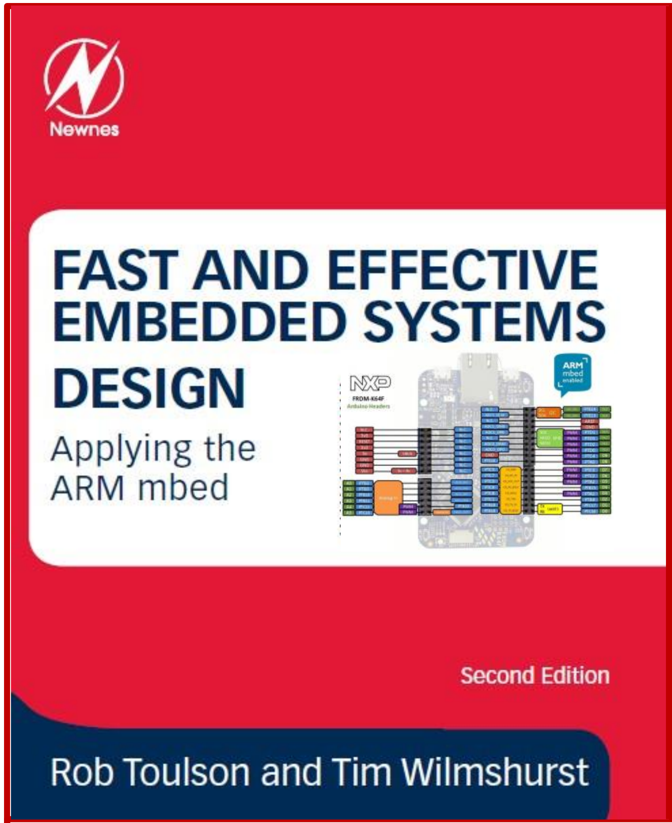


Newnes

FAST AND EFFECTIVE EMBEDDED SYSTEMS DESIGN

Applying the ARM mbed

Second Edition

Rob Toulson and Tim Wilmshurst

Chapter 11: Wireless Communication – Bluetooth Low Energy

tw rev. 26.8.16

If you use or reference these slides or the associated textbook, please cite the original authors' work as follows:

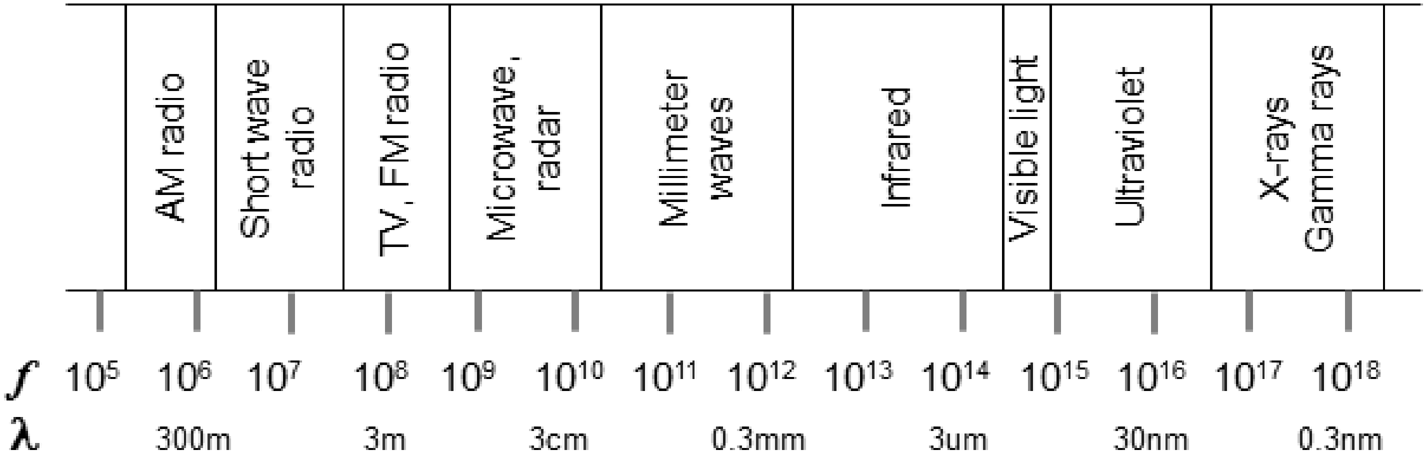
Toulson, R. & Wilmshurst, T. (2016). Fast and Effective Embedded Systems Design - Applying the ARM mbed (2nd edition), Newnes, Oxford, ISBN: 978-0-08-100880-5.

www.embedded-knowhow.co.uk

Some Wireless Preliminaries 1

The electromagnetic spectrum shows where various wireless activities sit. Any frequency on the spectrum, from the lowest frequency up to visible light, can be used for data communication. Almost all of this is very strictly regulated by national and international agencies.

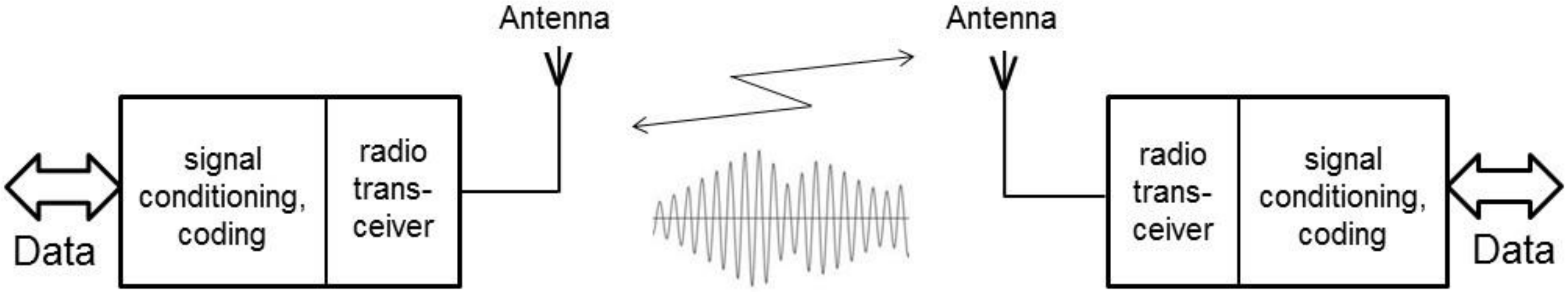
The International Telecommunication Union manages the allocation of the radio spectrum between different broadcasters and applications. It reserves certain frequency bands for Industrial, Scientific and Medical (ISM) applications. These are *unlicensed*, and some vary between countries. The 2.4 GHz band is however reserved for unlicensed use in all regions. It has become widely used, mainly for short-range, low power applications.



Some Wireless Preliminaries 2

While the spectrum represents the “pure” radio frequencies, they only become useful once they are carrying information. This is done by the process of *modulation*; the information to be carried is imprinted onto the carrier frequency, through one of a number of different techniques.

