浅析 BLE 协议栈

温碧伟[1+]

[1](广东海洋大学 信息学院 电气工程及其自动化 1113 班)

The Analysis of BLE Protocol Stack

Biwei Wen[1+]

[1](Class Electrical engineering and its automation 1113, Information Insititute, GuangDong Ocean University)

+ Corresponding author: Phone: 13726934904,E-mail: 397337354@qq.com ,

http://www.gdou.edu.cn

Abstract: This passage mainly describes the structure of the BLE protocol stack and the key part of the BLE bluetooth development.

Key words: Advertisement; Connection; GAP; GATT

摘　要：本文主要是介绍 BLE 协议栈的结构以及 BLE 蓝牙开发的中关键部分。

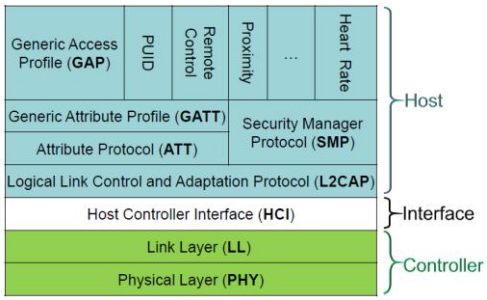
关键词：广播；连接；GAP；GATT

1 引言

BLE 蓝牙协议栈是由蓝牙技术联盟在蓝牙 4.0 的基础上推出的低功耗蓝牙协议，旨在占领低功耗设备与手持设备通信的市场。因此深入了解 BLE 协议栈还是很有必要的，特别对其通信和服务的配置等。本文旨在让初学蓝牙者对 BLE 协议栈结构和内容有一个大概的了解。

2 BLE 协议栈结构

BLE 协议栈结构入图一所示，主要有两成：Host(主机协议层)、Controller(控制协议层)。



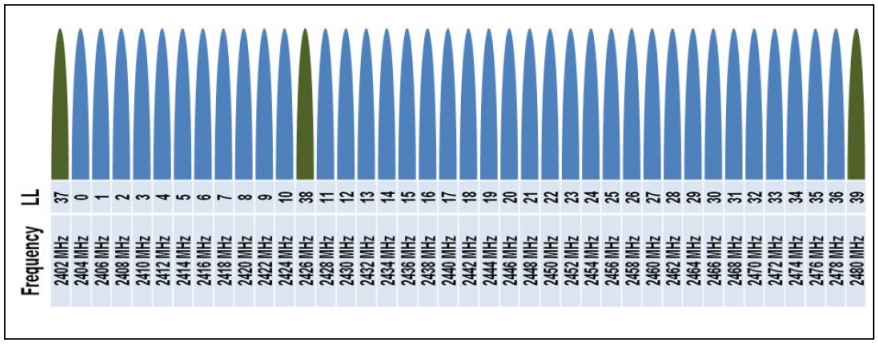
图一

3 控制协议层——Controller

控制协议层主要分为物理层（Physical Layer）和链接层(Link Layer)。

3.1 通道——Channels

一共 40 个通道，有 37 个数据通道用于两个连接的设备的通讯；3 个混合通道分别是 37、38、39，用于发现设备（Scanning devices）、初始化连接（initiating a connection）和广播数据（broadcasting date）。通道的具体频带分布如图二。



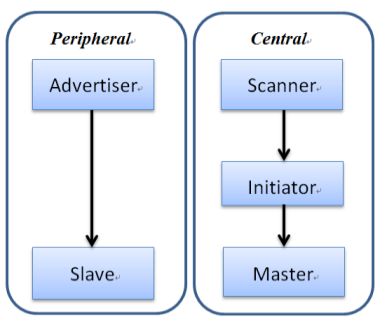
图二

3.2 链接层状态——Link Layer States

链接层状态有六个：

- 待机状态(Standby)：设备没有传输或接收任何数据，并且没有连接任何设备。
- 广播状态(Advertiser)：周期性地发射广播
- 扫描状态(Scanner)：主动寻找正在广播的设备
- 发起连接状态(Initiator):主动尝试开始和另一个设备连接
- 主设备(Master):作为主设备和另一个设备连接
- 从设备(Slave):作为从设备和另一个设备连接

