# 第一章 蓝牙基础

蓝牙（ Bluetooth）：是一种无线技术标准，可实现固定设备、移动设备和楼宇个人域网之间的短距离数据交换（使用 2.4—2.485GHz 的 ISM 波段的 UHF 无线电波）。

## 1. 蓝牙的一个分类

### 1.1. 传统蓝牙

传统蓝牙是在之前的 1.0.1.2，2.0+EDR,2.1+EDR,3.0+EDR 等基础上发展和完善起来的。

传统蓝牙可以用与数据量比较大的传输，如语音，音乐，较高数据量传输等，低功耗蓝牙这样应用于实时性要求比较高，但是数据速率比较低的产品，如遥控类的，如鼠标，键盘，遥控鼠标(Air Mouse)，传感设备的数据发送，如心跳带，血压计，温度传感器等。传统蓝牙有 3 个功率级别，Class1,Class2,Class3,分别支持 100m,10m,1m 的传输距离，而低功耗蓝牙无功率级别，一般发送功率在 7dBm，一般在空旷距离，达到 20m 应该是没有问题的。

由于苹果对经典蓝牙数据传输接口有限制（需要过 MFI 认证），加上功耗偏大，因此在目前移动互联应用中慢慢地被淘汰。

### 1.2. 低功耗蓝牙

低功耗蓝牙是 Nokia 的 Wibree 标准上发展起来的。

蓝牙低能耗(BLE)技术是低成本、短距离、可互操作的鲁棒性无线技术，工作在免许可的 2.4GHz ISM 射频频段。它从一开始就设计为超低功耗(ULP)无线技术。它利用许多智能手段最大限度地降低功耗。

蓝牙低能耗架构共有两种芯片构成：单模芯片和双模芯片。蓝牙单模芯片可以和其它单模芯片及双模芯片通信，此时后者需要使用自身架构中的蓝牙低能耗技术部分进行收发数据。双模芯片也能与标准蓝牙技术及使用传统蓝牙架构的其它双模芯片通信。

**此处我们将以低功耗蓝牙为主。**

## 2. 蓝牙专业词汇（原文）

### 2.1. SIG

蓝牙技术联盟(Bluetooth Special Interest Group)是一家贸易协会，由电信、计算机、汽车制造、工业自动化和网络行业的领先厂商组成。该小组致力于推动**蓝牙无线技术**的发展，为短距离连接移动设备制定低成本的无线规范，并将其推向市场。

### 2.2. Profile

Bluetooth 的一个很重要特性，就是所有的 Bluetooth 产品都无须实现全部的 Bluetooth 规范。为了更容易的保持 Bluetooth 设备之间的兼容，Bluetooth 规范中定义了 Profile。Profile 定义了设备如何实现一种连接或者应用，你可以把 Profile 理解为连接层或者应用层协议规范。

蓝牙组织规定了一些标准的 profile，例如 HID OVER GATT ，防丢器 ，心率计等。每个 profile 中会包含多个 service，每个 service 代表从机的一种能力。

### 2.3. service

service 可以理解为一个服务。在 BLE 从机中，通过有多个服务，例如电量信息服务、系统信息服务等，每个 service 中又包含多个 characteristic 特征值。每个具体的

characteristic 特征值才是 BLE 通信的主题。比如当前的电量是 80%，所以会通过电量的 characteristic 特征值存在从机的 profile 里，这样主机就可以通过这个 characteristic 来读取 80%这个数据。

### 2.4. characteristic

Characteristic 特征值。BLE 主从机的通信均是通过 characteristic 来实现，可以理解为一个标签，一个属性，通过这个标签可以获取或者写入想要的内容。

### 2.5. decriptor

Decriptor 描述符。描述符就是描述 Characteristic 的，描述符有读写属性，描述符也可以被读写。

### 2.6. UUID

UUID，统一识别码，我们刚才提到的 service 和 characteristic，都需要一个唯一的 UUID 来标识。

SIG 定义 UUID 共用了一个基本 UUID：0x0000xxxx-0000-1000-8000-00805F9B34FB，总共 128 位。为了进一步简化基本 UUID，每一个 SIG 定义的属性有一个唯一的 16 位 UUID，以代替上面的基本 UUID 的‘x’部分。使用 16 位的 UUID 便于记忆和操作，例如 SIG 定义了“Device Information”的 16 位 UUID 为 0x180A。

