

# BLE 简介（利用）

## 简介

本研讨会旨在利用 ARM® mbed™ 工具帮助大家熟悉 BLE（低功耗蓝牙）。如果您还不是很熟悉这些工具，请查看 [mbed 入门](#) 文档。

在本研讨会中，我们将从产品开发的角度的了解 BLE。为此，我们将论述现有标准、BLE 工作原理以及最大程度利用 BLE 系统的方式。因此，我们将略过许多低级别接口，这些接口只会由协议栈开发者看到，并且与应用层设计无关。相反地，我们将关注经过 GATT/GAP 和 SIG 许可的配置文件和服务。

参加本课程的前提条件是具备一台支持 BLE 的智能手机和一个 [支持 mbed 的 BLE 开发平台](#)。

最好以数字文档的形式查看本文档。此处为本文档的永久链接，方便您离线查看：

<http://goo.gl/VR2rki>

目录：

[简介](#)

[什么是 BLE？](#)

[BLE 堆栈](#)

[GAP 通用访问配置文件](#)

[GATT 通用属性配置文件](#)

[BLE 状态：广播与连接](#)

[物理层](#)

[BLE 数据包布局](#)

[GAP：广播有效负载 PDU](#)

[GATT：配置文件、服务和](#)

[特性](#)

[UUID 的](#)

[开发和测试](#)

[BLE 应用](#)

[示例](#)

[GAP：URIBeacon](#)

[GATT：心率配置文件](#)

[Evotings](#)

[示例：BLE Discovery](#)

[向前迈进](#)

[其他工具](#)

## 什么是 BLE？

低功耗蓝牙（亦称 BLE 或 Bluetooth Smart）是 Bluetooth v4.x 规格的一部分。BLE 不兼容传统蓝牙。它是一种低功耗、短距离的通信协议，是移动和嵌入式设备的理想之选。拥有不常用的较小数据包且最大比特率大约为 30KB/s 便可实现低功耗方面的特性。因此，不建议将 BLE 用于视频或音频流处理等高吞吐量应用。BLE 应用在纽扣电池上运行长达 1 年以上的现象很常见。

## BLE 协议栈

BLE 堆栈拥有多个层。但是，在本教程中，我们将仅论述最顶端的层：GATT 和 GAP。只有当您是堆栈开发者时，才需详细了解上述层下的其他层。如果是这种情况，我建议您阅读核心蓝牙规格。

**GAP：**通用访问配置文件

GAP 层负责执行多个控制功能。诸如安全性、连接管理和广播等功能均属 GAP 层的一部分。

**GATT：**通用属性配置文件

GATT 层负责执行数据交换和数据组织。GATT 层使用属性协议 (ATT) 作为在各设备间交换数据的通道。

## BLE 状态：广播广播与连接

BLE 在两种基本状态下运行：广播和连接。如果您查看 BLE 规格，实际上会发现每个用例（广播程序、启动程序、传播程序等）都有多个带有特殊名称的子模式，但是所有用例总体上可划分为上述两种状态。

在广播模式下，广播程序向正在侦听的任意扫描程序广播数据，有时有连接意向，而有时并没有建立连接的意向。它只是单纯地向任何要查找数据的用户广播，而不能保证数据正在传输。这就是一对多传输。广播模式使用 BLE 堆栈的 GAP 层。

在连接模式下，两台 BLE 设备会直接互连。在该连接状态下，服务器向客户端提供数据。术语“客户端”和“服务器”用于表示定向数据流，正因如此，这两台设备在双向信息流过程中将作为客户端和服务器。在连接状态的信息传输过程中，可以保证数据将被发送并通过校验和进行验证。这就是一对一传输。连接模式使用 BLE 堆栈的 GATT 层。

## 物理层

BLE 无线电使用 2.4 GHz 频带并且在频谱范围内将其划分为 40 个信道 (2.4000-2.4835 GHz)，包含 37 个数据传输通道和 3 个广播通道。正如下图所示，三个橙色通道用于进行广播，而蓝色通道则用于数据传输。