两个设备建立连接过程中，链接层状态切换如下图：



图三

3.3 广播——Advertisement

(1) 广播事件类型——Advertisement Event Types

广播事件类型分为 4 种，分别是：

- 可连接无定向广播事件——Connectable Undirected Event
- 可连接定向广播事件——Connectable directed Event
- 不可连接无定向广播事件——Non-connectable Undirected Event

◆ 可扫描无定向广播事件——Scannable Undirected

不同的广播类型对扫描请求和连接请求的不同结果如下图：

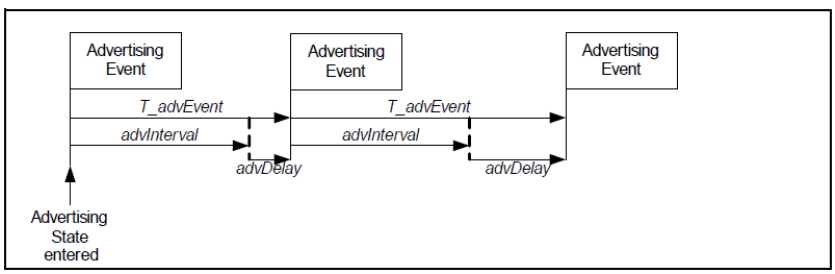
Advertising Event Type	PDU used in this advertising event type	Allowable response PDUs for advertising event	
		SCAN_REQ	CONNECT_REQ
Connectable Undirected Event	ADV_IND	YES	YES
Connectable Directed Event	ADV_DIRECT_IND	NO	YES*
Non-connectable Undirected Event	ADV_NONCONN_IND	NO	NO
Scannable Undirected Event	ADV_SCAN_IND	YES	NO

Table 4.1: Advertising event types, PDUs used and allowable response PDUs

* Only the correctly addressed initiator may respond.

图四

(2) 广播事件——Advertisement Events



图五

如图广播事件并不是连续的，而是有一个时间间隔——广播间隔（Advertisement Interval）。

在每次广播事件中，链路层会产生一个 0~10ms 的随机延迟，这个延迟时间被加到广播间隔中去，避免多器件的碰撞。一个广播事件的时间（T_advEvent）=广播间隔（advinterval）+广播延时（advDelay）。

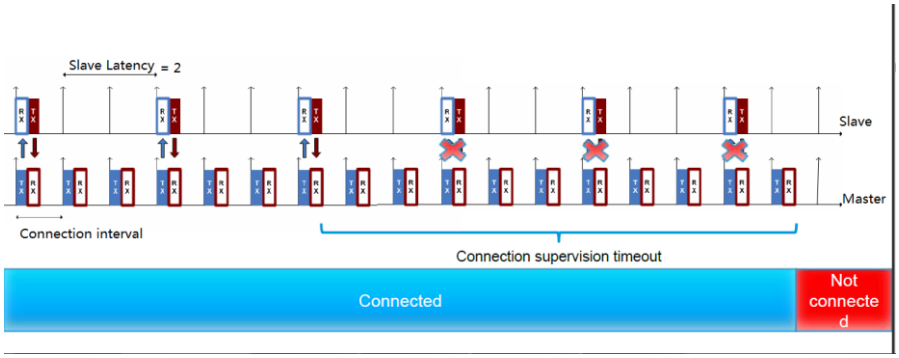
3.4 连接——Connection

(1) 连接初始化——Connection Initiation

在扫描设备扫描到一个可连接的广播消息后，扫描设备可以通过发送"Connection Request"数据包给广播设备，而成为(连接的)发起者连接参数——Connection Parameters。"Connection Request"包含从机设备一系列的链路层参数，这些参数声明连接时的从设备的MAC和通道以及时序要求。

如果广播设备接受了连接，那么这两个器件就进入了连接状态。而发起者就成为了主机，广播设备则成为了从机。

(2) 连接事件——Connection Events



图六

如图：两个设备建立连接后，空中的几个重要参数：

◆ 连接间隔——Connection interval

两个 Connection Event 之间的空闲值，单位：1.25ms，Range：7.5ms~4s。

◆ 从机（外设）延迟——Slave Latency

从机（外设）和主机建立连接后，主机会定期的向从机发送 request，而从机可以选择性的进行 response，这个 Slave Latency 就是跳过的主机发送的 request 次数，Range：0~499 次。

