



Android 快速开机 特性说明书

文档版本 00B01
发布日期 2015-03-04

版权所有 © 深圳市海思半导体有限公司 2015。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.hisilicon.com/cn/>

客户服务电话： 4008302118

客户服务邮箱： support@hisilicon.com



前 言

目的

本文档对特性的原理、功能、配置方法、测试要点、问题定位方法进行描述，用于指导客户对特性的应用和测试。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
HiAndroidSTB	V600R100C00SPC060 以及后续版本



读者对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 软件工程师
- 测试工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 说明	表示是正文的附加信息，是对正文的强调和补充。



作者信息

章节号	章节名称	作者信息
全文	全文	Z00272647

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2015-03-04	00B01	第 1 次正式发布。



目 录

前 言.....	i
1 特性描述.....	1
1.1 方案说明.....	1
1.1.1 定义 Definition.....	1
1.1.2 价值点 Benefits.....	3
1.1.3 可获得性 Availability.....	3
1.1.4 影响 Impact.....	3
1.2 应用场景.....	3
2 快速开机特性说明	5
2.1 特性修改点说明.....	5
2.1.1 新增分区	5
2.1.2 新增系统服务.....	5
2.1.3 数据重载	6
2.2 客户定制说明.....	6
2.2.1 Fastplay 的支持.....	6
2.2.2 镜像制作提示.....	6
2.2.3 生产模式	7
2.2.4 升级与恢复出厂设置.....	7
2.2.5 与正常开机之间的切换.....	7
2.2.6 镜像制作点定制.....	8
2.2.7 调试手段	8
2.2.8 定制系统服务.....	10
2.2.9 快速开机安全性.....	11
2.2.10 快速开机应用升级说明.....	11
2.2.11 快速开机中状态属性说明	12
3 配置描述.....	13
3.1 配置快速开机特性.....	13
3.1.1 使能与关闭	13
4 可维可测指导.....	15



4.1 测试环境要求.....	15
4.2 测试步骤.....	15
4.3 定位方法.....	15
4.4 Log 分析	15



1 特性描述

1.1 方案说明

快速开机特性是在 Android 正常启动的基础之上，利用镜像手段，将第一次 Android 系统启动后的内存分布，固化并压缩成一个镜像，存放在预置好的分区中。然后在下一次开机时，系统将已经固化和压缩了的镜像文件解压出来，恢复内存状态，不用再次经历 Android 系统启动的全部流程，从而加快开机速度，实现快速开机。

