软件概要设计说明书

Software Concept Design Document

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Date | Version | Description | Author |
| 1 | 2014.6.24 | V0.01 | 初始版本 | Zhonglong.Chen |
| 2 | 2014-7-29 | V0.02 | 补充收藏和历史 | Zhonglong.Chen |
|  |  |  |  |  |

目录

[1引言 3](#_Toc393352169)

[1.1背景 3](#_Toc393352170)

[1.2基线 3](#_Toc393352171)

[1.3范围 3](#_Toc393352172)

[1.4定义 3](#_Toc393352173)

[1.5参考资料 3](#_Toc393352174)

[2总体设计 4](#_Toc393352175)

[2.1概述 4](#_Toc393352176)

[2.1.1 需求规定 4](#_Toc393352177)

[2.1.2 运行环境 4](#_Toc393352178)

[2.1.3 拟复用的已有函数库 4](#_Toc393352179)

[2.2设计 5](#_Toc393352180)

[2.2.1设计构思 5](#_Toc393352181)

[2.2.2系统流程 (画出动态运行流程) 5](#_Toc393352182)

[2.2.3结构设计 5](#_Toc393352183)

[2.2.4功能需求与程序的关系 (具体需求与程序模块的对应关系) 7](#_Toc393352184)

[3系统数据结构设计 7](#_Toc393352185)

[3.1数据结构 7](#_Toc393352186)

[3.1.1 TVInfo 7](#_Toc393352187)

[3.1.2 AppInfo 8](#_Toc393352188)

[3.1.3 FavoriteInfo 8](#_Toc393352189)

[3.1.4 HistoryInfo 8](#_Toc393352190)

[4接口设计 8](#_Toc393352191)

[4.1用户接口 8](#_Toc393352192)

[4.2 外部接口 (与外部程序系统之间交互的接口) 9](#_Toc393352193)

[4.2.1 Intent 9](#_Toc393352194)

[4.3 内部接口(内部组件之间调用的接口) 9](#_Toc393352195)

[4.3.1 手机遥控器接口 9](#_Toc393352196)

[4.3.2 扩展命令接口 10](#_Toc393352197)

[4.3.3 我的云端接口 11](#_Toc393352198)

[5运行设计 13](#_Toc393352199)

[5.1 系统初始化（初始化流程图） 13](#_Toc393352200)

[5.2 运行模块组合 13](#_Toc393352201)

[5.3 运行时间 13](#_Toc393352202)

[6出错处理和维护 14](#_Toc393352203)

[6.1 出错信息 14](#_Toc393352204)

[6.2 补救措施 14](#_Toc393352205)

[6.3 系统维护 14](#_Toc393352206)

[7尚待解决的问题 14](#_Toc393352207)

# 1引言

## 1.1背景

本文是Flagship 2k15机种SmartRC应用（MyRemote服务端）的概要设计文档，是SmartRC详细设计和编码的指导文件。

## 1.2基线

规格基线：<http://172.16.144.44/CN_Flagship/Spec/0.Baseline/V1.0/MyRemote/2K15_MyRemote_FRS.xlsx> #297

## 1.3范围

## 1.4定义

MyRemote：手机助手客户端，安装在手机或平板上

SmartRC：手机助手服务端，安装在智能电视上

## 1.5参考资料

1. myRemote\_2k15\_android\_软件概要设计说明书.docx  
   <http://172.16.144.31:9090/ebony_ui/innovation/flagship_tv_2k15/myRemote_2k15_android/doc/myRemote_2k15_android_软件概要设计说明书.docx>
2. MyRemote键值列表.docx  
   <http://172.16.144.31:9090/ebony_ui/innovation/touchpad_RC_demo/doc/MyRemote键值列表.docx>
3. Ebony 2k14机种SmartRC概要设计  
   <http://172.16.144.31:9090/ebony_ui/product/mstar/ebony_2k14/Ebony_2k14_SmartRC/doc/XMIC_SmartRC_HLD.doc>
4. Ebony 2k14机种SmartRC详细设计  
   <http://172.16.144.31:9090/ebony_ui/product/mstar/ebony_2k14/Ebony_2k14_SmartRC/doc/XMIC_SmartRC_LLD.docx>

