

蓝牙基础知识

目录

- 1. 蓝牙基础知识 2
 - 1.1 蓝牙起源 2
 - 1.2 蓝牙发展历史 3
 - 1.3 蓝牙分类 5
 - 1.3.1 传统蓝牙 6
 - 1.3.2 BLE 7
 - 1.4 蓝牙 profile 7
- 2. 蓝牙物理特性 10

1. 蓝牙基础知识

1.1 蓝牙起源

蓝牙的历史实际上要追溯到第二次世界大战。蓝牙的核心是短距离无线电通讯，它的基础来自于跳频扩频（FHSS）技术，由好莱坞女演员 Hedy Lamarr 和钢琴家 George Antheil 在 1942 年 8 月申请的专利上提出。他们从钢琴的按键数量上得到启发，通过使用 88 种不同载波频率的无线电控制鱼雷，由于传输频率是不断跳变的，因此具有一定的保密能力和抗干扰能力。

起初该项技术并没有引起美国军方的重视，直到 20 世纪 80 年代才被军方用于战场上的无线通讯系统，跳频扩频（FHSS）技术后来在解决包括蓝牙、WiFi、3G 移动通讯系统在无线数据收发问题上发挥着关键作用。

蓝牙技术开始于爱立信在 1994 年创制的方案，该方案旨在研究移动电话和其他配件间进行低功耗、低成本无线通信连接的方法。发明者希望为设备间的无线通讯创造一组统一规则（标准化协议），以解决用户间互不兼容的移动电子设备的通信问题，用于替代 RS-232 串口通讯标准。

爱立信发现，解决兼容问题的方法是将各种不同的通信设备通过移动电话接入到蜂窝网上，而这种连接的最后一段就是短距离的无线连接。随着项目的进展，爱立信把大量资源投入到短距离无线通讯技术的研发上。

1998 年 5 月 20 日，爱立信联合 IBM、英特尔、诺基亚及东芝公司等 5 家著名厂商成立「特别兴趣小组」（Special Interest Group, SIG），即蓝牙技术联盟的前身，目标是开发一个成本低、效益高、可以在短距离范围内随意无线连接的蓝牙技术标准。当年蓝牙推出 0.7 规格，支持 Baseband 与 LMP（Link Manager Protocol）通讯协定两部分。

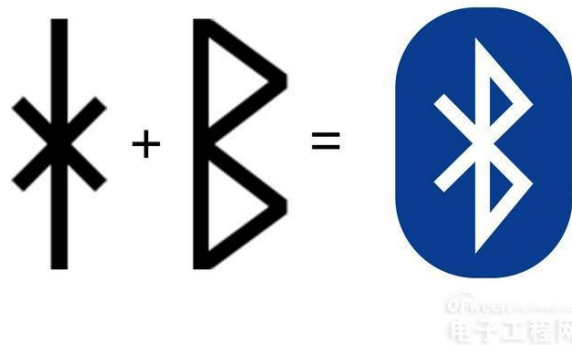
1999 年先后推出 0.8 版、0.9 版、1.0 Draft 版。完成了 SDP（Service Discovery Protocol）协定和 TCS（Telephony Control Specification）协定。

1999 年 7 月 26 日正式公布 1.0A 版，确定使用 2.4GHz 频段。和当时流行的红外线技术相比，蓝牙有着更高的传输速度，而且不需要像红外线那样进行接口对接口的连接，所有蓝牙设备基本上只要在有效通讯范围内使用，就可以进行随时连接。

1999 年下半年，微软、摩托罗拉、三星、朗讯与蓝牙特别小组的五家公司共同发起成立了蓝牙技术推广组织，从而在全球范围内掀起了一股「蓝牙」热潮。

到 2000 年 4 月，SIG 的成员数已超过 1500，其成长速度超过任何其他的无线联盟。

蓝牙为 bluetooth 的中文直译，而 bluetooth 又是斯堪的纳维亚语 Blåtand 的英语化，这个名字来源于中世纪丹麦国王 Harald Blåtand Gormsson（哈拉尔蓝牙王），这位国王统一了丹麦，虽然历史伟人与现代科技技术无关，但国王的成就影响到蓝牙技术开发者 Jim Kardach 的理念，他希望蓝牙也可以成为统一的通用传输标准。蓝牙的 logo 来自后弗萨克文的符文组合，代表 H 和 B，也就是哈拉尔蓝牙王名字的首字母。