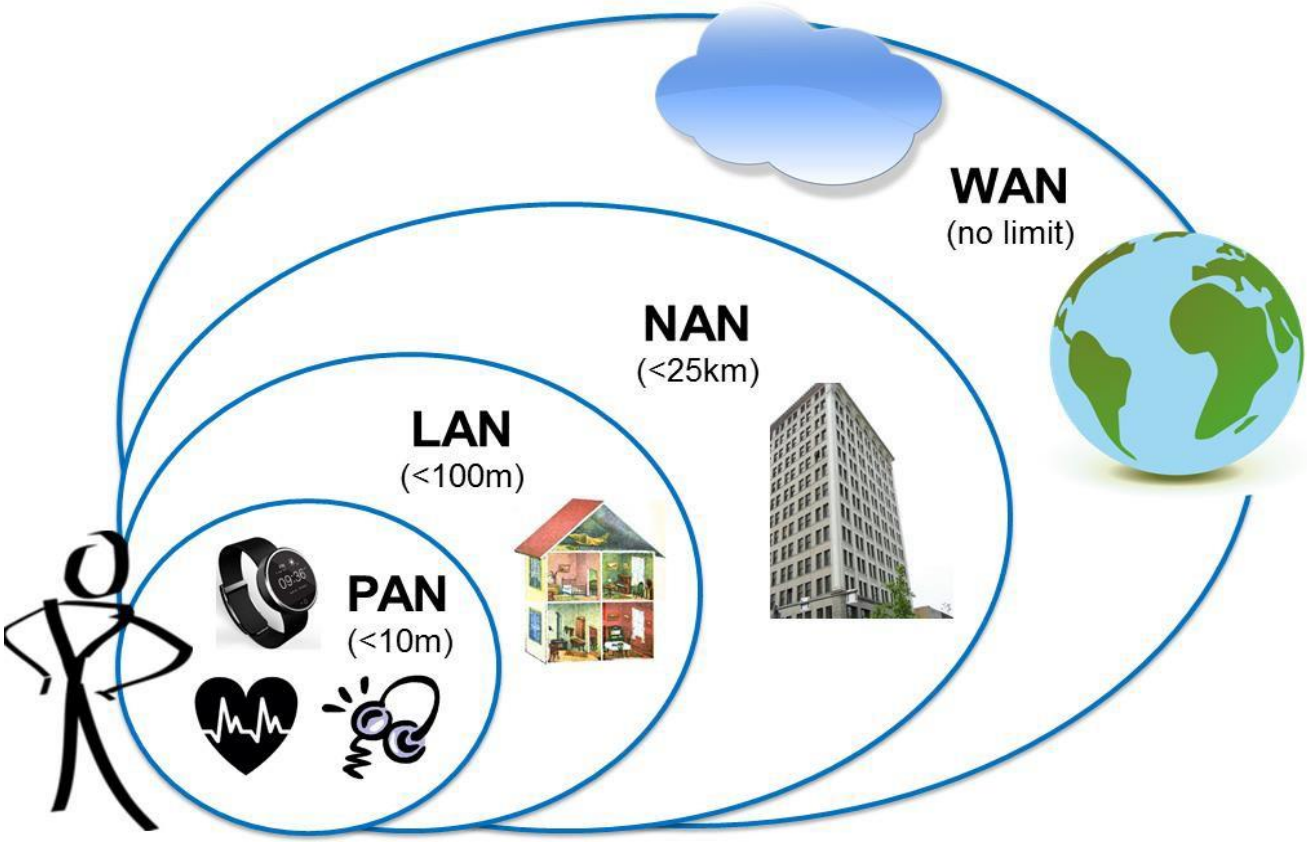
One effect of modulation is to cause fluctuations around the base frequency; thus if we say that a certain radio station can be found at the frequency of 103 MHz, in fact it is in a narrow band of frequencies centred on 103 MHz. The word *bandwidth* is used to define a range of frequencies, for example within which a particular transmission may be taking place.



Wireless Networks

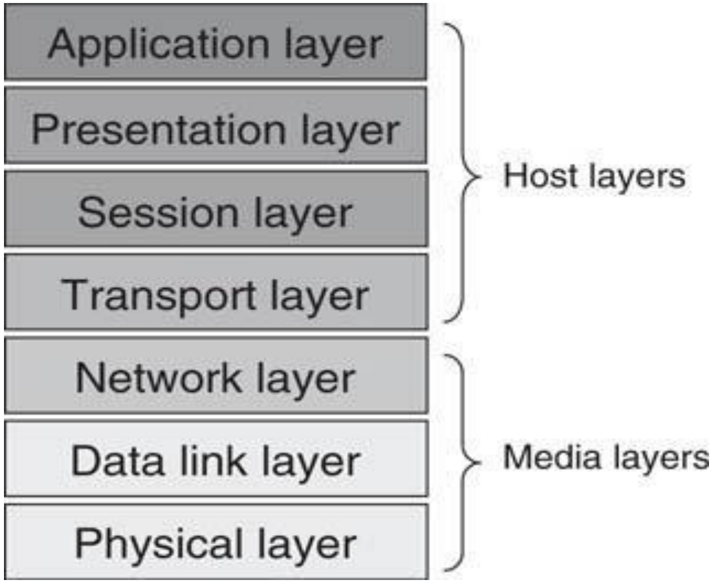
Networks can be divided into four categories:

- The Personal Area Network (PAN) usually relates to devices close to a person.
- The Local Area Network (LAN) typically applies to a single building.
- The Neighbourhood Area Network (NAN) could apply to a smart transport or smart energy system.
- The Wide Area Network (WAN) includes national or global systems, notably the Internet.



Protocols

With large networked systems, protocols can become very complicated, defining every aspect of the communication link. To aid in the process of defining a protocol, the International Organisation for Standardisation (ISO) devised a “protocol for protocols”, called the *Open Systems Interconnect* (OSI) model, as shown. Each layer of the OSI model provides a defined set of services to the layer above, and each therefore depends on the services of the layer below.



Protocols – IEEE Working Groups

The IEEE (the Institute of Electrical and Electronic Engineers) plays a major role in defining standards and protocols. It maintains a set of standards for Local Area Networks, allocated the number 802. A small number of these which are relevant to this and the next chapter are shown in the Table.

IEEE Working Group	Description
802.3	Ethernet
802.11	Wireless LAN, including Wi-Fi
802.15	Wireless PAN
802.15.1	Bluetooth
802.15.3	High-rate wireless PAN
802.15.4	Low-rate wireless PAN, e.g. Zigbee

Introducing Bluetooth

Bluetooth is a digital radio protocol, meant for PAN applications, and operating in the 2.4 GHz radio band. It provides wireless data links between devices such as mobile phones, wireless audio headsets, computer interface devices like mice and keyboards, and remote sensors.

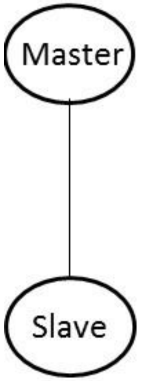
There are three Bluetooth classes, based on output power. Class 2 is the most common. The main characteristics are:

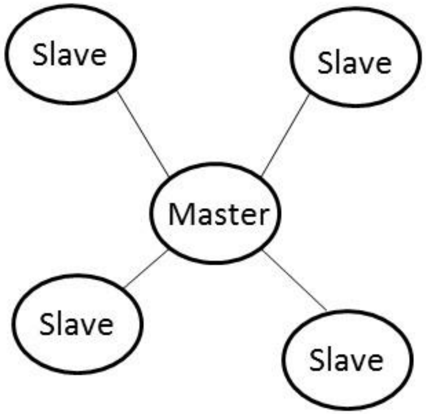
- The approximate communication range is up to 100 m for Class 1 Bluetooth devices, up to 10 m for Class 2 devices, and 1 m for Class 3.
- Bluetooth is relatively low power; devices of Classes 1 to 3 use around 100 mW, 2.5 mW and 1 mW respectively.
- Data rates up to 3 Mbps can be achieved. Recent higher data rate versions are being adopted.
- Up to 8 devices can be simultaneously linked, in a *piconet*. A Bluetooth device can belong to more than one piconet.
- Spread-spectrum frequency hopping is applied, with the transmitter changing frequency in a pseudo-random manner 1600 times per second.

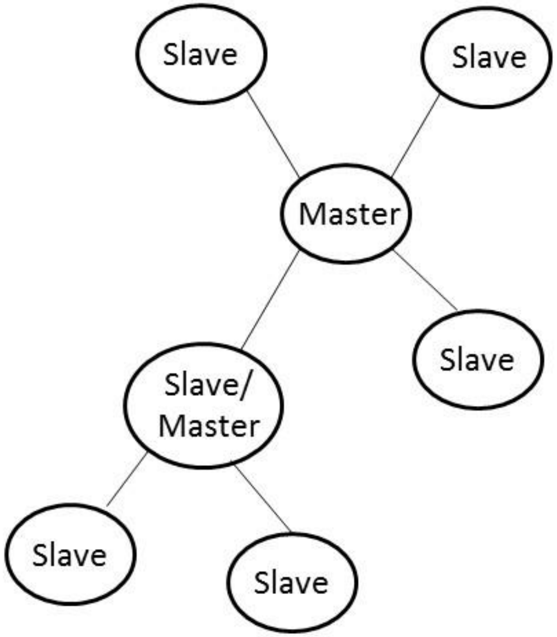
Introducing Bluetooth

When Bluetooth devices detect one another, they determine automatically whether they need to interact. Each device has a unique Media Access Control (MAC) address which communicating devices can recognise and initialise interaction if required. This process follows three phases:

- Discovery - a slave module broadcasts its name and MAC address, seeking for devices to link to.
- Pairing - slave and master exchange identification and authentication data, exploring whether a link should be established.
- Connecting - initiated by the master, through which a link is finally established.