## 3. 蓝牙的几个 Profile（原文）

在所有的 Profile 中，有四种是基本的 Profile，这些 Profile 会被其它的 Profile 使用，它们包括 GAP/SDAP/SPP/GOEP Profile。

### 3.1. GAP Profile

GAP Profile: Generic Access Profile，该 Profile 保证不同的 Bluetooth 产品可以互相发现对方并建立连接。GAP 规定的是一些一般性的运行任务。因此，它具有强制性，并作为所有其它蓝牙应用规范的**基础**。

### 3.2. SDAP Profile

SDAP Profile: Service Discovery Application Profile，通过该 Profile，一个 Bluetooth 设备可以找到其它 Bluetooth 设备提供的服务，以及查询相关的信息。

### 3.3. SPP Profile

全称 Serial Port Profile，定义了如何在两台 BT 设备之间建立虚拟串口并进行连接。例如，在两台电脑或者 Labtop 之间就可以建立这种连接。

### 3.4. GOEP Profile

GOEP Profile: Generic Object Exchange Profile，通用对象交换。这个 Profile 的名字有些费解，它定义的是数据的传输，包括同步，文件传输，或者推送其它的数据。可以理解为与内容无关的传输层协议，可以被任何应用用来传输自己定义的数据对象。

### 3.5. A2DP Profile

A2DP Profile 全名是 Advenced Audio Distribution Profile 蓝牙音频传输模型协定。

### 3.6. DUN Profile

DUN Profile 全称 Dial-up Networking (DUN) Profile，实现一台蓝牙设备通过另外一个带无线功能的蓝牙设备共享上网。

### 3.7. VRCP Profile

AVRCP（Audio/Video Remote Control Profile），也就是音频/视频远程控制配置文件。

### 3.8. HID Profile

HID 全称 Human Interface Device Profile，即人机接口设备 Profile.

## 4. 低功耗蓝牙

现在低功耗蓝牙（BLE）连接都是建立在 GATT (Generic Attribute Profile) 协议之上。GATT 是一个在蓝牙连接之上的发送和接收很短的数据段的通用规范，这些很短的数据段被称为属性（Attribute）。

GAP 使你的设备被其他设备可见，并决定了你的设备是否可以或者怎样与合同设备进行交互。例如 Beacon 设备就只是向外广播，不支持连接，小米手环就等设备就可以与中心设备连接。

### 4.1. 设备角色

GAP 给设备定义了若干角色，其中主要的两个是：外围设备（Peripheral）和中心设备（Central）。

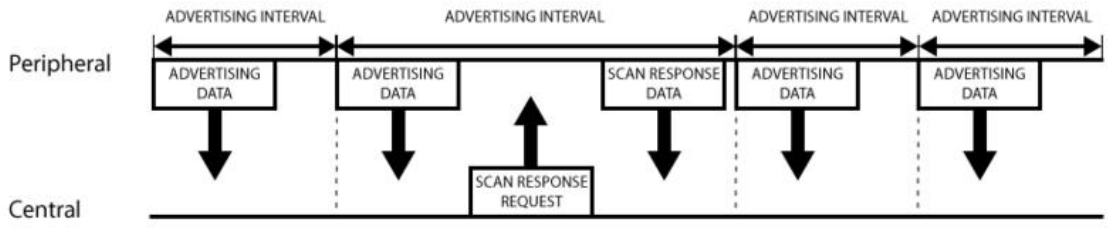
- 外围设备：这一般就是非常小或者简单的低功耗设备，用来提供数据，并连接到一个更加相对强大的中心设备。例如小米手环。
- 中心设备：中心设备相对比较强大，用来连接其他外围设备。例如手机等。

### 4.2. 广播数据

在 GAP 中外围设备通过两种方式向外广播数据： Advertising Data Payload（广播数据）和 Scan Response Data Payload（扫描回复），每种数据最长可以包含 31 byte。这里广播数据是必需的，因为外设必需不停的向外广播，让中心设备知道它的存在。扫描回复是可选的，中心设备可以向外设请求扫描回复，这里包含一些设备额外的信息，例如设备的名字。（广播的数据格式我将另外专门写一个篇博客来讲。）

