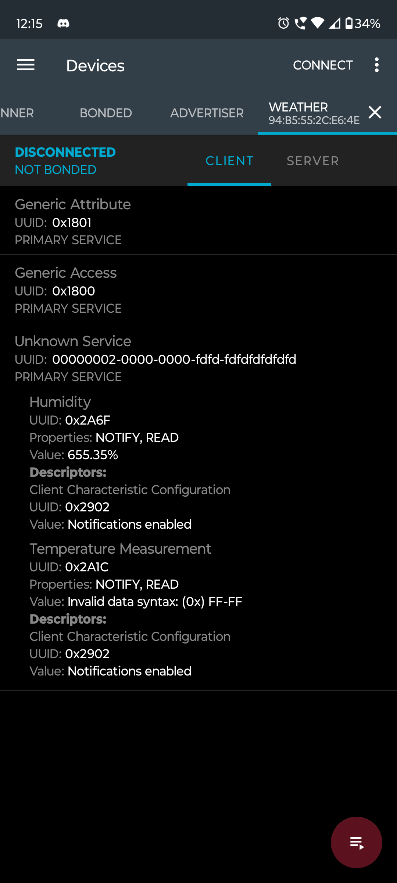
* **Used the callback function to restart advertising within the loop itself.**
* **Result: Successfully connected and reconnected multiple times with no issues (Check at the end of the** [**Project Overview Video**](https://drive.google.com/file/d/1l5og78Y-2ZODacyJ4YPPP19z49ZOXAzD/view?usp=sharing)**).**

**BLE Data Transmission Problem**

**Approach 1:**

* **Started with using custom characteristic UUIDs and tried to send the data as String values (I had worked on this before).**
* **Result: Successfully received both values Temperature Measurement & Humidity in Strings.**
* **Comment: This was not asked in the task. It was mandatory to use the default GATT UUIDs.**



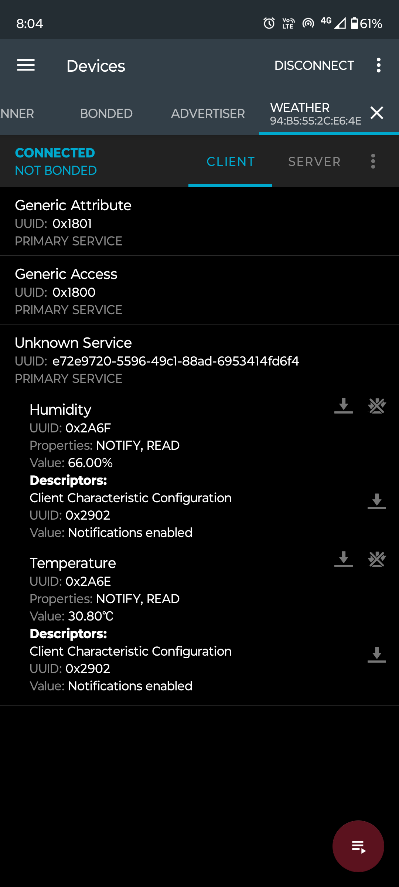


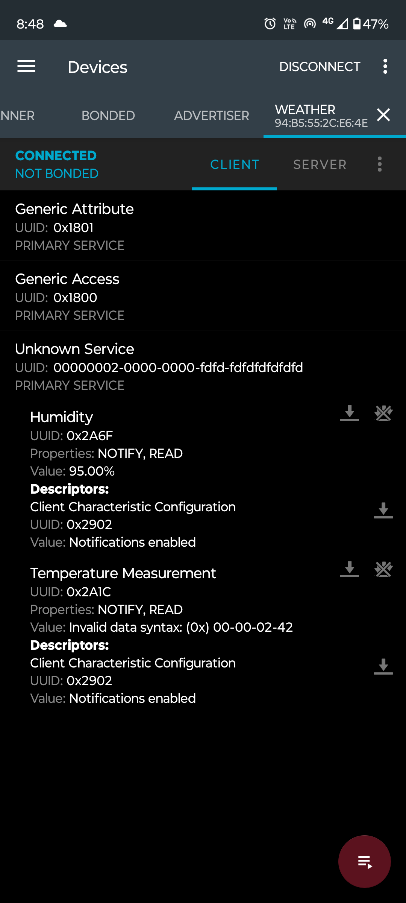
**Approach 2:**

* **Set the UUIDs to the required UUIDs (Temperature Measurement = 0x2A1C; Humidity = 0x2A6F) and tried to send the data as String values.**
* **Result: Error: Invalid Data Syntax @ nRF Application for Temperature Measurement value & Garbage value for Humidity value.**
* **Comment: This was not asked in the task. Getting the Humidity in % and Temperature Measurement in Celsius was mandatory.**

