

语音遥控器设计说明 发布 1.2.0

# 目录

1	文档介绍	1
	1.1 文档目的	1
	1.2 术语说明	1
	1.3 参考文档	1
	1.4 版本历史	2
2	遥控器应用设计	3
	2.1 功能需求	3
	2.2 关键性能规格需求	3
	2.3 遥控器应用总体架构	4
	2.4 遥控器模块流程概述	5
3	语音传送的 PROFILE 介绍	13
		13
		14
4	按键编码设计 1	16
5	按键行为规范	17
U		17
		17
		17
6	休眠与唤醒	۱9
U	11 - A A A A A A A A A A A A A A A A A A	19
		19
7	语音框架 2	20
1		20 21
		۷1 21

文档介绍

## **1.1** 文档目的

文主要介绍 Actions 语音遥控器方案的软件相关技术参数与行为规范,包括按键输入、语音采集、红外发射、BLE 广播、连接配对等行为规范,用以指导遥控器的开发。

## 1.2 术语说明

表 1.1: 术语说明

术语	说明
HID	Human Interface Device,人机接口设备,是 USB
	设备中常用的设备类型,是直接与人交互的 USB
	设备,例如键盘、鼠标与游戏杆等
BLE	蓝牙低功耗技术

## 1.3 参考文档

 $\bullet \ \, \rm http://docs.zephyrproject.org/$ 

# **1.4** 版本历史

表 1.2: 版本历史

日期	版本	注释	作者
2018-08-22	1.0	初始版本	ZS110A 项目组
2019-03-02	1.1	hid 语音遥控器 cache 版本	ZS110A 项目组
2019-04-10	1.2	红外自学习原理介绍及非 hid profile 介绍	ZS110A 项目组

## 遥控器应用设计

## 2.1 功能需求

- 按键输入: < 35 键
- 语音输入: 8k/16bit、16k/16bit
- 语音编码: 1/4、1/8、1/16 压缩算法
- 红外键码输出
- BLE 键码输出、BLE 语音输出
- 蓝牙断线后回连,并发送回连前的键值
- 电量检测及电量管理
- 红外自学习: 支持 9012 NEC RC5 RC6 50426 7461 50560 等主流协议,可扩展支持更多协议

## 2.2 关键性能规格需求

- BLE 传送性能 > 4kbyte/s
- 按键响应速度约 50ms
- 语音响应时间 < 200 ms

### 2.3 遥控器应用总体架构

遥控器总体架构分为四层,从上到下依次为应用层、应用框架层、硬件抽象层、底层驱动层



图 2.1: 遥控器应用总体架构框图

### 2.3.1 应用层

- 应用状态机
  - 事件触发让遥控器应用处理不同的状态。
- 应用定时器
  - 定时触发不同的事件,驱动遥控器正常运行
- 应用输入处理
  - 处理底层来的不同按键消息
- 应用音频输入处理
  - 将底层的音频处理消息,进行编码,然后通过蓝牙发送给对端设备
- BLE profile
  - HID profile, 提供按键输入输出接口服务
  - BAS service, 提供电池服务

- DIS service, 提供读取设备基本信息的接口服务
- ota profile, 提供 OTA 升级服务

#### 2.3.2 应用框架层

- 输入管理
  - 按键映射处理
  - 按键过滤机制
- 消息管理
  - 消息分配和释放
  - 消息发送和接收
- 内存管理
  - 动态内存管理
- 闪灯管理
  - 灯资源分配和释放
- 电池管理
  - 电量读取
  - 电量管理策略

### 2.3.3 硬件抽象层

将应用层和驱动层剥离开的中间件层。

### 2.3.4 底层驱动层

底层硬件操作接口,详细描述见驱动开发指南文档.

