**软件详细设计书-遥控器 SDK**

**Figure项目软件详细设计书**

版本号:V1.0.0



**中科创达软件股份有限公司**

**历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 作者/修订者 | 修改内容 | 评审者 |
| V1.0.0 | 2022/10/12 | 廖涛 | 初始版 |  |
|  |  |  |  |  |
|  |  |  |  |  |

**正式审批**

|  |  |  |
| --- | --- | --- |
| 职责 | 签 名 | 日 期 |
| 评审 |  |  |
| 批准 |  |  |

# 目录

**[目录 3](#_Toc954566257)**

**[1. 引言 4](#_Toc1692238309)**

[1.1. 编写目的 4](#_Toc175838495)

[1.2. 参考资料 4](#_Toc380087193)

**[2. 术语和定义 4](#_Toc1509086573)**

**[3. 软件功能概述 4](#_Toc1436161341)**

[3.1. 功能概述 4](#_Toc1994949554)

[3.2. 模块构成 4](#_Toc454973467)

[3.3. 需求与模块对应关系 4](#_Toc1697276549)

**[4. 常量和变量声明 5](#_Toc1101675942)**

[4.1. 常量 5](#_Toc263552760)

[4.2. 变量 7](#_Toc1419957206)

**[5. 详细设计说明 7](#_Toc234992131)**

[5.1. 软件架构设计 7](#_Toc290318884)

[5.2. 软件模块划分 9](#_Toc306637404)

[5.3. 软件数据结构 10](#_Toc1841579875)

[5.4. 软件接口定义 11](#_Toc2122113990)

**[6. 源程序文件设计 11](#_Toc961420554)**

[6.1. 遥控器服务端目录结构 11](#_Toc928291050)

[6.2. 遥控器 Android SDK 目录结构 13](#_Toc318981895)

[6.3. 遥控器 iOS SDK 目录结构 14](#_Toc1009526353)

**[7. 环境配置 16](#_Toc1988603571)**

1. **引言**
   1. **编写目的**

该详细设计说明书的编写目的主要为说明遥控器服务端、Android设备遥控器SDK、iOS设备遥控器SDK 开发的相关架构，以及核心功能逻辑，包括程序描述、输入/输出、算法、功能逻辑等，便于后续功能迭代和维护。

* 1. **参考资料**

无

1. **术语和定义**

|  |  |
| --- | --- |
| **缩写和术语** | **定义** |
| 遥控器服务端 | 副驾与后排娱乐系统里运行的软件，用于接受遥控器客户端发送过来的操作指令，并且对指令做出响应的应用程序。 |
| SDK | 遥控器SDK，为遥控器客户端提供遥控器基础服务的套件，遥控器开发商可以基于此套件开发出符合用户使用习惯的应用程序。如果没有写明是Android SDK 或者 iOS SDK 的情况下，通指包含Android 与 iOS的SDK。 |

1. **软件功能概述**
   1. **功能概述**

为解决车内驾驶员、乘客控制副驾或者后排娱乐系统。而开发的一款基于以太网通信的远程遥控器系统，简称车载娱乐系统遥控器系统。此系统包括遥控器服务端、Android设备遥控器SDK、iOS设备遥控器SDK。此遥控器系统的服务端由创通联达完成开发，而针对Android与iOS的手持设备，提供SDK，开发遥控器客户端的应用开发商，可以基于我们提供的SDK，开发符合用户使用的遥控器客户端APP。

* 1. **模块构成**

遥控器项目主要分为遥控器服务端、遥控器 Android SDK、遥控器 iOS SDK 三个子项目，其中遥控器 Android SDK 与遥控器 iOS SDK 为遥控器客户端 SDK 在 Android 与 iOS 平台上的不同实现。