**Approach 3:**

* **Changed the Temperature characteristic UUID to default temperature (Temperature = 0x2AE6) and tried to send the data as unsigned 16-bit int.**
* **Result: Successfully received both values Temperature Measurement & Humidity in unsigned 16-bit int values.**
* **Comment: Checked the BLE documentation for the UUIDs (Temperature = 0x2AE6; Humidity = 0x2A6F) and their required data syntax type required.**



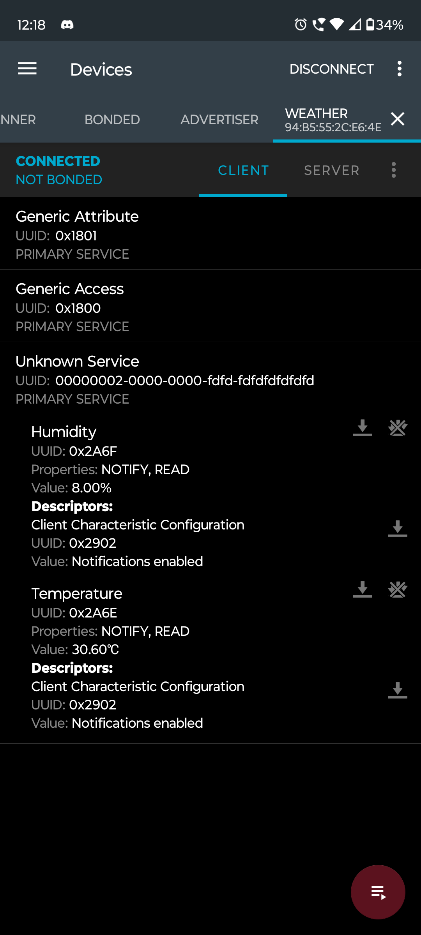


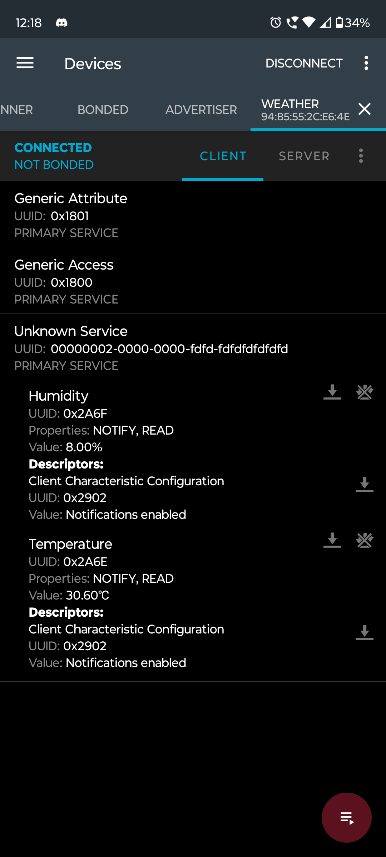
**Approach 4:**

* **Reset the Temperature characteristic UUID (Temperature Measurement = 0x2A1C) and tried to send the data as unsigned 16-bit int.**
* **Result: Error: Invalid Data Syntax @ nRF Application for Temperature Measurement value, but successfully received values for Humidity in unsigned 16-bit int values.**
* **Comment: Checked the BLE documentation for the Temperature Measurement UUID (Temperature Measurement = 0x2A1C) and their required data syntax type required.**

**Approach 5:**

* **Tried to convert the temperature data to 32-bit float and send the data along with the humidity data as unsigned 16-bit int.**
* **Result: Successfully received both values Temperature Measurement & Humidity in unsigned 16-bit int values, but Temperature Measurement unit was in Fahrenheit scale.**
* **Comment: Checked the BLE documentation for the Temperature Measurement UUID (Temperature Measurement = 0x2A1C) and their required data syntax type required and flags.**
* **Comment: Checked the conversion requires an IEEE11073 32-bit float data stream with 5 bytes (1 byte for C/F and 4 bytes for the actual value).**