### 2.4 遥控器模块流程概述

### 2.4.1 系统启动

- 系统相关初始化
- 板级相关外设初始化
- 蓝牙协议栈相关初始化
- HidApp 应用初始化,并进入 Main 主循环,等待消息处理

### 2.4.2 遥控器状态机

遥控器在运行过程中,主要靠如下3种状态维持他的正常运行。

表 2.1: 遥控器工作状态

序	状态	说明
号		
1	RMC_STATES_IDLE	处于无连接和无广播的空闲状态
2	RMC_STATES_ADVERTISING	处于发送广播的激活状态
3	RMC_STATES_CONNECTED	处于连接的工作状态

- 1. 触发遥控器进去空闲状态的事件:
  - >广播状态,没有连接成功,出现超时事件,进入 idle
  - > 连接状态,断开连接,如无操作主动断开连接,然后进入 idle
- 2. 触发遥控器进入激活状态的事件:
  - > 空闲状态,有按键、首次上电,进入激活状态
  - > 连接状态, 出现异常断开, 需要回连, 进入激活状态
- 3. 触发遥控器进入工作状态的事件:
  - > 激活状态下,配对成功或者回连成功,进入工作状态。

### 2.4.3 按键处理

由于遥控器的键值较多,通常用矩阵键盘方式以节省 pin 的使用。当使能 Key 模块后, Key 控制器就会处于矩阵扫描状态,当检测到外部按键有值时,就会产生中断,中断就 将按键信息上报给应用。

### 2.4.4 红外处理

- 在非连接状态下,按下按键,就会发射红外码,进而通过红外操作对端设备,如使用红外进行配对.
- IRC 协议上,最短的红外码重发时间为 108 ms,而按键的重复上报时间,可能小于 108ms,也可能大于 108ms,因此按键输入和红外发送模块时间上存在三种可能:
  - 1. 慢速点按

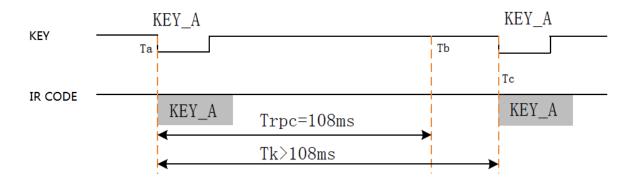


图 2.2: 慢速点按示意图

慢速点按动作特征是在大于 Trpc 时间后有多次的按键输入。在 Ta 时刻,发出初次按键值,在 Tb 时刻,不做任何响应,在 Tc 时刻,继续发送检测到的按键值,不会发送重复码.

#### 1. 快速点按

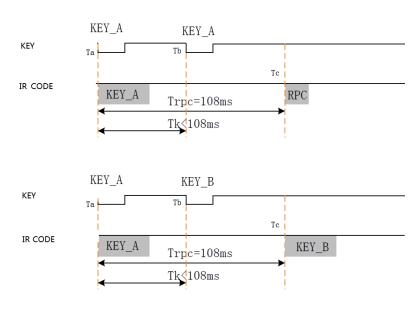


图 2.3: 快速点按示意图

快速点按的动作特征是用户在 Trpc 时间内有两次或者以上的按键按下弹起的动作. 在 Ta 时刻,将发送出初次按键,而 Tb 时刻并不发送按键值,在 Tc 时刻,如果按键仍然是按下状态,将发出按键值,否则将丢掉按键值.

#### 1. 长按

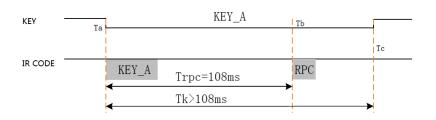


图 2.4: 长按示意图

长按的动作特征是按键按下后,一直不放开。此时 CPU 检测到按键的持续按下,则在 Ta 时刻发送出初次按键,发送出此时按键对应的红外键码,而在 Trpc 时间内没有检测到按键的松开,则在 Tb 时间输出重复码,直至检测到按键弹起为止.