* 1. **需求与模块对应关系**

|  |  |  |  |
| --- | --- | --- | --- |
| *需求编号* | *需求简述* | *模块* | *备注* |
| TC-RID: 1201-0206001 | 遥控器服务端 | RemoteControlServer |  |
| TC-RID: 1201-0206002 | 遥控器 Android 客户端 SDK | RemoteControlClients/Android |  |
| TC-RID: 1201-0206003 | 遥控器 iOS 客户端 SDK | RemoteControlClients/iOS |  |

1. **常量和变量声明**
   1. **常量**
      1. 遥控器命令 Code定义

package com.thundercomm.remotecontrolservice.Util;  
  
public class RemoteCommandConst {  
 public static final int REMOTE\_COMMAND\_TYPE\_CONNECT = 0x01;  
 public static final int REMOTE\_COMMAND\_TYPE\_DISCONNECT = 0x02;  
 public static final int REMOTE\_COMMAND\_TYPE\_MOUSE = 0x03;  
 public static final int REMOTE\_COMMAND\_TYPE\_MOUSE\_MOVE = 0x04;  
 public static final int REMOTE\_COMMAND\_TYPE\_KEY = 0x05;  
 public static final int REMOTE\_COMMAND\_TYPE\_INPUT = 0x06;  
 public static final int REMOTE\_COMMAND\_TYPE\_HEARTBEAT = 0x07;  
 public static final int REMOTE\_COMMAND\_TYPE\_CONNECTEDSTATUS = 0x08;  
 public static final int REMOTE\_COMMAND\_TYPE\_INPUTSYNC = 0x09;  
 public static final int REMOTE\_COMMAND\_TYPE\_DISCONNECTCONFIRM = 0x10;  
 public static final int REMOTE\_COMMAND\_TYPE\_FSEDISCONNECT = 0x11;  
 public static final int REMOTE\_COMMAND\_TYPE\_FSEDISCONNECTCONFIRM = 0x12;  
 public static final int REMOTE\_COMMAND\_TYPE\_SETVOLUME = 0x13;  
 public static final int REMOTE\_COMMAND\_TYPE\_ON\_VOLUME\_CHANGE = 0x14;  
 public static final int REMOTE\_COMMAND\_TYPE\_ERROR\_INFORMATION = 0x15;  
 public static final int REMOTE\_COMMAND\_TYPE\_SEEK = 0x16;  
 public static final int REMOTE\_COMMAND\_TYPE\_QUERY\_VOLUME = 0x17;  
 public static final int REMOTE\_COMMAND\_TYPE\_GET\_APP\_LIST = 0x18;  
 public static final int REMOTE\_COMMAND\_TYPE\_LAUNCH\_APP = 0x19;  
 public static final int REMOTE\_COMMAND\_TYPE\_GET\_QUICK\_SETTING\_LIST = 0x20;  
 public static final int REMOTE\_COMMAND\_TYPE\_CLICK\_QUICK\_SETTING = 0x21;  
 public static final int REMOTE\_COMMAND\_TYPE\_GET\_APP\_ICON = 0x22;  
 public static final int REMOTE\_COMMAND\_TYPE\_SWIPE = 0x23;  
}

* + 1. 遥控器错误消息 Code 定义

package com.thundercomm.remotecontrol.Util;

public class RemoteErrorConst {  
 public static final int ERROR\_CODE\_TOKEN\_ERROR = 1;  
 public static final int ERROR\_CODE\_DIGIST\_DISMATCH = 2;  
 public static final int ERROR\_CODE\_TOKEN\_EXPIRE = 3;  
 public static final int ERROR\_CODE\_TOKEN\_CHANGE = 4;  
 public static final int ERROR\_CODE\_CLIENT\_NULL = 5;  
 public static final int ERROR\_CODE\_CLIENT\_CONNECT\_MAX\_LIMIT = 6;  
 public static final int ERROR\_CODE\_APP\_LAUNCH\_INFO\_ERROR = 7;  
 public static final int ERROR\_CODE\_APP\_ICON\_INFO\_ERROR = 8;  
 public static final int ERROR\_CODE\_SERVER\_CONNECT\_FAILED = 60;  
 public static final int ERROR\_CODE\_SERVER\_NOT\_CONNECTED = 61;  
 public static final int ERROR\_CODE\_RECEIVE\_DATA\_IS\_NULL = 62;  
 public static final int ERROR\_CODE\_RECEIVE\_DATA\_CAN\_NOT\_PARSE = 63;  
 public static final int ERROR\_CODE\_TOKEN\_INVALID = 64;  
 public static final int ERROR\_CODE\_RESULT\_CAN\_NOT\_PARSE = 65;  
 public static final int ERROR\_CODE\_OTHER = 99;  
}