图 3: 频率层来源

BLE 使用跳频方案，其中无线电将定期按照指定的通道数跳跃，从而避免附近无线电之间出现冲突。这将支持 BLE 在群集的 2.4Ghz 频带下有效地广播数据。

## BLE 数据包布局

所有 BLE 数据包至多为 47 个字节。但是，这稍具欺骗性，因为在每次传输过程中，并不存在 47 个字节的用户可访问信息。相反，每个 BLE 堆栈层均会对这 47 个字节进行削减，并且通常情况下最终仅会为用户提供 20 个字节的可用信息。完整的 47B 布局如下所示。



PDU 节细分为两个字节标题和类型特定的有效负载。一些可用的有效负载类型如下所示：

*广播广播通道 PDU (CH 37,38,39):*

- 广播广播广播数据包 (GAP)
- 扫描程序请求/响应有效负载
- 连接启动有效负载

*数据通道 PDU (CH 0-36):*

- 链路层控制有效负载
- 数据有效负载
  - L2CAP 信号通道
  - 安全管理器协议
  - 属性协议 (GATT)

## GAP: 广播广播有效负载 PDU

广播广播有效负载用于广播 BLE 设备的相关信息。让我们深入了解下用于“广播广播有效负载”类型的 PDU。

39 个字节的广播有效负载 PDU 将缩减至 31B 的广播数据节。这一 31B 节包含多个广播结构，每个结构传输一个特定信息块。只要所有结构的总大小不超过 31B，便可以存在任意数量的 AD 结构。值得注意的是，每个 AD 结构拥有与其相关的 2B 开销。

有关可以放入“AD 类型”字段的内容的完整列表，请参阅 [SIG GAP 页面](#)。从技术层面来讲，AD 类型的长度为 N；但实际上，它的长度始终为 1B。以下简要列举了最常用的 AD 类型。

广播有效负载 AD 类型：

- 0xFF 制造商特定数据
- 0x01 关于可连接性的标记
- 0x08 缩短的本地名称
- 0x0A 发射功率电平
- [等等...](#)

几乎每个广播数据包都有 3 个字节被标记结构所占用，详细说明广播的广播模式。这就为带有 2B 开销的 26B 用户数据留下了余地。此空间的常见用途为，将自定义数据格式转换为“制造商特定数据”字段。这一用途见诸 [iBeacon](#) 和 [AltBeacon](#) 等技术中。

请记住，在“广播数据”节中传输数据并不保证送达；缺少 10-30% 的数据包并不少见。对于温度等数据而言，这并不是问题；但是对于加速计等实时数据而言，这会成为一个问题。

## GATT: 配置文件、服务和特性

GATT 层负责传输数据。还存在数据有效负载 PDU，但是我们不会对其进行深入研究，因为 GATT 的实际数据大小对于开发者而言并不重要。这是因为 GATT 层中的数据是按要求发送，而非一次性全部发送。这意味着只需发送请求的数据，其他数据则无需发送。

GATT 层将数据组织到一个服务和特性层级结构中。

服务拥有一个或多个特性。特性拥有一个小于 512B 的值，并且可以拥有 0 个或多个描述符。特性还拥有一个标题，用于定义特性值的读取/写入/通知。

理论上而言，描述符对于发现特性表示的内容十分有用，但是实际上，它们很少使用并且通常会被完全忽略。更为常见的是，用户定义自定义有效负载信息，并且只会发布其规格，而不会尝试使用描述符进行描述。对于已明确界定的服务（如心率监护仪），特性值已被广泛熟知，并且不包含描述符。

每个服务和特性均拥有 UUID（有关详情，请参阅下一部分）。

配置文件是另一层的抽象化。它们不是 GATT 技术规格的一部分，但是对于概念组织十分有用。

配置文件是一组服务，正如服务是一组特性一样。SIG 也会提供明确界定的配置文件。

其中一个很好的示例为心率监护仪配置文件。心率监护仪配置文件的结构如下：

S1: 心率服务

c1: 心率测量

c2: 传感器位置

S2: 设备信息服务

c1: 制造商名称

c2: 型号

c3: 修订号 c(n): 等

## UUID

UUID 代表通用唯一标识。在 BLE 中，一切都有 UUID。有两种大小的 UUID：2B/16 位（适用于向 SIG 付款的用户）和 16B/128 位（适用于自行提供的用户）。2B/16 位 UUID 实际上只是一个骗局；前两个字节用于服务/特性，而剩余的 14 个字节对于每个 SIG UUID 都相同的，因此可以隐含填充。

