# Mooresilicon Cygnus SDK遥控器业务设计

© 2022 Mooresilicon All rights reserved.

本文档版权归TCL摩星半导体所有，受相关法律法规的保护。未经书面许可不得复制传播。

# 修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 说明 |
| draft | 2022-04-25 | 区伟权 | 起草 |
| 1.0 | 2022-04-26 | 区伟权 | 初稿评审完成。修改点：   1. 增加无感对码描述 2. 删除MIC驱动； 3. 配置管理使用flash分区 4. 配置管理数据结构定义 |

目录

[Mooresilicon Cygnus SDK遥控器业务设计 1](#_Toc101875893)

[修订历史 2](#_Toc101875894)

[目的 4](#_Toc101875895)

[范围 4](#_Toc101875896)

[概述 4](#_Toc101875897)

[设计约束 4](#_Toc101875898)

[遥控器业务逻辑 4](#_Toc101875899)

[按键业务 4](#_Toc101875900)

[对码业务 5](#_Toc101875901)

[国内对码业务 5](#_Toc101875902)

[国外对码业务 6](#_Toc101875903)

[语音业务 8](#_Toc101875904)

[低电量提示 9](#_Toc101875905)

[架构设计 9](#_Toc101875906)

[运行设计 10](#_Toc101875907)

[配对流程 10](#_Toc101875908)

[蓝牙按键流程 11](#_Toc101875909)

[红外按键流程 11](#_Toc101875910)

[蓝牙语音传输流程 12](#_Toc101875911)

[模块描述 12](#_Toc101875912)

[BLE 12](#_Toc101875913)

[TV端接口 12](#_Toc101875914)

[按键管理端接口 16](#_Toc101875915)

[按键驱动 18](#_Toc101875916)

[接口 18](#_Toc101875917)

[模块结构 18](#_Toc101875918)

[ADC驱动 20](#_Toc101875919)

[DMA驱动 21](#_Toc101875920)

[PM 21](#_Toc101875921)

[红外驱动 21](#_Toc101875922)

[配置管理 22](#_Toc101875923)

[按键管理 23](#_Toc101875924)

[驱动分层设计 23](#_Toc101875925)

# 目的

本文主要描述TCL遥控器业务实现逻辑，指导遥控器DEMO开发和SDK API设计开发。

# 范围

读者范围是遥控器DEMO开发和SDK API设计开发的软件工程师。

# 概述

TCL遥控器业务是Cygnus SDK应用的典型场景。本文针对TCL遥控器技术规范和Google voice技术规范，分析TCL遥控器业务逻辑。最终确保SDK API满足遥控器应用场景需求。

# 设计约束

遵循规范

* 《RC902V FAR1蓝牙遥控器技术规范》
* 《智能硬件所-TCL蓝牙遥控器技术规范V1.5》
* 《Google Voice over BLE spec 1.0 [external]》

软件约束

* 基于FreeRTOS开发

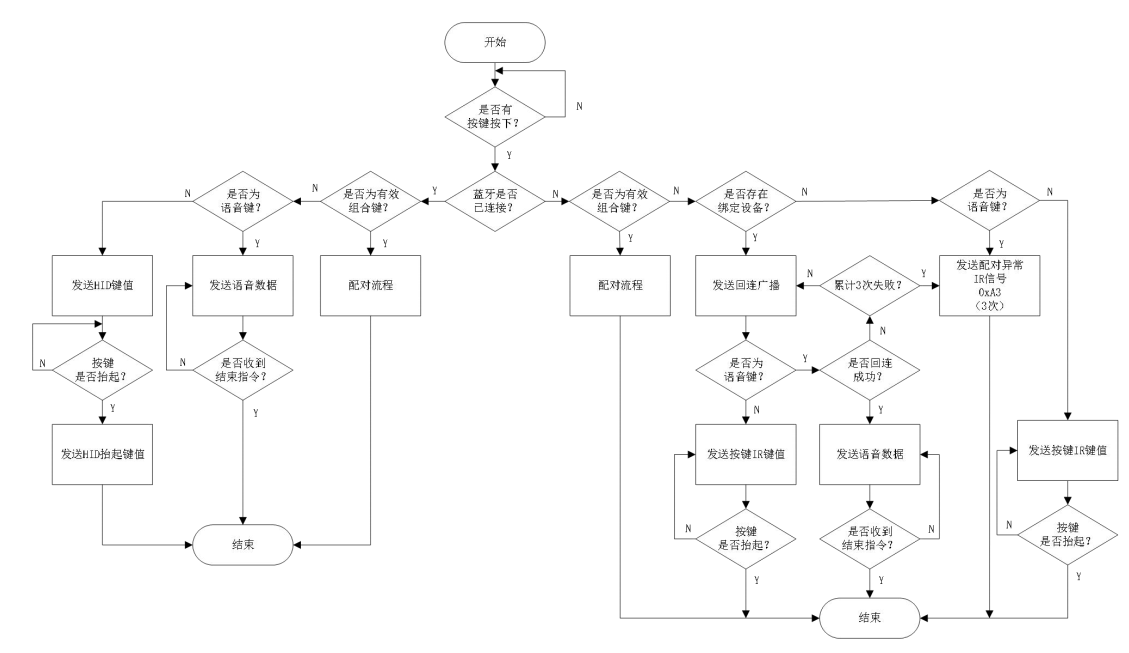
硬件约束

* 基于MS1008系列芯片

# 遥控器业务逻辑

## 按键业务

业务流程如下图所示：



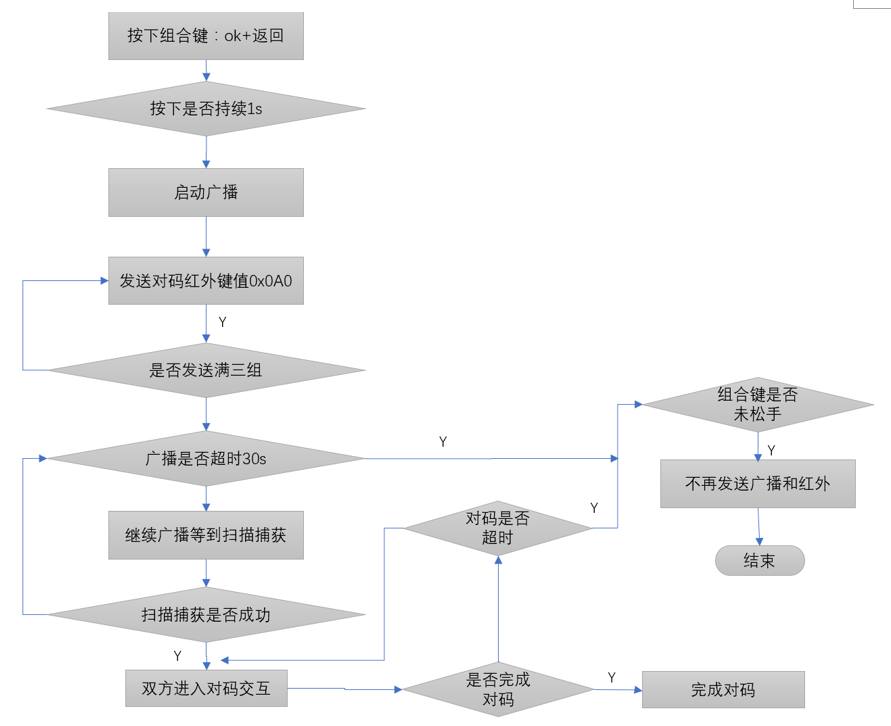
除了上述流程图中约束的按键逻辑， 遥控器按键响应还应满足如下设计要求：

1. 电源键只发红外键值；
2. 在蓝牙未连接状态时， 语音和鼠标按键应发 0xA3 键值的红外信号， 红外信号连续发 3 次；
3. 遥控器根据指定的组合键， 发出对应 0xA0 或 0xAD 键值的红外信号实现与电视自动配对， 红外信号连续发 3 次；
4. 遥控器在蓝牙未连接但有绑定设备信息时， 按下非语音键， 遥控器在发起回连广播时也要同时发送按键对应的红外信号；
5. 语音按键回连 3 次失败， 就发配对异常。