* + 1. 遥控器 SDK 命令定义

package com.thundercomm.remotecontrol.Util;  
  
public class RemoteCommandConst {  
 public static final String IntentConnectionCommand = "IntentConnectionCommand";  
 public static final String IntentDisConnectionCommand = "IntentDisConnectionCommand";  
 public static final String IntentSendKeyCodeCommand = "IntentSendKeyCodeCommand";  
 public static final String IntentQueryVolumeCommand = "IntentQueryVolumeCommand";  
 public static final String IntentSetVolumeCommand = "IntentSetVolumeCommand";  
 public static final String IntentGetAppListCommand = "IntentGetAppListCommand";  
 public static final String IntentStartAppCommand = "IntentStartAppCommand";  
 public static final String IntentGetSettingListCommand = "IntentGetSettingListCommand";  
 public static final String IntentChangeSettingCommand = "IntentChangeSettingCommand";  
 public static final String IntentGetAppIconCommand = "IntentGetAppIconCommand";  
 public static final String IntentMouseControlCommand = "IntentMouseControlCommand";  
 public static final String IntentSwipeControlCommand = "IntentSwipeControlCommand";  
  
 public static final String VALUE\_GET\_LIST\_WITHOUT\_IMG = "0";  
 public static final String VALUE\_GET\_LIST\_WITH\_IMG = "1";  
}

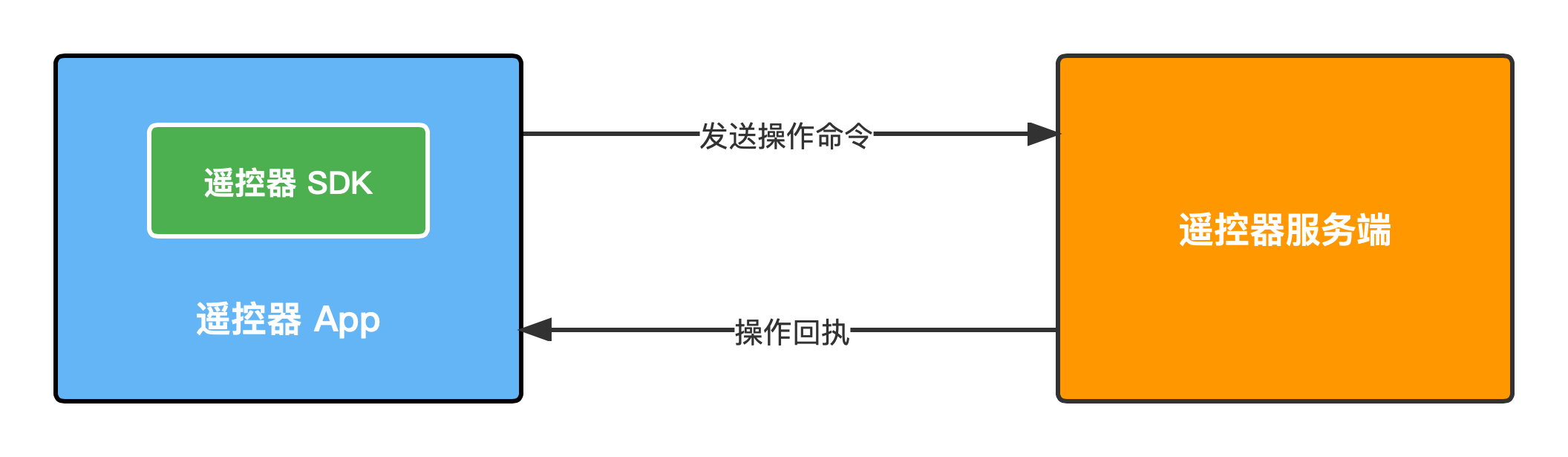
* + 1. 遥控器 SDK 按键命令键码 Key 定义

package com.thundercomm.remotecontrol;  
  