# 2总体设计

## 2.1概述

### 2.1.1 需求规定

SmartRC是手机MyRemote应用的服务端应用，配合MyRemote完成电视助手功能，包括：

（1）模拟按键和鼠标；

（2）快捷按键；

（3）文字输入；

（4）多点触控；

（5）推送影视和应用到电视；

（6）我的云端收藏和历史；

（7）我的应用。

### 2.1.2 运行环境

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 平台 | 操作系统 | 屏幕分辨率 | 存储空间占用 | 内存空间占用 | CPU占用 |
| *待定* | *Android 4.2+* | *全适配* | *<= 3M* | *<= 10M* | *<= 10%* |

### 2.1.3 拟复用的已有函数库

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 函数库名称 | 来源 | 预计节约工时（人力） |
| 1 | Anymote | 开源的手机遥控器协议，在以前机种中采用 | 64 |
| 2 | NanoHttpd | 开源的轻量级Web服务器： | 16 |
| 3 | Gson | 开源的JSON序列化和反序列化工具 | 8 |
|  |  |  |  |

## 2.2设计

### 2.2.1设计构思

本应用主要包含两个服务，RemoteControlService和MyCloudService。其中RemoteControlService用于实现设备广播、模拟按键和鼠标、快捷按键、文本输入、多点触控，以及打开推送的影视和应用、打开/删除收藏和历史，打开本地应用等功能；MyCloudService用于提供我的云端数据，包括收藏、历史和本地应用列表和图片等功能。

简单来说，RemoteControlService用于简单控制命令通信，MyCloudService用于大数据通信。双向通信，尤其是电视主动推送的内容只能通过RemoteControlService，因为HTTP实现长连接代价较高。本系统中的MyCloudService（Web服务器）只接受获取JSON列表和图片的HTTP GET请求。

### 2.2.2系统流程 (画出动态运行流程)

### 2.2.3结构设计

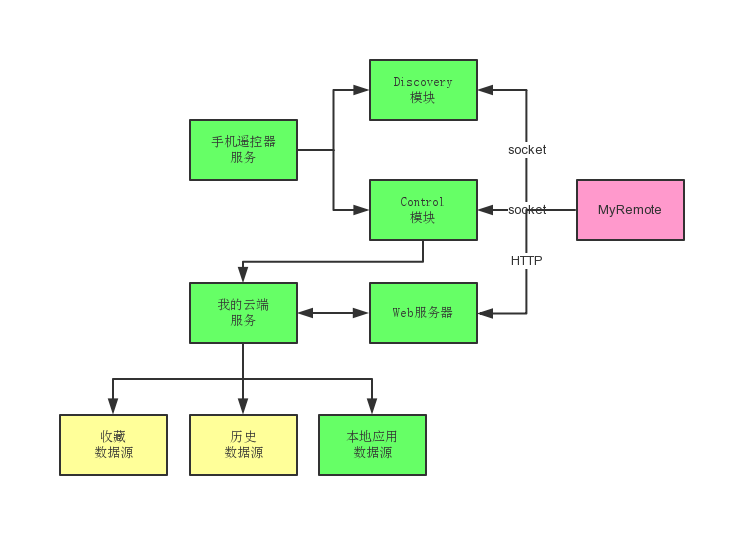


图1 总体结构图

如图2描述了系统详细功能点（绿色框框），以及各功能点依赖的内置应用或服务（黄色框框）。由于MyRemote只是一种控制方式，因此操作方式会与电视中内置的应用（紫色框框）一致。



图2 系统详细功能和依赖图

本系统主要包含两个服务。

包：com.tpv.xmic.smartrc.ots2k15

RemoteControlService：手机遥控器部分将沿用Ebony 2k14机种部分，但是移除配对相关功能，控制命令通道采用Socket而非SSLSocket。

