Challenge Bluetooth LE

Semester 4 IOT

Bluetooth LE on ESP32

March 19, 2023.

Student: Andre Sanao

Course: Smart Industry

Table of Contents

Acronyms

Acronym	*Meaning*
IOT	→ Internet of Things
BLE	→ Bluetooth Low Energy
ESP32	→ Expressif32
UUID	→ Universally Unique IDentifier
GATT	→ Generic ATTribute Profile

Table 1 – List of acronyms used throughout the report

Introduction

The assignment on which this document presents a small challenge of IOT subject. In this subject, we will learn how to use hardware to connect to the internet using internet protocols and such. The hardware that is going to be used to demonstrate these protocols is an ESP32. The ESP32 is a microcontroller that is used in embedded systems with an inbuilt wireless connectivity. In the following sections will provide the procedure and conclusion of the assignment.

Procedure

In this challenge, we will be using the Bluetooth LE library on the ESP32 and upload some basic example sketch. There are 2 sketches that we are going to use which Bluetooth scanner and server. The next sub sections will explain the outcome of these example.

Bluetooth scanner

In this sketch, we used the ESP32 to scan the number of devices in range and gather some information within the 5 seconds interval.

```
12:08:17.911 -> Advertised Device: Name: , Address: 58:a4:3a:dc:4d:5f, manufacturer data: e000013aca7a8f58, serviceUUID: 0000f
12:08:18.004 -> Advertised Device: Name: , Address: 48:26:39:66:e9:80, manufacturer data: 4c001006031dedbf6208, txPower: 5
12:08:18.051 -> Advertised Device: Name: , Address: da:42:58:34:61:65, manufacturer data: 4c0012025403
12:08:18.143 -> Advertised Device: Name: LE-Bose Revolve+ SoundLink, Address: 68:79:27:f2:5f:ad, manufacturer data: 0107021694
12:08:18.235 -> Advertised Device: Name: , Address: 79:d3:ee:0a:5e:dd, manufacturer data: 4c0010063a1d62794668, txPower: 8
12:08:18.328 -> Advertised Device: Name: , Address: cf:64:a3:d4:c7:f8, manufacturer data: 4c0012020002
12:08:18.467 -> Advertised Device: Name: , Address: c0:14:fa:e3:de:7c, manufacturer data: 4c00121900dfc08434c994a742bca85b2bb9
12:08:18.513 -> Advertised Device: Name: , Address: 6c:dd:0d:e4:98:b9, manufacturer data: 4c0010050618289734, txPower: 6
12:08:18.559 -> Advertised Device: Name: , Address: 7e:7e:cf:87:16:79, manufacturer data: 4c001006281e5cdbdc65, txPower: 7
12:08:18.698 -> Advertised Device: Name: , Address: a8:a7:95:0f:6f:2e, manufacturer data: 4c00021550765cb7d9ea4e2199a4fa879613
12:08:18.698 -> Advertised Device: Name: , Address: e2:8e:cb:b5:f1:44, manufacturer data: 4c0012020002
12:08:18.976 -> Advertised Device: Name: , Address: 5c:d0:e4:cf:8c:7b, manufacturer data: 4c0010050118b41313, txPower: 6
12:08:19.718 -> Advertised Device: Name: , Address: d1:a3:e3:dc:e9:8a, manufacturer data: 4c0012020002
12:08:19.718 -> Advertised Device: Name: , Address: c8:6f:9b:05:78:8f, manufacturer data: 4c0012020001
12:08:20.688 -> Advertised Device: Name: , Address: 50:e6:a6:69:b7:a4, manufacturer data: 4c001006361d9f47ff58, txPower: 8
12:08:22.868 -> Devices found: 15
12:08:22.868 -> Scan done!
```

Figure 1 Scanning results in serial monitor

In figure 1, you can see that we scanned all the devices in range and printed their information. Most of these devices has no device name because they are hidden by public scanning.

Bluetooth server

With the Bluetooth LE server, we can create server host where devices can connect and receive data provided by the server. In this example we just made a basic connection and used a smart phone to connect to the ESP32 server. We used an app called nRF Connect and search for devices that are available for connection.

