# Envelope知识总结

|  |  |  |
| --- | --- | --- |
| Date  日期 | Version  版本 | Comments  备注 |
| 2019-03-25 | 0.1 | First version |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Function  职位 | Name  姓名 | Date  日期 | Signature  签名 |
| Written by  拟定 | SW | 曾灿炫 | 2019-03-25 | 曾灿炫 |
| Verified by  审核 |  |  |  |  |
| Verified by  审核 |  |  |  |  |
| Approved by  批准 |  |  |  |  |

目录

[Envelope知识总结 1](#_Toc4681633)

[1. 概述 3](#_Toc4681634)

[**1.1.** **本文档的目的** 3](#_Toc4681635)

[**1.2.** **背景** 3](#_Toc4681636)

[**1.3.** **缩略语清单** 3](#_Toc4681637)

[**1.4.** **参考文献** 3](#_Toc4681638)

[2. Envelope基础概念 4](#_Toc4681639)

[2.1. 简介 4](#_Toc4681640)

[2.2. 应用机制 4](#_Toc4681641)

[2.3. 命令结构 4](#_Toc4681642)

[3. Modem端实现 5](#_Toc4681643)

[3.1 涉及模块 5](#_Toc4681644)

[3.2. 关键函数与流程 6](#_Toc4681645)

[3.3. 关键log 6](#_Toc4681646)

[4. 总结 9](#_Toc4681647)

# 概述

* 1. **本文档的目的**

*本文档作为一个学习输出，总结STK中Envelope相关的基础概念与高通的实现方式。*

* 1. **背景**

*STK即SIM Tool Kit，它提供一系列用于移动设备与SIM卡间交互的机制。通过这些机制，支持STK的手机可以操作SIM卡里的应用。这些机制除Profile Download, Proactivce Command之外，还有其他如Event Download，Menu Selection等需要Envelope Command的支持。*

* 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 |
| --- | --- |
|  |  |
|  |  |
|  |  |

* 1. **参考文献**

| Document | Explanation |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Envelope基础概念

## 简介

Envelope中文指信封，顾名思义，该命令使用来携带数据给SIM卡或者让SIM卡执行某些动作。与主动式命令相反，它是由UE端发起的命令；同时区别于Terminal Response，并不是某一条主动式命令的执行结果。Envelope涉及STK多个机制，如点击STK菜单条目进入下一级，下文将举例做简单的介绍。

## 应用机制

**Data Download:** 分为SMS-PP DATA DOWNLOAD与Cell Broadcast DATA DOWNLOAD，不管是哪一种，ME端在收到满足要求的message时都会将先其通过Envelope发给SIM卡处理，而不是显示给手机用户。

**Menu Selction:** SIM卡初始化时可通过set up menu提供多个菜单，当用户通过应用点击某一项或者求助选项，ME会通过Envelope将该点击事件发至SIM卡。该命令与主动式命令select item对应，SIM卡根据具体的选项发送二级菜单。

**Event Download：**这个涉及到主动式命令set up event list, SIM卡通过该命令让ME建立一个事件表，当对应的事件发生时，通过Envelope命令通知SIM卡。具体事件如WLAN Access status event，Location status event，Call disconnected event等。

**具体的其他机制可查阅3GPP TS31.111。**

## 命令结构

一条完整的Envelope命令应包含表中所列项（以Menu Selection为例，表格截至3GPP TS 11.14）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Section** | **M/O** | **Min** | **Length** |
| Menu Selection tag | 13.1 | M | Y | 1 |
| Length (A+B+C) | - | M | Y | 1 or 2 |
| Device identities | 12.7 | M | Y | A |
| Item identifier | 12.10 | M | Y | B |
| Help request | 12.21 | O | N | C |

主要关注点是Menu Selection tag、Device identities、Item identifier：

其中第一位为Command tag，这里是**Menu Selection tag**，具体值 在13.1中定义，查阅确定值为D3。

**Device identities** 这个值记录的是命令由谁发起，发到哪，具体值如下：

Coding: both Source and Destination device identities are coded as follows:

- '01' = Keypad

- '02' = Display

- '03' = Earpiece

- '10' to '17' = Additional Card Reader x (0 to 7). Value assigned by ME.