1.1.1 定义 Definition

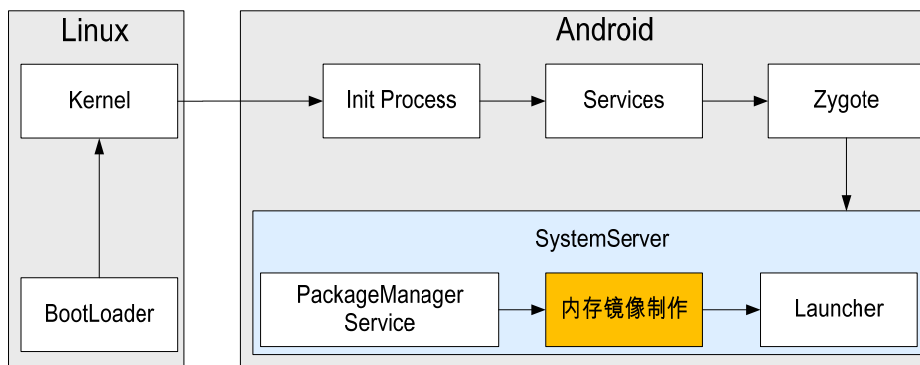
镜像制作

快速开机特性中，需要将 Android 系统正常启动后的内存分布做成镜像文件，并存储起来。

打开快速开机编译宏的版本，编译并烧录进单板后，第一次启动便会进入快速开机镜像制作，镜像制作成功后会正常启动到系统显示 Launcher。

快速开机镜像制作的流程如图 1-1 所示。

图1-1 快速开机镜像制作流程



上述内容主要是将正常 Android 启动的内存状态固化保存。



快速开机模式的启动

系统完成镜像制作流程后，后续的 Android 启动会进入快速开机模式，快速开机模式的启动步骤如下：

步骤 1 系统在经过 BootLoader 后，通过 qbflag 分区的信息，判断当前启动是否是快速启动模式。

- 如果是快速启动模式，会将存储在 qbdata 分区的镜像文件解压到内存中。完成内存镜像恢复。
- 如果不是，则按正常流程启动。

步骤 2 系统从镜像恢复点开始继续往下面执行，一直到 Launcher 启动，完成系统启动。

----结束

快速开机模式系统启动流程如图 1-2 所示。

图1-2 快速开机模式系统启动流程



数据重载

由于快速开机特性限制，每次系统都是从之前固化的镜像点开始恢复之前的内存状态，所以在使用系统的时候，当用户使用一些被固化在镜像里面的服务的时候，会出现下一次开机再次从镜像点恢复，而导致用户数据无法更新到最新的状态的情况。这时需要有一个数据重载的过程，即将数据刷新的最新的状态。



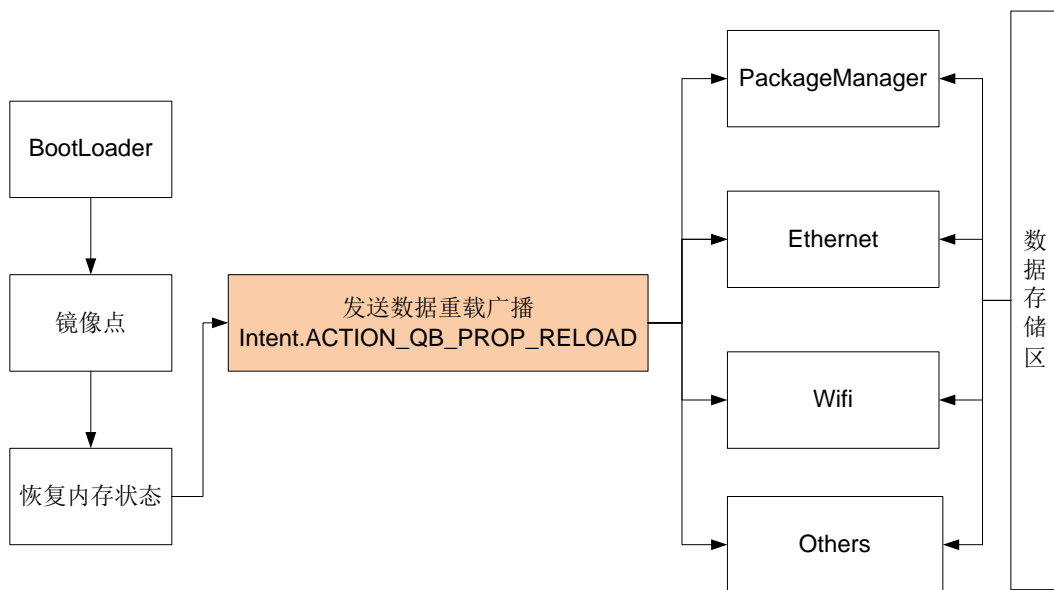
注意

镜像制作点之前的服务，需要进行数据重载，镜像制作点之后的服务不受该约束。

快速开机模式中，数据重载流程如图 1-3 所示。



图1-3 数据重载



1.1.2 价值点 Benefits

客户应用此功能，能够大大的缩短系统的启动时间，能让客户快速的进行人机交互，从而提升用户体验。

1.1.3 可获得性 Availability

HiAndroidSTBV600R100C00SPC060 版本以及后续的版本，在 Hi3798MV100 和 Hi3796MV100 芯片上可以支持此特性。

1.1.4 影响 Impact

由于快速特性方案是基于镜像技术，在每一块单板上面都需要进行一次正常开机做镜像，方可进入快速启动模式。

1.2 应用场景

当客户需要提升 STB Android 系统启动速度时候，可以通过配置项，打开快速开机配置，编译快速开机的版本并烧写。

使用海思的 HiAndroidSTBV600R100C00SPC060 版本，在 Hi3798MV100 的 Demo 板上正常启动速度约为 20s，打开快速开机后，使用 Android 默认的 Launcher，启动速度提升到 10s 以内，当前版本快速开机速度基本维持在 8s 左右。



