

Bluetooth Low Energy

WINTER 2017

Introduction to Bluetooth

- ▶ A brand owned by Bluetooth Special Interest Group (SIG)
- ▶ Globally recognizable with up to 97% consumer awareness
- ▶ Defines a global wireless communication protocol in the 2.4 GHz ISM Band
- ▶ The Bluetooth Core Specification is overseen by Bluetooth SIG, and regularly updated by the Bluetooth SIG Working Groups to meet evolving technology and market needs

Prevalent Types

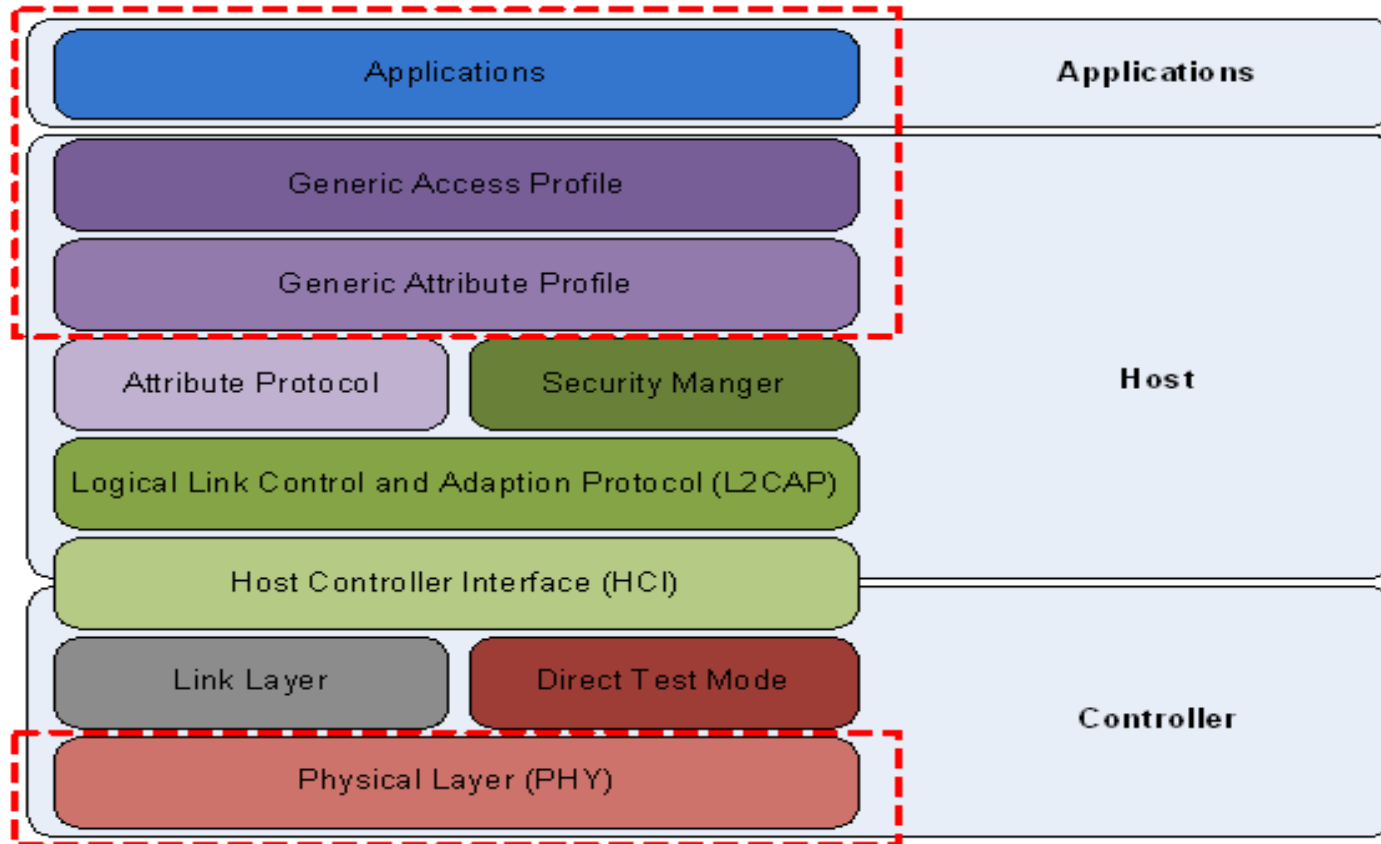
The two most prevalent types of the Bluetooth Specification:

- ▶ BR/EDR – Bluetooth Basic Rate/Enhanced Data Rate (V 2.0/2.1)
 - ▶ Establishes a relatively short-range, continuous wireless connection, which makes it ideal for use-cases such as streaming audio
- ▶ LE – Bluetooth with Low Energy (V 4.0/4.1/4.2)
 - ▶ Also known as Bluetooth Smart, it allows for short bursts of long-range radio connection, making it ideal for Internet of Things (IoT) applications that don't require continuous connection but depend on long battery life.