### 4.3. 广播流程

GAP 的广播工作流程如下图所示。

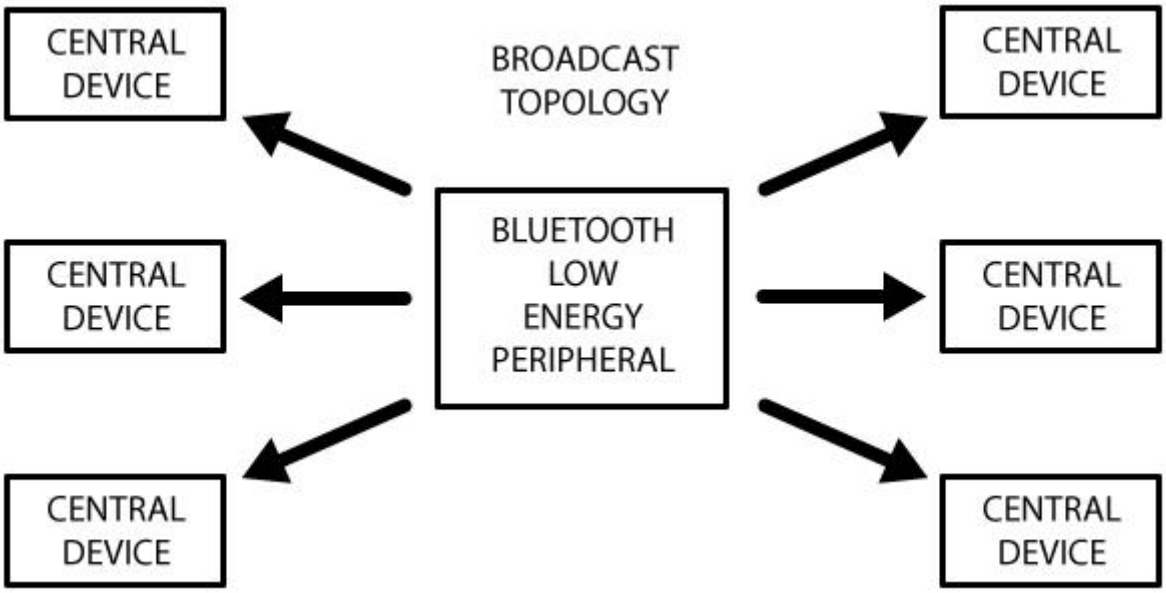



从图中我们可以清晰看出广播数据和扫描回复数据是怎么工作的。外围设备会设定一个广播间隔，每个广播间隔中，它会重新发送自己的广播数据。广播间隔越长，越省电，同时也不太容易扫描到。

### 4.4. 广播的网络拓扑结构

大部分情况下，外设通过广播自己来让中心设备发现自己，并建立 GATT 连接，从而进行更多的数据交换。也有些情况是不需要连接的，只要外设广播自己的数据即可。用这种方式主要目的是让外围设备，把自己的信息发送给多个中心设备。因为基于 GATT 连接的方式

的，只能是一个外设连接一个中心设备。使用广播这种方式最典型的应用就是苹果的 iBeacon。广播工作模式下的网络拓扑图如下：




## 4.5. GATT

GATT 的全名是 Generic Attribute Profile（普通属性协议），它定义两个 BLE 设备通过叫做 Service 和 Characteristic 的东西进行通信。GATT 就是使用了 ATT（Attribute Protocol）协议，ATT 协议把 Service, Characteristic 对应的数据保存在一个查找表中，查表使用 16 bit ID 作为每一项的索引。