1.2 蓝牙发展历史

蓝牙从 1998 年发布蓝牙 1.0 到现在，已发展到 Bluetooth 5.0，大体上的进步表现在传输速度、稳定性、安全性、交换协议、功耗表现等提升或改进。



年份	版本	主要特点
1999	1.0	1Mbps传输速度
2001	1.1	列入IEEE 802.15标准
2004	2.0	EDR技术、3Mbps速度
2007	2.1	优化配对、功耗表现
2009	3.0	AMP交替射频、24Mbps速度
2010	4.0	低功率
2014	4.1	批量数据交换、4G共存
2016	5.0	更快的传输速度、范围

A watermark '电子工程网' is visible at the bottom right.

蓝牙 1.x

蓝牙 1.0 在 1999 年提出，经历 1.0 Draft、1.0 和 1.0B 三个版本。蓝牙 1.0 Draft 完成了 SDP 服务发现协议和 TCS 电话控制协议，到蓝牙 1.0 为正式公布的版本，确定使用 2.4GHz 频段，理论传输速度为 1Mbps，实际传输速度最高为 721Kbps（约 90KB/s），随后的 1.0B 增加对 WAP 无线应用协议的支持。

到 2001 年，蓝牙 1.1 被列入 IEEE 标准，批准为 IEEE 802.15.1，蓝牙 1.1 修正了 1.0B 的错误，加入信号强度指示。蓝牙 1.2 为 802.15.1a，改善了抗干扰和连接能力，可向下兼容蓝牙 1.1。

蓝牙 1.x 只能单工方式工作，而传输的音频频率响应也不高，所以当时的蓝牙耳机大多都是单声道单耳式，音质也较差。如果要流媒体薄立体声音频，则需要 A2DP 技术的支持。

蓝牙 2.x

在 2004 年推出的蓝牙 2.0+EDR，最大改进便是加入 EDR（Enhanced Data Rate）技术，加快了传输速率，理论传输速度为 3Mbps（375KB/s），实际为 2.1Mbps（262.5KB/s），而且 EDR 技术还可以通过减少工作负载循环来降低功耗。EDR 是个可选技术，蓝牙 2.0 本身的改进是支持到双工方式，即可以语音的同时还可以传输数据。

3 年后的 2007 年，蓝牙 2.1+EDR 推出，主要亮点改善配对过程，不再需要用户输入配对码，而是自动提供配对码进行配对确认。另外有 SSP（Secure Simple Pairing），提高配对的安全性。还有 Sniff Subrating 技术，可以延长两个设备之间确认讯号的发送间隔时长，以降低功耗。

蓝牙 3.x

蓝牙 3.0+HS 在 2009 年发布，HS 是 High Speed，理论传输速度为 24Mbps（3MB/s），这其实并不是蓝牙本身的速度，而是来自 IEEE 802.11，也就是 Wi-Fi，蓝牙 3.0 的重要技术在 AMP（Alternative MAC/PHY，交替射频），该技术可以让蓝牙 3.0 在需要大量和高速的数据传输时调用 Wi-Fi 来作为传输连接。

如果设备只有蓝牙而没有 Wi-Fi，或是没有“HS”后缀的蓝牙 3.0，那么传输其实还是蓝牙 2.0 的速度。蓝牙 3.0 本身加入 L2CAP 增强模式、单播无连接数据和 EPC（增强电源控制）机制。

蓝牙 4.x

在 2010 年提出的蓝牙 4.0 包括了 Classic Bluetooth、Bluetooth High Speed 和 Bluetooth Low Energy 三种协议，Classic Bluetooth 和 Bluetooth High Speed 基本延续蓝牙 3.x 带不带 HS 后缀的两个版本，所以 Bluetooth Low Energy 才算是蓝牙 4.0 新特性，即为 BLE、Bluetooth Smart、低功耗蓝牙，相比标准蓝牙在功耗上降低到 0.01-0.5W，可以减少蓝牙对移动设备带来的电力负担。