## 对码业务

### 国内对码业务

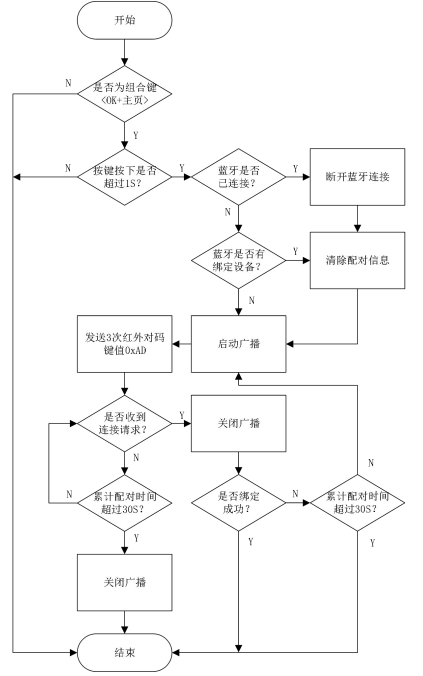
遥控器每次长按ok＋返回1s以上，启动配对广播进入配对模式，并间隔8ms发送红外键值0xA0,连续发送三组通知电视扫描，配对广播超时限制为30S；对码超时后如检测到组合按键未松开，不再发红外和配对广播。流程如下图所示：



### 国外对码业务

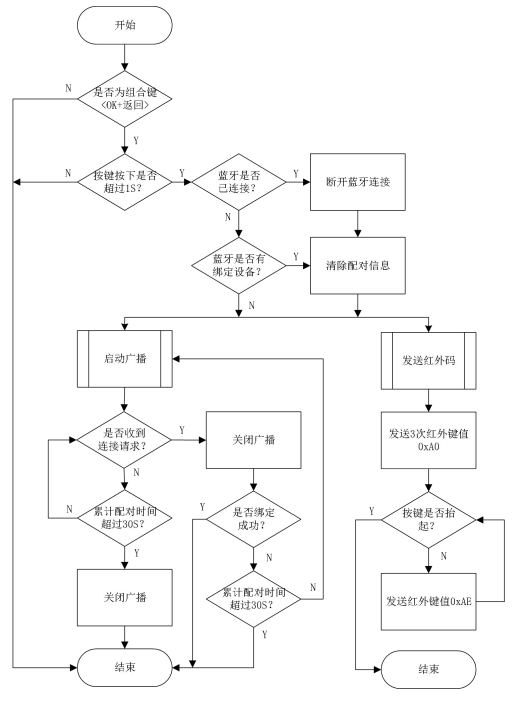
#### 模式一

遥控器长按【OK】 +【主页】 组合键 1 秒以上， 启动配对广播及进入配对模式， 并连续发送三组红外配对码（码值： 0xAD） 通知电视开启扫描， 配对广播应设置 30S 超时时间限制；遥控器每次按下【OK】 +【主页】 组合键， 仅触发一次对码流程。流程如下图所示：



#### 模式二

遥控器长按【OK】 +【返回】 组合键 1s 以上， 启动配对广播进入配对模式，并连续发送三组红外配对码（码值： 0xA0） 通知电视开启扫描， 配对广播应设置 30S 超时时间限制；遥控器发送完配对码 0xA0 后， 若组合键仍然为按下状态， 则改发红外码0xAE， 直到组合键松开； 但每次按下【OK】 +【返回】 组合键只启动一次对码广播流程。流程如下图所示：



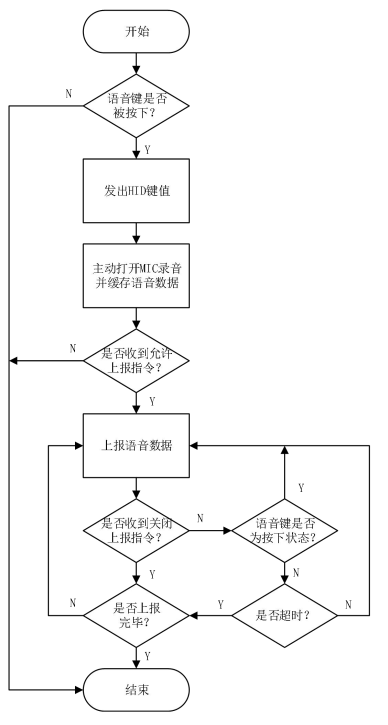
### 无感对码流程

TV开机进入初始向导界面，可扫描配对蓝牙遥控器。遥控器在未配对情况下，按下任意键可发出不定向配对广播以及对应的红外码值，（语音键还是配对红外码值）。遥控器在已配对情况下，会根据回连是否超时，后续按照一定的策略发出定向配对广播以及对应的红外码值，（语音键还是配对红外码值）；遥控器按“返回”+“OK”也可自动触发扫描配对。