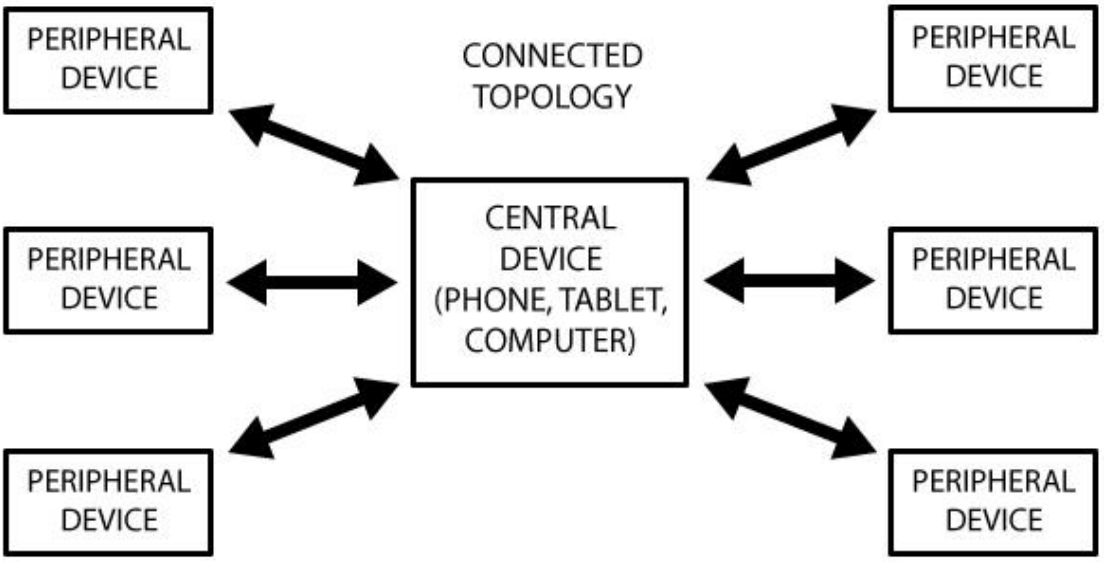
一旦两个设备建立起了连接，GATT 就开始起作用了，这也意味着，你必需完成前面的 GAP 协议。这里需要说明的是 **GATT 连接，必需先经过 GAP 协议**。实际上，我们在 Android 开发中，可以直接使用设备的 MAC 地址，发起连接，可以不经过扫描的步骤。这并不意味不需要经过 GAP，实际上在芯片级别已经给你做好了，蓝牙芯片发起连接，总是先扫描设备，扫描到了才会发起连接。

GATT 连接需要特别注意的是：GATT 连接是独占的。也就是一个 BLE 外设同时只能被一个中心设备连接。一旦外设被连接，它就会马上停止广播，这样它就对其他设备不可见了。当设备断开，它又开始广播。

**中心设备和外设需要双向通信的话，唯一的方式就是建立 GATT 连接。**

## 4.6. GATT 连接的网络拓扑

一个外设只能连接一个中心设备，而一个中心设备可以连接多个外设。一旦建立起了连接，通信就是双向的了，对比前面的 GAP 广播的网络拓扑，GAP 通信是单向的。如果你要让两个设备外设能通信，就只能通过中心设备中转。

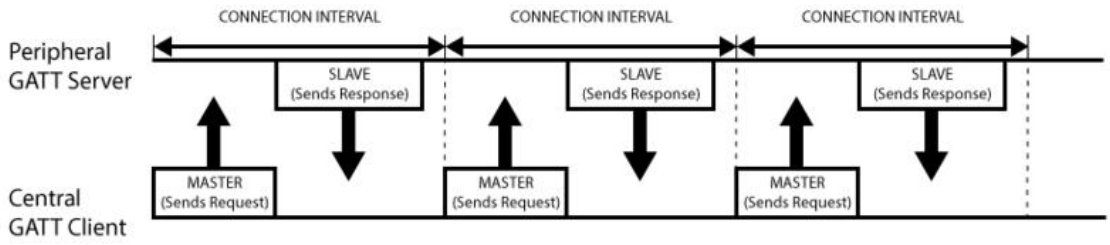



## 4.7. GATT 通信事务

GATT 通信的双方是 C/S 关系。外设作为 GATT 服务端（Server），它维持了 ATT 的查找表以及 service 和 characteristic 的定义。中心设备是 GATT 客户端（Client），它向 Server 发起请求。需要注意的是，所有的通信事件，都是由客户端（也叫主设备，Master）发起，并且接收服务端（也叫从设备，Slave）的响应。