MyCloudService：我的云端服务，提供收藏、历史和本地应用列表和图片，采用NanoHttpd库来实现Web服务功能，采用Gson库将列表序列化成JSON格式的数据。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 功能 |
| 1 | Remote Control Service | Service | 手机遥控器服务，主要包含两个线程：Discovery线程和Control线程 |
| 1 | Advertise Server | Thread | 搜索设备的线程，用于向手机客户端反馈本机机种名、电视名称等信息 |
| 2 | Request Receiver Implement | Class | Anymote协议中RequestReceiver的实现类，用于控制电视，各平台控制方式有差异 |
| 3 | My Cloud Service | Service | 我的云端服务，包括收藏、历史和本地应用 |
| 4 | Web Server | Class | NanoHTTPD的派生类，用于实现Web服务器， |
| 5 | Favorite Source | Class | 收藏数据中间层，用于提供格式化的数据到Web服务器 |
| 6 | History Source | Class | 历史数据中间层，用于提供格式化的数据到Web服务器 |
| 7 | App Source | Class | 本地应用数据中间层，用于提供格式化的数据到Web服务器 |

### 2.2.4功能需求与程序的关系 (具体需求与程序模块的对应关系)

|  |  |  |
| --- | --- | --- |
| 序号 | 功能需求(functional spec 和 UI Tree上面体现出来的需求) | 程序组件(对应的实现组件) |
| 1 | 连接（1） | Advertise Server |
| 2 | 遥控（2） | Remote Request Implement |
| R | 推送（3） | MA |
| 4 | 个人云端（4）- 个人云端界面(4) | Web Server |
| 5 | 个人云端（4）- 收藏界面（4.4） | Favorite Source |
| 6 | 个人云端（4）- 收藏TV界面（4.41） |
| 7 | 个人云端（4）- 切换账号(4.2) | Remote Request Implement |
| 8 | 个人云端（4）- 删除收藏界面（4.42） | Remote Request Implement |
| 9 | 个人云端（4）- 历史(4.3) | History Source |
| 10 | 个人云端（4）- 历史（4.31） |
| 11 | 屏幕映射（6） | NA |
| 12 | SimplyShare（7） | NA |
| 13 | 我的应用（8） | App Source |

# 3系统数据结构设计

## 3.1数据结构

### 3.1.1 TVInfo

电视信息

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 说明 |
| 1 | Header | String | 协议头，默认为“\_anymote.\_tcp” |
| 2 | mTVName | String | 电视名称 |
| 3 | mVersionCode | Int | 软体版本号 |
| 4 | TVModelId | String | 机种名 |
| 5 | mac | String | 有线网络MAC地址 |
| 6 | mControlServerPort | Int | 控制命令端口 |

### 3.1.2 AppInfo

本地应用信息

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 说明 |
| 1 | label | String | 应用标题 |
| 2 | Icon | String | 应用图标的URL地址 |
| 3 | action | String | 启动应用程序的参数 |

### 3.1.3 FavoriteInfo

收藏记录信息，同Favorite Widget定义，参考收藏的概要设计文档。

Git路径：[git@172.20.30.2:ots-flagship-2k15/ots-favouritewidget-2k15.git](mailto:git@172.20.30.2:ots-flagship-2k15/ots-favouritewidget-2k15.git)。

### 3.1.4 HistoryInfo

历史记录信息，同History Widget定义，参考历史的概要设计文档。

Git路径：[git@172.20.30.2:ots-flagship-2k15/ots-historywidget-2k15.git](mailto:git@172.20.30.2:ots-flagship-2k15/ots-historywidget-2k15.git)。

# 4接口设计

## 4.1用户接口

本系统采用后台服务，用户无法直接进行交互。

## 4.2 外部接口 (与外部程序系统之间交互的接口)

### 4.2.1 Intent

本系统可以接收Intent请求来启动后台服务，可以使用“am startservice -a”命令手动启动服务，目前支持的intent如下

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | Action | Data | 备注 |
| 1 | com.tpv.xmic.smartrc.RCSERVICE | NA | 启动手机遥控服务 |
| 2 | com.tpv.xmic.smartrc.MYCLOUDSERVICE | NA | 启动我的云端服务 |
|  |  |  |  |

## 4.3 内部接口(内部组件之间调用的接口)

### 4.3.1 手机遥控器接口

RequestReceiver类的回调函数中实现手机遥控器的主要功能接口，包括：

接口1：onConnect

形式：void onConnect(ConnectInfo connectInfo)

输入：connectInfo：连接信息

输出：无

返回： 无

描述：控制服务连接成功的回调函数，未使用

接口2：onData

形式：void onData(String s1, String s2)