详细流程请参考《智能硬件所-TCL蓝牙遥控器技术规范V1.5》 3.8章节。

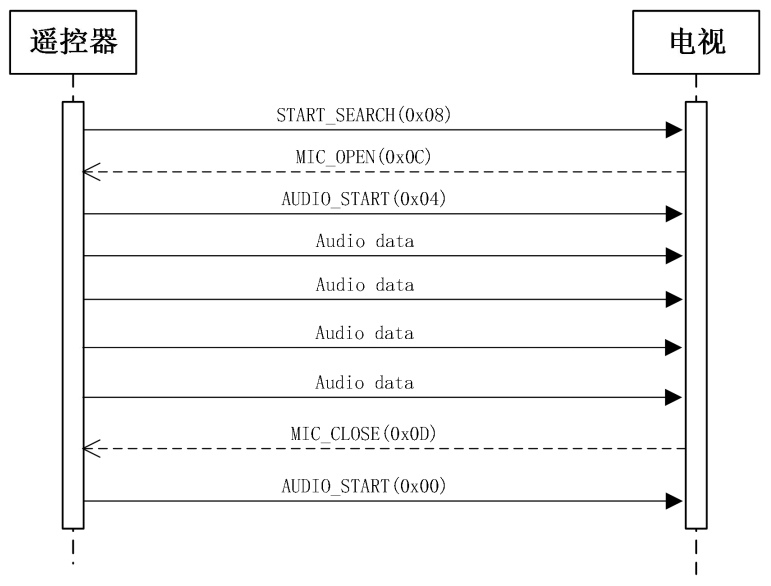
## 语音业务

蓝牙连接状态下， 遥控器语音按键触发的逻辑流程如下图所示。

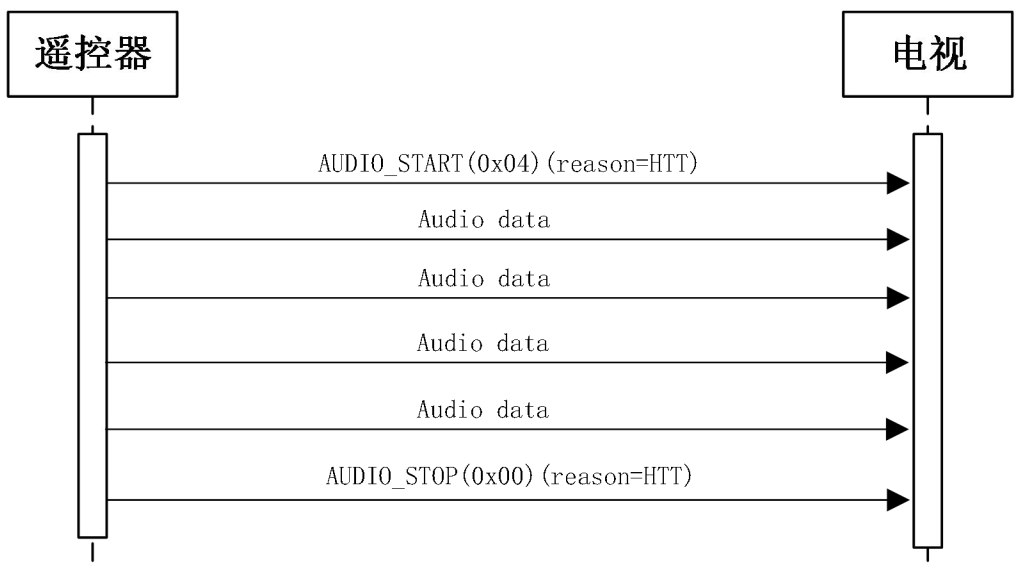


语音上报支持Google Voice Profile，包括三种模式：on-request、 HTT、 PTT 。

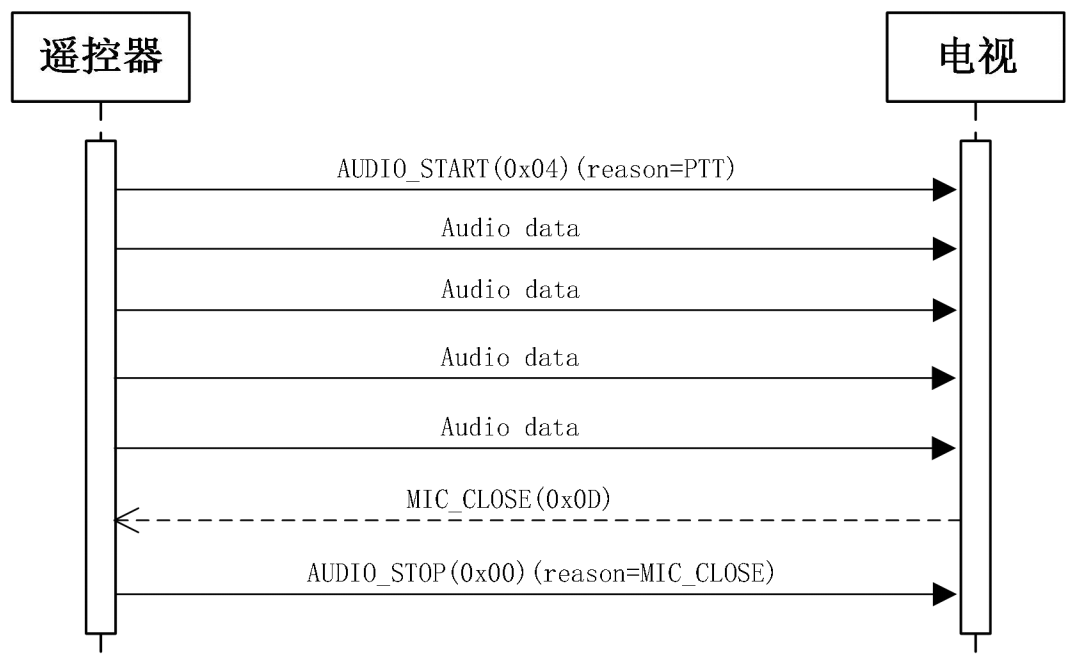
On-Request 时序图：



HTT 时序图：

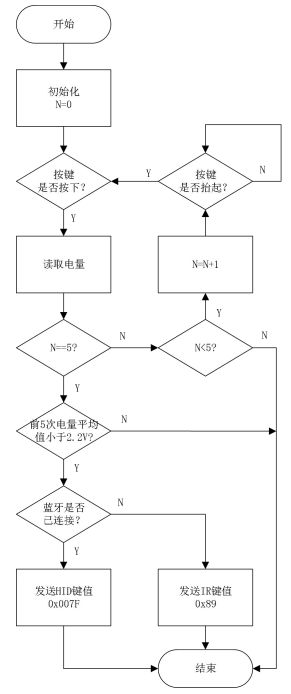


PTT 时序图：



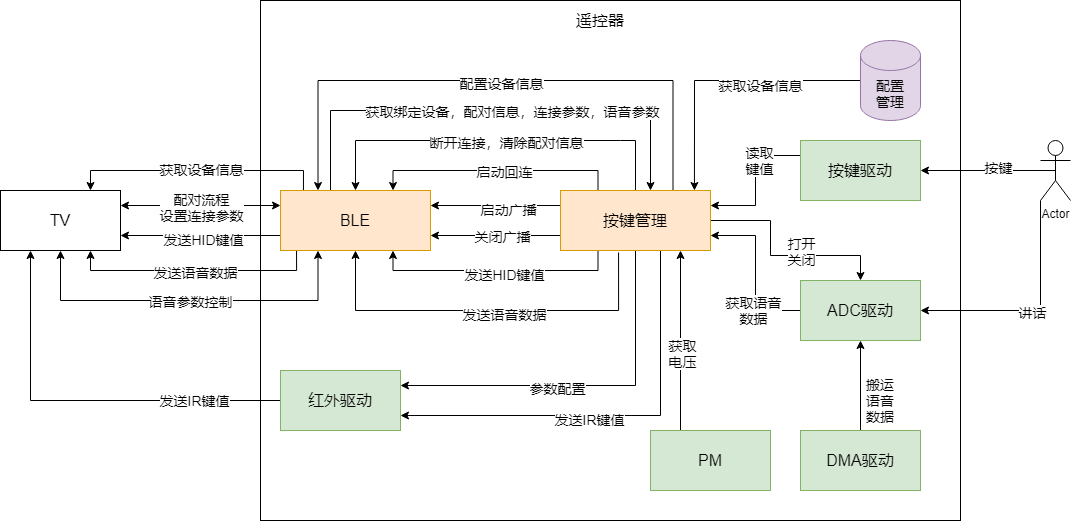
## 低电量提示

流程如下图所示：



# 架构设计

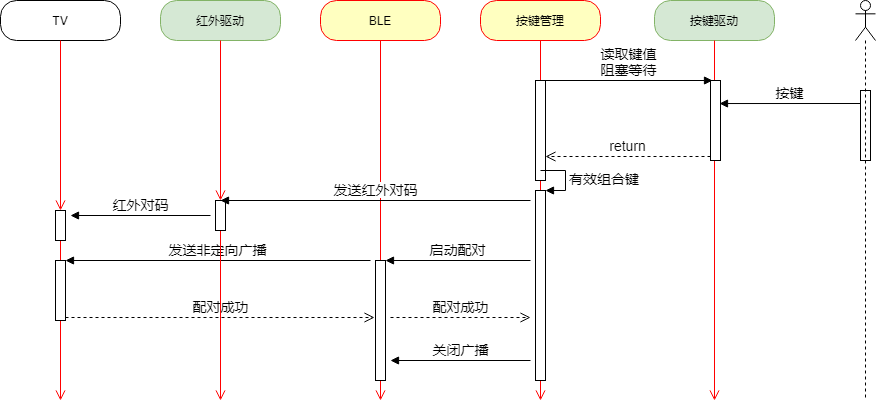
遥控器业务软件架构如下图所示：



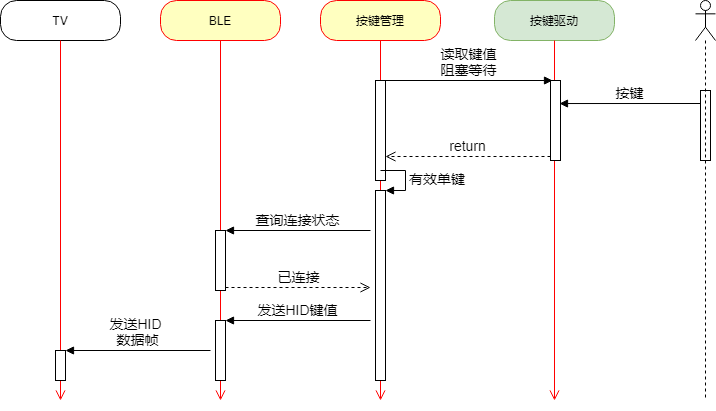
|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 类型 | 交付范围 | 描述 |
| BLE | 任务 | SDK | 实现BLE蓝牙协议栈，与TV端蓝牙对接；实现遥控器与TV的数据和管理通道； |
| 按键管理 | 任务 | DEMO | 遥控器业务逻辑实现 |
| 配置管理 | 配置文件 | DEMO | 存储遥控器配置信息 |
| 按键驱动 | 驱动 | SDK | 提供API接口获取按键数据 |
| ADC驱动 | 驱动 | SDK | 提供API接口获取语音压缩数据，支持算法：SBC/ ADPCM |
| DMA驱动 | 驱动 | SDK | 给ADC模块搬运语音数据 |
| PM | 模块组件 | SDK | 提供API接口获取电压 |
| 红外驱动 | 驱动 | SDK | 提供API接口配置红外参数和发送IR键值 |