注意

实际的客户环境下，单板的 EMMC 器件的读写速度，客户定制化的 Launcher，都会对快速开机的时间有影响。当前快速开机特性，仅仅支持 EMMC 器件。



2 快速开机特性说明

2.1 特性修改点说明

2.1.1 新增分区

根据快速开机的特性要求，增加了四个分区：

- **hibdrv**：存放 **hibdrv.bin** 文件，主要用来做镜像以及解压镜像的文件。
- **userapi**：针对不同的存储器件，比如不同种类的 EMMC，实现两个功能：
 - 将压缩的快速开机镜像写入到存放镜像的分区
 - 将分区中已经做好的镜像信息读取出来
- **qbflag**：快速开机的标志位，系统刚起来时读取这个分区的信息，判断是否需要快速开机：
 - 若是需要进行快速开机，则需要将已经做好的镜像从存储器件中读出解压
 - 否则，进行正常的开机流程
- **qbdata**：存放快速开机内存镜像的分区，系统做完镜像后，存入这个分区。当进行快速开机模式时，系统会在这个分区中进行镜像读取。

2.1.2 新增系统服务

根据快速开机特性的要求，在系统服务（SystemServer）中新增了一个关键性服务：

`QbAndroidManagerService`

这个服务的具体源码位于如下目录中：

`device/hisilicon/bigfish/frameworks/qb`

在 `SystemServer.java` 中按顺序启动系统各个服务，其中有新增如下代码：

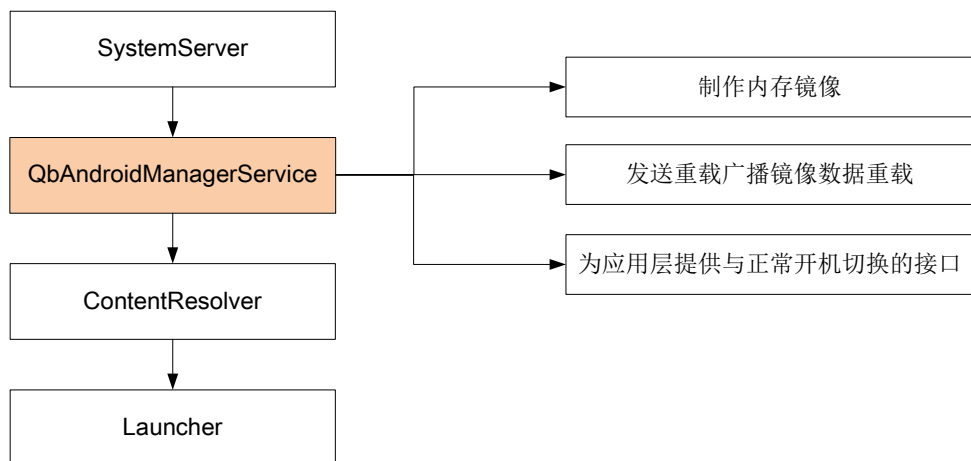
```
startQbService(context);
```

该函数主要功能的是启动 `QbAndroidManagerService` 服务，并根据当前的系统 `property` 的状态，来进行做镜像的操作和发送系统数据重载的广播，通知各个模块进行数据重载。