Example Bluetooth networks

Introducing Bluetooth Low Energy

Bluetooth Low Energy (**BLE**, also marketed as **Bluetooth Smart**) started as part of the Bluetooth 4.0 Core Specification. It's tempting to present BLE as a smaller, highly opti-mized version of its bigger brother, **classic Bluetooth**, but in reality, BLE has an entirely different lineage and design goals.

Background

- Originally introduced under the name **Wibree** by Nokia in 2006.
- Merged into the main Bluetooth standard in 2010 with the adoption of the Bluetooth Core Specification Version 4.0.
- The beginning focus was to design a radio standard with the **lowest possible power consumption**, specifically optimized for **low cost, low bandwidth, low power, and low complexity.**

What Makes BLE Different?

Compared to other wireless standards, the rapid growth of BLE is relatively easy to explain: BLE has gone further faster because its fate is so intimately tied to the phenomenal growth in smartphones, tablets, and mobile computing.

Key Benefits

- low power consumption - Designed for coin cells (15ma peak transmit, 1uA sleep)
- Low latency connection (3ms)
- Designed to send small packets of data (Connect->transmit->disconnect->sleep)
- Small size and low cost
- Low bandwidth(~100 kbps)

Bluetooth Classic vs SMART

An actual battery-life comparison
Innova’s anti-loss products

vs

Protag G1 (Classic)
Released: 2012
Battery Capacity: 3.7V, 270mAh
Battery Life: 1 - 2 weeks

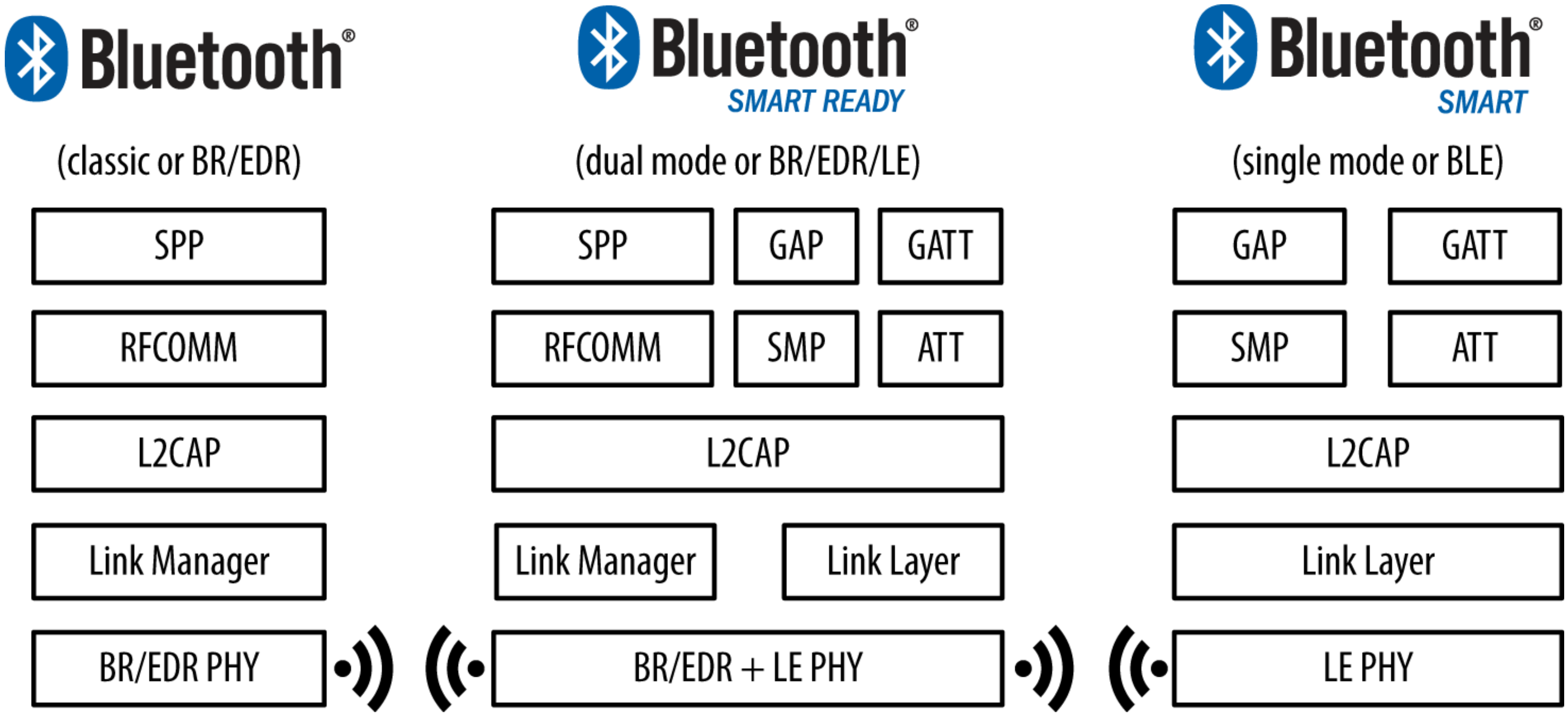
Protag Elite (SMART)
Released: 2013
Battery Capacity: 3.7V, 150mAh
Battery Life: 6 months to 1 year

Classic Bluetooth vs BLE

Specifications	Bluetooth	BLE(Bluetooth Low Energy)
Network/Topology	Scatternet	Star Bus
Power consumption	Low (less than 30 mA)	Very Low (less than 15 mA)
Speed	700 Kbps	1 Mbps
Range	<30 m	50 meters(150 meters in open field)
RF Frequency band	2400 MHz	2400 MHz
Frequency Channels	79 channels from 2.400 GHz to 2.4835 GHz with 1 MHz spacing	40 channels from 2402MHz to 2480 MHz (includes 3 advertising and 37 data channels)
Modulation	GFSK (modulation index 0.35) , π/4 DQPSK, 8DPSK	GFSK (modulation index 0.5)
Latency in data transfer between two devices	Approx. 100 ms	Approx. 3 ms
Spreading	FHSS (1MHz channel)	FHSS (2MHz channel)
Link layer	TDMA	TDMA
message size(bytes)	358 (Max)	8 to 47
Error detection/correction	8 bit CRC(header), 16 bit CRC, 2/3 FEC(payload), ACKs	24 bit CRC, ACKs
Security	64b/128b, user defined application layer	128 bits AES, user defined application layer
Application throughput	0.7 to 2.1 Mbps	less than 0.3 Mbps
Nodes/Active Slaves	7	Unlimited

Specification configurations

Device	BR/EDR (classic Bluetooth) support	BLE (Bluetooth Low Energy) support
Pre-4.0 Bluetooth	Yes	No
4.x Single-Mode (Bluetooth Smart)	No	Yes
4.x Dual-Mode (Bluetooth Smart Ready)	Yes	Yes



Bluetooth

(classic or BR/EDR)

SPP

RFCOMM

L2CAP

Link Manager

BR/EDR PHY

Bluetooth SMART READY

(dual mode or BR/EDR/LE)

SPP | GAP | GATT

RFCOMM | SMP | ATT

L2CAP

Link Manager | Link Layer

BR/EDR + LE PHY

Bluetooth SMART

(single mode or BLE)

