# Profile Download知识总结（高通）

|  |  |  |
| --- | --- | --- |
| Date  日期 | Version  版本 | Comments  备注 |
| 2019-01-09 | 0.1 | First version |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Function  职位 | Name  姓名 | Date  日期 | Signature  签名 |
| Written by  拟定 | SW | 曾灿炫 | 2018/01/19 | 曾灿炫 |
| Verified by  审核 |  |  |  |  |
| Verified by  审核 |  |  |  |  |
| Approved by  批准 |  |  |  |  |

目录

[Profile Download知识总结（高通） 1](#_Toc535592849)

[1. 概述 3](#_Toc535592850)

[**1.1.** **本文档的目的** 3](#_Toc535592851)

[**1.2.** **背景** 3](#_Toc535592852)

[**1.3.** **缩略语清单** 3](#_Toc535592853)

[**1.4.** **参考文献** 3](#_Toc535592854)

[2. 基本概念 4](#_Toc535592855)

[2.1 简介 4](#_Toc535592856)

[2.2 Terminal Profile解读 4](#_Toc535592857)

[2.2.1 定义 4](#_Toc535592858)

[2.2.2 实现 4](#_Toc535592859)

[3. 关键流程分析 5](#_Toc535592860)

[3.1 整体流程图 5](#_Toc535592861)

[3.2 modem端流程 5](#_Toc535592862)

[3.2 代码分析 6](#_Toc535592863)

[3.2.1 mmgsdi->gstk（default TP） 6](#_Toc535592864)

[3.2.2 gstk->uim（发送TP） 7](#_Toc535592865)

[3.2.3 uim->gstk 8](#_Toc535592866)

[3.2.4 gstk->mmgsdi 10](#_Toc535592867)

[4. 经验分享 11](#_Toc535592868)

[5. 总结 11](#_Toc535592869)

# 概述

* 1. **本文档的目的**

本文档介绍STK模块中的Profile Download运作流程，并基于高通项目做代码的跟进。

* 1. **背景**

STK(SIM Tool Kit)，也叫SAT(Sim Application Toolkit) 允许SIM卡运行自己的应用, 在手机中形成业务菜单,前提是手机支持STK。这就需要由手机来通知SIM卡它支持的功能，否则SIM卡默认手机不支持STK，也不会向其发送任何主动式命令。Profile Download便是实现该功能的过程。

* 1. **缩略语清单**

*列出文中使用的术语的定义和缩略（语）词的英文全名和中文解释。*

| Term | Explanation |
| --- | --- |
| TP | Terminal Profile |
|  |  |
|  |  |

* 1. **参考文献**

| Document | Explanation |
| --- | --- |
| [3GPP TS 11.11] | Technical Specification Group Terminals Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface. |
| [3GPP TS 11.14] | Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment(SIM - ME) interface. |
| 80-ng610-1\_k\_gstk\_overview | 高通文档 |

# 基本概念

## 2.1 简介

在SIM卡的初始化过程中，ME通过读取SIM卡的EFPHASE文件确定是否需要完成Profile Download。Profile Download即向SIM卡发送Terminal Profile命令，该命令通知SIM卡ME所能支持的SIM卡功能。

## 2.2 Terminal Profile解读

### 2.2.1 定义

Terminal Profile 包含30多个字节，每个字节都涉及8个功能，具体定义可查看3GPP TS 31.111。

如第一个字节：

First byte (Download):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | b8 | | b7 | | b6 | | b5 | | b4 | | b3 | | b2 | | b1 | |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | See TS 102 223 [32] clause 5.2 |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | SMS-PP data download |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | Cell Broadcast data download |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | See TS 102 223 [32] clause 5.2 |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | Bit = 1 if SMS-PP data download is supported |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | See TS 102 223 [32] clause 5.2 |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | Bit = 1 if Call Control by USIM is supported |
|  |  | |  | |  | |  | |  | |  | |  | |  | |  | | Bit = 1 if Call Control by USIM is supported |

### 2.2.2 实现

在具体的项目中，相关功能是否支持由pics值决定，Benz的pics表为80-PB971-5。

相关值在modem的gstk\_exp.h中定义(在defect7311073中出现pics表与具体代码值不同的情况，解决方案由高通提供)