每个服务、特性和描述符均拥有一个 UUID。但是配置文件没有。UUID 用于描述 BLE 事务中的一切，因为它们进行压缩的效率远高于对英文描述进行压缩的效率。这使 BLE 调试看起来像是个巨大的 UUID 查找游戏。在大多数服务中，至少有一个特性将是可选的，这意味着它可以包括在内，也可以不包括在内。这会导致无法使用特性索引来解读其含义，因此改用 UUID 来定义具体的特性。

有关 UUID 的完整列表，请参见 [GATT 规格选项卡](http://developer.bluetooth.org)，网址为 [developer.bluetooth.org](http://developer.bluetooth.org)。

- [服务](#)
- [特性](#)
- [描述符](#)

## 开发和测试

要开发和测试 BLE 应用，您将需要两件事物：客户端和服务端。我们应用中内置的服务器为 Nordic Semiconductor 的 [nRF51822 mkit](#)，但是任何内置有 BLE 的支持 [mbed](#) 的平台也将适用。对于客户端而言，您需要一台内置有 BLE 应用且支持 BLE 的智能手机。

我们将继续关注心率配置文件，因为它相对而言比较简单且受到广泛支持。

### BLE 应用

由于 BLE 拥有标准服务和特性，因此任意符合标准的 BLE 应用均可以用于与基于标准的设备进行通信。快速搜索应用商店应该就能找到大量适合的应用。一些尤为知名的应用为 nRF Master Control Panel、nRF Toolbox for Android 和 LightBlue for iOS，后者还拥有一个能够在 OSX 上运行的配套应用。

**nRF Master Control Panel** 可以很好地显示您周围的所有 BLE 设备，并且能够按信号强度进行组织。

**LightBlue** 带有一个简单易用的界面，以及一个很好的“UUID-至-英文”转换器。



## 示例

我们将介绍两个示例，一个用于 GAP，一个用于 GATT。请确保您已将内置有 BLE 且支持 mbed 的平台添加至编译器。如果您忘记了如何执行此操作，请参阅之前的文档 [mbed 入门](#)。

### GAP: URIBeacon

让我们先查看下 [URIBeacons](#)，它属于[物理 Web](#) 项目的一部分。URIBeacons 使用 BLE 广播重新定向至网站或其他 URI 的短 URL。从最基本的形式来看，您可以将其视为基于 BLE 的 QR 代码。URIBeacons 使用 GAP 广播数据包来广播 URL。

方法为：

1. 下载适用于 [iOS](#) 或 [Android](#) 的物理 Web 应用。
2. 将 mbed [URIBeacon 程序](#) 导入您的编译器。
3. 编译该程序，然后将其加载至开发板。
4. 打开物理 Web 应用后，您应当看到针对“<http://www.mbed.org>”的 URIBeacon。物理 Web 应用还将从互联网上抓取大量关于该 URL 的信息。

现在，尝试更改 URI。请记住，URI 的最大长度为 17 个字节。编译代码，然后将其加载至开发板。完成编程后，务必按下“重置”。使用物理 Web 应用来查看您的信标。

扫描 URIBeacons 的物理 Web 应用 显示信标的物理 Web 应用

URIBeacon 是广播数据包 PDU 的特殊用途。具体而言，它使用广播数据字段的制造商特定数据字段。URIBeacon 的规格如下所示。

请注意 ADLenth 和 ADType 字段占用的前两个字节，正如任何其他广播数据字段一样，您实际上只有初始 31 个字节中的 26 个可以使用。URIBeacon 规格随后将 26 个字节中的 8 个字节用于开销，从而将 17 个字节的空间留给 URI。如果您查阅[规格](#)，您将发现前缀（'www.'、'http://'）和后缀（'.com/'、'.org/'）均已减至一个字节，每个字节均是规格开销的一部分。