多协议似的蓝牙 4.0 支持双模式和单模式的部署方式，双模式可以让 BLE 集成到经典蓝牙，或在原来的蓝牙 2.1+EDR、3.0+HS 上加入 BLE，双模式蓝牙 4.0 可以向下兼容以前的蓝牙版本，而单模式下就只允许两个蓝牙 4.0 设备之间的传输。

蓝牙 4.1 只是个软件升级，重要功能是批量数据交换，这允许蓝牙可以同时连接多个设备，主要针对智能穿戴设备和手机之间的连接，蓝牙 4.1 支持这类设备通过 IPv6 连接到网网络，还有与 4G 共存特性，可以在两种无线网络都进行传输下，降低互相之间的干扰。在 2014 年推出的蓝牙 4.2 针对 IoT 物联网加入了一些功能。

蓝牙 5.0

BIG 在 2016 年 6 月公布了蓝牙 5.0，宣称新一代蓝牙相比蓝牙 4.2 LE 有 4 倍传输范围提升、2 倍传输速度提升，还支持室内定位技术，但目前还没有很详细的参数和技术讲解，也没有相关设备搭载，但蓝牙 5.0 必然是未来两三年智能手机、智能设备的标配技术。

蓝牙 5.0 在低功耗模式下具备更快更远的传输能力，传输速率是蓝牙 4.2 的两倍（速度上限为 2Mbps），有效传输距离是蓝牙 4.2 的四倍（理论上可达 300 米），数据包容量是蓝牙 4.2 的八倍。

支持室内定位导航功能，结合 WiFi 可以实现精度小于 1 米的室内定位。

针对 IoT 物联网进行底层优化，力求以更低的功耗和更高的性能为智能家居服务。

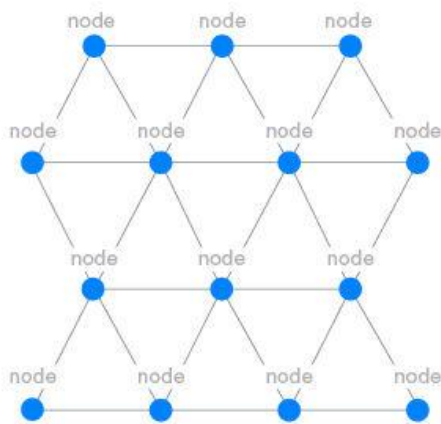
Mesh 网状网络：实现物联网的关键「钥匙」

Mesh 网状网络是一项独立研发的网络技术，它能够将蓝牙设备作为信号中继站，将数据覆盖到非常大的物理区域，兼容蓝牙 4 和 5 系列的协议。

传统的蓝牙连接是通过一台设备到另一台设备的「配对」实现的，建立「一对一」或「一对多」的微型网络关系。

而 Mesh 网络能够使设备实现「多对多」的关系。Mesh 网络中每个设备节点都能发送和接收信息，只要有一个设备连上网关，信息就能够在节点之间被中继，从而让消息传输至比无线电波正常传输距离更远的位置。

这样，Mesh 网络就可以分布在制造工厂、办公楼、购物中心、商业园区以及更广的场景中，为照明设备、工业自动化设备、安防摄像机、烟雾探测器和环境传感器提供更稳定的控制方案。



1.3 蓝牙分类

蓝牙是一种短距的无线通讯技术，可实现固定设备、移动设备之间的数据交换。一般将蓝牙 3.0 之前的 BR/EDR(Basic Rate/Enhanced Data Rate)蓝牙称为传统蓝牙，而将蓝牙 4.0 规范下的 LE 蓝牙称为低功耗蓝牙。



图 6 蓝牙分类










	Low Energy (LE)	Classic Basic Rate & EDR
Channels	40 channels with 2 MHz spacing	79 channels with 1 MHz spacing
Data Rate	BLE 5: 2 Mbps BLE 4.2: 1 Mbps BLE 5 Long Range (S=2): 500 Mbps BLE 5 Long Range (S=8): 125 Mbps	EDR (8DPSK): 3 Mbps EDR (π/4 DQPSK): 2 Mbps Basic Data Rate: 1 Mbps
Power & Radio Profiles	Class 1: 100 mW (+20 dBm) Class 1.5: 10 mW (+10 dBm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)	Class 1: 100 mW (+20 dBm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)
Power Consumption	~0.01x to 0.5x of Classic	Based on radio class.
Network Topologies	Point-to-Point (including piconet) Broadcast Mesh	Point-to-Point (including piconet)