具体的TP解析则需要借助文档GCF-CC-3720-F3.4\_USAT support进行转化。



# 关键流程分析

## 3.1 整体流程图

Profile Download流程图如下：

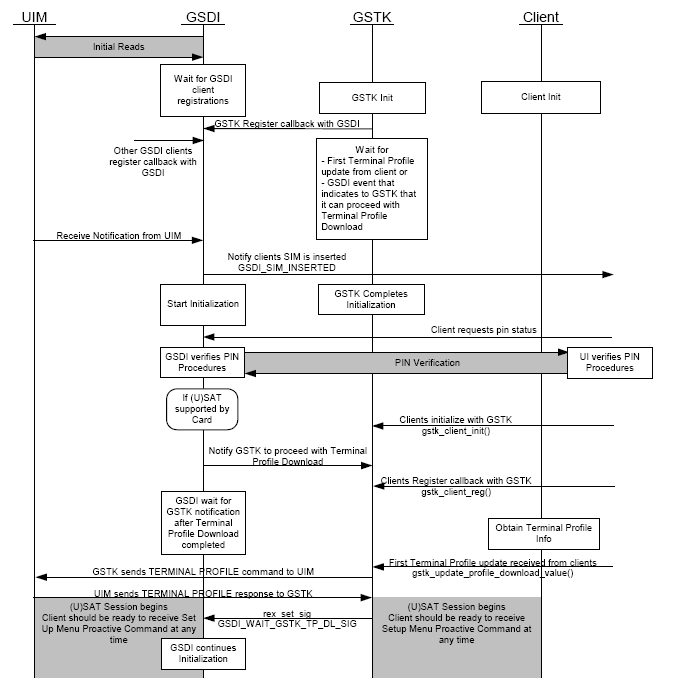


Phase文件指的是SIM卡所支持的GSM标准的级数，值为3则指SIM卡支持**GSM phase 2和profile download(3GPP 11.11)**。

1. ME通过读取EFphase文件确定SIM卡所提供的功能。
2. 向SIM发送Termianl Profile命令，告知SIM卡ME所能支持的功能。

## 3.2 modem端流程

高通modem端的完整时序图(从卡插入到发送TP)如下：



1-TP时序图(80-ng610-1\_k\_gstk\_overview)

## 3.2 代码分析

通过log的追踪，在modem端进行具体的代码跟进。

### 3.2.1 mmgsdi->gstk（构建TP）



如上为mmgsdi部分的时序图，mmgsdi通知gstk去构建TP并提供回调函数用于通知自身TP已发送完毕。

**mmgsdi\_card)init.c:**

1. 调用mmgsdi\_card\_init\_gstk\_terminal\_profile\_download去促使gstk发送TP

2. 函数调用mmgsdi\_card\_init\_is\_toolkit\_allowed去读取service table确定当前SIM卡是否支持STK。

3. 返回值为true则调用gstklib.c的gstk\_gsdi\_set\_fetch\_status设置fetch的状态(能否fetch命令)。

4. 调用gstklib.c的gstk\_slot\_send\_terminal\_profile\_dl\_command让gstk完成Terminal Profile的流程。这里提供**mmgsdi\_gstk\_terminal\_profile\_cb**回调函数，使gstk完成TP发送后通知mmgsdi。

**gstklib.c：**

1. 获取一个gstk的消息buffer(gstk\_task\_cmd\_type)，构建该buf并调用gstk.c的gstk\_task\_put\_cmd\_buf(设置其comman\_group为GSTK\_MMGSDI\_CMD)将其发送至gstk的命令队列。

### 3.2.2 gstk->uim（发送TP）

进入gstk，发送TP的时序图如下：



**gstk.c :**

1. gstk\_init\_post\_task\_startup gstk初始化时调用gstk\_update\_default\_app\_specific\_tp构建初始TP.(该函数内可跟进TP各个位的初始值)

2.gstk\_task获取队列的command，调用gstk\_process\_state(gstk\_state.c)检测当前状态。

**gstk\_state.c:**

1.读取当前gstk状态为GSTK\_NO\_SIM\_S，gstk\_state\_no\_sim( gstk\_s\_no\_sim.c)

**gstk\_s\_no\_sim.c：**

1.读取command\_group，为mmgsdi传来的GSTK\_MMGSDI\_CMD，进而确定command\_id为 GSTK\_MMGSDI\_TERMINAL\_PROFILE\_DL\_EVT，调用gstk\_pack\_tp\_cmd\_to\_uim

(gstk\_term\_profile)开始打包发送TP。

**gstk\_term\_profile.c:**

1.调用gstk\_update\_base\_tp更新gstk初始化时构建的TP，根据cat\_version限定TP的长度。

2.调用gstk\_send\_cmd\_to\_uim\_server(gstkutil.c)开始发送uim cmd。

**gstkutil.c:**

1.调用uim\_cmd将cmd\_ptr发送至uim的消息队列。

2.根据command类型，提供函数gstk\_uim\_terminal\_profile\_report(gstk\_term\_profile.c)，

当uim收到sim卡的respone时调用该函数构建gstk\_term\_profile\_rsp\_type，改变gstk状态，从GSTK\_NO\_SIM\_S到GSTK\_TERM\_PROFILE\_RSP\_WAIT\_S。gstk的状态机再根据命令类型执行对应的流程。