有效的连接间隙=连接间隔*(Slave Latency 次数+1)；注意：Connection interval*(最大 Slave Latency 次数+1) < 32s。

◆ 管理超时——Supervision Timeout

两个设备在连接的这段时间内没有发生通讯导致连接自动断开的时间，单位 10ms，Range：10ms~32s；Supervision Timeout 必须大于有效连接间隙，即 Supervision Timeout > Connection interval*(最大 Slave Latency 次数+1) < 32s。

4 主机协议层——Host

主机协议层包括的逻辑连接控制和适配层（L2CAP）、属性协议层（ATT）、通用属性配置层（GATT）、通用访问文件配置层（GAP）和安全协议层（SMP），本文主要介绍开发中常见的三个协议层 GATT、GAP、ATT 层。

4.1 通用访问配置文件概述——Generic Access Profile(GAP) Overview

GAP 层一共有四种设备角色可以配置：

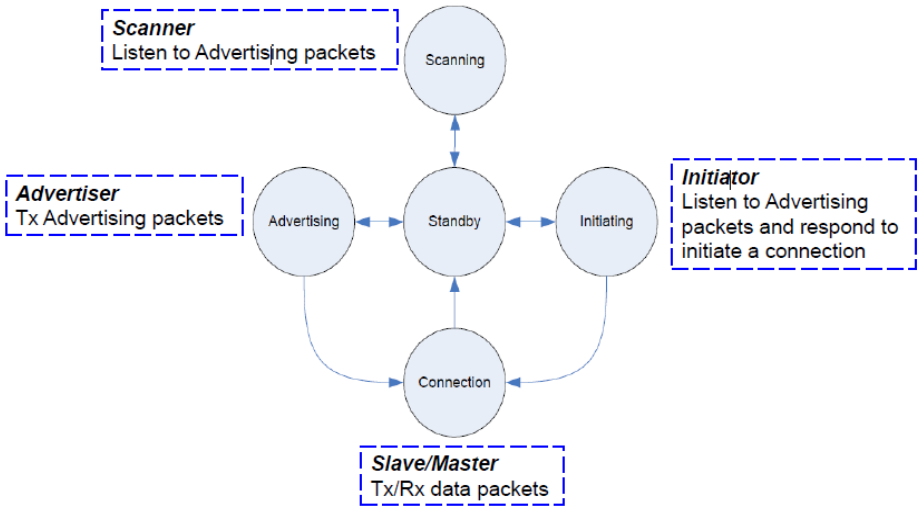
◆ 广播者(Broadcaster)：不可连接的广告设备;

◆ 观察者(Observer)：扫描广播，但不发起建立连接;

◆ 外设(Peripheral)：可连接的设备，可以在单个链路层中作为从机;

◆ 主机(Central)：扫描广告设备并发起连接，在单个链路层中作为主机，最多连接三个外设。

这几种角色的关系如下图：



图七

4.2 属性协议概述——Attribute Protocol(ATT) Overview

ATT 是离散值，与之相关的有以下三个属性：

◆ ATT handle——属性句柄，每个属性都有一个独一无二的 handle；

◆ ATT Type——属性类型，由 UUID 定义或用户自定义；

◆ A set of permission——一组权限，规定客户端对服务器属性的访问权限。

4.2.1 ATT 客户端与服务端结构——Client/Server Architecture

Server（服务端）有数据，Client（客户端）就是访问这些数据。

注：它与链接层的主/从设备是两个概念，也不同于 GAP 的外设和主机的角色。一个主设备既可以是客户端，也可以是服务端；从设备也是如此。不过主设备一般做 Client，从设备一般做 Server。

4.2.2 属性示例——Attribute Table Example

Handle	Type	Permissions	Value
39	0x2800 (GATT Service UUID)	Read	E0:FF(2 bytes)
40	0x2803 (GATT Characteristic UUID)	Read	10:29:00:E1:FF(5 bytes)
41	0xFFE1 (Simple Key State)	(none)	00 (1 byte)
42	0x2902 (GATT Client Characteristic Configuration UUID)	Read and Write	00:00 (2 bytes)