# 运行设计

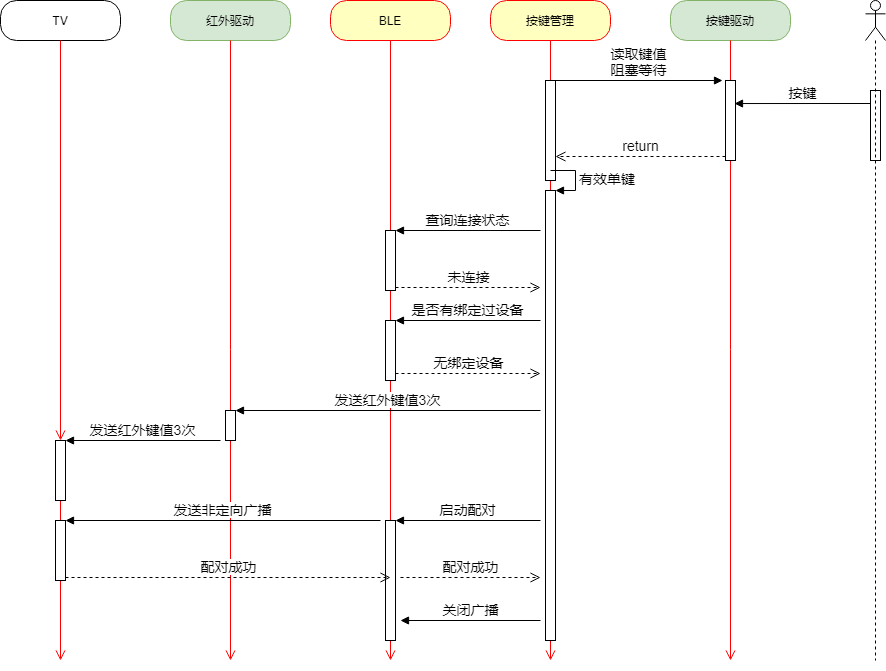
## 配对流程



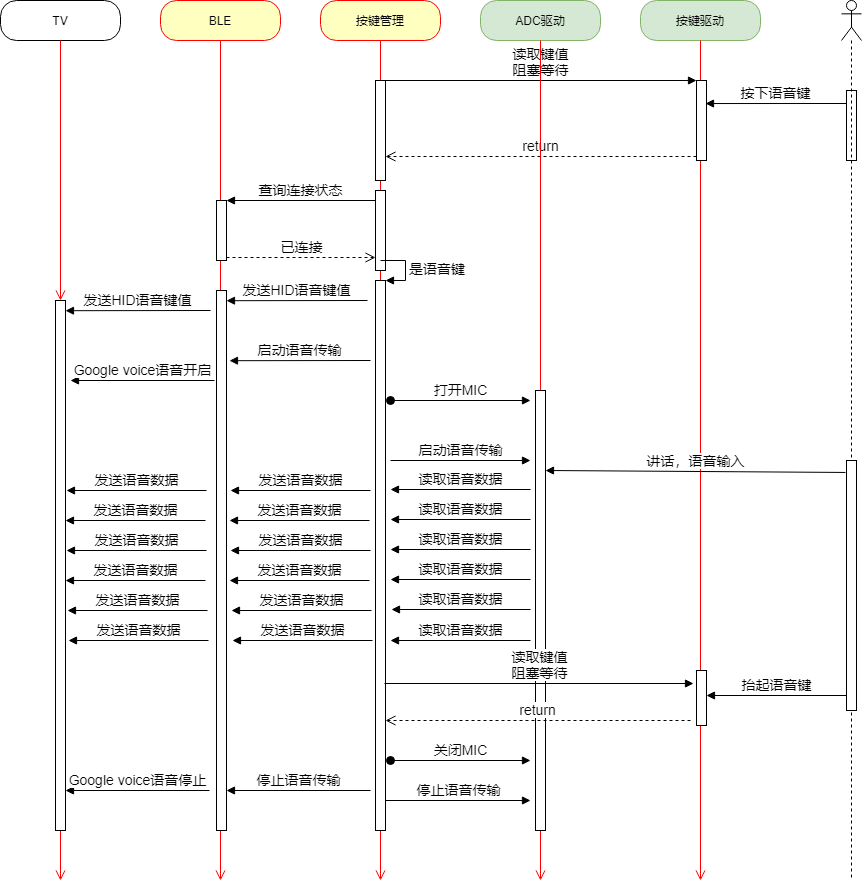
## 蓝牙按键流程



## 无感对码+红外按键流程



## 蓝牙语音传输流程



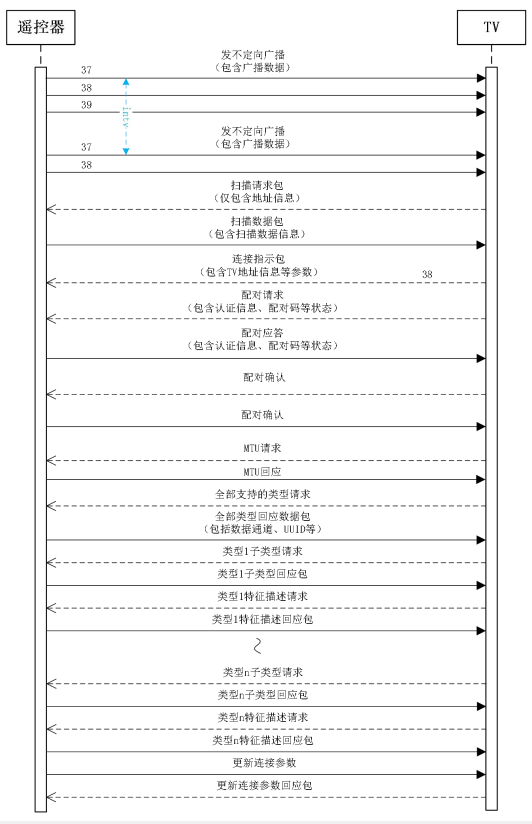
# 模块描述

## BLE

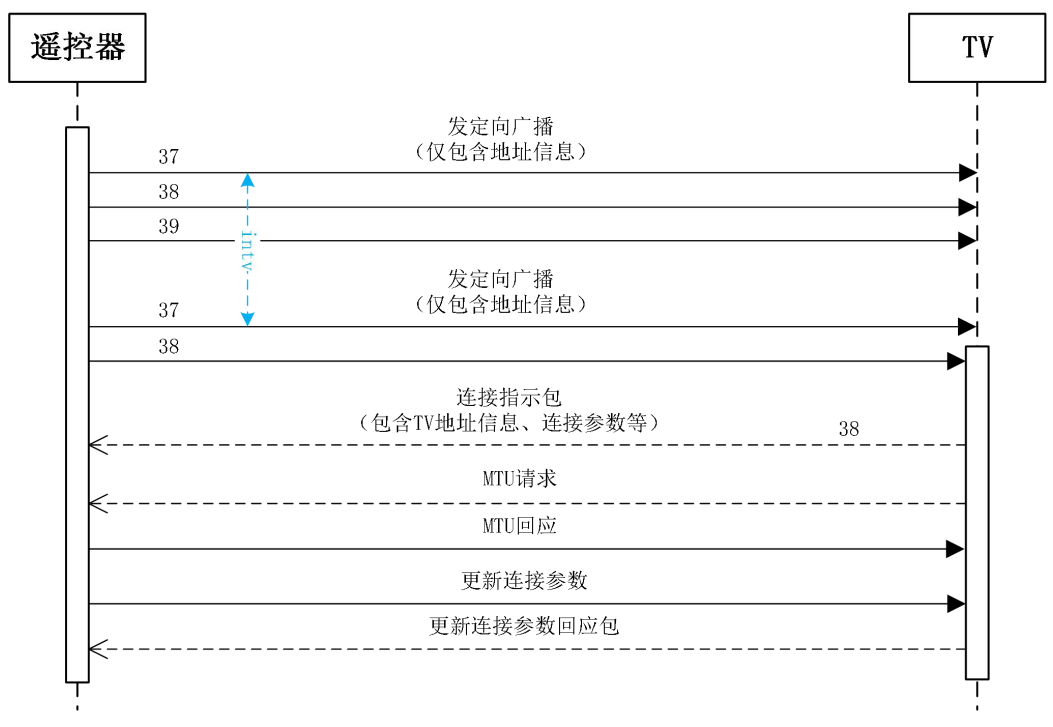
### TV端接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口 | 类型 | 方向 | 描述 |
| 配对流程 | 蓝牙协议 | 双向 |  |
| 设置连接参数 | 蓝牙协议 | TV到遥控器 |  |
| 发送HID键值 | 蓝牙协议 | 遥控器到TV | HID OVER GATT Profile (HOGP) |
| 下发语音控制 | 蓝牙协议 | TV到遥控器 | Google Voice Profile  Write Characteristic 通道 |
| 上报语音控制 | 蓝牙协议 | 遥控器到TV | Google Voice Profile  Control Characteristic 通道 |
| 发送语音数据 | 蓝牙协议 | 遥控器到TV | Google Voice Profile  Read Characteristic 通道 |
| 获取设备信息 | 蓝牙协议 | TV到遥控器 | Device Information Service（DIS） |

非定向广播匹配流程



定向广播匹配流程



设置连接参数

|  |  |
| --- | --- |
| 连接参数 | 参数配置 |
| 最小连接间隔 | 0x0A（对应 12.5ms） |
| 最大连接间隔 | 0x0A（对应 12.5ms） |
| 连接潜伏值 | 64 |
| 超时监视时间 | 500（对应 5s） |

TV下发语音控制

|  |  |  |
| --- | --- | --- |
| CMD | Msg | 描述 |
| 0x0A | GET CAPS | 获取遥控器相关能力信息 |
| 0x0C | MIC OPEN | 麦克风打开请求 |
| 0x0D | MIC CLOSE | 麦克风关闭请求 |
| 0x0E | MIC EXTEND | 启动音频传输时间超时 |

遥控器上报语音控制

|  |  |  |
| --- | --- | --- |
| CMD | Msg | 描述 |
| 0x00 | AUDIO STOP | 音频关闭 |
| 0x04 | AUDIO START | 音频启动 |
| 0x08 | START SEARCH | “ 助手” 按钮通知 |
| 0x0A | AUDIO SYNC | 音频同步通知 |
| 0x0B | CAPS RESP | 能力集请求查询回应 |
| 0x0C | MIC OPEN ERROR | 麦克风打开失败回应 |

能力集

|  |  |
| --- | --- |
| GET CAPS 净荷 | |
| version (2) | The highest spec version that Android TV Device supports.  0x100：version 1.0 |
| legacy\_0x0003 (2) | The legacy constant to comply with spec 0.4e.  0x0003: constant value that is used for compatibility with the previous spec. |
| supported assistant interaction models (1) | Supported Assistant interaction model. The value of this parameter depends on the platform.  0x00: only On-request model is supported;  0x01: Press-to-Talk and On-request models are supported;  0x03: Hold-to-Talk, Press-to-Talk and On-request models are supported. |