输入：s1：命令字符串，s2：参数字符串

输出：无

返回：无

描述：扩展命令的回调函数

接口3：onFling

形式：boolean onFling(String strUri)

输入：strUri：网址

输出：boolean

返回：打开网址的结果，成功返回true，失败返回false

描述：打开网址的回调函数

接口4：onKeyEvent

形式：void onKeyEvent(Key.Code code, Key.Action action)

输入：code：按键键值，action：按键动作

输出：无

返回：无

描述：虚拟遥控器按键的回调函数

接口5：onMouseEvent

形式：void onMouseEvent(int paramInt1, int paramInt2)

输入：paramInt1：鼠标X方向偏移，paramInt2：鼠标Y方向偏移

输出：无

返回：无

描述：虚拟鼠标的回调函数

接口6：onMouseWheel

形式：void onMouseWheel(int paramInt1, int paramInt2)

输入：paramInt1：鼠标X方向偏移，paramInt2：鼠标Y方向偏移

输出：无

返回：无

描述：虚拟鼠标滚轮的回调函数

接口7：onMutilTouch

形式：void onMutilTouch(List<Integer> pointsX, List<Integer> pointsY, List<Integer> pointsId, int action, int pointCount)

输入：pointsX：触摸点的X坐标列表，poinstsY：触摸点的Y坐标列表，pointsId：每个触摸点的唯一标记列表，action：触摸动作，pointCount：触摸点数量

输出：无

返回：无

描述：多点触控的回调函数

### 4.3.2 扩展命令接口

客户端通过sendData命令发送的控制命令：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 命令 | 参数 | 返回值 | 说明 |
| 1 | com.google.tv.string | <text to input> | 无 | 文本输入 |
| 2 | com.xmic.START\_ACTIVITY | <package>/<activity> | 无 | 启动应用 |
| 3 | com.xmic.PUSH\_VIDEO | URL | 无 | 打开推送的视频 |
| 4 | com.xmic.PUSH\_APP | URL | 无 | 打开推送的应用 |
| 5 | com.xmic.DELETE\_FAVORITE | ID |  | 删除收藏 |
| 6 | com.xmic.DELETE\_HISTORY | ID |  | 删除历史 |
| 7 | com.xmic.VIEW\_FAVORITE | <action> |  | 打开指定收藏记录 |
| 8 | com.xmic.VIEW\_HISTORY | <action> |  | 打来指定历史激励 |
| 9 | com.xmic.GET\_FAVORITES | 无 | Web链接地址 | 获取收藏列表链接 |
| 10 | com.xmic.GET\_HISTORYS | 无 | Web链接地址 | 获取历史列表链接 |
| 11 | com.xmic.GET\_TOKEN | 无 | 账户Token | 获取账户Token |
| 12 | com.xmic.ATTACH\_VOICE | 无 | 无 | 激活语音引擎 |
| 13 | com.xmic.DETACH\_VOICE | 无 | 无 | 断开语音引擎 |

服务端通过sendData命令通知客户端的信息：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 命令 | 参数 | 返回值 | 说明 |
| 1 | com.xmic.ON\_ACCOUNT\_LOGIN | <account name> | 无 | 账户登录 |
| 2 | com.xmic.ON\_ACCOUNT\_LOGOUT | NA | 无 | 账户登出 |
| 3 | com.xmic.ON\_FAVORITE\_DELETED | ID | 无 | 删除收藏 |
| 4 | com.xmic.ON\_HISTORY\_DELETED | ID | 无 | 删除历史 |