图八

4.3 Bluetooth 服务的 UUID

UUID 含义是通用唯一识别码——Universally Unique Identifier，这里的 UUID 主要是关于 Bluetooth 服务的方面标识，由 Bluetooth 技术联盟定义推出。

UUID 在表中有两种形式，一种是 16Bit 的，一种是 128Bit 的。其中 16Bit 在最终传输的时候也是 128Bit。UUID 由 Alias 和 Base 两部分。例如：UUID 为 0000xxxx-0000-1000-8000-00805F9B34FB，其中 xxxx 就是 Alias 部分，剩余部分为 Base 部分。

Bluetooth SIG定义的UUID就是16Bit的Alias部分，它们使用相同的Base部分

自定的UUID，关键是Alias部分在表中必须是增长的趋势。在决定服务的UUID之前，最好查看有没有Bluetooth SIG有没有定义好。

4.4 通用属性配置文件概述——Generic Attribute Profile(GATT) Overview

两个设备应用数据的通信时通过协议栈的 GATT 层实现。因为 GATT 层是 ATT 的配置层，所以 ATT 层的属性与 GATT 所说的属性“特性”是同一个概念。

4.4.1 GATT 客户端与服务端结构——Client/Server Architecture

从 GATT 角度来看，当两个设备建立连接后，他们处于以下角色之一：

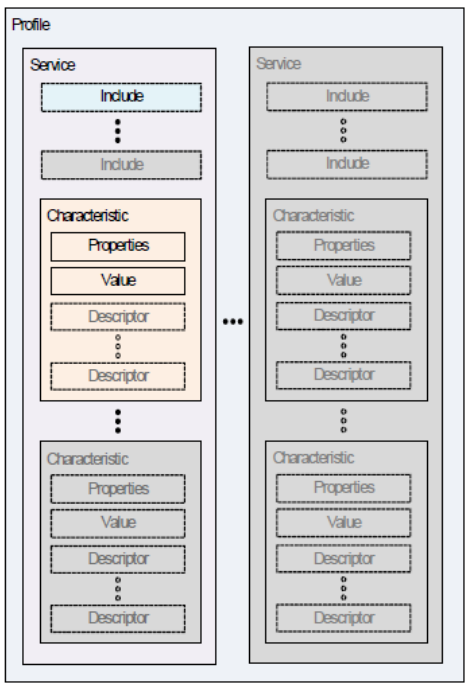
GATT 服务器——它是为 GATT 客户端提供数据服务的设备；

GATT 客户端——它是从 GATT 服务器读写应用数据的设备。

这个 GATT 客户端/服务器和 ATT 层的角色是一样的，两者必须相同。因为 GATT 是 ATT 层属性的定义者。

4.4.2 GATT 配置文件层次结构——GATT Profile Hierarchy

一个 GATT 服务器中可以包含多个 GATT 服务，这些服务可以是 GAP Service，GATT Service……，而要管理这些服务就需要用到 Profile，其层次结构如下图：



图九

4.4.3 服务端示例——Server Example

如上图可知，每一个服务都有很多属性组成，常见的有Include、Primary Service、Characteristic、Characteristic Format、Characteristic User Description、Client Characteristic Configuration Descriptor等组成。而在协议栈（程序中）就需要一张服务表来管理这些服务属性了。例如下表：

Handle	UUID (Type)	Value (Attribute Value)
0x0001	0x2800 (Service)	0x1800 (GAP Service)
0x0002	0x2803 (Characteristic)	{0x0A, 0x0003, 0x2A00}
0x0003	0x2A00 (Device Name)	"Example Device"
0x0010	0x2800 (Service)	0x1801 (GATT Service)
0x0100	0x2800 (Service)	0x180A (Battery State Service)
0x0101	0x2803 (Characteristic)	{0x02, 0x0102, 0x2A19}
0x0102	0x2A19 (Battery Level)	0x04
0x0200	0x2800 (Service)	0x5AB20001-B355-4D8A-96EF-2963812DD0B8 (Proprietary Thermometer Humidity Service)
0x0201	0x2803 (Characteristic)	{0x12, 0x0202, 0x5AB2FF01-B355-4D8A-96EF-2963812DD0B8}
0x0202	0x5AB2FF01-B355-4D8A-96EF-2963812DD0B8 (Proprietary Temperature Characteristic)	0x028A
0x0203	0x2904 (Characteristic Format)	{0x0E, 0xFE, «Celsius», «Outside»}
0x0204	0x2901 (Characteristic User Description)	"Outside Temperature"
0x0205	0x2902 (Client Characteristic Configuration Descriptor)	0x0000
0x0210	0x2803 (Characteristic)	{0x12, 0x0211, 0x5AB2FF02-B355-4D8A-96EF-2963812DD0B8}
0x0211	0x5AB2FF02-B355-4D8A-96EF-2963812DD0B8 (Proprietary Humidity Characteristic)	0x27
0x0212	0x2904 (Characteristic Format)	{0x04, 0x00, «Percent», «Outside»}
0x0213	0x2901 (Characteristic User Description)	"Outside Relative Humidity"
0x0214	0x2902 (Client Characteristic Configuration Descriptor)	0x0000