- '21' to '27' = Channel x (1 to 7). Value assigned by ME (if class "e" is supported).

- '81' = SIM

- '82' = ME

- '83' = Network

根据定义，一个Device identities需包含Device identities tag，Length = '02'，Source device identity，Destination device identity共4位。

**Item identifier**指用户在STK应用中所点击的选项，是该命令要传输的值。这一值占3位，即tag，length，itentifier。

# Modem端实现

## 3.1 涉及模块



上图为Envelope主要涉及的模块，可以看到与Proactive Command经过相同模块的处理，即UIM,GSTK,QMI\_CAT。其中GSTK做STK的主要逻辑处理，其余两模块负责与上下层对接。

## 3.2. 关键函数与流程

下图为高通文档80-ng610-1\_k\_gstk\_overview 提供的Menu Selection的完整处理流程，从上层至SIM卡依次涉及的关键函数为：

**QMI\_CAT:**

1. gstk\_send\_raw\_envelope\_command:这里与图示函数有差异，Benz项目里以raw代指所有的命令类型，直接发至GSTK，不分命令类型做处理。
2. qmi\_cati\_gstk\_raw\_env\_cb env\_rsp\_ptr：回调函数，供GSTK发送Envelope Respnse。

**GSTK:**

1. gstk\_send\_cmd\_to\_uim\_server：发送命令至UIM。
2. gstk\_uim\_envelope\_report：回调函数，供UIM发送Envelope Response。

**UIM:**

1. uim\_send\_command:发送命令至SIM卡。
2. uim\_generic\_command\_response：处理SIM卡返回的两位状态字。



## 3.3. 代码跟进

### 3.3.1.关键log

以Menu Selection为例，分析Benz的modem端log，这里只截取了关键的log。

**发送Envelope:**

1.[ qmi\_cat.c 8684] Handling cmd:0x22

2.[ gstklib.c 1931] GSTK client send raw envelope of len 0x9

3.[gstk.c 4552] Queuing cmd: 0x2 0x74

4.[gstk\_s\_idle.c 724] Received Envelope cmd

5.[ gstk\_envelope\_cmd.c 3989] gstk\_process\_envelope\_cmd: command = 0x74

6.[ gstk\_envelope\_cmd.c 4045] Rec'd Raw Envelope: GSTK\_RAW\_ENVELOPE\_IND

7.[gstkutil.c 7722] 0xd3 0x7 0x82 0x2 0x1 0x81 0x90 0x1 0x1

8.[uimgen.c 2722] UIM\_1: Received Envelope command 0x110

9.[uimdrv.c 2391] UIM\_1: uim\_send\_command

**处理Envelope Response:**

10.[uim.c 1796] UIM\_1: cmd status 0x0 SW1 0x91,SW2 0x83, Response data length 0x0

11.[gstk\_envelope\_rsp.c 297] GSTK recv UIM envelope rsp, 0x91, 0x83, Report Status 0x1

12.[gstklib.c 3528] Setting proactive\_cmd\_in\_progress flag to 0x1

13.[uimgen.c 2738] UIM\_1: Received Internal Fetch command 0x111

14.[uimdrv.c 2391] UIM\_1: uim\_send\_command

15.[gstk\_envelope\_rsp.c 814] IN GSTK\_RAW\_ENVELOPE\_RSP()

16.[qmi\_cat.c 6842] Sending response for raw env rsp

17.[qmi\_cat.c 12608] Sending QMI CAT Indication to QMI Framework: clid:0x2, ind\_id:0x22

**发送Proactive Command:**

18.[uim.c 1796] UIM\_1: cmd status 0x0 SW1 0x90,SW2 0x0, Response data length 0x83

19.[uimgen.c 8777] UIM\_1: Processsing uim\_generic\_command\_response for state 0x23

20.[gstk\_s\_idle.c 686] Received Proactive cmd

21.[gstk\_proactive\_cmd.c 598] COMMAND DETAILS CMD NUM: 0x1, TYPE OF CMD: 0x24, QUALIFIER: 0x0