| CAPS\_RESP净荷 | |
| version (2) | The spec version that is implemented on the Remote Device.  0x0100: version 1.0;  0x0004: version 0.4e |
| codecs supported (1) | Audio codecs that are supported on the Remote Device.  0x01: [0000\_0001B] ADPCM 8khz/16bit;  0x02: [0000\_0010B] ADPCM 16khz/16bit (recommended);  0x03: [0000\_0011B] ADPCM 8khz/16bit & 16khz/16bit. The value should be used if the Remote Device supports |
| assistant interaction model (1) | Assistant interaction model that is used by this Remote Device. The returned value should be aligned with “supported assistant interaction models” value provided by the Android TV Device.  0x00:​ On-request model (should be supported by all remotes);  0x01:​ Press-to-Talk model enabled;  0x03:​ Hold-to-Talk model enabled. |
| audio frame size (2) | The desired audio packet size. The value is used in audio frame counting and usually matches the maximum payload size of a single notification, but can be any arbitrary number.  Some examples:  0x0014:​ 20 bytes (default);  0x00A0:​ 160 bytes (recommended for 16kHz audio codec and 20ms connection interval). |
| extra configuration (1) | 0x01: [0000\_0001B] If set, host will attempt to enable DLE (and negotiate a new ATT\_MTU size) which would allow to transfer an entire audio frame in a single bluetooth packet.  It’s recommended for the Remote to implement the “GATT Client” role and negotiate ATT MTU and send DLE before sending CAPS\_RESP message. Then there is no need to request DLE from the host side. |
| reserved(1) |  |
| remote fw data (any) | Optional field that remote can use to export data about the firmware installed on the remote. This information will be attached to any bug report and could help to identify problems specific to the FW version of the remote. |