public class RemoteControlConst {  
 public static final int REMOTE\_KEY\_POWER = 0;  
 public static final int REMOTE\_KEY\_SCREEN = 1;  
 public static final int REMOTE\_KEY\_HOME = 2;  
 public static final int REMOTE\_KEY\_BACK = 3;  
  
 public static final int REMOTE\_KEY\_DIRECT\_UP = 4;  
 public static final int REMOTE\_KEY\_DIRECT\_DOWN = 5;  
 public static final int REMOTE\_KEY\_DIRECT\_LEFT = 6;  
 public static final int REMOTE\_KEY\_DIRECT\_RIGHT = 7;  
 public static final int REMOTE\_KEY\_ACTION\_OK = 8;  
}

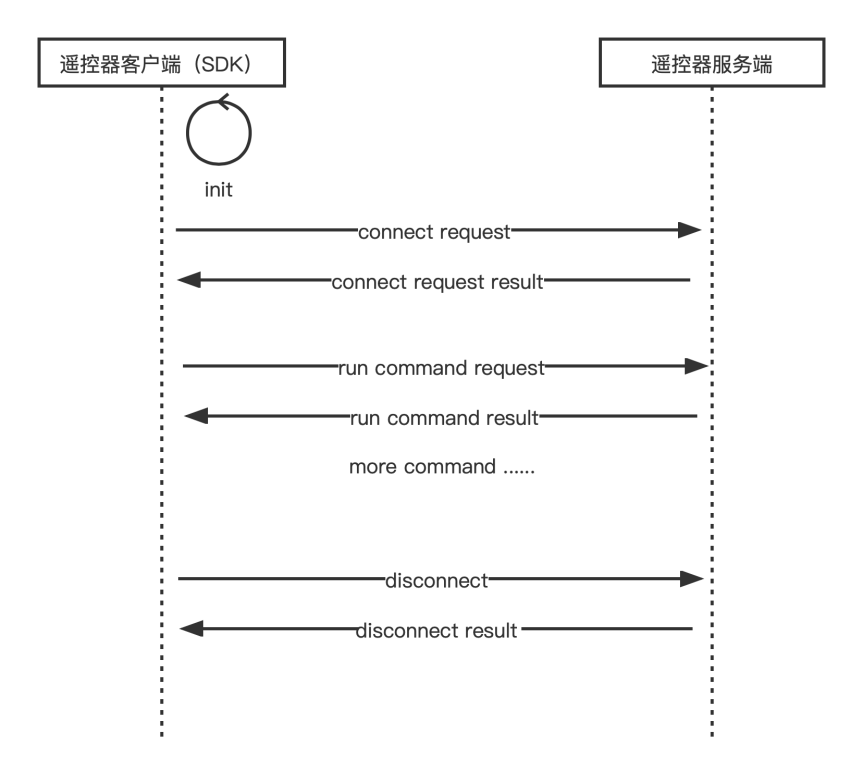
* 1. **变量**

无

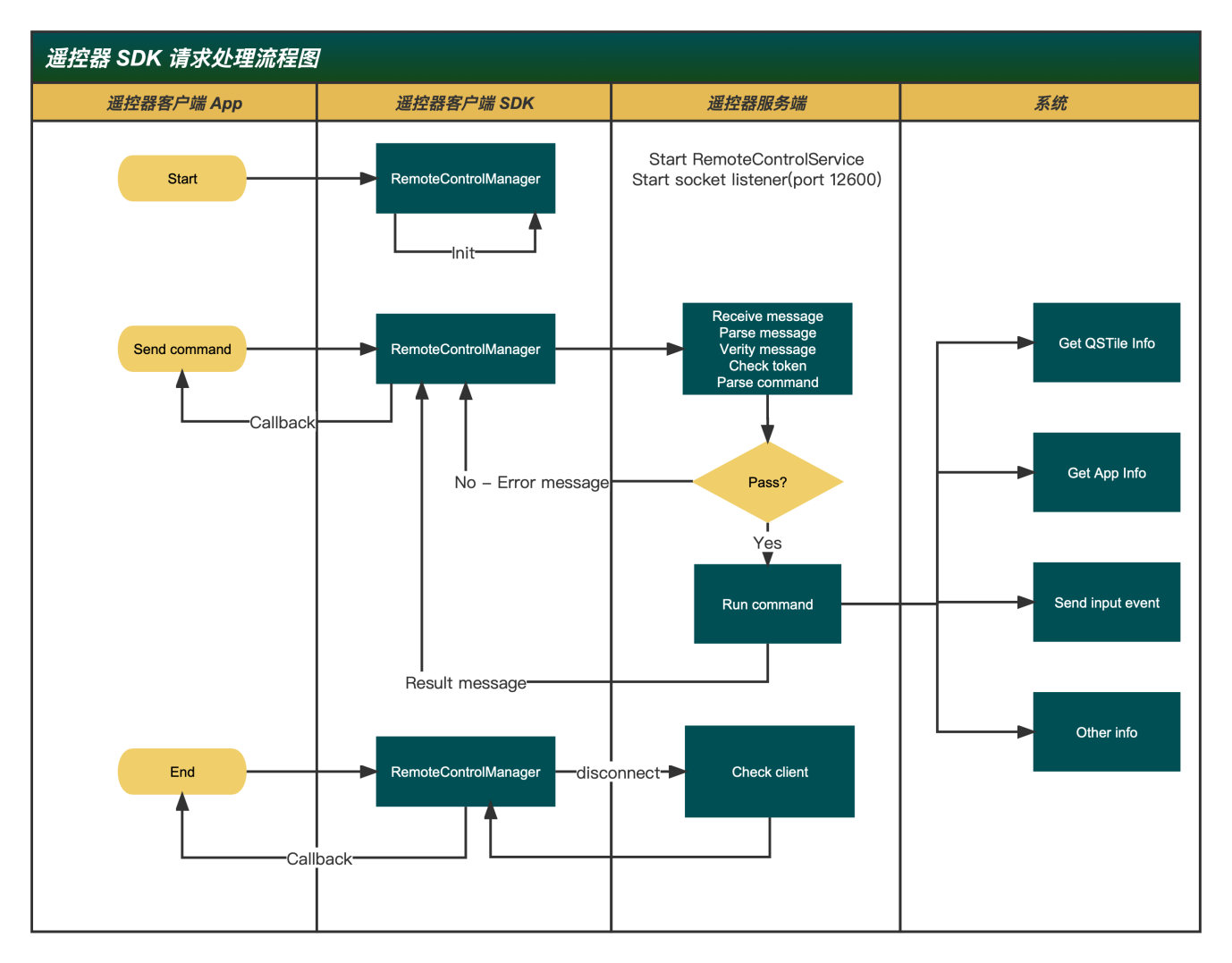
1. **详细设计说明**
   1. **软件架构设计**
      1. 客户端与服务端通信示意图



* + 1. 客户端与服务端通信时序图



* + 1. 客户端与服务端通信流程图



* + 1. 服务端整体架构

遥控器服务端，RemoteControlService 继承了 android.app.Service，并添加到 SystemServer 中，在系统启动时由 SystemServer 启动遥控器常驻服务。