图十

可知，这张服务表由三个部分组成，分别是Handle、UUID（Attribute Type）、Value（Attribute Value）。

Handle的范围为0x0001 to 0xFFFF中的任意数，但是不能重复，并且在服务表中是处于增长的趋势，允许不连续增长，所以Handle的值在服务表中是独一无二的。为了方便管理，同一个服务的各个属性最好按照一定的间隔增长。从中可以得出GATT最多只能有65536个属性。

在一个服务的Handle区间，服务的属性类型就要以UUID（即universal unique identifier）来解释了。在每一个服务开始的Attribute Type必须是Primary Service属性或者是Secondary Service属性，接着就是Include或Characteristic了，再接下去就要根据不同的服务功能来分了。由Bluetooth SIG定义的UUID如下：

表1：

Attribute Type	UUID	Reference specification	Description
«Generic Access Profile»	0x1800	«蓝牙核心规格»第3卷C部分第12节	GATT服务
«Generic Attribute Profile»	0x1801	«蓝牙核心规格»第3卷G部分第7节	
«Primary Service»	0x2800	«蓝牙核心规格»第3卷G部分第3.1节	GATT属性类型
«Secondary Service»	0x2801	«蓝牙核心规格»第3卷G部分第3.1节	
«Include»	0x2802	«蓝牙核心规格»第3卷G部分第3.2节	
«Characteristic»	0x2803	«蓝牙核心规格»第3卷G部分第3.2节	
«Characteristic Extended Properties»	0x2900	«蓝牙核心规格»第3卷G部分第3.3.3.1节	GATT特征描述符
«Characteristic User Description»	0x2901	«蓝牙核心规格»第3卷G部分第3.3.3.2节	

«Client Characteristic Configuration»	0x2902	«蓝牙核心规格»第3卷G部分第3.3.3.3节	
«Server Characteristic Configuration»	0x2903	«蓝牙核心规格»第3卷G部分第3.3.3.4节	
«Characteristic Format»	0x2904	«蓝牙核心规格»第3卷G部分第3.3.3.5节	
«Characteristic Aggregate Format»	0x2905	«蓝牙核心规格»第3卷G部分第3.3.3.6节	
«Device Name»	0x2A00	«蓝牙核心规格»第3卷C部分第12.1节	GATT特征类型
«Appearance »	0x2A01	«蓝牙核心规格»第3卷C部分第12.2节	
«Peripheral Privacy Flag »	0x2A02	«蓝牙核心规格»第3卷C部分第12.3节	
«Reconnection Address »	0x2A03	«蓝牙核心规格»第3卷C部分第12.4节	
«Peripheral Preferred Connection Parameters»	0x2A04	«蓝牙核心规格»第3卷C部分第12.5节	
«Service Changed»	0x2A05	«蓝牙核心规格»第3卷G部分第7.1节	

4.4.4 GATT 服务的基本组成

一个基本的GATT服务在除了Handle之外，就包括GATT属性类型和服务内容两部分。而根据在不同的GATT服务中属性类型的Characteristic的详细程度，又可延伸出Characteristic Descriptor（特性描述符）和Characteristic Type（特性类型）。所以对于一个服务内容复杂性，可以根据Characteristic的详细程度来判断。

4.4.5 GATT 属性类型

由上表可以知道GATT属性类型包括«Primary Service»、«Secondary Service»、«Include»、«Characteristic»四类。这里简单介绍常用的«Primary Service»和«Characteristic»两个