DIS消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | UUID | Property | Field | Value | Length |
| Device Information Server | 0x180A |  |  |  |  |
| Manufacturer Name Characteristic | 0x2A29 | Read | Manufacturer Name | 厂商名  （ 例） “ TCL BT” | 厂商名长度  （ 例） 6B |
| Model Number String Characteristic | 0x2A24 | Read | Model Number | 类型名， 与广播名一致  （ 例） “ TCL\_RC802D2” | 类型名长度  (例)11B |
| Serial Number String Characteristic | 0x2A25 | Read | Serial Number | 序列号， 软件发布日期  （ 例） “ 20220101” | 序列号长度  （ 例） 8B |
| Hardware Revision Characteristic | 0x2A27 | Read | Hardware Revision | 硬件 PCB 版本  （ 例） “ 1.1.20210801” | 硬件 PCB 版本长度  （ 例） 12B |
| Firmware Revisiong Characteristic | 0x2A26 | Read | Firmware Revision | SDK 版本号/中间件版本号 | SDK 版本号/中间件版本号长度 |
| Software Revision String Characteristic | 0x2A28 | Read | Software Revision | APP 版本号  （ 例） “APP\_V0003” | APP 版本号长度  （ 例） 9B |
| System ID Characteristic | 0x2A23 | Read | System ID | 0x00 0x01 0x02 0x00 0x00 0x03 0x04 0x05 | 8B |
| IEEE List Characteristic | 0x2A2A | Read | IEEE | IEEE 列表 | 列表长度 |
| PnP ID | 0x2A50 | Read | APP DIS PNP ID | vendor ID + product ID  如 0x01 0x17 0x04 0x04 0x03 0x03 0x00 | 7B |

### 按键管理端接口

|  |  |  |
| --- | --- | --- |
| 接口 | 类型 | 描述 |
| 设置广播参数 | API消息接口 | GAPM\_ACTIVITY\_CREATE\_CMD  不定向广播 |
| 启动广播 | API消息接口 | GAPM\_ACTIVITY\_START\_CMD |
| 关闭广播 | API消息接口 | GAPM\_ACTIVITY\_STOP\_CMD |
| 启动回连 | API消息接口 | GAPM\_ACTIVITY\_CREATE\_CMD  定向广播 |
| 断开连接 | API消息接口 | GAPC\_DISCONNECT\_CMD |
| 清除配对信息 | API消息接口 | GAPM\_ACTIVITY\_DELETE\_CMD |
| 获取绑定设备 | API消息接口 | GAPC\_GET\_DEV\_INFO\_CFM |
| 获取配对信息 | API消息接口 | GAPC\_GET\_INFO\_CMD |
| 获取连接参数 | API消息接口 | GAPC\_GET\_INFO\_CMD |
| 获取语音参数 | API消息接口 | Google Voice Profile  MIC EXTEND启动音频传输时间超时 |
| 发送HID键值 | API消息接口 | HID OVER GATT Profile (HOGP) |
| 发送语音数据 | API消息接口 | Google Voice Profile  Read Characteristic  SEND VOICE |
| 配置设备信息 | API接口 |  |

定向广播参数

|  |  |
| --- | --- |
| 广播参数 | 参数配置 |
| 最小广播间隔（intv\_min） | 0x06 （对应 3.75ms） |
| 最大广播间隔（intv\_max） | 0x06 （对应 3.75ms） |
| 广播超时时间（timeout） | 1.28s |
| 码速 | 1M |
| 广播信道 | 0x07 （对应所有广播信道） |

不定向广播

|  |  |
| --- | --- |
| 广播参数 | 参数配置 |
| 最小广播间隔（intv\_min） | 0x20 （对应 20ms） |
| 最大广播间隔（intv\_max） | 0x20 （对应 20ms） |
| 广播超时时间（timeout） | 30s |
| 码速 | 1M |
| 广播信道 | 0x07 （对应所有广播信道） |

配置设备信息参数

|  |  |
| --- | --- |
| Field | Value |
| Manufacturer Name | 厂商名（ 例） “ TCL BT” |
| Model Number | 类型名， 与广播名一致（ 例） “ TCL\_RC802D2” |
| Serial Number | 序列号， 软件发布日期（ 例） “ 20220101” |
| Hardware Revision | 硬件 PCB 版本（ 例） “ 1.1.20210801” |
| Firmware Revision | SDK 版本号/中间件版本号 |
| Software Revision | APP 版本号（ 例） “APP\_V0003” |
| System ID | 0x00 0x01 0x02 0x00 0x00 0x03 0x04 0x05 |
| IEEE | IEEE 列表 |
| vendor ID | 厂商ID |
| product ID | 产品ID |
| Device Address | 6字节MAC地址 |
| assistant interaction model | 0x00:​ On-request model (should be supported by all remotes);  0x01:​ Press-to-Talk model enabled;  0x03:​ Hold-to-Talk model enabled. |
| codecs supported (1) | 0x01: [0000\_0001B] ADPCM 8khz/16bit;  0x02: [0000\_0010B] ADPCM 16khz/16bit (recommended);  0x03: [0000\_0011B] ADPCM 8khz/16bit & 16khz/16bit. |
| Google Voice Profile version | Version 1.0 |

## 按键驱动

支持功能：

* 按键事件读取；
* 按下和抬起的识别；

### 接口

enum key state {

KEY\_RELEASE = 0,

KEY\_PRESS = 1,

};

struct key\_value {

enum key\_sate state;

uint32 key;

};

|  |  |  |  |
| --- | --- | --- | --- |
| 接口 | 类型 | 描述 | 参数 |
| 配置键值表 | API函数接口 | 初始化配置键值与行列的对应关系 | 输入  键值表指针；键值表是5行6列，共5\*6=30个键值； |
| 读取键值 | API函数接口 | 阻塞读取键值 | 输出  struct key\_value value; |

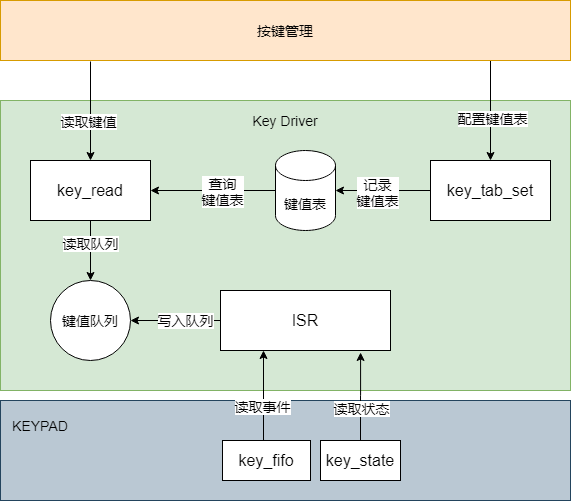
键值输出如上数据结构所示：

|  |  |
| --- | --- |
| 参数 | 描述 |
| state | KEY\_RELEASE 键值抬起；  KEY\_PRESS 键值按下； |
| key1 | 键值 |