1.3.1 传统蓝牙

蓝牙 4.0 标准包括两个蓝牙标准：一个是双模标准，它包括传统蓝牙（或称经典蓝牙）、高速蓝牙和低功耗蓝牙部分；另一个就是单独的低功耗标准。这个两部分适用于不同的应用条件和应用环境。

传统蓝牙（或称经典蓝牙）可以用于数据量较大的传输，如语音、音乐等，低功耗蓝牙也是建立在传统蓝牙基础之上发展起来的，并区别于传统模块，最大的特点就是成本和功耗降低，应用于实时性要求比较高但数据速率相对较低的产品中，如传感设备的数据发送和遥控装置设备等。

蓝牙商标有三种，可用来区分产品所采用的蓝牙类型。制造商会在产品本身或其包装上使用这些商标。可以认为 Bluetooth Smart Ready 是双模标准，Bluetooth Smart 是低功耗蓝牙。

If your product bears this logo...	It's compatible with products bearing any of these logos...
	  
	 
	

1.3.2 BLE

蓝牙低功耗（BLE）技术，也称为 Bluetooth® Smart，是蓝牙 V4.0 及以上核心规范的一部分，它可以满足小型电池供电的设备进行低功耗无线连接的要求，并大大延长电池寿命。主要应用包括：定位标签，资产跟踪，运动及健身传感器，医疗传感器，智能手表，遥控器，玩具等。

蓝牙 BLE 相比于经典蓝牙的优点：

- 峰值电流、平均电流及空闲模式下电流消耗少
- 搜索、连接的速度更快，无需配对
- 低成本
- 不同供应商认证设备之间的互操作性

蓝牙技术联盟沿用经典蓝牙的规范内容，为蓝牙低功耗定义了一些 profile，这些 profile 定义了一个设备在特定应用情景下如何工作。制造商应通过在实现中遵循特定的 profile 以确保兼容性。一台设备可以使用多个 profile。

当前所有低功耗应用 profile 都基于通用属性规范（GATT）。GATT 定义了属性，作为通用的封装数据的单位，并定义了如何通过蓝牙连接传输属性从而达到传输数据的目的。蓝牙 4.0 能够提供低功耗的较高比特率传输。

1.4 蓝牙 profile

什么是 Profile？

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。

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，你可根据所需要的产品实现需要的 Profile，不必给开发带来更大的开销。

这就是说当需要利用蓝牙提供数据传输功能时必须建立对应的 Profile，

Profile 定义了一种基于蓝牙的应用，每个 Profile 规范主要包括针对开发者的接口，消息的格式和标准（例如音频压缩），使用蓝牙协议栈的组件等。

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。

1.1 GAP

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。

一般访问应用规范（GAP）定义了蓝牙设备如何发现 and 建立与其他设备的安全（或不安全）连接。它处理一些一般模式的业务（如询问、命名和搜索）和一些安全性问题（如担保），同时还处理一些有关连接的业务（如链路建立、信道和连接建立）。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。

1.2 SDAP

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。

1.3 SPP

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。

1.4 GOEP

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。

1.5 A2DP

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

A2DP 规定了使用蓝牙非同步传输信道方式，传输高质量音乐文件数据的协议堆栈软件和使用方法，基于该协议就能通过以蓝牙方式传输高品质的音乐了。这个技术可以利用立体声蓝牙耳机来收听手机中的音乐了。

所谓的蓝牙其实包括很多版本以及小的功能规范子集，其中一个叫做 A2DP（Advanced Audio Distribution Profile）的，是专门为使用蓝牙传送立体声音乐而制定的只要两个蓝牙设备均支持这种规范，那么即可以实现无线传送立体声音乐。

1.6 DUN

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。

1.7 AVRCP

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

AVRCP 设计用于提供控制 TV、Hi-Fi 设备等的标准接口。此配置文件用于许可单个远程控制设备（或其它设备）控制所有用户可以接入的 A/V 设备。它可以与 A2DP 或 VDP 配合使用。