`QbAndroidManagerService` 的作用如图 2-1 所示。



图2-1 QbAndroidManagerService 作用



2.1.3 数据重载

当单板端在运行快速开机版本的时候，由于使用的是内存镜像技术，故需要有一个数据重载的过程，开机后将数据刷新的上次关机之前的状态。

涉及到改动的一些模块包括 Ethernet、Wifi 等需要有保存设置的地方。



注意

在海思的基础版本中已经适配了 Android 系统中的数据重载。客户只需要关注的是客户自己添加的系统服务地方是否需要数据重载。

2.2 客户定制说明

2.2.1 Fastplay 的支持

当前快速开机版本中不会重新走完整 Kernel 的启动流程，不能支持 Fastplay。

2.2.2 镜像制作提示

问题描述

由于在制作镜像的时候，系统 Kernel 层会将所有的驱动 Suspend，所以这个时候系统无法输出视频信息，从用户的角度上看，在做镜像的这段时间内，电视无任何输出（大约 5s 左右）。



解决办法

在制作镜像之前会先经过 Android 的动画，因此这个时候可以通过制作一个动画提示用户“正在制作镜像，请稍后”等类似提示。

相关方案海思已经制作完毕，提供了 Demo，客户可以参照海思的 Demo，按照如下步骤操作：

步骤 1 按照 Android 标准的 Bootanimation 的制作方式，制作一个 bootanimation.zip。

制作 Android 标准的 Bootanimation 方式可以参照：

步骤 2 重命名制作的 bootanimation.zip 为 quickboot.zip

步骤 3 将 quickboot.zip 放置在系统的 system/media/目录下即可。

海思版本中有预置的海思的 Bootanimation，名为 quickboot.zip，遵循标准 Android 的 Bootanimation 的制作流程，存放于如下目录：

```
device/hisilicon/bigfish/prebuilt
```

用户可以参考这个包和上述链接，进行用户的定制化的 Bootanimation 的制作。

----结束

2.2.3 生产模式

由于每个单板上的有些器件（比如 EMMC 存储器件）不同，都具有全球唯一标识 ID，Kernel 中都要使用该 ID 信息，因此当前快速开机的启动镜像不能直接在某一片上面制作好之后，烧录到其他单板上使用，而需要针对每个单板都完整的走一次快速开机制作镜像的流程，然后才能正常的使用快速开机特性。

因此在生产环节都需要先进行本地的开机、做镜像、再启动的流程，这样才算是完成了快速开机的制作。



注意

在生产流程上，在烧写流程上，和正常版本一致。快速开机版本需要等待镜像制作完成。

2.2.4 升级与恢复出厂设置

当升级或恢复出厂设置的时候，系统会去执行恢复出厂的操作，恢复出厂完成后，需要重新制作快速开机镜像。

2.2.5 与正常开机之间的切换

当前快速开机版本特性中支持动态切换开机模式，可以由快速开机模式切换到正常版本模式，也可以由正常版本模式切换到快速开机版本模式。



可以通过调用 QbAndroidManagerService 中的 setQbEnable(boolean flag)接口来实现快速开机和正常开机之前的切换。

- 当 setQbEnable(true)时候, 切换到快速开机版本。
- 当 setQbEnable(false)时候, 切换到正常版本。

当前版本中已经有相应的 Demo APK。位于目录:

`device/hisilicon/bigfish/development/apps/HiFastboot`

当打开快速开机版本的编译宏的时候, 该 APK 被编译进系统。

2.2.6 镜像制作点定制

由于制作镜像的时间点直接关系到镜像的大小、镜像恢复时系统的状态、数据重载的流程、系统是否需要解数据库与 data 分区的依赖等重要的因素, 因此当前镜像制作时间点海思已经进行优化处理, 快速开机版本暂不支持用户自由配置制作镜像的时间点。

2.2.7 调试手段

由于快速开机版本中开机都需要取恢复内存镜像, 所以在系统调试的时候, 首先需要将系统恢复到正常系统的版本 (通过 2.2.5 中叙述的手段), 然后在正常版本基础之上添加调试信息, 然后去重新制作镜像 (通过 2.2.5 中叙述的手段)。这个时候修改的文件已经被做到镜像里面了, 可以通过 logcat 的方式进行 Log 信息抓取, 调试代码。