遥控器一般是12行3列的按键布局。需要有一个键值转换的逻辑。此逻辑在按键管理模块中完成。

### 模块结构

如下图所示：



**ISR** 处理芯片KEYPAD模块产生的按键中断。中断和事件产生逻辑如下表所示。每按下或者抬起一个键都会产生相应的事件和中断。已按下的键，不会重复产生事件和中断。ISR处理步骤如下：

1. 禁止KEYPAD中断；
2. KEYPAD的key\_fifo缓存key\_event，把fifo读空，把key\_event写入键值队列；
3. 打开KEYPAD中断；
4. 结束返回；

**键值队列** 是固定长度数组，深度可缓存三个事件，支持三键。

**key\_read** 函数，处理步骤如下：

1. 读取键值队列，如果队列空，则阻塞当前任务；
2. 如果队列非空，则读队列头，获取事件；
3. 读取键值表，把事件的行列转换为键值；
4. 函数输出返回；

**key\_tab\_set** 函数，配置键值表；

**键值表** 5行6列的数组，数组保存键值；

## ADC驱动

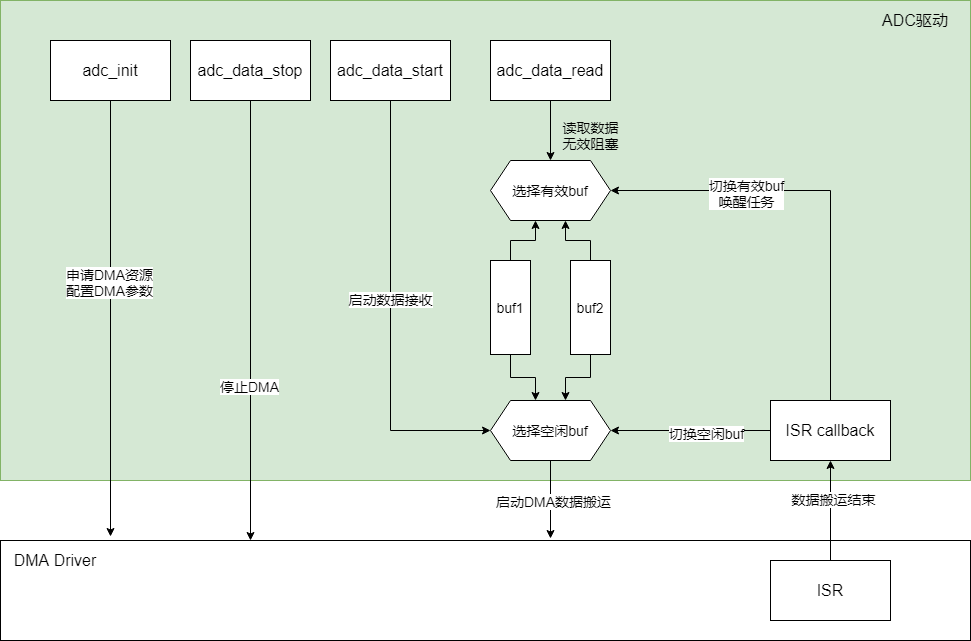
#### 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口 | 类型 | 描述 | 参数 |
| adc\_init | API函数接口 | 初始化ADC，申请DMA资源，配置DMA参数 | NA |
| adc\_data\_start | API函数接口 | 打开采样，启动数据传输 | NA |
| adc\_data\_stop | API函数接口 | 关闭采样，停止数据传输 | NA |
| adc\_data\_read | API函数接口 | 获取语音数据 | uchar8 \*data |

缓存语音数据，待按键管理任务读取。支持codec 16kHz/16bits采样率，即每秒32000bytes数据。BLE单通道带宽是1Mbps，支持160bytes语音帧长，考虑语音压缩和帧头开销，在15ms内要完成一帧发送。即15ms周期内，语音数据缓存是480bytes 。

#### 模块结构

考虑数据处理和数据接收的并行，使用pingpong fifo做数据缓存。模块结构如下图所示。



## DMA驱动

|  |  |  |
| --- | --- | --- |
| 接口 | 类型 | 描述 |
| ms\_dmac\_mgmt\_alloc | API函数接口 | 分配DMA资源 |
| ms\_dmac\_init | API函数接口 | 初始化DMA参数，注册ISR回调函数 |
| ms\_dmac\_start\_chx\_xfer\_it | API函数接口 | 启动DMA数据搬运 |
| ms\_dmac\_stop\_chx\_xfer | API函数接口 | 停止DMA数据搬运 |

## PM

|  |  |  |
| --- | --- | --- |
| 接口 | 类型 | 描述 |
| 获取电压值 | API函数接口 |  |

## 红外驱动

|  |  |  |
| --- | --- | --- |
| 接口 | 类型 | 描述 |
| 配置红外参数 | API函数接口 |  |
| 发送红外键值 | API函数接口 |  |

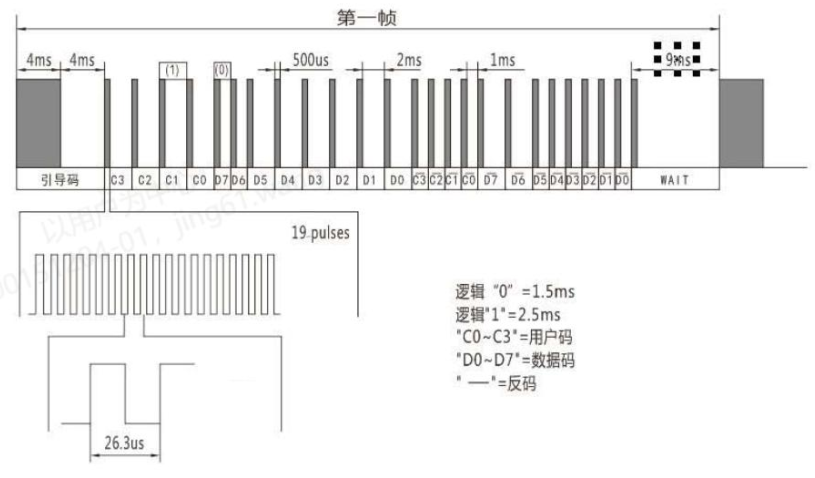
**红外参数**

|  |  |
| --- | --- |
| 参数 | 参数配置（样例） |
| 载波频率 | 38KHz ± 1.5% |
| 脉冲间隔 / 载波周期 | 26.3us |
| 脉冲占比 | 1:2 |
| 调制方式 | PPM |
| 数据发送顺序 | MSB->LSB |
| 引导码 | 4000us（±1.5%） 载波 + 4000us（±1.5%） 低电平 |
| 数据 0 | 500us（±1.5%） 载波 + 1000us（±1.5%） 低电平 |
| 数据 1 | 500us（±1.5%） 载波 + 2000us（±1.5%） 低电平 |
| 结束码 | 500us（±1.5%） 载波 + 8000us（±1.5%） 低电平 |
| 用户码 | 0FH |

