Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Center for Project-Based Learning

P&S: PBL

LAB
Introduction to Bluetooth Low Energy (BLE)

17.02.2021

1 Bluetooth Low Energy (BLE)

Bluetooth is a wireless communication protocol that allows devices to be connected at a short distance. The first official version of Bluetooth was released in 1994 by the Ericsson Mobile company.

There are two different types of Bluetooth: The first is known as **Bluetooth Classic** and is mainly used where a large bandwidth is required (data transport, audio and multimedia systems). The second type is **Bluetooth Low Energy**. Since December 2009 this extension complements the industry standard Bluetooth, published as part of the specification 4.0. BLE is used where only small data, such as in sensor applications, needs to be transmitted and where there are strict resource limitations (limited energy availability, e.g. battery-powered devices).

The two Bluetooth types are incompatible with each other. Certain devices, such as smartphones, support both standards. The specifications of both standards can be found in the official Bluetooth Core specifications document [1].

The most important features of BLE in its latest version 5.2 are:

- Max data-rate of 2 Mbit/s
- Applications throughput 1360 kbit/s
- Point-to-point and Mesh topology
- Long-range mode with a range of up to 400 m, max-range in a free field is around 1000 m (outdoor) [2]
- Direction Finding
- LE Audio (as alternative to audio transmission in Bluetooth Classic)

1.1 Architecture

The three main blocks in the architecture [3, 19] of a BLE device are: the application, the host and the controller

1.1.1 Application

The application layer [3, 20] is implemented above the Generic Access Profile (GAP) and Generic Attribute Profile (GATT). It defines how an application should react to various events, such as read requests, write requests and notifications (e.g. turn a light on, send temperature, ...). The controller is often a separate IC or core (e.g SPBTLE-RF[1], the

[1] https://www.st.com/content/st_com/en/products/wireless-transceivers-mcus-and-modules/bluetooth-bluetooth-low-energy/spbtle-rf.html

network core of the STM32WB - a STM Microcontroller Unit (MCU) with integrated BLE - or the in the SensorTile used BlueNRG-MS[2]) but there exists also SoC's where the Host and Controller is build on the same microcontroller (e.g. the by Nordic Semiconductor produced nRF52840[3]).

1.1.2 Host

The host[3, 20] contains the following layers:

- Generic Access Profile (GAP)
- Generic Attribute Profile (GATT)
- Attribute Protocol (ATT)
- Security Manager (SM)
- Logical Link Control and Adaption Protocol (L2CAP)
- Host Controller Interface (HCI) - Host side

1.1.3 Controller

The controller[3, 20] contains the following layers:

- Physical Layer (PHY)
- Link Layer
- Direct Test Mode
- Host Controller Interface (HCI) - Controller side

The controller is the block which interacts directly with the radio. This part will not be discussed further, as it is implemented in hardware on the BlueNRG-MS.

[2] https://www.st.com/en/wireless-transceivers-mcus-and-modules/bluenrg-ms.html
[3] https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840

1.2 BLE Peripherals and Centrals

In BLE there are different roles[3, 27], the most frequently used are central and peripheral.

To put it simply, peripherals act like servers, and centrals like clients. Thus, peripherals provide data, such as sensor values. Centrals, on the other hand, make requests for the data (peripherals cannot initiate a connection). In many applications, centrals are smartphones, tablets or computers.

It is possible for devices to have multiple roles simultaneously (f.e broadcaster and peripheral).

1.3 Advertising: Generic Access Profile (GAP)

In the advertising mode, the devices involved are not connected to each other. Peripherals and broadcasters use the advertising mode to signal their existence to other BLE devices. GAP is the framework that defines how and what BLE devices advertise. It contains the following informations:

- Modes and roles of BLE devices
- Advertising parameters, advertising data
- Connection establishment (initiating, accepting, connection parameters)
- Supported security modes and levels

Usually peripherals advertise their name and their primary services. Often, GAP is only used so that the peripheral can be recognized by other centrals and they can connect to it (allowing faster and bidirectional transmission).

1.4 Connection: Attribute Protocol (ATT) and Generic Attribute Profile (GATT)

To establish a connection, two devices have to make following steps [3, 40]:

- The peripheral needs to send out advertisements, including the information that he is connectable (GAP)
- The central needs to scan for those advertisements
- If the central receives an advertisement, he can send a connection request packet
- The peripheral listens for a short interval on the same advertising channel after sending out its advertisement. On receiving a connection request it can connect to the central device.