AVRCP 定义了如何控制流媒体的特征。包括暂停、停止、启动重放、音量控制及其它类型的远程控制操作。AVRCP 定义了两个角色，即控制器和目标设备。控制器通常为远程控制设备，而目标设备为特征可以更改的设备。在 AVRCP 中，控制器将检测到的用户操作翻译为 A/V 控制信号，然后再将其传输至远程 Bluetooth 设备。对于“随身听”类型的媒体播放器，控制设备可以是允许

跳过音轨的耳机，而目标设备则是实际的播放器。常规红外遥控器的可用功能可以在此协议中实现。

AVRCP 协议规定了 AV/C 数字接口命令集（AV/C 命令集，由 1394 行业协会定义）的应用范围，实现了简化实施和易操作性。此协议为控制消息采用了 AV/C 设备模式和命令格式，这些消息可以通过音频/视频控制传输协议（AVCTP）传输。

1.8 HID

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

HID 是标准的键盘、鼠标等的输入输出，例如可以用这个 Profile 来实现一些简单的远程按键控制。

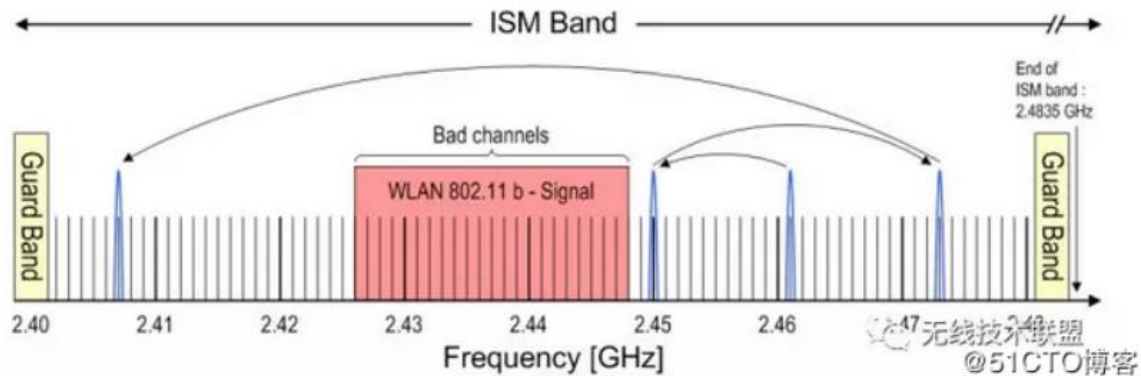
Android 种 HID 的事件捕获与 backKey 等方式相同，可以在使用 View.OnKeyListener 的 onKey 来捕获对应的 keyCode。

2. 蓝牙物理特性

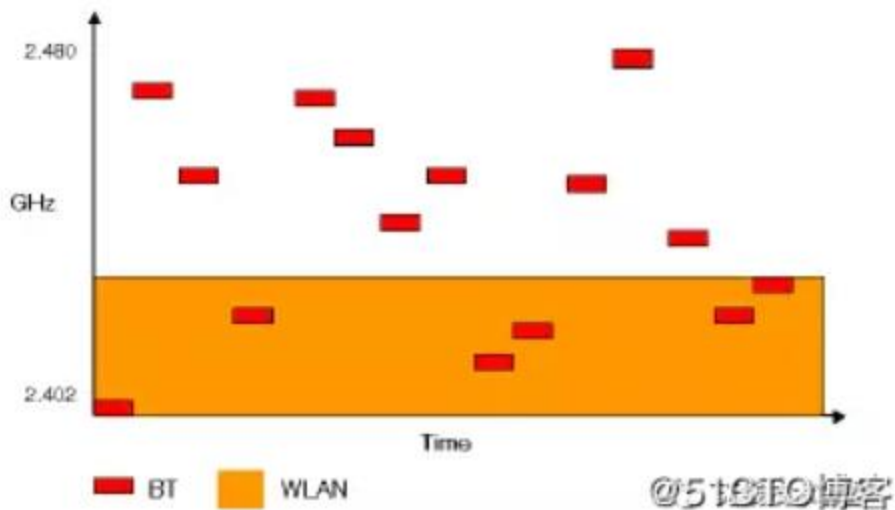
蓝牙工作在无需许可证的，全球统一开放的 2.4GHz ISM（工业，科学，医学，Industrial，Scientific and Medical）频段上（无线电也是一种资源，该频道使用无须向各国的无线电资源管理部门申请许可证）。