#### 2.4.5 红外自学习

- 当前有 2 种红外自学习方式
  - 1. 硬件方式
  - 2. 软件方式
- 硬件自学习当前 IC 只支持 9012, NEC, RC5, RC6 这四种主流格式, 而软件自学习可以扩展支持多种格式, 默认使用软件自学习.
- 软件红外接收原理

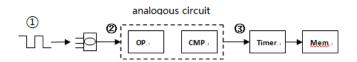


图 2.5: 软件红外接收通路示意图

红外信号经过接收头后的整个硬件通路流程如上图所示, 其中,

- 1. 首先在空气中传播的红外码流光信号,传播到接收头, idle 为低电平
- 2. 高电平通过接收管时产生电流进行导通且有反向特性,得到码流是未解调的载波波形
- 3. 从接收头出来后进入模拟电路进行反向放大、高低阈值检测滤波, 得到滤波后的载波波形

软件自学习过程就是将模拟电路滤波后的载波波形,通过使用一个二级 timer 方式,实现去载波、载波频率识别以及脉冲序列的时长统计

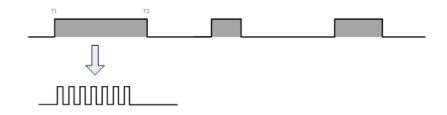


图 2.6: 模拟电路得到的载波波形示意图

假如当前模拟电路得到红外载波输入的序列如上图所示,我们需要完成以下几步工作:

- 1. 统计灰色框的时间长度,得出值 T H LVL
- 2. 统计灰色框的信号高低电平翻转次数,得出 CNT NUM
- 3. 统计非灰色框长度, 得出值 T\_L\_LVL
- 4. 重复上述操作,顺序记录下 T\_H\_LVL 和 \_L\_LVL 序列, 记录到 SPI NOR 储存

实现上述功能,需要使用两个 timer 实现,一个为普通延迟 timer,一个为 capture timer. 普通延时 timer,使用 reload 读 VAL 值的方式,而 capture timer 主要是要抓取电平翻转次数,详细过程如下所示:

- 1. 当 T1 时刻,通过检测电平值翻转,检测周期为check\_period,保留参数可以配置,目前预置值为5us,并且记录初始timer读值为TCNT1
- 2. 在灰色框的持续时间内, 检测电平值翻转次数 CNT\_NUM, 检测周期依然为 check period
- 3. 当 T2 时刻时,如果连续 n 次检测电平不再发生翻转,即 认为当前灰色框载波信号终止,n 目前预置值为 150us。即 150us 的载波超时检测时间,并且记录下当前 timer 读值 TCNT2
- 4. 再次回到第一步,检测电平翻转,记录当前 timer 读值为 TCNT3,如此循环下去
- 5. 上述步骤中,可以恢复出来的载波持续长度序列为 T\_mod = TCNT1-TCNT2, 非载波持续长度序列为 TCNT2-TCNT3
- 软件红外发送原理

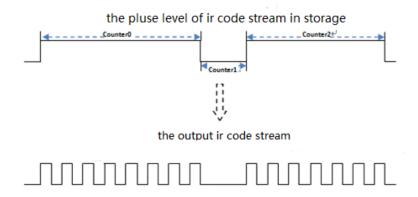


图 2.7: 软件红外发送示意图

通过 PWM 方式进行载波发送,mfp 切换来实现载波部分和非载波部分的转换,大致流程如下:

- 1. 读取储存在 SPI NOR 中红外码流宽度序列
- 2. 配置载波占空比和频率,使能 PWM 进行载波发送,码流宽度由 SPI NOR 中存储的值来决定
- 3. 载波部分结束后将 PWM mfp 切换成 GPIO, 使之为默认低电平, 拉低时长同样由 SPI NOR 中存储的值来决定
- 4. 重复上述过程,即完成红外码的发送。
- 红外自学习功能操作过程
  - 1. 长按遥控器红外学习按键区的学习键约 5 秒钟, 直至遥控器指示灯从快闪状态变为长亮状态, 此时进入学习状态
  - 2. 按一下要学习的按键, 指示灯从常亮状态变为慢闪状态, 此时处于学习接收状态
  - 3. 将被学习的遥控器和我们的遥控器对准大约 3~5 mm, 按下被学习的按键
  - 4. 学习成功后指示灯快速闪烁 3 下后继续保持常亮, 可以再次选择其他学习按键进行学习
  - 5. 如学习失败则指示灯闪烁 6 下, 返回学习状态, 重复步骤 2-4
  - 6. 学习完毕后, 按红外学习按键区以外的任意其它按键退出学习状态, 红外指示灯灯熄灭

#### 2.4.6 语音采集

- 当启动 Voice Key 后, ADC 开始采集
- 采集的数据通过 DMA 搬运到应用的循环 buffer 中,同时发送消息给 Main 线程, 让其处理语音数据。

• 如果 Main 线程处理速度不够快, audioin 驱动就会因为分不到 buffer, 而将采集的语音数据丢弃。

详细流程参看语音章节

#### 2.4.7 BLE 数据传送

- 将 audioin 驱动发送上来的数据进行编码压缩。
- 然后将编码后的压缩数据切成几个 20byte 的数据包
- 最后通过 hid profile notify 接口发送给 BLE 协议栈

详细流程参看语音章节

### 2.4.8 应用软件 Timer 管理模块

目前遥控器定义了几个软件 timer, 用途如下表所示。

	L-41.	
序	名称	功能
号		
		\H_1 \= -
1	ID_ADV_TIMEOUT	设定 Undirect advertising 的
		时间,Timeout 后停止广播
		的发送
2	ID_DIRECT_ADV_TIMEOUT	设定 direct advertising 的时
		间,Timeout 后停止广播的
		发送
3	ID_NO_ACT_TIMEOUT	遥控器连接上后,一段时间
		内没有按键或者语音动作,
		断开连接
4	ID_ADC_BATTERY_TIMEOUT	遥控器定时采样 battery 电
		压, 在有连接的情况下, 发
		送给对端
5	ID PAIR COMB KEY TIMEOUT	检测组合按键按住的时间,
		Timeout 后启动配对流程
6	ID_RMC_SEC_TIMEOUT	设置从连接到加密成功的时
		间,Timeout 后断开连接
7	ID_IR_LEARN_KEY_TIMEOUT	设置红外学习键进入学习状
		态的时间,Timeout 后进入学
		习状态

表 2.2: 遥控器使用的软件 timer

### 2.4.9 LED 管理模块

遥控器定义了几种 LED 指示灯,用于指示遥控器的一些状态,如下表所示

通常遥控器只有一个物理的 LED 灯用于各种场景的指示,这就需要软件上让其分时复用,如果同时需要显示两种状态,状态需要定义优先级,优先级高的状态先指示。如在配对模式下,处于闪灯状态下,这时候按下按键,那么灯还是处于快闪状态。

### 2.4.10 OTA 升级模块

当前 OTA 设计在应用中,系统随时可以升级,升级过程中,系统的其他模块仍然可以正常使用,升级完后,由对端设备决定是否重启系统,如果对端设备没有重启系统,系统将在下一次上电的时候,使用新的固件。详细见 OTA 相关文档。

#### 2.4.11 电池管理模块

电量检测采用标准的 BAS Service,实现电量的读取(Read)和上报(Notification)。

目前定义的电池电压检测范围  $1.8\sim3.25\mathrm{V}$ ,线性等分对应 0%-100%,由于没有关机操作,当前配置了一个超低电门槛  $1.8\mathrm{V}$ ,低于超低门槛后,系统不再允许 Nor 操作,蓝牙连接也会自动断开。

BAS 的 notification configuration characteristic 使能后,定时器每 N 分钟读取一次电量并通过 Notification 上报。