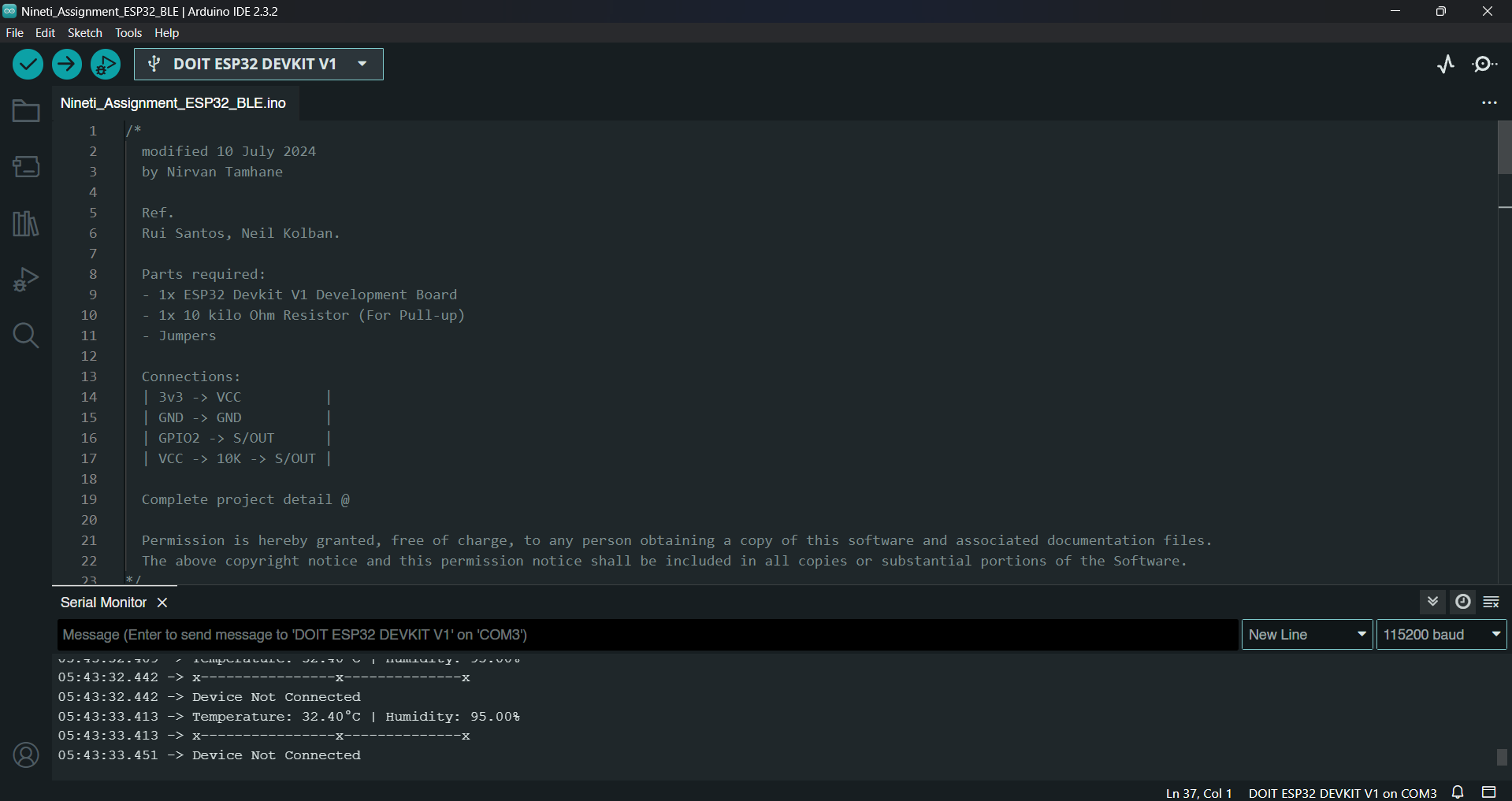


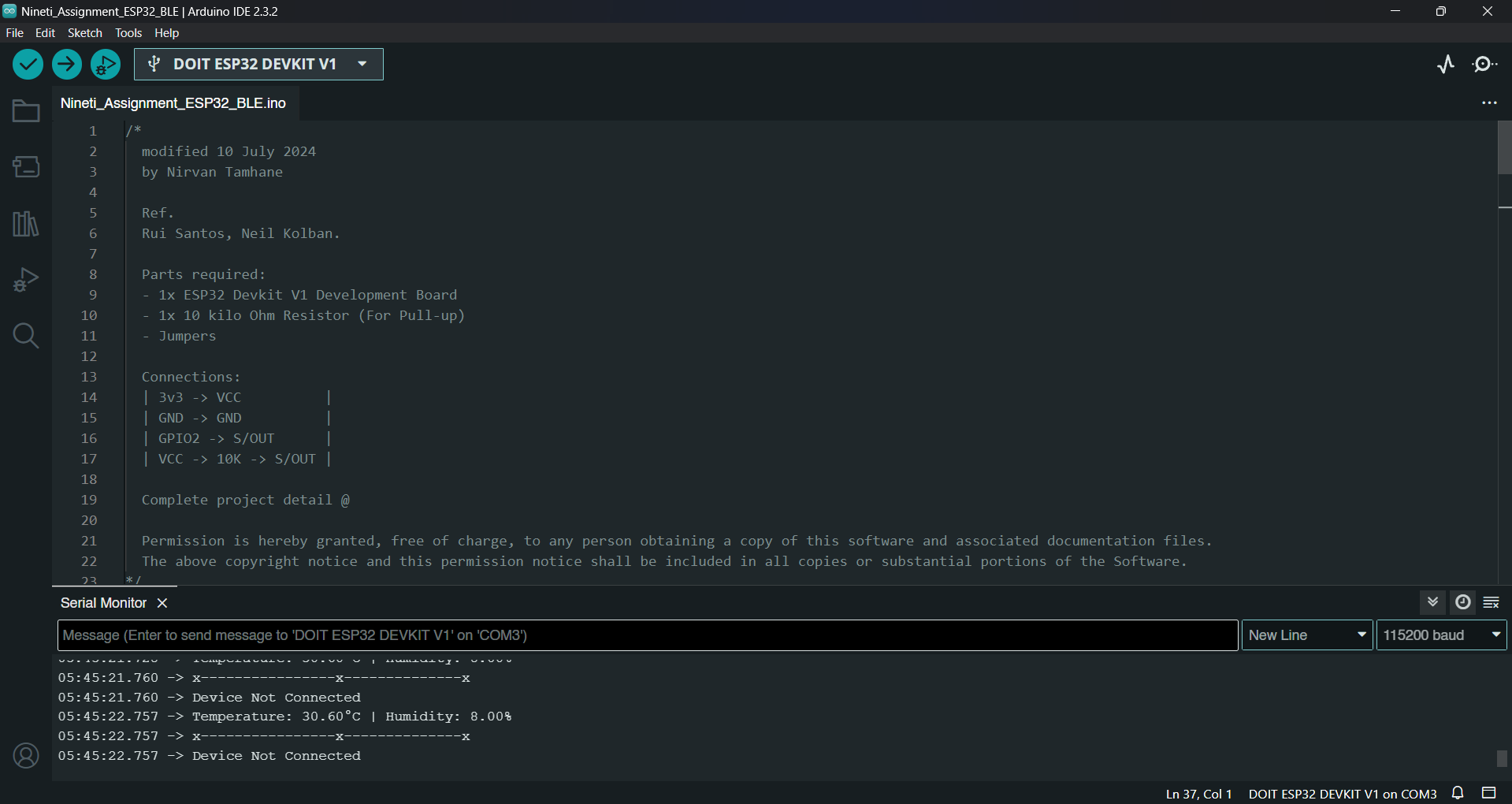
**Approach 6:**

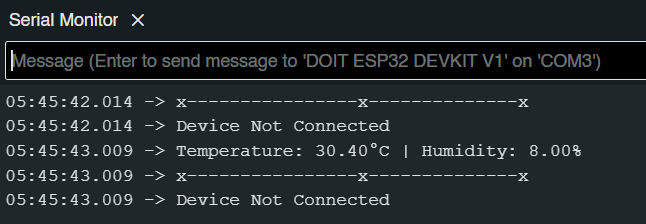
* **Changed the IEEE11073 32-bit float data stream with 5 bytes (1 byte for C/F and 4 bytes for the actual value) and set it to 0x00 flag for Celsius unit scale.**
* **Result: Successfully received both values Temperature Measurement & Humidity values on the nRF Connect Application with correct unit scales.**
* **Comment: Checked the BLE documentation for the Temperature Measurement UUID (Temperature Measurement = 0x2A1C) and their required data syntax type required and flags.**