The connection is now created, but not yet established. A connection is considered established, as soon as the devices receives a packet from the peer. At this moment, the central is called master and the peripheral slave. Once a connection is established, the advertisement process will stop, and GATT services and characteristics are used to communicate in both directions.

1.4.1 Generic Attribute Profile (GATT)

GATT defines how two BLE devices transfer data between each other, using services and their characteristics. GATT uses a data protocol called Attribute Protocol (ATT), which is used to store services, characteristics and related data in a simple lookup table using 16 bit or 128 bit IDs for each entry in the table [4].

A peripheral can only connect to one central at a time. The peripheral is also known as the GATT server, which holds the ATT table and the service/characteristic definitions and the central is known as the GATT client, which sends requests to the server. Transactions of data can only be initiated by the client.

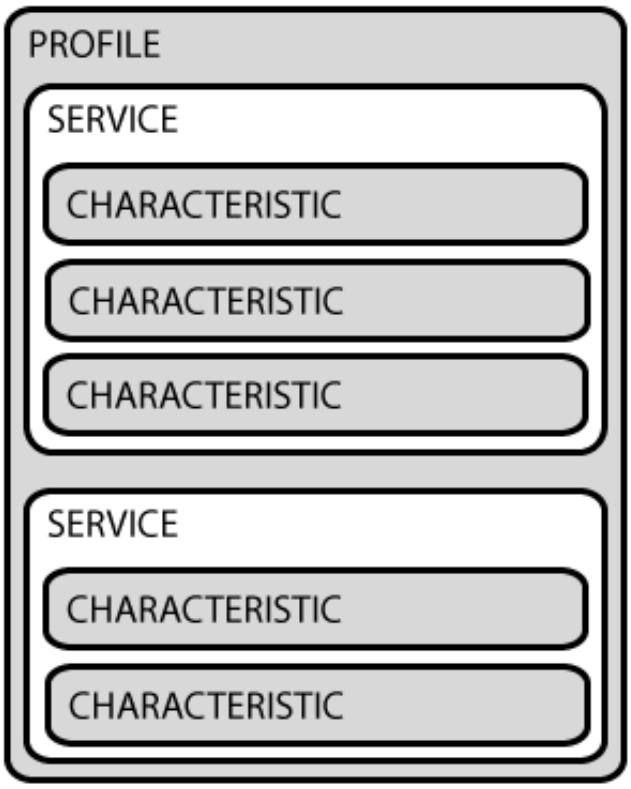

Figure 1: BLE, Generic Attribute Profile

- **Profiles**

  A profile does not actually exists on the peripheral. It is a predefined collection of services. There are official ones from the Bluetooth SIG [5], but they are usually created by the developer.

- **Services**

  A service is a collection of different values (i.e. sensor data), called Characteristics, assigned to a unique identifier (UUID). There are official service definitions from the Bluetooth SIG (16 bit), but they can also be created by the developer (128 bit).

- **Characteristics**

  A characteristic is a single value or an array of values, i.e. the readings of an accelerometer. This means that f.e the array $\begin{bmatrix} a_x & a_y & a_z \end{bmatrix}$ can be a single characteristic. If then another sensor, like a gyroscope, would be added, this would be a new characteristic. Just like every service has its own UUID, the same is true for the characteristics.

  Characteristics can have different properties. They define how a characteristic value can be used. Some of them are read, write, notify and indicate.

References

[1] Unknown, "Bluetooth core specification," Bluetooth SIG, Tech. Rep., 12 2019, rev. v5.2. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726

[2] J. G. Sponås, "Things you should know about bluetooth range," 02 2018. [Online]. Available: https://blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range

[3] M. Afaneh, *Intro to Bluetooth Low Energy: The easiest way to learn BLE*, 1st ed. Novel Bits, LLC, 2018.

[4] K. Townsend, "Introduction to bluetooth low energy," 03 2014. [Online]. Available: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt

[5] Unknown, "Gatt specifications," 04 2020. [Online]. Available: https://www.bluetooth.com/specifications/gatt/

List of Acronyms

MCU Microcontroller Unit

IC Integrated circuit

BLE Bluetooth Low Energy

PHY Physical Layer

UUID Universally unique identifier

HCI Host Controller Interface

GAP Generic Access Profile

GATT Generic Attribute Profile

SoC System-on-a-chip

ATT Attribute Protocol

L2CAP Logical Link Control and Adaption Protocol

SM Security Manager