RemoteControlService 启动后监听网络状态的变化，当设备连接到网络后，创建 Socket 服务端（RemoteSocketService），并监听客户端连接并接收消息。

当收到客户端的连接请求并验证通过后，创建一个客户端（RemoteControllerClient）并添加到 RemoteControlManager 中进行管理。

当收到客户端发送的命令消息后，由 RemoteControlReader 进行处理，其中包括消息的解析，验证，解密等，然后通过 RemoteCommandFactory 工厂方法创建对应命令实例并执行，命令执行完成后发送回执消息到客户端。

* + 1. 客户端 SDK 整体架构

客户端 SDK 主要通过 RemoteControlManager 进行相关命令操作。

在客户端 App 进行初始化时，需要同时调用 RemoteControlManager.init(Context context) 进行 SDK 初始化操作。

后续通过 RemoteControlManager 的公开方法进行相关命令操作的调用。在调用命令操作前，需要给 RemoteControlManager 设置回调监听（setCallback(RemoteControlCallback callback)），通过回调可以接收命令操作回执、客户端和服务端连接状态变化消息及错误消息。

* 1. **软件模块划分**
     1. 遥控器服务端

遥控器服务端主要分为常驻服务模块（**RemoteControlService**）、消息处理模块（**RemoteControlReader**）、客户端管理模块（**RemoteControlManager**）。

* + - 1. 常驻服务模块

主要包含 RemoteControlService、RemoteSocketService 类。

核心功能如下：

* 常驻系统服务；
* Socket 服务端创建及维持，接收客户端连接请求及消息；
* 实现与设备相关操作功能（包含与 Native 层的交互）；
  + - 1. 消息处理模块

主要包含 RemoteControlReader、RemoteCommandFactory 类。

核心功能如下：

* 客户端消息请求的解析（验证、加解密等）；
* 通过 RemoteCommandFactory 工厂类将消息转换为具体的命令实例；
* 执行相关命令；
  + - 1. 客户端管理模块

主要包含 RemoteControlManager、RemoteControllerClient 类。

核心功能如下：

* 客户端连接处理；
* 客户端管理（连接限制、心跳维护等）；
  + 1. 遥控器 SDK

遥控器 SDK 主要分为控制中心模块（**RemoteControlManager**）、消息发送处理模块（**SocketMessageUtil**、**MessageReader**）。

* + - 1. 控制中心模块

主要包含 RemoteControlManager 。

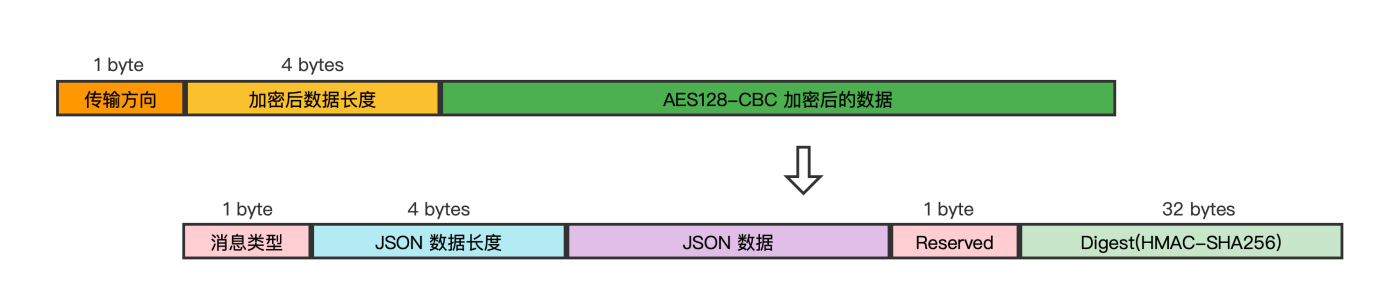
核心功能如下：

* 通过公开方法提供对外调用命令的接口；
* 通过 RemoteCommandFactory 工厂类转换为对应的命令并发送命令消息到服务端；
* 管理客户端连接（连接状态、心跳等）；
  + - 1. 消息发送处理模块