电池等级定义	电压 (V)	电压比例	意义	特殊处理
		(%)		
BATT_FULL	3.25	100%	满电电压	无
BATT_LOW	2.00	13%	低电压	电池低电报警
BATT_VLOW	1.80	0.00%	低电异常处理	1. 电池低电报警 2.
			电压	不再允许 Nor flash
				操作 3. 连接自动断
				一

表 2.3: 电池处理

## 语音传送的 PROFILE 介绍

## 3.1 HID profile 介绍

• GATT 服务需求

表 3.1: HID Profile 的服务需求

服务	是否需要	是否支持
HID service	必须	是
Battery Service	必须	是
Device Information Service	必须	是

• Report Map 介绍

Report Map 属性主要用于定义在 HID Service 和 report Host 端传送的 input report、output report、feature report 格式。每个 HID service 只能包含一个 report map 属性,属性值长度为小于等于 512byte。

• HID Reports 介绍

当前定义 3 种 HID input report IDs:

- Remote report用于传送遥控器按键信息
- Voice report
  用于传送语音信息,每次 report 的数据为 20byte
- Mouse report
  用于传送空鼠信息,暂未使用

- Hid Profile 语音传输基本过程
  - Client 端进行 BLE 连接、配对及服务发现,并使能所有 input 属性通知属性。
  - Server 端通过 HID profile 的 Remote report 发送一个开始按键到 Client 端, Client 准备接收 audio Data。
  - Server 端通过 HID profile 的 Audio Out 属性发送 voice Data 到 Client 端。
  - Server 端通过 HID profile 的 Remote report 发送一个结束按键, Client 停止接收 audio Data.

## 3.2 Actions Audio profile 介绍

- Actions audio service 基本功能概述
  - 1. 实现语音传送功能: 发送语音、接收语音等
  - 2. 实现 Audio 的传送控制功能: 开始、结束同步命令、编码方式通知
  - 3. 实现模块 BLE 配置功能:广播参数、连接参数等
- Actions audio service 属性介绍

自定义 audio 服务的 UUID 定义为 0x001120A0-2233-4455-6677-889912345678, 定义了 3 个 UUID 属性特征值, 分别语音发送通道、语音接收通道, 命令控制通道, 一些细节如下表所示:

表 3.2: Actions audio service 属性

type	uuid	properties	description
Service	20A0	RD	Audio 自定义服务
Audio Out charac-	20A1	NTF	Server 端传送 audio 到
teristic			Client 端
Audio In charac-	20A2	WR	Client 端传送 audio 到
teristic			Server 端
Control character-	20A3	WR&NTF	配置 BLE 参数及控制
istic			audio 的传送

注解: RD: read, WR: write, NTF: notify

1. 接收语音通道 (UUID: 20A1)

描述: Server 端传送 audio 到 Client 端

默认值: 20

长度: 每次最多接收 20 字节

通知: 使能通知情况下才能接收到数据

2. 发送语音通道 (UUID: 20A2)

描述: Client 端传送 audio 到 Server 端

默认值: 20

长度: 一次最多传送 20 个字节

写操作: 为了透传效果, 此写为无响应写

3. 命令控制通道 (UUID: 20A3)

描述:通过写 HCI 扩展命令实现对参数配置及行为控制

默认值: 0

长度:可变长度,根据命令需要

写操作: 无响应写

通知: 响应命令应答事件通过通知行为发回

• Actions audio service 语音传送过程

介绍一下 Server 端通过 audio Profile 传送 audio 的基本过程,大致过程如下: Client 端进行 BLE 连接、配对及服务发现,并使能 Audio Out 通知属性。语音传送:

- 1. Server 端通过 audio profile 的 Control 属性发送一个开始命令 (0x020c) 到 Client 端, Client 准备接收 audio Data。
- 2. Server 端通过 audio profile 的 Audio Out 属性发送 voice Data 到 Client 端。
- 3. Server 端通过 audio profile 的 Control 属性发送一个结束命令 (0xc-ccc), Client 停止接收 audio Data.