GAP | GATT

SMP | ATT

L2CAP

Link Layer

LE PHY

BLE Platform Support

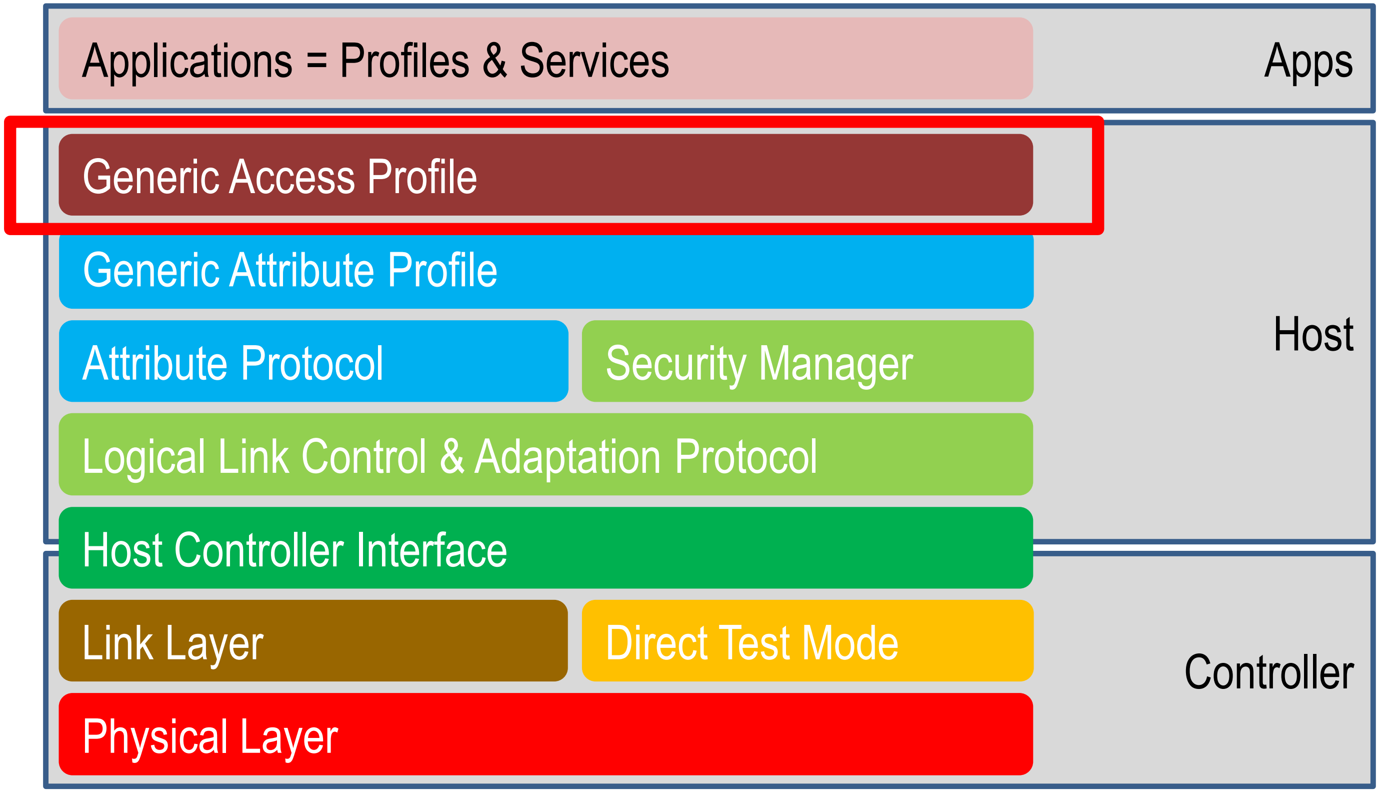
Support for Bluetooth 4.0 and Bluetooth Low Energy (which is a subset of BT 4.0) is available on most major platforms as of the versions listed below:

- iOS5+ (iOS7+ preferred)
- Android 4.3+ (numerous bug fixes in 4.4+)
- Apple OS X 10.6+
- Windows 8 (XP, Vista and 7 only support Bluetooth 2.1)
- GNU/Linux Vanilla BlueZ 4.93+

Protocol Stack

GAP - Generic Access Profile

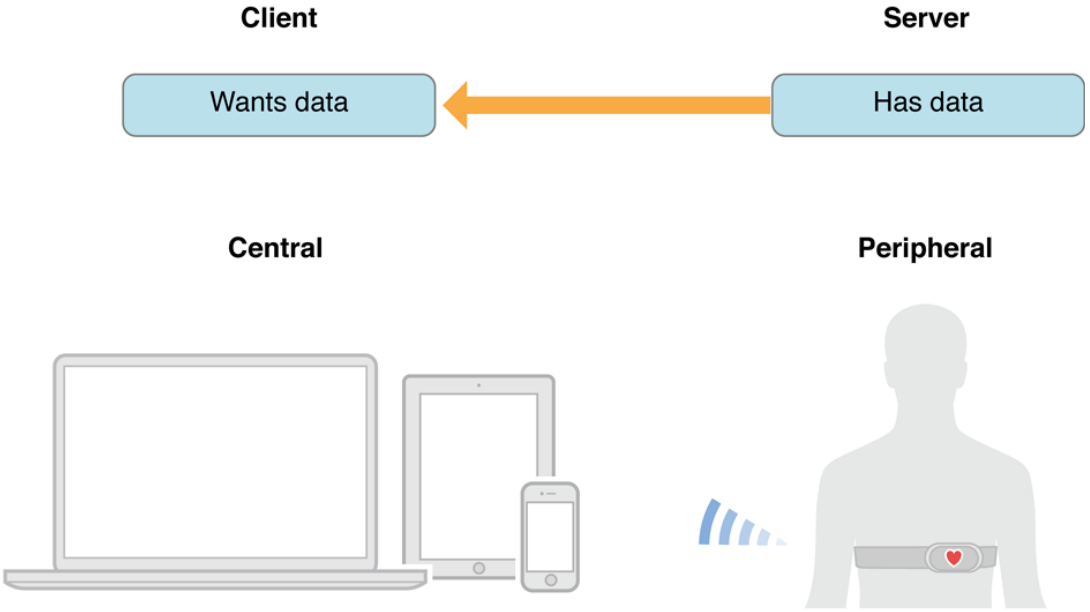
GAP controls connections and advertising in Bluetooth. GAP is what makes your device visible to the outside world, and determines how two devices can (or can't) interact with each other.


Applications = Profiles & Services
Apps
Generic Access Profile
Generic Attribute Profile
Attribute Protocol
Security Manager
Host
Logical Link Control & Adaptation Protocol
Host Controller Interface
Link Layer
Direct Test Mode
Controller
Physical Layer

GAP - Generic Access Profile

Device role

GAP defines various roles for devices, but the two key concepts to keep in mind are **Central** devices and **Peripheral** devices.



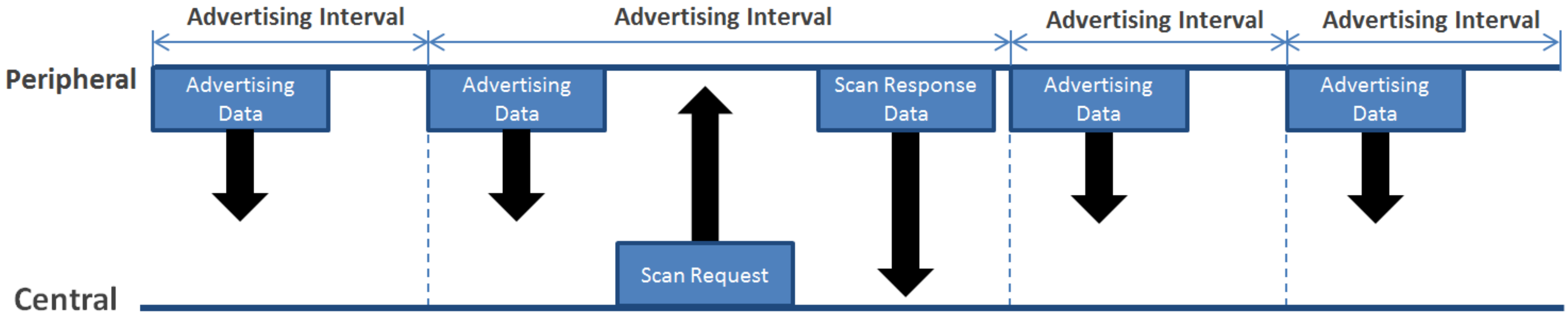
Peripheral devices are small, low power, resource constrained devices that can connect to a much more powerful central device.