主要包含 SocketMessageUtil、MessageReader 。

核心功能如下：

* SocketMessageUtil 包装、加密及发送消息；
* MessageReader 解析处理消息；
  1. **软件数据结构**
     1. Socket 通信数据结构定义



* + 1. 通信数据加解密及验证方式

通信数据采用 AES128-CBC 进行加解密，加解密 Key 使用固定值。

JSON 数据采用 HMAC-SHA256 进行 Digest 操作后获得 32 bytes 长度数据作为校验数据，在收到数据后，需要以相同的方式及 Key（Key 使用固定值）对 JSON 数据进行 Digest 操作，并且与收到的 Digest 数据进行比较，一致表示数据有效，否则数据无效，需要丢弃。

* + 1. JSON 数据结构定义

{

"source": 0, //请求或响应来源，中控或手持设备

"type": "xxx", //操作类型

"token":"xxxxxxxx", //Token，用以确定客户端唯一性及状态

"value": 1 //请求参数或响应数据

}

* 1. **软件接口定义**

|  |  |  |  |
| --- | --- | --- | --- |
| 命令描述 | 命令方法 | 类型 | 回调方法 |
| 请求连接 | connectRequest | 0x01 | void connectStateCallBack(Boolean connected) |
| 断开链接 | disConnected | 0x02 | void connectStateCallBack(Boolean connected) |
| 设备开关 | sendKeyCode | 0x05 | void sendKeycodeCallBack(Boolean success) |
| 回到主页 |
| 全局返回 |
| 查询音量 | queryVolume | 0x17 | void queryVolumeCallBack(VolumeInfo volumeInfo) |
| 音量设置 | turnUpVolume/ turnDownVolume | 0x13 | void setVolumeCallBack(VolumeInfo volumeInfo) |
| 应用列表 | getAppList | 0x18 | void getAppListCallBack(List<AppInfo> appInfoList, boolean withImage) |
| 应用图标 | getAppIcon | 0x22 | void getAppIconCallBack(AppInfo appInfo) |
| 启动应用 | appLaunch | 0x19 | void appLaunchCallBack(Boolean success) |
| 快捷设置列表 | getQuickSettingsList | 0x20 | void getQuickSettingsCallBack(List<QuickSettingInfo> quickSettingList) |
| 点击快捷设置（切换开关状态） | changeQuickSetting | 0x21 | void changeQuickSettingsCallback(Boolean success) |
| 鼠标操作 | mouseControl | 0x03 | void mouseControlCallBack(Boolean success) |
| 触摸操作 | swipeControl | 0x23 | void swipeControlCallBack(Boolean success, Point screenSize) |
| 心跳信息 | startHeartBeatTimer stopHeartBeatTimer 连接和关闭连接时进行操作 | 0x07 | none |
| 错误消息 | none | 0x15 | void onError(int code, String message) |