需要注意的是, 快速开机模式下, 不支持 PC 端的 adb remount。因此若是需要调试快速开机版本, 并希望更新 system/目录下的文件的话, 需要按照如下步骤操作:

步骤 1 通过 2.2.5 中叙述的手段, 将系统恢复到正常开机模式。

步骤 2 将 PC 端的 adb 与板端的连上。

步骤 3 将需要更改的文件通过 adb push 到/data/local/tmp/这个可读写的目录下。

步骤 4 在板端通过串口执行命令:

```
mount -o remount /system /system
```

将/system 目录系统 remount

步骤 5 将/data/local/tmp/目录下的文件拷贝至需要的目录。

步骤 6 在单板端通过串口执行命令:

```
sync
```

步骤 7 重启单板。

步骤 8 系统起来后, 通过 2.2.5 中叙述的手段, 将系统切换到快速开机模式进行调试。

----结束



注意

编译的快速开机系统是 user 版本的，与 eng 版本不一样，user 版本编译出来的 jar 包文件会进行校验，比如 eng 版本编译出来在系统目录/system/framework/下有 framework.jar，而 user 版本编译出来在/system/framework/目录下有 framework.jar 和 framework.odex。故，在添加调试信息的时候，不但需要替换.jar 文件，同时也要将编译出来的对应的.odex 文件进行替换。

user 版本调试过程中，若调试文件编译生成.jar 和.odex，则需要将编译.jar 和.odex 同时替换，这里需要屏蔽掉 Android 原生系统的校验，若不进行该操作，推入替换的.jar 和.odex 重新制作镜像，系统将无法启动。屏蔽 Android 原生 user 版本的校验方式如下：

步骤 1 打开 dalvik/vm/analysis/ DexPrepare.cpp 文件。

步骤 2 注释掉 1306 行。

```
//const ul* signature = getSignature(cpe);
```

步骤 3 注释掉 1327 行~1332 行。

```
/*  
if (memcmp(signature, ptr, kSHA1DigestLen) != 0) {  
    ALOGI("DexOpt: mismatch dep signature for '%s'", cacheFileName);  
    goto bail;  
}  
*/
```

步骤 4 最终版本调试完成后，将上述代码改回即可。

----结束

调试的时候的几个基本步骤：

步骤 1 将上述校验注释掉，编译快速开机的版本并烧写。

步骤 2 需要调试时候，将系统通过 2.2.5 中介绍的方式切换到正常启动模式。

步骤 3 将调试需要替换的文件通过 adb 推送到：

```
/data/local/tmp/
```

步骤 4 若是需要更新系统 system 目录下的文件，需要在板端串口 remount：

```
mount -o remount /system /system
```

步骤 5 将需要替换的文件 cp 到相应的目录下然后键入 sync 命令。

步骤 6 重启系统，系统正常启动后，通过 2.2.5 方式重新制作镜像，开启快速开机。

步骤 7 然后就可以通过 logcat 查看系统日志，进行调试。



----结束

2.2.8 定制系统服务

由于快速开机特性中，镜像制作点的不可定制化，若是想添加自己的系统服务需要注意以下问题：

- 客户若是希望在 SystemServer 中镜像制作点之前添加系统服务，需要在系统启动后做数据重载。

比如，用户希望在 SystemServer 中在制作镜像制作点前（startQbService 调用之前）新增一个 DVBSerivce，这个 Service 中存有 user、password 等数据。当系统重启后，这些参数又会恢复之前镜像点的那个状态。

上述问题的解决方案是：在这个 DVBSerivce 服务中，设置监听接收 Intent.ACTION_QB_PROP_RELOAD 广播，收到该广播后，将存储的数据取出并刷新到当前的值。