Central devices are usually the mobile phone or tablet that you connect to with far more processing power and memory.

GAP - Generic Access Profile

Advertising and Scan Response Data

There are two ways to send advertising out with GAP. The **Advertising Data payload** and the **Scan Response payload**.



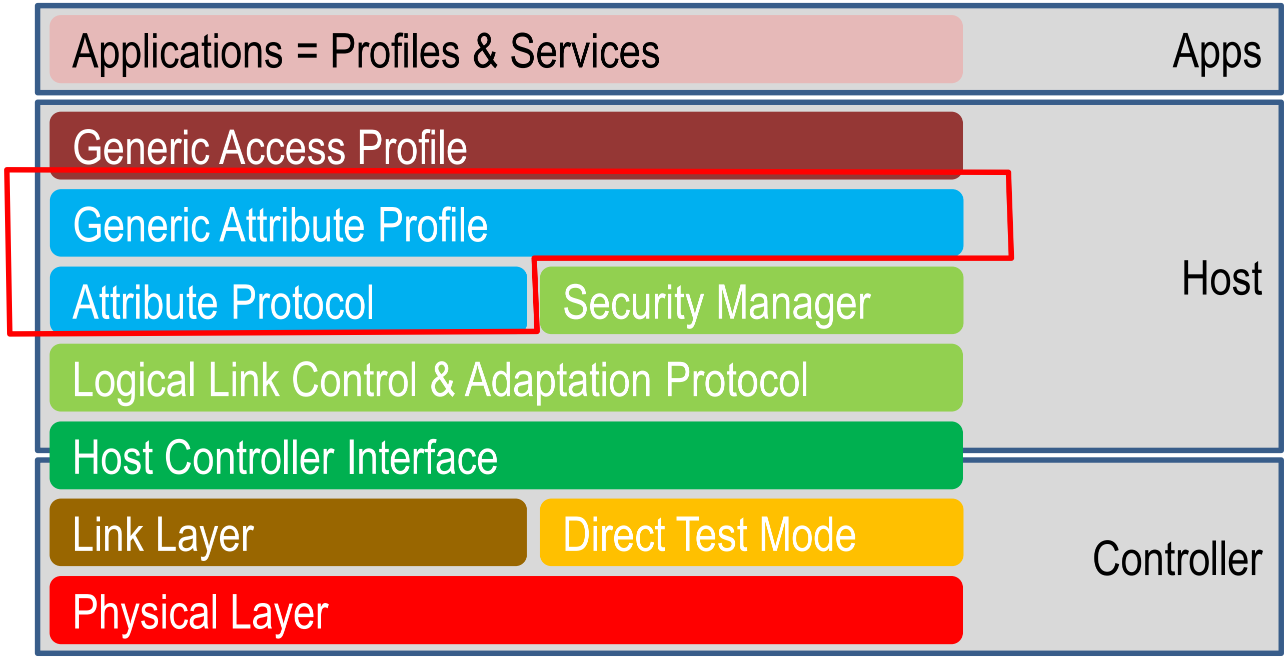
Peripheral will set a specific advertising interval, and every time this interval passes, it will retransmit it's main advertising packet.

If a listening device is interested in the scan response payload (and it is available on the peripheral) it can optionally request the scan response payload, and the **peripheral** will respond with the additional data.

Protocol Stack

GATT - Generic Attribute Profile

GATT defines the way that two Bluetooth LE devices transfer data back and forth using concepts called **Services** and **Characteristics**.

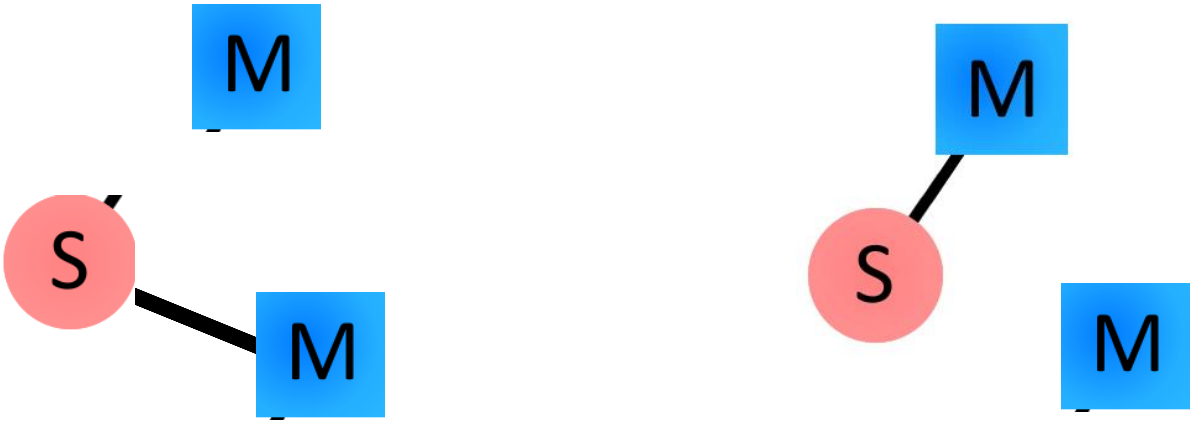

Applications = Profiles & Services
Apps
Generic Access Profile
Generic Attribute Profile
Attribute Protocol
Security Manager
Host
Logical Link Control & Adaptation Protocol
Host Controller Interface
Link Layer
Direct Test Mode
Controller
Physical Layer

It makes use of a generic data protocol called the **Attribute Protocol**(ATT), which is used to store Services, Characteristics and related data in a simple lookup table using **16-bit IDs** for each entry in the table.

GATT - Generic Attribute Profile

Connected Network Topology

The most important thing to keep in mind with GATT and connections is that *connections are exclusive*. What is meant by that is that **a BLE peripheral can o nly be co nnected to one central device (a mobile phone, etc.) at a time!**

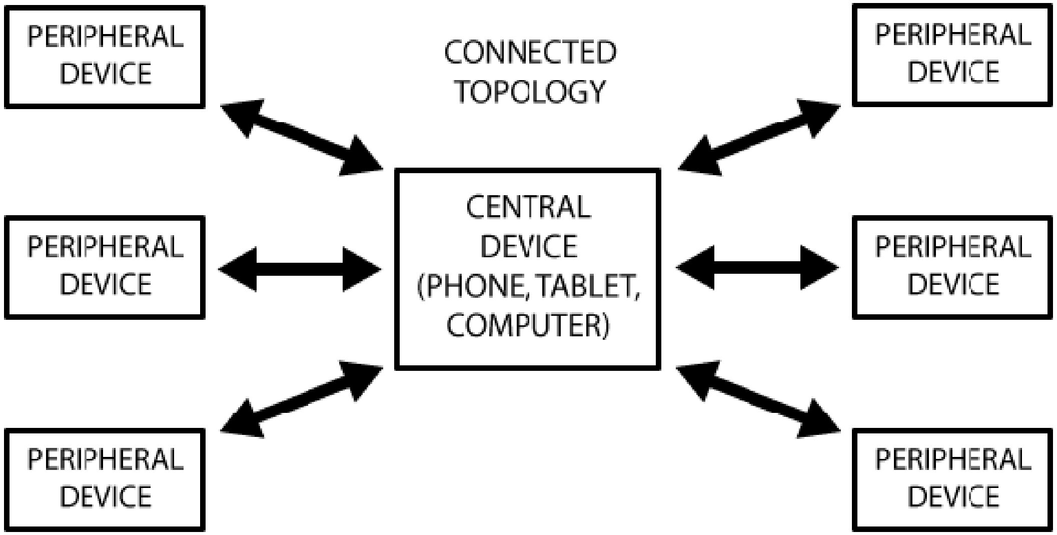


As soon as a peripheral connects to a central device, it will stop advertising itself and other devices will no longer be able to see it or connect to it until the existing connection is broken.