1. **源程序文件设计**
   1. **遥控器服务端目录结构**

- com.thundercomm.remotecontrolservice

- bean

- BaseRequestCommandBean

- BaseResponseCommandBean

- CommandMessageBean

- ErrorInfomationBean

- InputCommandBean

- MouseCommandBean

- RemoteControllerRegisterInfo

- SocketRawEvent

- SwipeCommandBean

- VolumeCommandBean

- VolumeCurrentBean

- remotecommand

- BaseCommand

- ConnectionCommand

- DisconnectCommand

- HeartBeatCommand

- RemoteAppIconCommand

- RemoteAppLaunchCommand

- RemoteAppListCommand

- RemoteCommandFactory

- RemoteCommandInterface

- RemoteKeyCommand

- RemoteMouseCommand

- RemoteQuickSettingActionCommand

- RemoteQuickSettingListCommand

- RemoteSwipeCommand

- RemoteVolumeCommand

- RemoteVolumeQueryCommand

- Util

- AesEncryptUtil

- AppListHelper

- BitmapUtil

- ByteDataUtil

- InjectEventHelper

- RemoteCommandConst

- SocketMessageUtil

- VolumeChangeHelper

- QSTileControlManager

- RemoteControllerClient

- RemoteControlManager

- RemoteControlReader

- RemoteControlService

- RemoteSocketService

- TLog

* 1. **遥控器 Android SDK 目录结构**

- com.thundercomm.remotecontrol

- bean

- AppInfo

- BaseMessageBean

- CommandMessageBean

- ErrorResponseInfo

- JSONStringInter

- MouseControlBean

- QuickSettingInfo

- RemoteControllerRegisterInfo

- ResponseInfo

- SocketRawEvent

- SwipeControlBean

- VolumeInfo

- remotecommand

- BaseCommand

- ChangeQuickSettingCommand

- ConnectionCommand

- DisConnectionCommand

- GetAppIconCommand

- GetAppListCommnad

- GetQuickSettingListCommand

- KeycodeCommand

- MouseControlCommand

- QueryVolumeCommand

- RemoteCommandFactory

- RemoteCommandInterface

- SetVolumeCommand

- StartAppCommand

- SwipeControlCommand

- Util

- AesEncryptUtil

- ByteDataUtil

- MessageReader

- RemoteCommandConst

- RemoteErrorConst

- SocketMessageUtil

- RemoteControlCallback

- RemoteControlConst

- RemoteControllerTimerTask

- RemoteControlManager

- TLog

* 1. **遥控器 iOS SDK 目录结构**

- RemoteControlKit

- Util

- SocketMessageUtil.h

- SocketMessageUtil.m

- ByteDataUtil.h

- ByteDataUtil.m

- NSString+Extension.h

- NSString+Extension.m

- NSObject+Extension.h

- NSObject+Extension.m

- Model

- BaseMessageModel.h

- BaseMessageModel.m

- AppInfoModel.h

- AppInfoModel.m

- MouseControlModel.h

- MouseControlModel.m

- RCRegisterInfoModel.h

- RCRegisterInfoModel.m

- SocketRawEvent.h

- SocketRawEvent.m

- CommandMessageModel.h

- CommandMessageModel.m

- VolumeInfoModel.h

- VolumeInfoModel.m

- QuickSettingInfoModel.h

- QuickSettingInfoModel.m

- ErrorMessageModel.h

- ErrorMessageModel.m

- SwipeControlModel.h

- SwipeControlModel.m

- JSONProtocol.h

- RemoteCommand

- RCommandConnect.h

- RCommandConnect.m

- RCommandBase.h

- RCommandBase.m

- RCommandDisconnect.h

- RCommandDisconnect.m

- RCommandHeartBeat.h

- RCommandHeartBeat.m

- RCommandKeycode.h

- RCommandKeycode.m

- RCommandQueryVolume.h

- RCommandQueryVolume.m

- RCommandGetAppList.h

- RCommandGetAppList.m

- RCommandStartApp.h

- RCommandStartApp.m

- RCommandGetQuickSettingList.h

- RCommandGetQuickSettingList.m

- RCommandChangeQuickSetting.h

- RCommandChangeQuickSetting.m

- RCommandMouseControl.h

- RCommandMouseControl.m

- RCommandGetAppIcon.h

- RCommandGetAppIcon.m

- RCommandSetVolume.h

- RCommandSetVolume.m

- RCommandSwipeControl.h

- RCommandSwipeControl.m

- RemoteControlKit.h

- RemoteControlManager.h

- RemoteControlManager.m

- RCommandFactory.h

- RCommandFactory.m

- RCommandActionDefine.h

- RCPublicHeader.h

- RCPrefixHeader.h

- RCKitHeader.h

1. **环境配置**

遥控器服务端：QCS 8250 & Android 12

Android 遥控器SDK：Android 10 与 Android 12中间的版本

iOS 遥控器SDK：
iOS 12 及以上系统