**红外键值**

TCL 红外数据包由引导码、 用户码（4bit） 、 数据码（8bit） 、 用户反码（4bit）、

数据反码（8bit） 组成， 码型波形图可参考下图。遥控器在发码时务必确保每组遥控码完整发出， 不允许发不完整码。当需要连续发码时， 每组遥控码间应有等待间隔（即结束码） 。



|  |  |
| --- | --- |
| 参数 | 参数配置 |
| 每组码之间等待间隔 | 6ms-10ms， 典型值为 8ms |
| 用户码 | 4bit, 取 1111 |
| 数据码 | 8bit， 参考 “红外键值表” |

## 配置管理

使用Flash分区存储管理配置信息。

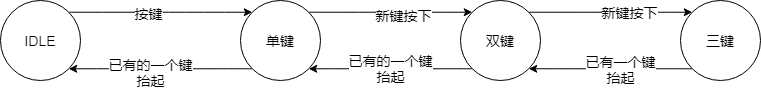
|  |  |  |
| --- | --- | --- |
| 接口 | 类型 | 描述 |
| 获取配置信息 | API函数接口 | 输出配置参数结构体指针  struct device\_info \*info;  结构体内容如下表所示 |

|  |  |
| --- | --- |
| Field | Value |
| Manufacturer Name | 厂商名（ 例） “ TCL BT” |
| Model Number | 类型名， 与广播名一致（ 例） “ TCL\_RC802D2” |
| Serial Number | 序列号， 软件发布日期（ 例） “ 20220101” |
| Hardware Revision | 硬件 PCB 版本（ 例） “ 1.1.20210801” |
| Firmware Revision | SDK 版本号/中间件版本号 |
| Software Revision | APP 版本号（ 例） “APP\_V0003” |
| System ID | 0x00 0x01 0x02 0x00 0x00 0x03 0x04 0x05 |
| IEEE | IEEE 列表 |
| vendor ID | 厂商ID |
| product ID | 产品ID |
| Device Address | 6字节MAC地址 |
| assistant interaction model | 0x00:​ On-request model (should be supported by all remotes);  0x01:​ Press-to-Talk model enabled;  0x03:​ Hold-to-Talk model enabled. |
| codecs supported (1) | 0x01: [0000\_0001B] ADPCM 8khz/16bit;  0x02: [0000\_0010B] ADPCM 16khz/16bit (recommended);  0x03: [0000\_0011B] ADPCM 8khz/16bit & 16khz/16bit. |
| Google Voice Profile version | Version 1.0 |
| 键值表1 | 键值与芯片KEYPAD行列的对应关系 |
| 键值表2 | 遥控器12\*3键值与键值表1的对应关系 |
| 键值表3 | HID report map |

## 按键管理

实现遥控器业务逻辑描述的业务流程。

**按键识别逻辑** 如下图状态机所示。



三键状态用于产出模式。

**语音业务传输** 任务处理。任务创建条件：

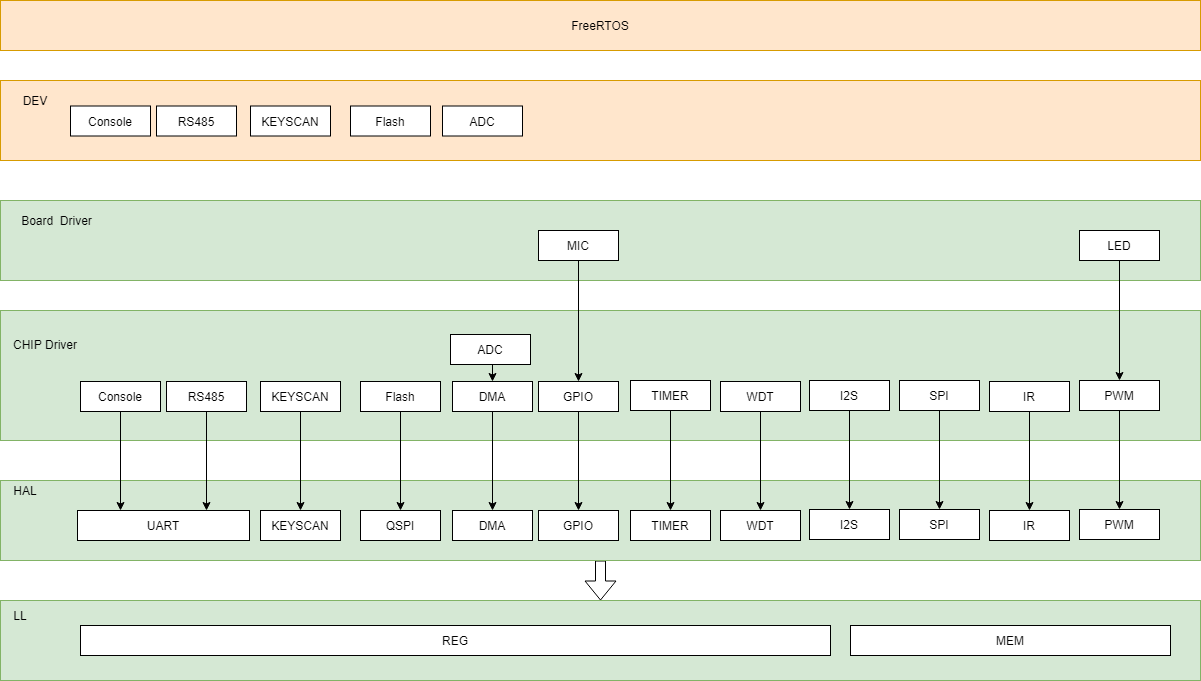
* HTT、PTT模式下，检测到语音键按下；
* On-request模式下，收到MIC\_OPEN消息；

任务结束条件：

* 设置定时器，超时结束语音传输，结束任务；
* HTT模式下，语音键抬起；
* On-request、PTT模式下，收到BLE的MIC\_CLOSE消息；

# 驱动分层设计

驱动架构如下图所示。



LL屏蔽芯片版本寄存器和表项差异，如：

* 寄存器和表项起始地址；
* 寄存器和表项地址偏移；
* 寄存器和表项的定义；
* 对HAL提供统一的寄存器和表项操作接口；

HAL屏蔽系列芯片管理流程的差异

* 使用LL的寄存器和表项操作接口，实现芯片各模块的配置流程；
* 对Driver提供芯片各模块的功能接口；

CHIP Driver提供芯片驱动业务接口

* 对APP提供统一的业务接口；
* 调用HAL的芯片模块功能接口，实现各接口的业务逻辑；
* 实现驱动资源管理，如：硬件资源分配、缓存管理等；
* 业务模块与HAL模块可以是多对一关系，如Console和RS485都是使用HAL UART的功能；
* 定义驱动的ISR；
* 提供接口，DEV可注册ISR回调；

Board Driver提供板级驱动业务接口

* 芯片外的设备驱动，调用相应的外设接口驱动；如MIC驱动调用GPIO驱动接口；LED驱动调用PWM接口，等；

DEV 基于FreeRTOS开发的设备驱动

* 实现任务间同步互斥的驱动接口；
* 调用Driver的接口实现业务；
* 注册ISR回调，获取中断处理；