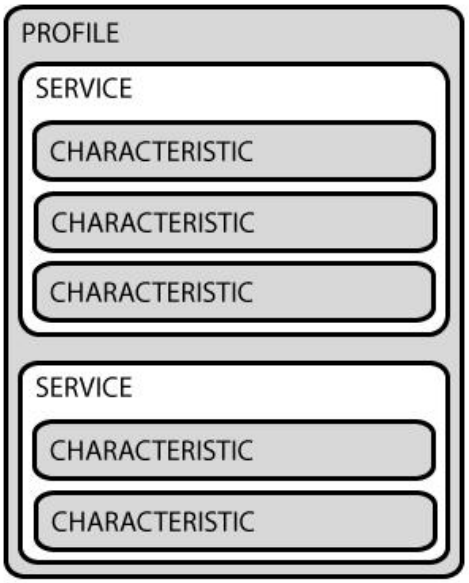
一旦连接建立，外设将会给中心设备建议一个连接间隔（Connection Interval），这样，中心设备就会在每个连接间隔尝试去重新连接，检查是否有新的数据。但是，这个连接间隔只是一个建议，你的中心设备可能并不会严格按照这个间隔来执行，例如你的中心设备正在忙于连接其他的外设，或者中心设备资源太忙。

下图展示一个外设（GATT 服务端）和中心设备（GATT 客户端）之间的数据交换流程，可以看到的是，每次都是主设备发起请求：




## 4.8. GATT 结构

GATT 事务是建立在嵌套的 Profiles, Services 和 Characteristics 之上的的，如下图所示：




- Profile

并不是实际存在于 BLE 外设上的，它只是一个被 Bluetooth SIG 或者外设设计者预先定义的 Service 的集合。例如心率 Profile（Heart Rate Profile）就是结合了 Heart Rate Service 和 Device Information Service。所有官方通过 GATT Profile 的列表可以从这里找到。

- Service

是把数据分成一个个的独立逻辑项，它包含一个或者多个 Characteristic。每个 Service 有一个 UUID 唯一标识。 UUID 有 16 bit 的，或者 128 bit 的。16 bit 的 UUID 是官方通过认证的，需要花钱购买，128 bit 是自定义的，这个就可以自己随便设置。

官方通过了一些标准 Service，完整列表在这里。以 心率 Heart Rate Service 为例，可以看到它的官方通过 16 bit UUID 是 0x180D，包含 3 个 Characteristic：Heart Rate Measurement, Body Sensor Location 和 Heart Rate Control Point，并且定义了只有第一个是必须的，它是可选实现的。

- Characteristic

在 GATT 事务中的最低界别的是 Characteristic，Characteristic 是最小的逻辑数据单元，当然它可能包含一个组关联的数据，例如加速度计的 X/Y/Z 三轴值。

与 Service 类似，每个 Characteristic 用 16 bit 或者 128 bit 的 UUID 唯一标识。你可以免费使用 Bluetooth SIG 官方定义的标准 Characteristic，使用官方定义的，可以确保 BLE 的软件和硬件能相互理解。当然，你可以自定义 Characteristic，这样的话，就只有你自己的软件和外设能够相互理解。

**举个例子**，心率 Heart Rate Measurement Characteristic，这是上面提到的 Heart Rate Service 必需实现的 Characteristic，它的 UUID 是 0x2A37。它的数据结构是，开始 8 bit 定义心率数据格式（是 UINT8 还是 UINT16？），接下来就是对应格式的实际心率数据。

实际上，和 BLE 外设打交道，主要是通过 Characteristic。你可以从 Characteristic 读取数据，也可以往 Characteristic 写数据。这样就实现了双向的通信。所以你可以自己实现一个类似串口（UART）的 Sevice，这个 Service

中包含两个 Characteristic，一个被配置只读的通道（RX），另一个配置为只写的通道（TX）。

## 5. 更多内容

**Bluetooth SIG 官方文档，如果想深入了解，可以精读。**

- 蓝牙核心协议文档
- Bluetooth Developer Portal
- 官方通过的 BLE Profile
- 官方通过的 BLE Service
- 官方通过的 BLE Characteristic

## 6. 总结

- 给大家推荐一本书《低功耗蓝牙权威指南》
- 教程源码地址：https://github.com/HX-IoT/