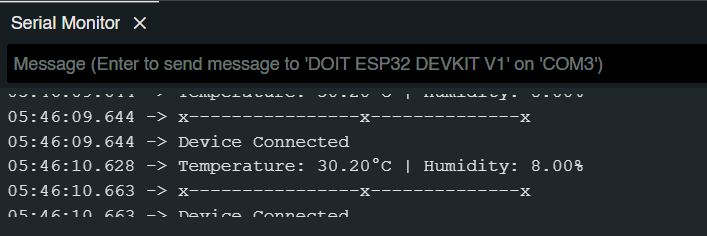
**Outputs**

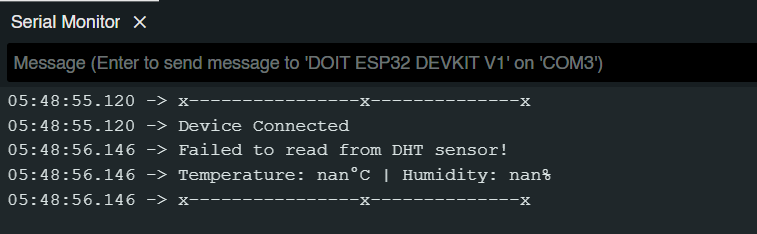
**Serial Monitor Messages**

****

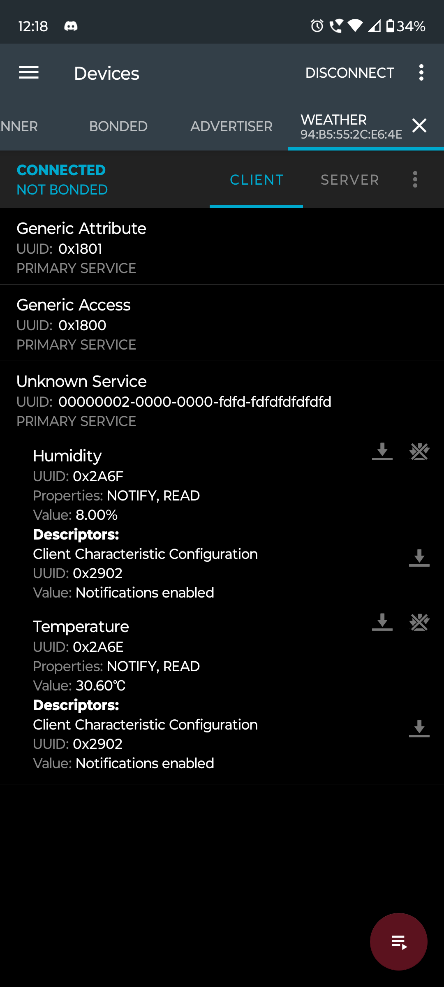
****

****

****

****

**nRF Connect Application interface**

****

**References**

1. [**ESP32 BLE Arduino**](https://github.com/nkolban/ESP32_BLE_Arduino)
2. [**RuiSantosdotme**](https://github.com/RuiSantosdotme)
3. [**Random Nerd Tutorials: ESP32 BLE Server Environmental Sensing Service**](https://randomnerdtutorials.com/esp32-ble-server-environmental-sensing-service/)
4. [**Makerhero: ESP32 BLE DHT11**](https://github.com/makerhero/esp32-ble-dht11)
5. [**ATC MiThermometer Issues**](https://github.com/atc1441/ATC_MiThermometer/issues/150)
6. [**Converting Two Bytes to IEEE 11073 16-bit sfloat in C#**](https://stackoverflow.com/questions/28899195/converting-two-bytes-to-an-ieee-11073-16-bit-sfloat-in-c-sharp/28901867#28901867)
7. [**Dart: Convert IEEE 11073 32-bit Float to Simple Double**](https://stackoverflow.com/questions/60841331/dart-convert-ieee-11073-32-bit-float-to-a-simple-double/71732994#71732994)
8. [**Bosch IoT Suite Bluetooth LE Driver API**](https://docs.bosch-iot-suite.com/edge/edge-services/API/modules/com.prosyst.mbs.bluetooth.le.driver.api/allclasses-noframe.html)
9. [**GATT XML: Temperature Measurement**](https://github.com/oesmith/gatt-xml/blob/master/org.bluetooth.characteristic.temperature_measurement.xml)

**…**