Bluetooth Low Energy (BLE) - Introduction

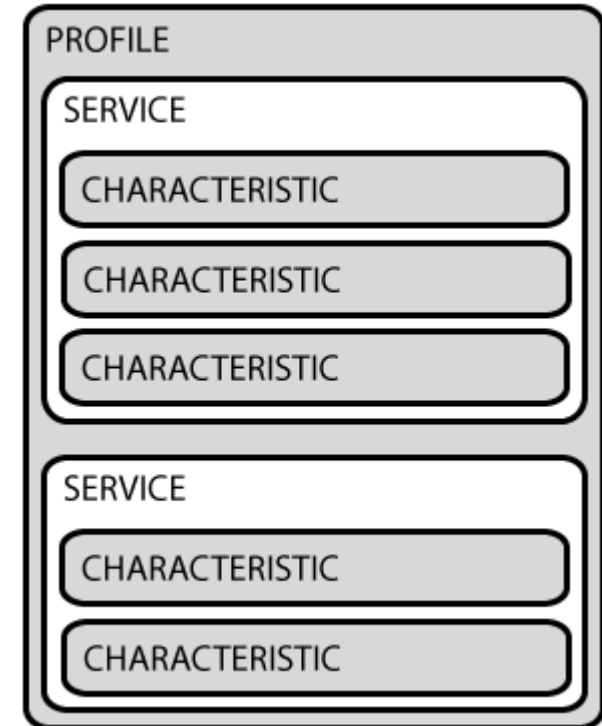
- ▶ Enables low-power consumption of device by using short-burst transmission of data.
- ▶ Useful for devices that send data infrequently, or in small amounts
- ▶ Good standard for Internet of Things (IoT) applications as power-efficient, and devices don't typically need continuous connection

BLE Stack



Overview (1)

- ▶ BLE devices typically have a predefined “profile” associated with them.
 - ▶ Ex. Heart Rate Profile
- ▶ Profiles have a collection of “Services” which they offer. These are typically logical extensions of the profile
 - ▶ Ex. A device with a Heart Rate profile would offer a Heart Rate service
- ▶ Each service contains “characteristics”; these are lowest denominations of data that they can provide to facilitate the service they offer
 - ▶ Ex. Heart Rate service has 3 characteristics: Heart Rate Measurement, Body Sensor Location, Heart Rate Control Point
- ▶ Some services (including the above) are predefined standards.



Overview (2)

- ▶ Devices scan the radio channel, while complementary devices broadcast/advertise. Processed facilitated by **Generic Access Profile (GAP)**
 - ▶ Details: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap>
- ▶ Three channels to avoid interference, scanned sequentially. Detection may not occur instantaneously
- ▶ Upon detection of each other, devices can sync. At this point, the scanning device becomes the “master”, while the broadcasting the “slave”
- ▶ “Master” device drives communication

GATT (1)

- ▶ All BLE devices use the **Generic Attribute Profile (GATT)**. Android/iOS API revolves around GATT concepts.
- ▶ Main concepts:
 - ▶ Client – Initiates requests and accepts responses (ex. Smartphone)
 - ▶ Server – Receives requests and returns responses (ex. Sensors)
- ▶ Referring to previous slide, the “Client” typically acts as the “master” (drives communications) while the “Server” broadcasts/advertises and responds (ie. acts as “slave”)

GATT (2)

- ▶ In addition to these, other key concepts include:
 - ▶ Characteristic – The piece of data that's transferred between client and server. This can be the sensor data.
 - ▶ Descriptor – Complements characteristics; typically provides additional information
 - ▶ Service – A collection of related characteristics. This “collection” is typically defined at a higher-level. An example is a *Health Service* that returns heart-rate and perspiration characteristics
- ▶ Will typically be working with Characteristics for data, of which the two main types of operation are **Read** and **Write**

Identifiers

- ▶ Services, characteristics, descriptors typically referred to as “attributes”
- ▶ Each attribute has a **UUID**
 - ▶ Unique 128-bit identifier
 - ▶ Range of UUIDs **reserved** by Bluetooth SIG for standard attributes
 - ▶ xxxxxxxx-0000-1000-8000-00805F9B34FB
 - ▶ As such, UUIDs typically represented as 16 or 32-bit values
 - ▶ Can define full custom UUID

Conclusion

- ▶ Consult complementary slides for more information
- ▶ Base project provided for Nucleo board along with documentation. Use these as basis
- ▶ For Android, use BLE API. Good starting point:
 - ▶ <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>