添加的 Intent.ACTION_QB_PROP_RELOAD 在如下路径：

`frameworks/base/core/java/android/content/Intent.java`

可以通过自己定义 BroadcastReceiver 广播接收器，来监听如下广播：

`Intent.ACTION_QB_PROP_RELOAD`

监听到该广播后，进行数据重载

上述流程如图 2-2 所示。

图2-2 SystemServer 镜像制作点之前新增服务示意图（1）

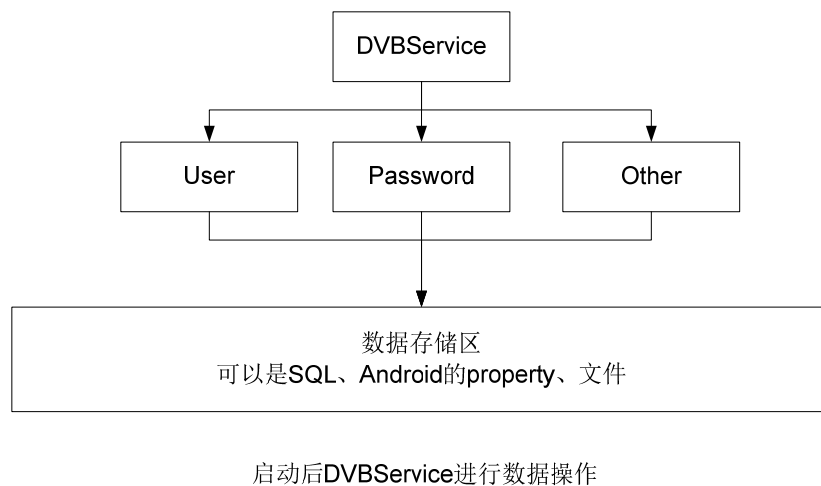
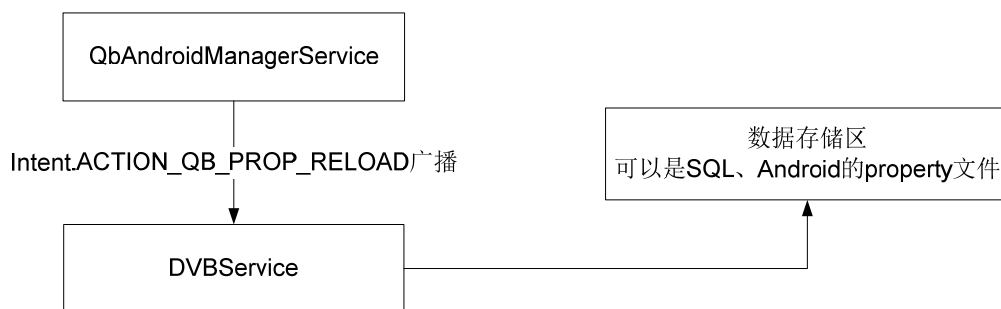




图2-3 SystemServer 镜像制作点之前新增服务示意图（2）



快速启动时DVBService重载数据

- 若是在 SystemServer 中添加的系统服务在镜像制作点之后，无需做上述操作。
- 针对在 init.rc 中添加服务的情况：

由于做镜像时刻需要将分区 unmount 掉，因此需要保证在做镜像时刻点，对挂载上的分区没有直接的读写依赖操作。若是需要在 init.rc 中添加所需的任务，则需要保证这些任务中在制作镜像点的时刻，没有读写 data 分区和 cache 分区的操作。否则会导致镜像制作不成功。

由于数据库的操作是连接上后，持续的进行，因此不要在 init.rc 中添加有数据库操作的服务。

可以在 init.rc 添加读写文件的操作的服务，但是需要保证文件读写在做镜像时刻是断开的。建议不要或者尽量短时间的进行文件读写的操作。
- 若是客户需要自定义、添加驱动程序，则需要联系海思进行适配。
- 客户可以根据自己的业务情况新增分区，不会影响快速开机特性。