工作在 ISM 频段的无线电设备有很多种，比如家用微波炉，WLAN 等，为了抵抗来自这些设备的干扰，蓝牙采用了跳频（Frequency Hopping）方式来扩展频谱（Spread Spectrum），将 2.402~2.48GHz 频段分成 79 个频点，相邻频点间隔 1MHz，蓝牙设备在某个频点发送数据后，再跳到另一个频点发送，而频点的排列顺序则是伪随机的，每秒钟频率改变 1600 次，每个频率持续 625us，称为间隙（Time Slot）。

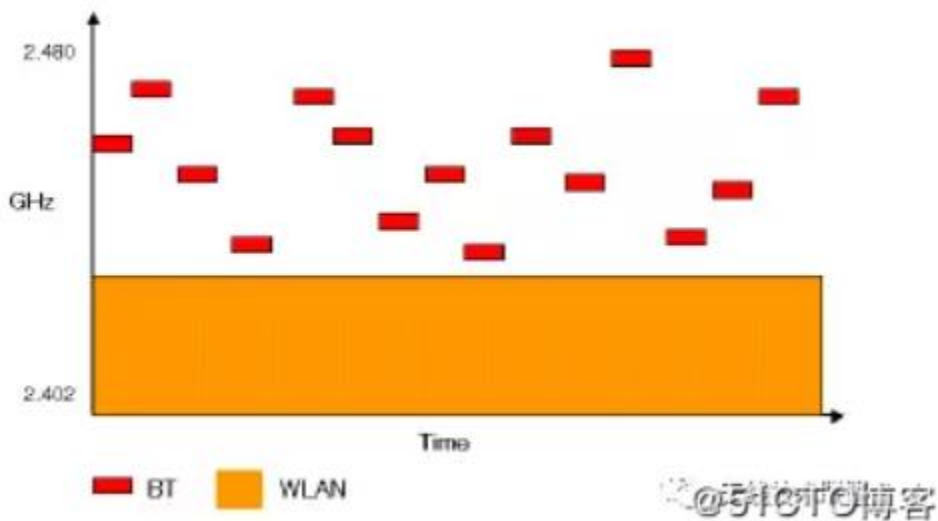
为了改善在该环境中的相同规范设备的通讯性能，SIG 引入了自适应跳频的技术 AFH（advance frequency hopping），以减少这种干扰的影响。该跳频技术可以通过各种方法实现，每种方法都有其固有的优点和缺点。



在 AFH 解决方案出现之前开发的蓝牙产品采用另一种形式的跳频，其跳频在设计上是随机的。这些第一代蓝牙设备使用 2.4 GHz 频段中的 79 个可用频道，以随机方式跳过这些频道，速率为每秒 1600 次。一旦将另一个无线设备引入环境中，这种类型的跳跃就会导致偶尔的冲突。没有 AFH 蓝牙缺乏避免这些冲突的能力，从而适应其环境。结果如下图所示，显示了蓝牙（BT）和无线 LAN（WLAN）都在运行的环境。



与上述相反，自适应跳频 AFH 允许蓝牙通过识别固定的干扰源并将其从可用信道列表中排除来适应环境。这种重新映射过程还涉及减少蓝牙使用的信道数量。蓝牙规范要求至少 20 个信道的最小集合。下图显示了与上图相同的环境，但现在使用了自适应跳频后的蓝牙通讯信道。



经典蓝牙跳频框架如下所示：有一个 Channel map，即为跳频表，一个跳频阶跃；根据跳频表和跳频阶跃和当前通讯频点，即可计算出主从机下一次数据通讯的频点。

蓝牙跳频表算法各家的算法略有不同，但都需要解决两个问题

1. 信道评估：

SIG 规范没有规定如何识别不良信道，目前有两种主要的方法用于执行具有自适应跳频的信道评估：RSSI（接收信号强度指示）和 PER（分组错误率）。

RSSI 和 PER 都是众所周知的用于确定哪些信道可能已被占用的技术。然而，当涉及监听当前信道状态时，这两种方法不同。PER 用于反复测试和重新评估不良信道的方法不如 RSSI 准确，并且可能导致临时挫折。然而，在使用 RSSI 时还存在许多其他问题，例如 RSSI 消耗的功率大于 PER。当缺少可用的时隙时，RSSI 还可以要求从其他功能获取带宽。