22.[qmi\_cat.c 6204] qmi\_cat\_cache\_raw\_proactive\_command: command\_id=0x24, slot\_index=0x0, is\_recovery\_cache=0x1

23.[qmi\_cat.c 12608] Sending QMI CAT Indication to QMI Framework: clid:0x2, ind\_id:0x1

### 3.3.2.log分析

**发送Envelope:**

1. qmi\_cat调用qmi\_cat\_process\_framework\_cmd处理上层传来的命令，cmd = 0x22。查看结构体qmi\_cati\_cmd\_callbacks确定0x22为CATI\_CMD\_VAL\_SEND\_ENVELOPE\_CMD，调用qmi\_cati\_send\_envelope\_cmd处理消息。

static qmi\_svc\_cmd\_hdlr\_type qmi\_cati\_cmd\_callbacks[] =

{

......

CAT\_HDLR( **CATI\_CMD\_VAL\_SEND\_ENVELOPE\_CMD**, qmi\_cati\_send\_envelope\_cmd ),

......

/\* NOTE: When adding an item to this table, also add it to

qmi\_cat\_is\_command\_disabled() \*/

};

1. gstk\_send\_raw\_envelope\_command被qmi\_cati\_send\_envelope\_cmd调用，将命令传至gstk quene。
2. 确定当前处理命令为0x74，查看对应的结构体gstk\_cmd\_enum\_type确定0x74为GSTK\_RAW\_ENVELOPE\_IND，如上文所说，这里也不分具体的命令类型，作为raw envelope command统一处理。

typedef enum {

.......

GSTK\_RAW\_ENVELOPE\_IND = 0x74, /\*\*< Envelope command: Raw Envelope Command. \*/

.......

}

gstk\_cmd\_enum\_type;

1. gstk处于空闲模式，进入gstk\_s\_idle.c开始处理Envelope command。
2. 根据命令类型调用gstk\_envelope\_cmd.c的gstk\_process\_envelope\_cmd。
3. 打印命令具体数据：d3 07 82 02 01 81 90 01 01

根据2.3所提做命令的解析：

1. Menu Selection tag：d3
2. 长度：07，（Menu Selectiong tag 与长度除外）
3. Device identities：82 02 01 81

其中82为Device identitiies tag， 02 为长度，01 81指该命令由keypad发至SIM。

1. 收到gstk传来的命令，0x110即UIM\_ENVELOPE\_F。（uim\_cmd\_name\_type）

/\* Command types \*/

typedef enum {

......

UIM\_ENVELOPE\_F, /\* Send a toolkit command to the UIM \*/

......

} uim\_cmd\_name\_type;

1. 由uimdrv.c将命令发至SIM卡。

**处理Envelope Response:**

1. uim收到状态字9183，即有长度为83的主动式命令。
2. uim调用gstk\_uim\_envelope\_report开始处理response。
3. 准备处理主动式命令，设置proactive\_cmd\_in\_progress flag为1。
4. uim收到gstk发来的fetch请求，0x111即UIM\_INTERNAL\_FETCH\_F。
5. 执行fetch命令。

15-17.将Envelope response发至上层。

**发送Proactive Command:**

18-23．收到Proactive Command，发至上层。

在返回状态字9000后，modem开始处理主动式命令的流程，具体的主动式命令是Select Item，用于返回二级菜单。（0x25）

根据log分析，画出对应的时序图如下：



### 3.3.3.小结

1. Envelope在不同模块中具体的定义值不同，分别是

QMI:0x22（qmi\_cati\_cmd\_callbacks）;

GSTK:0x74（gstk\_cmd\_enum\_type）;

UIM:0x110（uim\_cmd\_name\_type）

1. SIM卡在收到Envelope命令后会返回状态字91XX，通知Modem下发fetch。
2. 与Proactive Command类似，以raw代替具体的命令类型，即不同的Envelope命令的处理过程大部分是相同的。

# 总结

以上便是Envelope相关的概念与Modem端的处理流程，Modem端主要作用是将该命令发至SIM卡。目前没接触过Envelope的相关问题，但从log分析可以确定涉及到Modem端的问题分析主要集中在命令是否收发成功、命令携带的信息是否正确。命令携带的信息需通过3GPP TS 11.14确定具体的内容。