## GATT: 心率配置文件

为演示 GATT 连接，我们将使用 mbed [BLE HeartRate](#) 示例。

1. 请将该示例添加至编译器并对其进行编译，确保程序不含任何错误。
2. 更改设备名称，以便能够轻松识别。

将下列行更改为您能够轻松识别的内容：

3. 打开您选择的 BLE 扫描仪应用，然后连接至您的设备。
- 提示：建议使用 nRF Toolbox 应用，但是所有符合标准的 BLE 应用应该都可行。

请注意，iOS 会本地缓存名称，因此如果您事先已连接至 BLE 设备，则可能会缓存旧名称并且您将不会看到新名称。

连接成功后，您应当能够查看心率服务的心率测量特性。值会呈上升趋势（从 100 上升至 175），然后重置。

尝试修改 mbed 代码，使其能够以 5 的倍数而非 1 的倍数增长。

## Evthings

Evthings 是一个开源项目，可以为 Cordova 智能手机平台提供 BLE 插件。它支持您使用标准 BLE API 在 JavaScript 中创建与硬件无关的应用。这意味着，您可以在 Evthings 中写入应用，然后在 iOS 或 Android 上运行，并且只会在平台切换时产生少量开销或者不产生开销。

Evthings 拥有可以在 OSX、Windows 和 Linux 上运行的方便的可执行程序，适用于快速进行原型设计和开发。

1. 从相应网站上下载 [Evthings Studio Workbench](#)。
2. 从应用商店（[iOS](#) 和 [Android](#)）下载 Evthings Client 应用。
3. 将软件包解压缩至您偏好的文件夹中，然后启动 Evthings Workbench。

图 6: 桌面上的 Evothings Workbench

4. 打开手机上的客户端应用，它将自动开始扫描已打开 Workbench 的计算机。
5. 当客户端识别工作站后，它将连接至该工作站。
6. 现在，您可以单击应用列表旁的**运行**按钮，在 Workbench 上运行任何应用。
7. 您还可以利用现有项目，然后对其进行修改。所有内容均出现在示例文件夹中。该应用只是适用于代码的 JavaScript 文件和适用于 GUI 的 index.html 文件。

### 示例：BLE Discovery

BLE Discovery 程序是一款很好入门的应用。它提供连接至任何附近 BLE 设备并探索其服务和特性的选项。您可以使用此应用连接至您的设备，然后仔细查看它显示的 UUID 是否符合您的预期。

例如，**0x1800** 是指通用访问服务，而 **0x180F** 则是指电池服务。如果您感兴趣，可以在[此处](#)找到完整的服务列表。

如果您对创建自定义 UUID 感兴趣，则可以在[此处](#)找到最佳实践。

图 7: 智能手机上的 BLE Discovery 应用

## 向前迈进

希望您目前对于如何构建 BLE 堆栈有了一个更好的了解。如果您希望在 `mbed` 上探索更多示例程序，则可以在[此处](#)找到所有官方示例。下载一些示例，然后逐个尝试，看看您是否已了解代码的含义，然后试着单独行动。记住 - 如果您有任何问题，您始终可以在[问题论坛](#)上发布以联系我们的 `mbed/BLE` 专家。祝您编码顺利！

## 其他工具

其他平台也会提供大量用于 BLE 开发的工具。以下简要列举了一些目前出现的最佳工具。

### 智能设备

- 软件: nRF Master Control Panel、nRF ToolBox 和 BLE Explorer 等。
- 硬件: 内置。

### OSX

- 软件: lightBlue - 不错的可视化工具，但是它使用 UUID，而非人类可读名称拥有适用于 iOS 的对等软件。XCode 拥有一些不错的插件。
- 硬件: USB 适配器或内置。

## Windows

- 软件：Wireshark 或 TI 的工具集（需要 CC215x BLE 适配器）。上述两种选项均非常耗费资源。Windows 没有本地 BLE API。
- 硬件：USB 适配器或内置。

## Linux

- 软件：BlueZ hciconfig 和 gatttool。全是命令行，并不十分适合初学者，但是它很有用并且能够轻松进行脚本制作/扩展。此处是一个[不错的教程](#)。
- 硬件：USB 适配器或内置。

## 其他

Python 资料库提供大量 BLE 串口设备。它们可以插入计算机，并且能够通过命令行进行控制。

- [BlueGiga](#)
- LightBlue Bean
- [iteadStudio](#)