2. 同一信道数据通讯：

蓝牙 AFH 规定，主设备和从设备都通过同一频道进行通信。这样做是为了避免主设备在“好”信道上发送而从设备响应“坏”信道（反之亦然）的情况，因为这将导致多次重传（其他协议 AFH 的数据收发是在不同信道，会产生发数据正常，接收信道干扰导致无法响应主产生的多次重传）。由于主设备和从设备在相同频率上发送接收数据，因此信道跳频率降低 50%至每秒 800 次。虽然这可以使蓝牙设备对来自其他蓝牙设备的干扰更敏感，但迄今为止所带来的好处超过了这个小缺点。

2.1. AFH 的原理

AFH 技术是对原始蓝牙跳频序列的一种改进，它允许蓝牙设备缩减跳频点的数量，其基本原理是将信道分成 unknown, bad or good 三类，从而避免使用 bad 信道，减少受干扰的程度。当蓝牙微微网进入 AFH 状态后，其跳频序列可使用的跳频点 N 的数量是动态变化的，但规定必须有一个最小值 N_{min} ，即 $N_{min} \leq N \leq 79$ 。Bluetooth 协议中将 N_{min} 定义为 20。AFH 只用于连接状态，而不会改变寻呼、查询等状态时的跳频序列。自适应跳频选择机制的实现是基于原 79 跳系统的频率选择核心，在其基础上增加了 AFH_mode 和 AFH_channel_map 两个参数。AFH_mode 指出当前选频核心是否可以使用自适应跳频序列；AFH_channel_map 中指明哪些信道是可用的，哪些信道是不可用的。首先，原选频核心生成一个信道，如果这个信道是 AFH_channel_map 中定义的可用信道，则不作任何调整，直接作为跳频序列的输出；如果此信道包含在不可用信道中，则通过重定位函数将其映射成一个可用的信道。这种映射关系是一一对应的，就是说，如果给定了蓝牙地址、时钟以及 AFH_channel_map，一个不可用的射频信道将被唯一地转换为一可用信道，这样保证了在同一微微网中使用 AFH 机制的主从设备能够保持跳频序列的同步。

2.2. AFH 状态的控制

在这种实现机制下，非自适应的 79 跳系统的跳频序列等于将全部信道设为可用的 AFH 选频核心产生的频率序列，这一属性使得可以方便地与原非 AFH 设备保持兼容。AFH 技术的另一点改变是：在原跳频系统中，主从设备分别采用不同的频率发送数据；当处于 AFH 状态时，在一次主从对话期间，Slave 使用与 Master 相同的射频信道向 Master 响应数据包，这被称作 AFH 的相同信道机制。使用相同信道机制主要是由于在网中存在干扰的情况下，减少跳频可以防止 Slave 在发送响应分组时跳到可能发生冲突的信道上，保证至少在一次主从对话的过程中数据不易受到干扰，达到提高吞吐率的目的。

如果是进行更新信道信息操作，在 AFH_instant 之前未收到确认信息的情况下，主设备将恢复使用全部信道作为可用信道的跳频序列，并继续向从设备发送 AFH 命令直到收到响应，方可以使用更新后的自适应跳频序列。

2.3. 可用信道的确定

AFH_channel_map 参数中的可用信道是由以下几点共同确定的：

- (1) 主设备本身对于干扰的测量。其主要依据有分组包丢失率(PLR)、接收信号强度(RSSI)、循环冗余纠错(CRC)、混合纠错(HEC)、前向纠错(FEC)等。
- (2) Host 端可以通过 HCI 命令 HCI_set_AFH_Channel_Classification 向蓝牙设备声明信道状态信息。
- (3) 从设备也可以把其掌握的信道信息通过 LMP 层的 PDU 报告给主设备。

主设备将对这些信息资源进行理的分析 and 推算并最终生成一个可信赖的 AFH_channel_map，但无论如何可用信道的数量一定要大于最小信道数 N_{min} (20)。假设推算得到可用的 good 信道数为 NG，若 $NG \geq N_{min}$ ，则可以使用全部 good 信道；若 $N_{min} > NG$ ，则必须使用部分 unknown/bad 信道。