到了这一步，TP已经发送至SIM卡了，接下来就是等待SIM卡的response，执行发送后的流程。

### 3.2.3 uim->gstk（接收完成）

gstk\_uim\_terminal\_profile\_report主要部分如下：

task\_cmd = gstk\_task\_get\_cmd\_buf();

if ( task\_cmd != NULL ) { /\* successfully obtained command buffer \*/

/\* build the terminal profile response \*/

task\_cmd->cmd.terminal\_profile\_rsp.message\_header.sim\_slot\_id =

(gstk\_slot\_id\_enum\_type)report->slot;

task\_cmd->cmd.terminal\_profile\_rsp.message\_header.command\_group = GSTK\_TERMINAL\_PROFILE\_RSP;

task\_cmd->cmd.terminal\_profile\_rsp.message\_header.command\_id = (int)GSTK\_PROFILE\_DL\_IND\_RSP;

task\_cmd->cmd.terminal\_profile\_rsp.message\_header.user\_data = report->user\_data;

if (report->rpt\_status == UIM\_PASS) {

task\_cmd->cmd.terminal\_profile\_rsp.success = TRUE;

if (GSTK\_IS\_VALID\_SLOT\_ID(report->slot)) {

GSTK\_RETURN\_IF\_NULL\_PTR(gstk\_instances\_ptr[(uint8)report->slot - 1]);

gstk\_instances\_ptr[(uint8)report->slot - 1]->is\_tp\_rsp\_received = TRUE;

}

}

/\* set GSTK\_CMD\_Q\_SIG and put on GSTK queue \*/

gstk\_task\_put\_cmd\_buf(task\_cmd);

}

该函数与mmgsdi通知gstk的原理相似，都是获取gstk消息体gstk\_task\_cmd\_type并进行构建发送至队列。

回到gstk\_status.c，进入不同分支，执行发送TP后的流程，时序图如下：



**gstk.c :**

1.gstk\_task获取队列的command，调用gstk\_process\_state(gstk\_state.c)检测当前状态。

**gstk\_state.c:**

1.读取当前gstk状态为GSTK\_TERMINAL\_PROFILE\_RSP，调用gstk\_state\_tp\_rsp\_wait

(gstk\_s\_term\_profile\_rsp\_wait.c)

**gstk\_s\_term\_profile\_rsp\_wait.c :**

1.调用uim\_set\_proactive\_uim\_slot(uim.c)指示UIM为主动式uim

2.调用gstk\_util\_send\_message\_to\_clients(gstkutil.c)向所有注册了该事件的client发送response。

**gstkutil.c:**

1.调用estk\_bip\_full\_clean\_up清除BIP table

2.通过回调函数estk\_gstk\_evt\_cb清除estk存储的关于刚处理完的指令的数据。

### 3.2.4 gstk->mmgsdi（发送完成）

回到3.2.1第三步，当gstk\_slot\_send\_terminal\_profile\_dl\_command执行完毕，回调mmgsdi\_gstk\_terminal\_profile\_cb进入TP发送完毕的处理流程。该函数主要是构建一个mmgsdi的消息结构体mmgsdi\_task\_cmd\_type(MMGSDI\_TP\_DOWNLOAD\_COMPLETE\_REQ)，传入来自gstk的TP response，并放到mmgsdi的command quene当中。

进入mmgsdi的command处理函数mmgsdi\_process\_command，可看到根据command为MMGSDI\_TP\_DOWNLOAD\_COMPLETE\_REQ，函数又调用了mmgsdi\_process\_tp\_download\_complete，该函数主要部分如下：

mmgsdi\_status = mmgsdi\_util\_get\_client\_request\_table\_free\_index(&index);

if (mmgsdi\_status != MMGSDI\_SUCCESS)

{

return mmgsdi\_status;

}

mmgsdi\_status = mmgsdi\_util\_populate\_client\_request\_table\_info(

index,

&req\_ptr->request\_header,

NULL,

NULL);

if (mmgsdi\_status != MMGSDI\_SUCCESS)

{

mmgsdi\_util\_free\_client\_request\_table\_index(index);

return mmgsdi\_status;

}

mmgsdi\_status = mmgsdi\_util\_queue\_mmgsdi\_uim\_report\_rsp(

index,

MMGSDI\_TP\_DOWNLOAD\_COMPLETE\_REQ,

mmgsdi\_status);

可看到函数获取了client request table中可用的索引，然后通过该索引对mmgsdi\_client\_req\_table\_info\_ptr进行数据的填充，并调用mmgsdi\_util\_queue\_mmgsdi\_uim\_report\_rsp构建对应的mmgsdi\_task\_cmd\_type发至comand quene(MMGSDI\_UIM\_REPORT\_RSP)

进入到mmgsdi\_process\_response，根据command类型为MMGSDI\_TP\_DOWNLOAD\_COMPLETE\_REQ调用mmgsdi\_process\_tp\_dl\_complete\_response。

该函数检测卡当前可用的卡类型,并根据卡profile发送card inserted事件。

# 经验分享

涉及Terminal Profile的defect主要是通过查看pics表与代码里的具体值。

# 总结

Terminal Profile相关的协议文档只做了定义规范，具体实现需要查看相关的pics表。而高通文档对于该部分的流程介绍不多，这个模块的学习更多依赖的是modem代码的跟进。且该流程是在modem端运作，有必要去了解modem的架构与相关模块的作用。