GATT - Generic Attribute Profile

Connected Network Topology

A peripheral can only be connected to one central at a time, but the central device can be connected to multiple peripherals. If data needs to be exchanged between two peripherals, a custom mailbox system will need to be implemented where all messages pass through the central device.

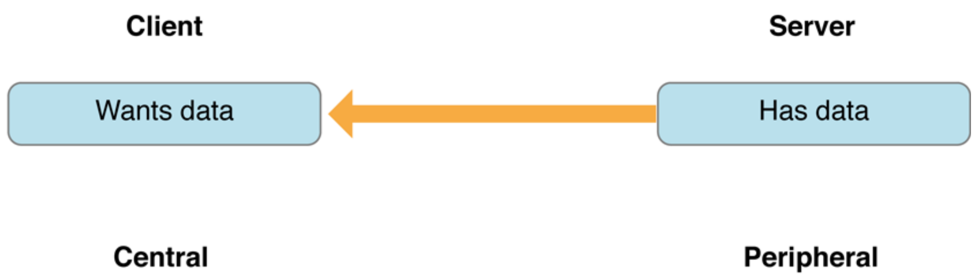


Once a connection is established between a peripherals and central device, however, communication can take place in both directions.

GATT - Generic Attribute Profile

GATT Transactions

The peripheral is known as the **GATT Server**, which holds the ATT lookup data and service and characteristic definitions, and the **GATT Client** (the phone/tablet), which sends requests to this server.



Client

Wants data

Server

Has data

Central

Peripheral

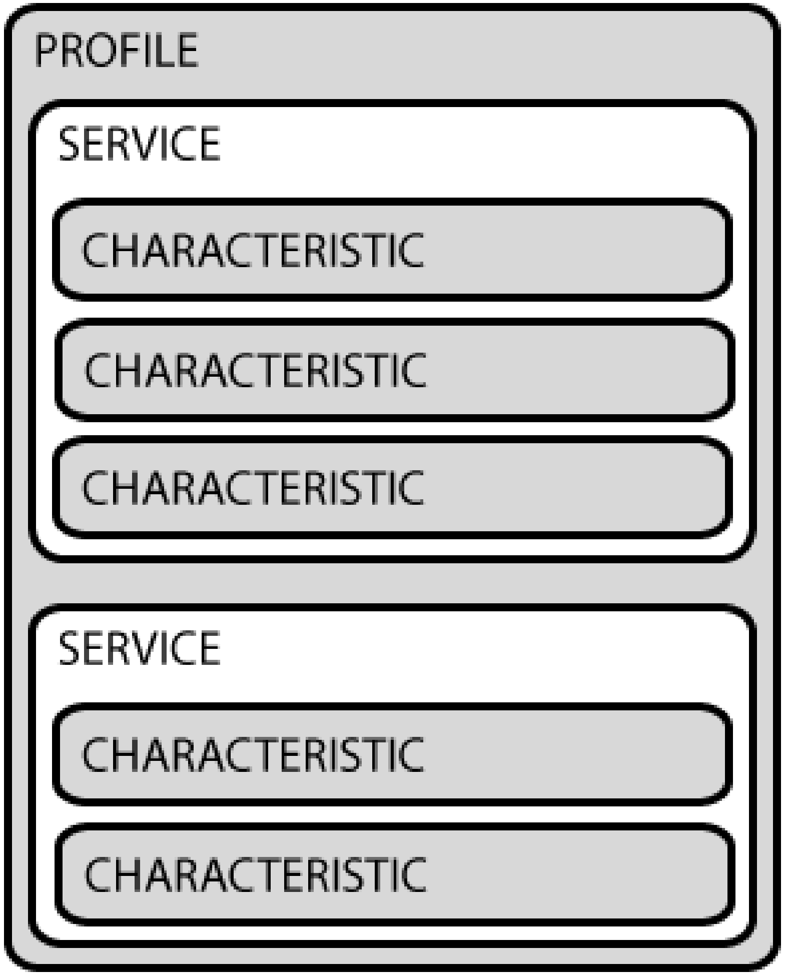
All transactions are started by the master device, the GATT Client, which receives response from the slave device, the GATT Server.

GATT - Generic Attribute Profile

Services and Characteristics

GATT transactions in BLE are based on high-level, nested objects called

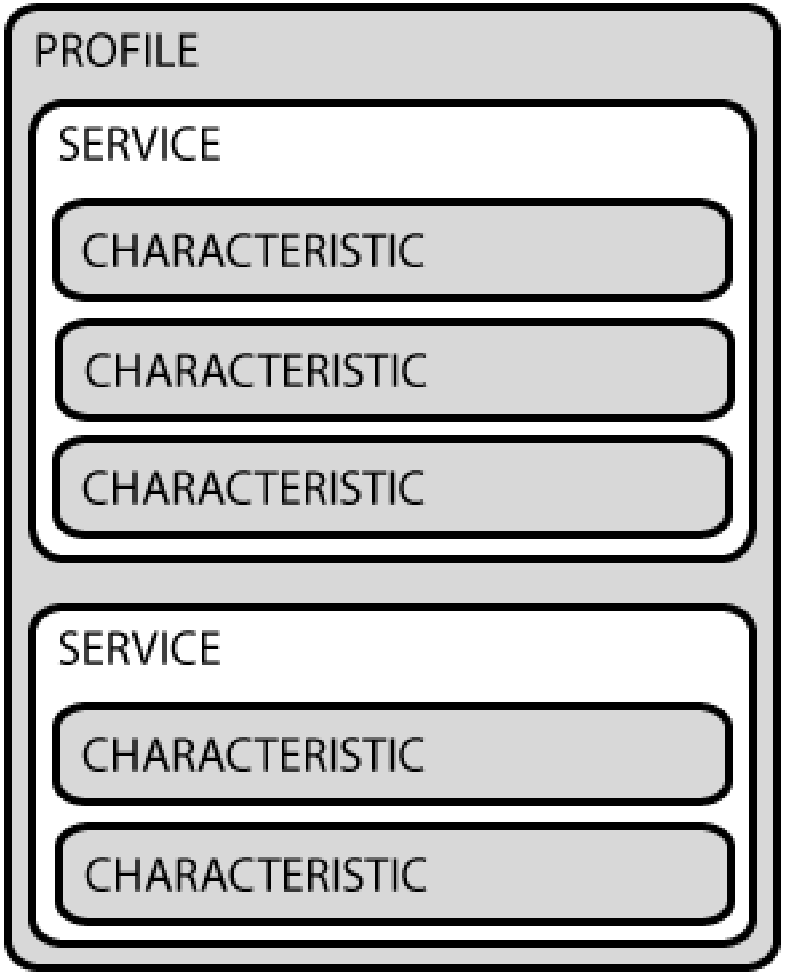
- **Profiles**
- **Services**
- **Characteristics**



Services and Characteristics

- **Profiles**
  A Profile doesn't actually exist on the BLE peripheral itself, it's simple a pre-defined collection of Services that has been compiled by either the Bluetooth SIG or by the peripheral designers.

- Services
- Characteristics

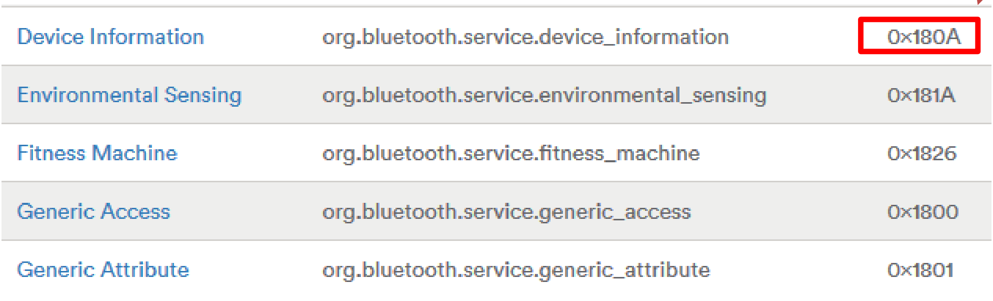


Note: The example document about the Heart Rate Profile is available on the course website.