（1）Primary Service 格式

Primary Service 的值就是向 Client 端说明这是一个什么服务，也就是一个服务名字，这个名字也是 16Bit 的 UUID。Bluetooth SIG 定义两个通用的服务 GAP 和 GATT 服务。这个值是可以自定的。但是经过查找资料发现，要想自己服务具有通用性，被其他蓝牙设备识别，是要向 Bluetooth SIG 提交申请的。Bluetooth SIG 已经通过了很多服务名字的 UUID，在定义这个值的时候最好查询一下，看看是否存在。

（2）Characteristic 格式

这是GATT最重要的一个属性类型了。它包含了三个部分：Properties、Handle、UUID。Properties描述的是服务性质，具有作用。其常用值如下表：

Properties	Value
Broadcast	0x01
Read	0x02
Write Without Response	0x04
Write	0x08
Notify	0x10
Indicate	0x20
Authenticated Signed Writes	0x40
Extended Properties	0x80

图十一

这个值可以是表中的性质的叠加，比如一个服务具有Notify和Read性质，那这个就是0x10+0x02=0x12。

Handle 就是服务内容在 GATT 服务表中具体的值。这里的 UUID 就是服务内容在 GATT 服务表中具体的 UUID。可以自定义（按照 UUID 格式），也可以是 Bluetooth SIG 规范中的。

4.4.6 GATT 定义的 GATT 服务器与客户端之间的通信子过程

在蓝牙核心规范文档中，一共定义了 11 个 features，每个 feature 对应一种或多种 Sub-procedures。这 11 个 features 与对应的 Sub-procedures 如下表：

No.	Feature	Sub-Procedures	Support in Client	Support in Server
1	Server Configuration	Exchange MTU	O	O
2	Primary Service Discovery	Discover All Primary Services Discover Primary Services By Service UUID	O O	M M
3	Relationship Discovery	Find Included Services	O	M
4	Characteristic Discovery	Discover All Characteristic of a Service Discover Characteristic by UUID	O O	M M
5	Characteristic Descriptor Discovery	Discover All Characteristic Descriptors	O	M
6	Characteristic Value Read	Read Characteristic Value Read Using Characteristic UUID Read Long Characteristic Values Read Multiple Characteristic Values	O O O O	M M O O
7	Characteristic Value Write	Write Without Response Signed Write Without Response Write Characteristic Value Write Long Characteristic Values Characteristic Value Reliable Writes	O O O O O	C1 O C2 O O
8	Characteristic Value Notification	Notifications	O	O
9	Characteristic Value Indication	Indications	M	C3
10	Characteristic Descriptor Value Read	Read Characteristic Descriptors Read Long Characteristic Descriptors	O O	O O
11	Characteristic Descriptor Value Write	Write Characteristic Descriptors Write Long Characteristic Descriptors	O O	O O

O: 'O' for optional to support (used for capabilities that can be used in the profile);
M: 'M' for mandatory to support (used for capabilities that shall be used in the profile);
C1: Write Without Response is mandatory if Signed Write Without Response is supported otherwise optional
C2: Write Characteristic Value is mandatory if Write Long Characteristic Values is supported otherwise optional
C3: If *Service Change Characteristic* is present, this feature is mandatory, otherwise optional.

5 总结

本文只是对蓝牙协议栈有一个深入的理解，主要是蓝牙的角色和建立连接时的状态切换，以及 GATT 协议层中 Profile 的服务格式，这是在蓝牙 BLE 通信的核心，因为 BLE 数据传输主要通过读取服务中的 Characteristic 值。限于篇幅，文章进行了压缩， 部分内容未能详细叙述，而且并未介绍 BLE 协议栈的实例开发。如果有需要，可以去 TI 公司官网下载 CC2540 的开发 Demo，或者联系笔者索要具体开发实例和资料。

参考文献：

[1] http://developer.bluetooth.cn/

[2] http://www.bluetooth.com/

[3] www.ti.com/blestack

[4] Specification of the Bluetooth System 4.0.pdf

温碧伟(1991-)广东河源人，广东海洋大学，信息学院，电气工程及其自动化 1113 班，兴趣方向为嵌入式系统开发，自动化控制