### 4.3.3 我的云端接口

以收藏为例，接口访问时序如图：

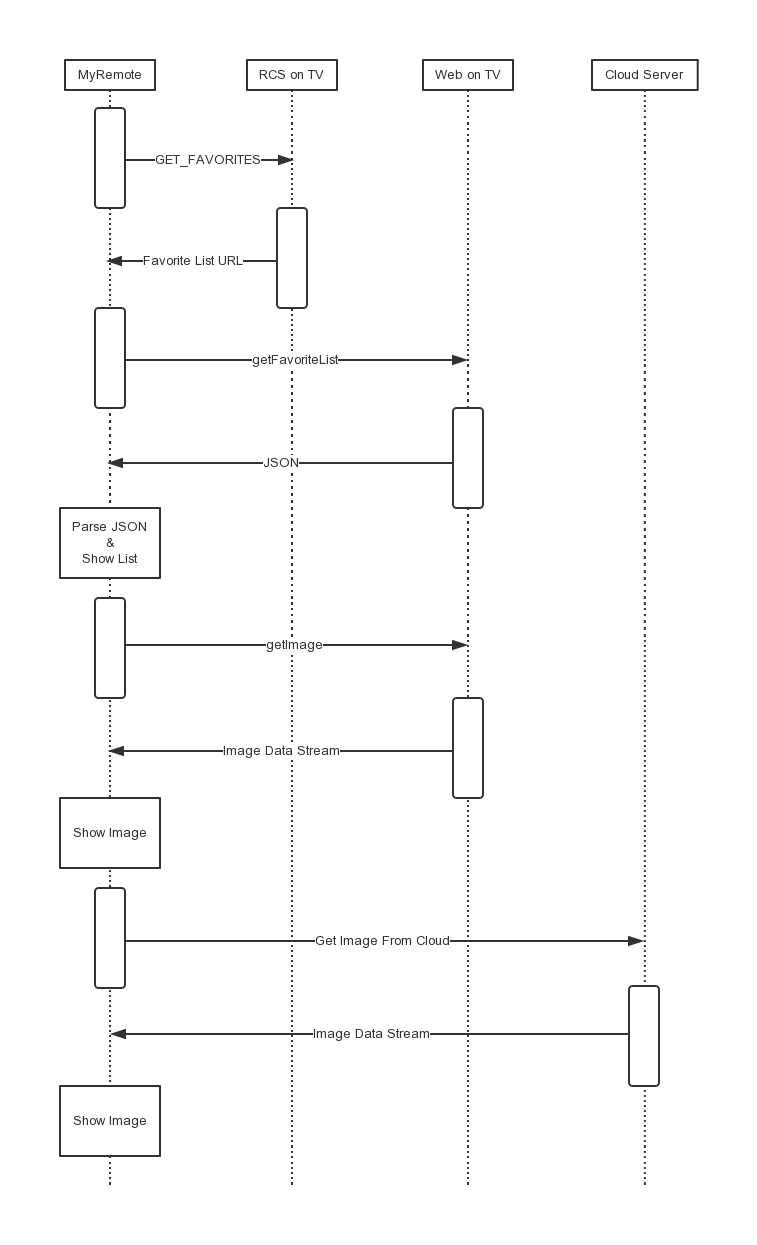


图1 收藏功能的接口访问时序

Web接口用于提供收藏和历史列表（JSON格式），以及相关图片。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 命令 | 参数 | 返回值 | 说明 |
| 1 | getFavoriteList | type：类型 | JSON数据 | 获取收藏列表 |
| 2 | getHistoryList | type：类型  dayago：几天前 | JSON数据 | 获取历史列表 |
| 3 | getAppList | 无 | JSON数据 | 获取应用列表 |
| 4 | getImage | Id：标记符 | 图片流 | 获取图片 |

备注：

1. type类型有四种，分别是APP（应用），MUSIC（音乐），TV（数字电视）和VOD（视频点播），字符串不区分大小写，类型必须指定；
2. dayago表示几天前的时间，包括0（今天），1（昨天），2（前天），3（三天前），如果没有指定则默认是今天。

示例：

http://<TVURL>/getAppList（获取本地应用列表）

http://<TVURL>/getFavoriteList?type=VOD（获取收藏的视频点播列表）

http://<TVURL>/getHistoryList?type=VOD&dayago=1（获取昨天的视频点播历史列表）

获取的JSON数据格式由数据源定义，与内置的Favorite/History Widget应用保持一致。

JSON数据中包含图片URL，这个URL可以是外网的图片URL，也可以是电视端Web服务器的URL，手机端都可以通过HTTP协议下载图片。如果下载电视端的图片，才会访问getImage接口链接。

# 5运行设计

## 5.1 系统初始化（初始化流程图）

## 5.2 运行模块组合

## 5.3 运行时间

# 6出错处理和维护

## 6.1 出错信息

## 6.2 补救措施

## 6.3 系统维护

# 7尚待解决的问题

1. 收藏和历史的数据源（格式、存储方式等）待定；