Bluez源码分析

Bluez是linux 的官方蓝牙协议栈。

Bluez由两部分组成一部分工作在kernel层，位于linux 的kernel源码中；另外一部分工作在用户空间，可以在bluez官网直接下载。

1. Kernel部分代码：位于Kernel的代码分为驱动代码和协议代码

**什么是HCI**

HCI (Host Control Interface) 是 Bluetooth SIG 设计用来提供Host与Bluetooth连接的软件接口（Software Interface)。而实际连接的硬件上目前也已经制定了传输方式，使Host与BT Module硬件连接有所依据，目前已经制定的传输方式有：

* UART
* USB
* SD
* 3-Wire UART

HCI有四种基本形态的封包，在不同的传输层通过不同的方式区别：

* Commands：用于Host对BT Module 的控制
  + 在UART Transport使用0x01放到packet的开头作为识别
  + USB Transport是control endpoint（endpoint 0）（控制断点）
* Asynchronous connectionless (ACL) Data：用于传输一般的资料，具有基本的资料可靠性保证
  + UART transport 使用 0x02
  + USB transport用Bulk endpoint 传输
* Synchronous connection-oriented (SCO) Data：用于语音资料的传输
  + UART transport 使用 0x03
  + USB transport用Isochronous endpoint 传输
* Event：用于 BT Module 回应 Host 的控制
  + UART transport 使用 0x04
  + USB transport 则用 Interrupt endpoint（中断断点）

HCI驱动代码位于kernel\drivers\bluetooth，

hci\_ldisc.c 提供基于 UART 的标准 HCI 接口。

hci\_bcsp.c hci\_h4.c HCI H4和 HCI BCSP协议

H4 和 BCSP 协议之间有什么区别呢？

蓝牙规范V1.2定义了一系列通信层和协议，允许主机程序发送接收数据和控制蓝牙连接。以下内容简短的介绍下H4和BCSP这两个基于UART的协议。

1. H4

H4 是最简单的主机传输协议。它被规定为没有奇偶校验的 RS232。但是硬件流控制是必须的。

* Commands
* Event
* Synchronous connection-oriented (SCO) Data
* Asynchronous connectionless (ACL) Data

H4 的缺点在于错误检测上面，由于没有奇偶校验，它只能检测以下错误：

* 错误的HCI 包
* 损坏的HCI 命令
* HCI 长度超过了范围

缺乏错误恢复机制，一旦出现错误，唯一的恢复方式就是复位总线和重新启动传输。这通常意味着重整BT连接。这种方式的优点就是

1. BCSP

BCSP（BlueCore Serial Protocol）是 CSR 开发用来替代 H4 的一种可靠性传输协议。这个协议定义用3-wire UART连接。不过，BCSP 也可以选择有两路流控制的 5-wire UART。

* HCI 封包中的消息也增加了对以下功能的支持：
* 可选可靠连接或者不可靠连接
* 对于不可靠连接有软件的流控制和序列控制
* 对于可靠连接每一个封包都有应答信号
* 所有的消息都可以选择是否使用 CRC
* 封包的头部有8位的校验和
* 简单的错误恢复策略

任何传输之前开始,一个链接的建立过程允许同步没有硬件流控制线路。BCSP封包同样包含有：Commond，Event，SCO，ALC。额外的通道分配也允许上层的RFCOMM 直接通过制造商的私有频道命令来通信。

BCSP 堆栈运行在主机和主机控制器上。

* 一个可靠的双向数据服务
* 一个不可靠的双向数据服务

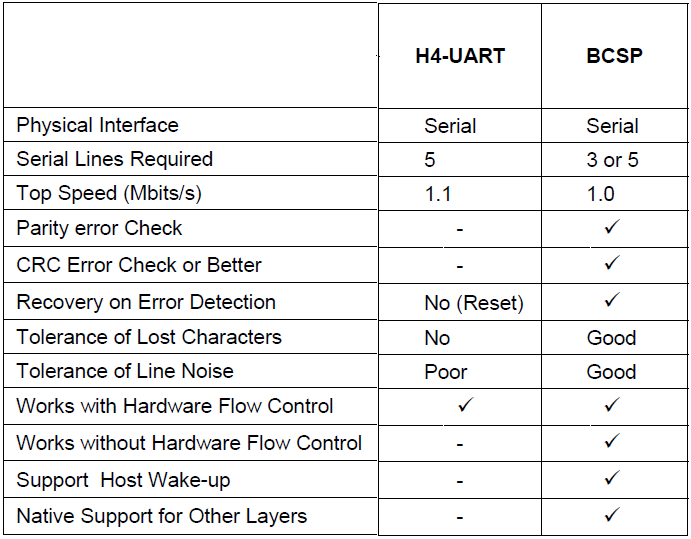
更高层次的堆栈可以基于上面的两种数据服务。

BlueCore 设备在使用 BCSP 的时候，包含专用硬件来提高性能。BCSP封包中的额外信息使得这种可能能够实现。

硬件检测包的完整性、正确性，包括校验和和CRC，使用DMA将负载运输到目的地，这种方式是的处理器有两种好处：

* 更多能力用来处理其它任务
* 更多机会进入睡眠

1. 对比表格



参考资料：

<https://bluegiga.zendesk.com/entries/25173052-HCI-What-are-the-differences-between-H4-and-BCSP-protocol->

<https://kunyichen.wordpress.com/2006/11/28/bluetooth-hci-data-format/>

1. 协议代码位于kernel\net\bluetooth

HCI 协议

L2CAP协议

SCO

RFCOMM

1. 资料搜集

Profiles

<https://en.wikipedia.org/wiki/List_of_Bluetooth_profiles>

<https://en.wikipedia.org/wiki/List_of_Bluetooth_protocols>

Linux 应用

蓝牙初始化流程

Bluez使用的是dbus，因此先打开dbus 需要的服务

# turn on dbus-daemon service

mkdir -p /tmp/dbus

mkdir -p /var/lib/dbus

dbus-uuidgen > /var/lib/dbus/machine-id

dbus-daemon --config-file=/etc/dbus-1/system.conf

打开蓝牙电源

# bluetooth power on

echo 1 > /sys/class/rfkill/rfkill0/state

sleep 1 # if invoke this script in c with system(), must sleep for a while!!!!!

初始化uart口，下载固件到蓝牙芯片

不同的芯片这部分流程有差异

# attach firmware

brcm\_patchram\_plus --enable\_hci --baudrate 3000000 --no2bytes --patchram /lib/firmware/BCM4343A0\_001.001.034.0056.0221\_26M\_ORC.hcd /dev/ttyS0 --tosleep=50000 --use\_baudrate\_for\_download --enable\_lpm &

打开 HCI ???

hciconfig hci0 up

打开蓝牙后台服务

# turn on bluetooth service

bluetoothd

复位HCI，开启扫描

hciconfig hci0 reset

sleep 3

# open scan and be-scan func

hciconfig hci0 piscan up

启动代理服务，能够自动重新连接

# reponse connection request automatic

agent --adapter hci0 0000 &

bluetoothd 作为后台服务，提供蓝牙服务给客户使用，使用的通信方式为 dbus。

alsa + bluez

用户通过 libasound.lib 来调用通用的 alsa 接口使用。而libasound.lib 则通过 libasound\_module\_pcm\_bluetooth.so

libasound\_module\_ctl\_bluetooth.so

来调用真正的播放过程。

依赖一个文件，/etc/asound.conf

pcm.device\_name{

type bluetooth

device btaddr

}