## 按键编码设计

表 4.1: 按键编码设计表

遥控器	Usage ID	Consumer 上对应功能	对应设备
Power	0x30	Power	HID Consumer
menu	0x40	Menu	HID Consumer
OK	0x41	Menu Pick	HID Consumer
Up	0x42	Menu Up	HID Consumer
Down	0x43	Menu Down	HID Consumer
Left	0x44	Menu Left	HID Consumer
Right	0x45	Menu Right	HID Consumer
Help	0x95	Help	HID Consumer
Mute	0xe2	Mute	HID Consumer
vol+	0xe9	Volume Increment	HID Consumer
vol-	0xea	Volume Decrement	HID Consumer
MIC			
Search	0x0221	AC Search	HID Consumer
Home	0x0223	AC Home	HID Consumer
Back	0x0224	AC Back	HID Consumer

- 1. 此按键编码方式除了 MIC 键是扩展的, 其它按键编码都是标准的 consumer 键.
- 2. 标准协议栈参考 HID Usage Tables Version1.12

按键行为规范

### 5.1 普通按键的行为

- 1. 在非连接状态,唤醒系统,发送回连包或者配对广播包,并通过红外发送红外码
- 2. 在连接状态下,发送蓝牙码

发送给远端设备的按键行为大致为:

 $down->down->\cdots->up$ 

设备端并不处理长按或者短按逻辑,交由远端设备处理。

## 5.2 语音按键的行为

- 1. 在非连接状态,唤醒系统,发送回连包或者配对广播包
- 2. 在连接状态下,Down 状态发送语音开始标识, up 状态发送语音结束标识。

发给远端设备的按键行为大致为:

Down—>up

设备端过滤掉中间的 down。

### 5.3 配对组合键的行为

配对组合按键会强制进入配对模式,发送配对广播包触发配对流程。 持续按住配对组合按键 N s 以上,进入配对模式,之后的流程如下:

- 1. 如果有 link key, 先删除 link key
- 2. 如果是连接状态,断开连接,如果是非连接状态,发送配对广播包
- 3. 组合键键值不会发送给远端设备

休眠与唤醒

### 6.1 休眠

- 1. 在广播状态下, Host CPU 会进入 deepsleep 状态, BLE 控制器处于广播状态.
- 2. 在连接状态下,不发送按键/语音,Host CPU 进入 deepsleep,BLE 控制器处于连接状态,如果 connSlaveLatency = 44,连接间隔为 10ms,那么 BLE 控制器进入 deepsleep,每隔 450 ms 醒来一次发一次空包回应对端,然后进入 deepsleep 状态.
- 3. 在连接状态下,发送按键,按键发送完,进入2状态.
- 4. 在连接状态下,发送语音,Host CPU 不会进入 deepsleep, BLE 控制器处于连接状态.
- 5. 在无连接和广播状态,Host CPU 和 BLE 控制器都会进去 deepsleep 状态,即最省电模式,所以应用层还设计了在 N 分钟内没有按键和语音操作,遥控器会主动断开连接进入 deepsleep 模式,达到最省电模式。

### 6.2 唤醒

- 1. 遥控器 Host CPU 可以由蓝牙事件唤醒,也可以由按键事件唤醒,还可以由 timer 事件唤醒
- 2. 在连接状态下,按键操作唤醒 Host CPU 然后将键码/语音发出
- 3. 在没有连接状态下,按键操作会触发进入回连模式,在回连成功后,将键码发出

## 语音框架

总体框架如下图所示,大致包括2个过程:

- BLE Server 语音采集、编码及传输
- BLE Client 语音接收、解码及识别

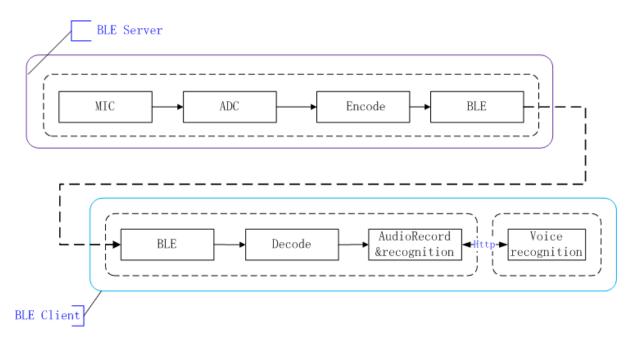


图 7.1: 语音传送框架图

### 7.1 BLE Server 语音采集、编码及传输

#### 7.1.1 MIC 语音采集

音频驱动使用 DMA 方式接收,配置一个 N×SIZE 的 BufferQueue, N 个 buffer 循环配置给 DMA 使用。当 BufferQueue buffer 消耗完,最后一个 buffer 被覆盖使用。

#### 7.1.2 语音编码

- 语音编码通常采用经典的 IMA adpcm 编码, 压缩率为 1/4.
  - 该算法优点: 消耗 CPU 和 RAM 相对比较少
  - 该算法缺点: 压缩率低, 有损压缩, 音质一般
- 当前也设计支持了高压缩高音质的 1/8、1/16 音频编码算法

#### 7.1.3 BLE 语音传送

BLE 通过 HID profile, 按照每包 20 个 byte 数据将编码后的数据传送给对端 BLE Hid Profile。

### 7.2 BLE Client 端语音接收、解码及识别

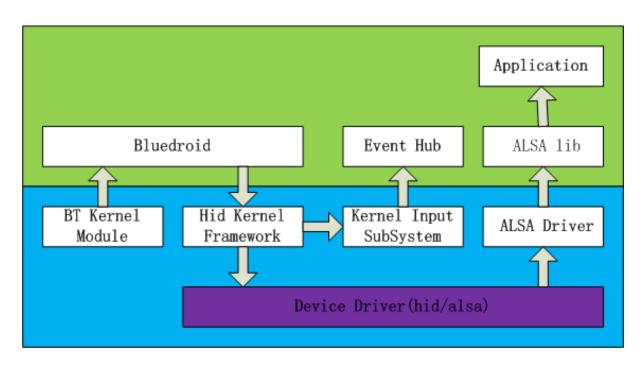


图 7.2: android 设备 HID 处理框架图

#### 7.2.1 BLE 语音接收

Android hid input 驱动收到 BLE 数据,通过 event 事件上报给 input 子系统, input 子系统再将事件上报给适配的事件处理 handler。

#### **7.2.2 Audio** 驱动解码

当 BLE hid 驱动收到遥控器开始语音按键后,就会开启 Audio 设备(虚拟 mic 驱动), Audio 会开启一个缓冲 buffer 用于接收 BLE 语音数据,当 buffer 快满的时候,Audio 驱动启动一个 N ms 的 timer 从 buffer 取 M byte 进行解码,解码后得到 K byte 数据,然后放到 pcm buffer 中。

### 7.2.3 录音识别

如果 Android 上层已经开启录音设备,录音 APK 就能从 audioTrack 读取 audio HAL 层语音数据,进而读取 Audio 设备驱动的 pcm buffer 音频数据,录音 APK 得到录音数据,通过 http 传送远程识别服务器,最终得到识别结果返回给本地。

# List of Figures

2.1	遥控器应用总体架构框图 4
2.2	慢速点按示意图
2.3	快速点按示意图
2.4	长按示意图 8
2.5	软件红外接收通路示意图 8
2.6	模拟电路得到的载波波形示意图 9
2.7	软件红外发送示意图 10
	语音传送框架图

## List of Tables

	术语说明	
2.2	遥控器工作状态	1
	HID Profile 的服务需求   1.     Actions audio service 属性   1.	
4.1	按键编码设计表	6