2.2.9 快速开机安全性

在快速开机特性中，支持遥控器特定按键强制切换为正常启动。

- 此功能默认是关闭的。打开此功能的配置为修改 SDK 配置项，如文件 hi3798mdmo_hi3798mv100_android_cfg.mak，然后把 `# CFG_HI_BUILD_WITH_IR is not set` 修改为 `CFG_HI_BUILD_WITH_IR=y`
- 打开此功能后，一旦强制切换为正常启动，此后系统启动就切换为正常启动模式了。
- 此功能默认键值是遥控器“返回值”，如要适配，请修改文件 `/device/hisilicon/bigfish/sdk/source/boot/product/android/customer_android.c`，在函数 `Android_Qb_CheckRemote(void)` 里，修改 `0x6f90ff00` 为特定遥控器的键值即可。

2.2.10 快速开机应用升级说明

快速开机特性中，针对应用升级的场景，分为两种情况：



1. 系统运行中，用户安装并通过与用户交互来启动的应用（比如：一些视频客户端的应用）。针对这部分应用需要升级的情况，海思已经做了相应的适配，客户无需关注。
2. 系统 Launcher 一类的应用。由于快速开机对应用的更新，是进入 Launcher 后，开始对应用进行后扫描并进行更新，故客户若是希望升级自己的 Launcher 一类，或是随 Launcher 一起启动的应用，建议的做法是：更新了应用后，重新做一次快速开机的镜像，将更新的应用做到镜像中。

2.2.11 快速开机中状态属性说明

快速开机特性中，海思新增并适配了一些系统属性，具体描述如下表所示：

表2-1 快速开机状态属性说明

状态	属性取值	状态含义
判断快速开机启动模式	<code>persist.sys.qb.enable</code> true	当前是快速开机版本
	<code>persist.sys.qb.enable</code> false	当前是正常版本
快速开机制作镜像状态	<code>persist.sys.qb.enable</code> true	快速开机准备制作镜像
	<code>persist.sys.qb.flag</code> false	
	<code>persist.sys.firstboot.flag</code> false	
	<code>persist.sys.qb.enable</code> true	快速开机镜像制作完成
	<code>persist.sys.qb.flag</code> true	
	<code>persist.sys.firstboot.flag</code> false	



3 配置描述

3.1 配置快速开机特性

3.1.1 使能与关闭

配置关闭快速开机特性

HiAndroidSTBV600R100C00SPC060 版本中默认是关闭该特性。

配置使能快速开机特性

步骤 1 打开快速开机的配置项

进入源代码目录：

```
/device/hisilicon/芯片目录
```

芯片目录指诸如 Hi3798MV100, Hi3796MV100。比如，当前是 Hi3798MV100 芯片，则进入：

```
/device/hisilicon/Hi3798MV100
```

打开 customer.mk 文件

修改配置：

```
BOARD_QBSUPPORT := true
```

步骤 2 编译快速开机版本

配置完后，回到源码根目录下，编译快速开机版本。当前快速开机仅支持 user 版源码目录下输入：

```
source build/envsetup.sh
```

然后输入：

```
lunch 芯片-user
```

其中芯片指的是编译上述步骤 1 中的芯片类型的版本，如编译 Hi3798MV100 的，输入：



```
lunch Hi3798MV100-user
```

步骤 3 最后将编译所生成的镜像烧写进单板，就完成了整个快速开机的配置。

----结束



4 可维可测指导

4.1 测试环境要求

快速开机版本的测试环境和普通 STB Android 版本的测试环境一致。测试环境参照 STB Android 正常版本测试环境。

4.2 测试步骤

步骤 1 编译并烧写快速开机版本至单板。

步骤 2 按照 STB Android 正常版本的测试步骤进行测试即可。

----结束

4.3 定位方法

问题定位的手段主要还是以 Log 分析为主，通过输入 logcat 来打印日志。

4.4 Log 分析

根据打印日志进行 Log 分析，具体的步骤参见 [2.2.7](#) 中调试。