Services and Characteristics

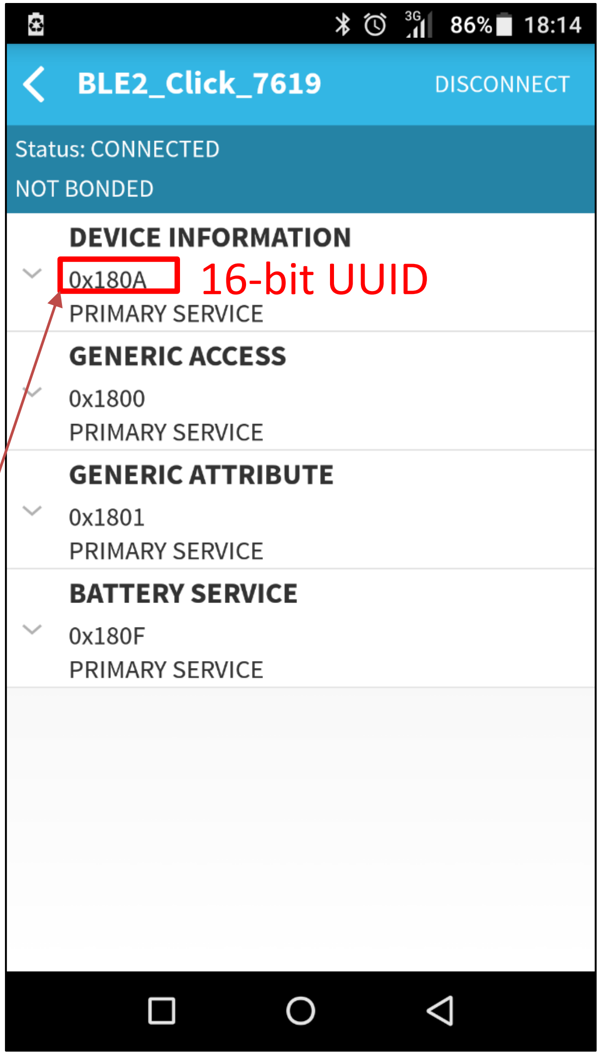
- **Services**
  Services are used to break data up into logic entities, and contain specific chunks of data called characteristics.

A service distinguishes itself from other services by means of a unique numeric ID called a UUID, which can be either 16-bit (official adopted BLE Services) or 128-bit (Customized services).



Device Information	org.bluetooth.service.device_information	0x180A
Environmental Sensing	org.bluetooth.service.environmental_sensing	0x181A
Fitness Machine	org.bluetooth.service.fitness_machine	0x1826
Generic Access	org.bluetooth.service.generic_access	0x1800
Generic Attribute	org.bluetooth.service.generic_attribute	0x1801

GATT Services on **Bluetooth Technology Website**



Status: CONNECTED
NOT BONDED

DEVICE INFORMATION
0x180A
PRIMARY SERVICE

16-bit UUID

GENERIC ACCESS
0x1800
PRIMARY SERVICE

GENERIC ATTRIBUTE
0x1801
PRIMARY SERVICE

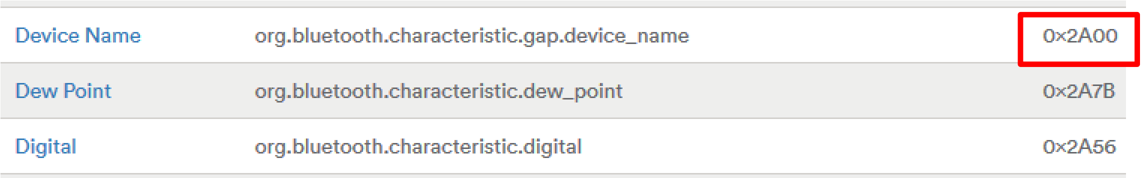
BATTERY SERVICE
0x180F
PRIMARY SERVICE

Service list on BLE2 click

Services and Characteristics

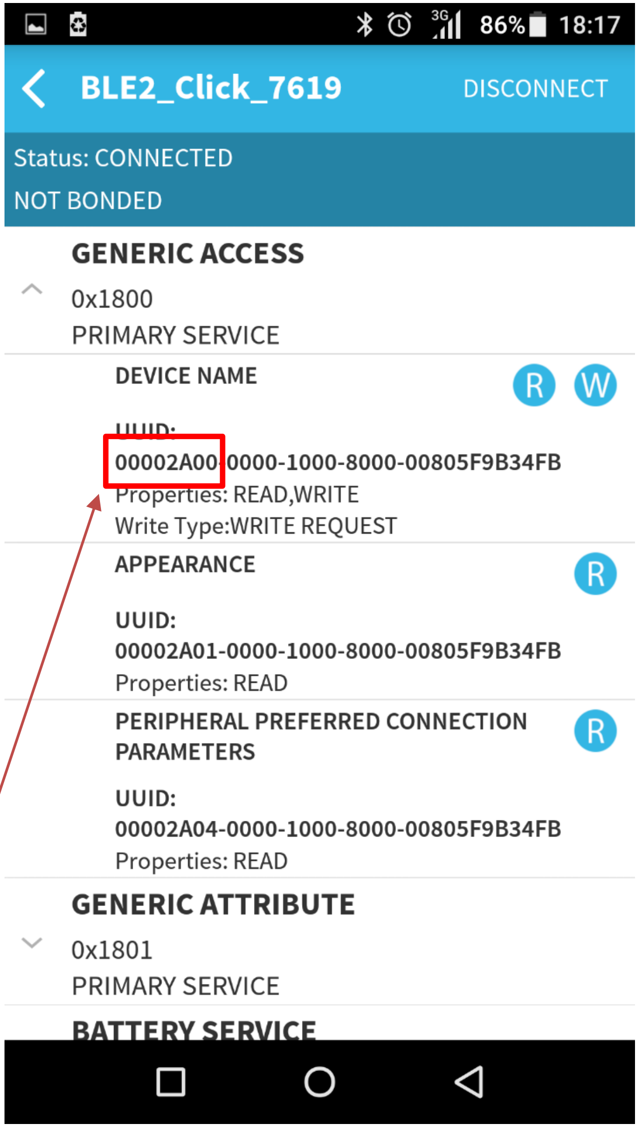
- **Characteristics**
  The lowest level concept in GATT transactions is the Characteristic, which encapsulates a single data point.

Similarly to Services, each Characteristic distinguishes itself via a pre-defined 16-bit or 128-bit UUID.
It may contain an array of related data, for example, the device name.



Device Name	org.bluetooth.characteristic.gap.device_name	0x2A00
Dew Point	org.bluetooth.characteristic.dew_point	0x2A7B
Digital	org.bluetooth.characteristic.digital	0x2A56

GATT Characteristics on [Bluetooth Technology Website]()



86% 18:17

BLE2_Click_7619 DISCONNECT

Status: CONNECTED
NOT BONDED

GENERIC ACCESS
0x1800
PRIMARY SERVICE

DEVICE NAME R W
UUID:
00002A00-0000-1000-8000-00805F9B34FB
Properties: READ,WRITE
Write Type:WRITE REQUEST

APPEARANCE R
UUID:
00002A01-0000-1000-8000-00805F9B34FB
Properties: READ

PERIPHERAL PREFERRED CONNECTION PARAMETERS R
UUID:
00002A04-0000-1000-8000-00805F9B34FB
Properties: READ

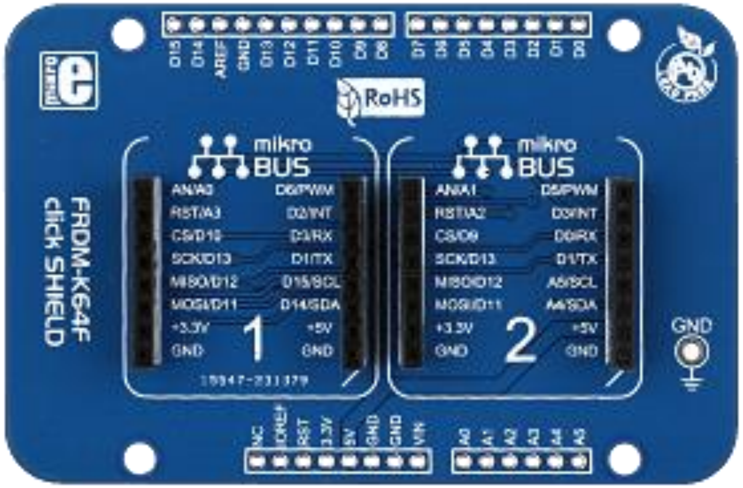
GENERIC ATTRIBUTE
0x1801
PRIMARY SERVICE

BATTERY SERVICE

Characteristic list under the service

The RN4020 Bluetooth Low Energy Module

The simplest use of the RN4020 module, BLE2 click features the RN4020 module from Microchip, that integrates RF, a baseband controller, and a command API processor. The click communicates with the target board MCU through mikroBUS™ RX, TX and AN (CMD), PWM (con.), and RST (wake) lines. The board is designed to use 3.3V power supply only. And the click shield is used here to connected to K64F.