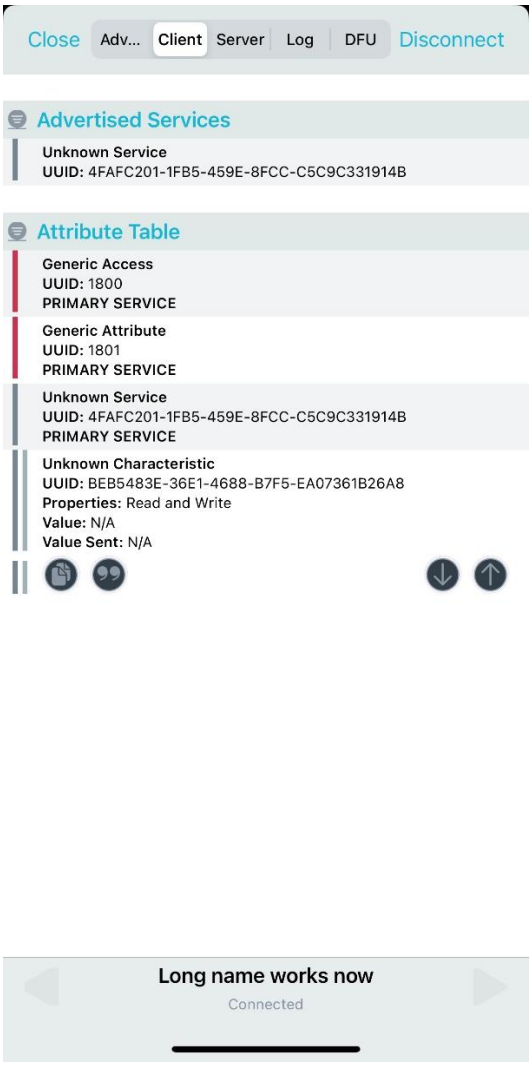

Figure 2 nRF ESP32 server details

In figure 2, you can see the default service and characteristic UUID. Bluetooth LE has a large table of assigned UUID and these can be used to combine with other UUID to create an application or sensor

Advance Assignment

After gaining some basic knowledge of how to scan and create a server, I tried to take on the advance assignment in which we are to add an environmental BLE service using a DHT22 module to send temperature and humidity sensor data through Bluetooth. For this assignment, I used DHT22 header file for the sensor and the server sketch and modify the service and characteristics corresponding to the requirements.

Service Name	**UUID**
Environmental Sensing service	0x181A
Characteristic Name	**UUID**
Temperature	0x2A6E
Humidity	0x2A6F

Table 2 Bluetooth UUID information

The information on table 2 defines the way that two Bluetooth LE devices transfer data back and forth using these concepts called services and characteristics. This concept is also know as GATT.

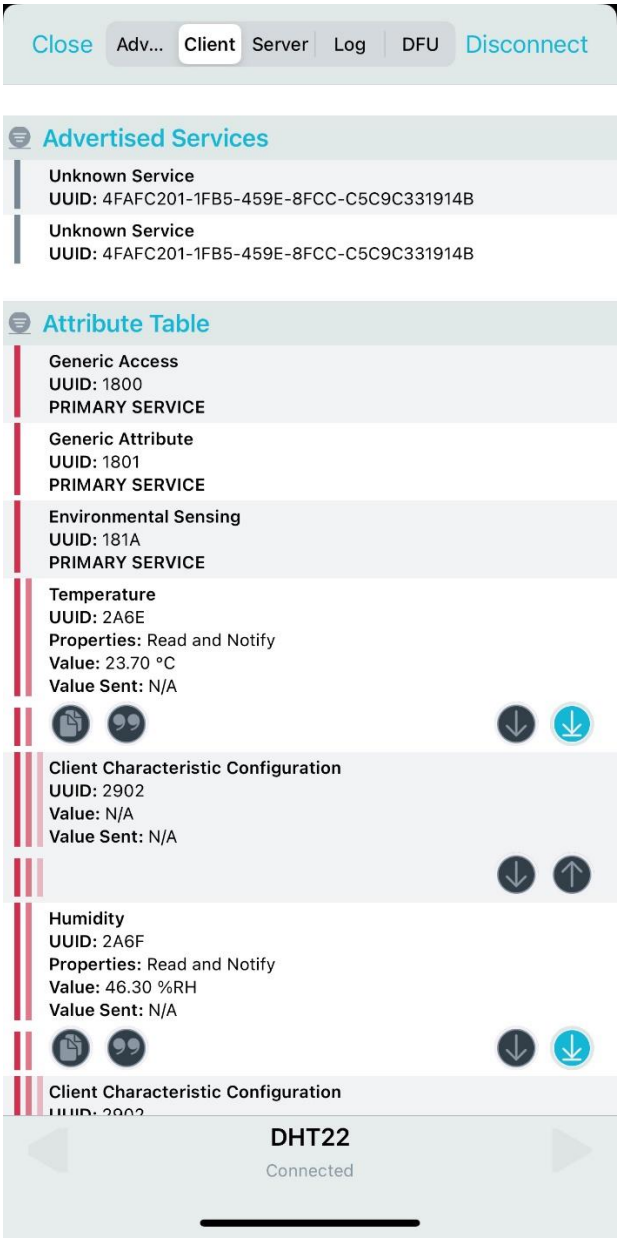

Figure 3 nRF Connect attribute table

In figure 3, you can see temperature and humidity values from the DHT22 being transferred through BLE connection and these values are updated in a 2 second interval.

Conclusion

To conclude this assignment, I have gained knowledge for short distance protocol for ESP32. BLE can be used in short distance projects that will benefit from BLE due to low energy consumption. It took some research to understand the UUID and the outcome was the advance part of this assignment. I look forward in using knowledge gained from this assignment in future projects.

Reference

Assigned numbers. Bluetooth® Technology Website. (n.d.). Retrieved March 19, 2023, from https://www.bluetooth.com/specifications/assigned-numbers/

Wisintainer, M., Sheldon, Germán, Santos, R., Giovanni, Tawil, Y., Bello, S., Santos, S., Túlio, Escasa, D., Michele, Paul, JanWillem, Iturry, E., Ali, Joel, Karad, vikas, Kunal, Ion, … Samuel. (2019, June 4). Esp32 Bluetooth Low Energy (BLE) on Arduino Ide. Random Nerd Tutorials. Retrieved March 19, 2023, from https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/