Simple Bluetooth LE: Receive device name from rn4020(1)

```
/* Program Example 11.1: Read the device name from the Bluetooth LE.
*/
#include "mbed.h"
#include <string>
Serial rn4020(D1, D0);
Serial pc(USBTX, USBRX);
DigitalOut AWAKE(A3);
void sendString(string msg);
string getString(void);
char getChar(void);
string buff;

int main() {
    AWAKE = 0;
    rn4020.baud(115200);
    wait(1.0);
    AWAKE = 1;
    pc.printf("CMD mode: %s",getString());
    sendString("GN\r\n"); // Get device name
    pc.printf("Device name: %s",getString());
    sendString("A\r\n");  // Start advertisement
    pc.printf("Annotation start: %s",getString());
}
```

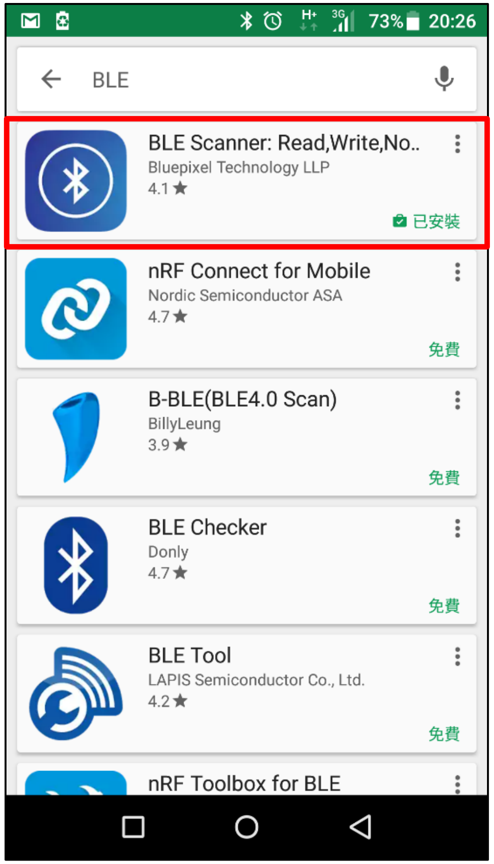
……

Simple Bluetooth LE: Receive device name from rn4020(2)

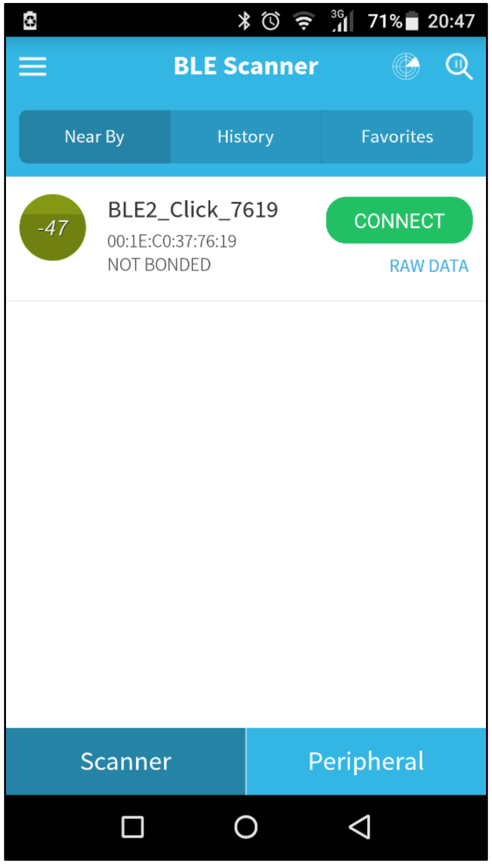
```
void sendString(string msg){
    rn4020.printf("%s",msg);
}
char getChar(){
    return rn4020.getc();
}
string getString(){
    string msg = "";
    char prev = ' ';
    char curr = ' ';
    while(1){
        if(rn4020.readable()){
            prev = curr;
            curr = getChar();
            msg += curr;
            if(prev=='\r' and curr=='\n'){
                break; // Break when receive “\r\n”
            }
        }
    }
    return msg;
}
```

Simple Bluetooth LE: Receive device name from rn4020(3)

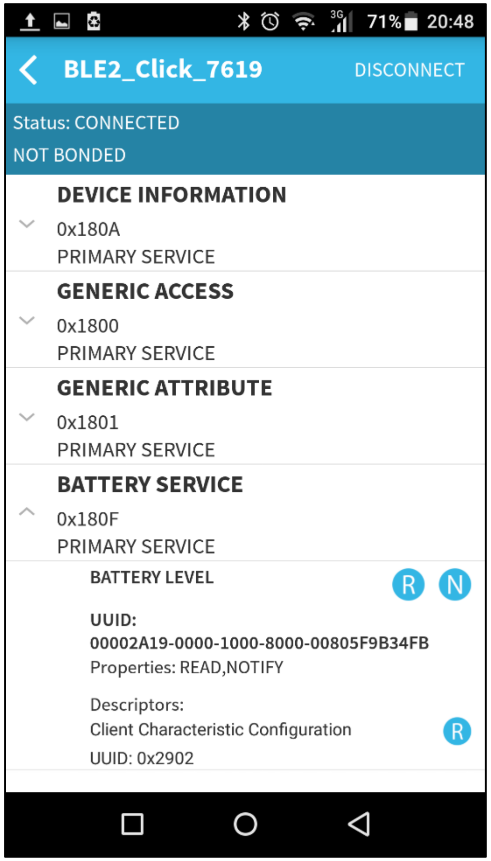
After start the advertisement, we could use smart phone with BLE app to scan rn4020 on both Android and iOS platform. We take the android platform as an example.



Installing the app on smart phone



Scanning the advertising device



Status: CONNECTED
NOT BONDED

DEVICE INFORMATION
0x180A
PRIMARY SERVICE

GENERIC ACCESS
0x1800
PRIMARY SERVICE

GENERIC ATTRIBUTE
0x1801
PRIMARY SERVICE

BATTERY SERVICE
0x180F
PRIMARY SERVICE

BATTERY LEVEL R N
UUID:
00002A19-0000-1000-8000-00805F9B34FB
Properties: READ,NOTIFY

Descriptors:
Client Characteristic Configuration R
UUID: 0x2902

Connect to the device

Evaluating Bluetooth LE & Bluetooth

- Bluetooth LE & Bluetooth are the exciting technology that allows short range wireless communication. They have many valuable applications where wires are intrusive, expensive or difficult to install.
- Recent enhancements to Bluetooth have enabled streaming of high quality audio data and increased range, so the opportunities and applications for Bluetooth are continuously growing.
- Bluetooth Low Energy is intended for low power applications. It is interesting to note that some mbed enabled boards have this capability, and there is support information on the mbed web site

Chapter Review

- Wireless links exploit the characteristics of the electromagnetic spectrum, notably in radio, infra-red or visible light.
- A wide range of protocols and technologies exist to implement wireless links across personal, local, neighbourhood and wide area networks.
- Bluetooth is a complex yet effective protocol defined within the IEEE 802 group, which allows Bluetooth-enabled devices to connect and transfer data wirelessly, with potentially high data rates.
- The RN4020 module can be used to give an mbed Bluetooth LE capability.