

ICA

ICA联盟白皮书

AT 指令集规范

AT Command Set Specification

V0.6 – 2018/09/06

2018 – 09 – 06 发布

ICA 发布

文件修订历史

作者	操作	说明	日期	版本
长氢	创建	根据需求起草 AT 指令集规范，包含 MQTT 和 COAP 指令	2017/9/8	0.1.0
长氢	更新	增加 OTA 指令和日志指令	2017/9/27	0.2.0
长氢、安勍	更新	增加 HTTP 指令，增加收发 MQTT 和 COAP 二进制消息的指令，增加通用指令；增加概述章节和 AT 指令语法	2017/11/28	0.3.0
长氢	更新	ICOAPRCVRSR 和 ICOAPRCVRSR 指令的参数 path 改为可选； HTTP 的 TIMEOUT 时间单位改为秒，和 MQTT 的统一； 删除 MQTT 连接状态查询指令中为“1”的情况； 细化回复消息格式； 增加二进制数据发送指令中未指定长度的情况； 在发送数据后返回的 id 或者 token 前加上相应的指令。	2018/2/23	0.4.0
长氢	更新	增加设置服务器地址和端口命令； 增加参数配置鉴权模式； 增加参数配置数据发送和接收的格式及分包计数； 网络状态变化时，增加主动上报网络状态命令。	2018/7/6	0.5.0
长氢	更新	更新 IOTABLOCK 指令的消息体为 HEX 字符串格式	2018/9/6	0.6.0

目 录

目录	3
1 范围	5
2 术语、定义和缩略语	5
2.1 缩略语	5
3 概述	5
4 AT 指令语法	6
5 MQTT 指令	7
5.1 命令总览	7
5.2 设置连接方式 +IMQTTMODE	7
5.3 设置服务器地址和端口 +IMQTTOPEN	8
5.4 设备鉴权 +IMQTTAUTH	8
5.5 设置 MQTT 参数 +IMQTTPARA	9
5.6 MQTT 连接 +IMQTTCONN	9
5.7 发布消息 +IMQTTPUB	10
5.8 发布消息 (二进制格式) +IMQTTPUBIN	11
5.9 订阅消息 +IMQTTSUB	12
5.10 取消订阅 +IMQTTUNSUB	13
5.11 收到 publish 消息 +IMQTTTRCV PUB	14
5.12 收到 publish 消息 (二进制格式) +IMQTTTRCV PUBIN	14
5.13 查询 MQTT 连接状态 +IMQTTSTATE	15
5.14 断开 MQTT 连接 +IMQTTDISCONN	15
5.15 设置调试参数 +IMQTTDBGPARA	16
5.16 收到 ping response 消息 +IMQTTPINGRSP	16
6 CoAP 指令	17
6.1 命令总览	17
6.2 设置参数 +ICOAPARA	17
6.3 设置服务器地址和端口 +ICOAPOPEN	17
6.4 设备鉴权 +ICOAPAUTH	18
6.5 发送 request 消息 +ICOAPSENDREQ	18
6.6 发送 request 消息 (二进制格式) +ICOAPSEENDBIN	20
6.7 发送不需要被确认的 request 消息 +ICOAPSENDREQN	21
6.8 发送不需要被确认的 request 消息 (二进制格式) +ICOAPSENDNBIN	22
6.9 收到 response 消息 +ICOAPRCVRSP	23
6.10 收到 response 消息 (二进制格式) +ICOAPRCV BIN	23
7 HTTP 指令	24
7.1 命令总览	24
7.2 设置服务器地址和端口 +IHTTPOPEN	24
7.3 设置参数 +IHTTPARA	25
7.4 设备鉴权 +IHTTPAUTH	25
7.5 发送消息并同步接收响应 +IHTTPSEND	26
7.6 发送 request 消息 +IHTTPSENDREQ	27
7.7 发送 request 消息 (二进制格式) +IHTTPSEENDBIN	28
7.8 收到 response 消息 +IHTTPRCVRSP	29

7.9 收到 response 消息 (二进制格式) +IHTTPRCVBIN.....	30
7.10 断开 HTTP 连接 +IHTTPDISCONN.....	30
8 OTA 指令.....	31
8.1 命令总览.....	31
8.2 设置参数 +IOTAPARA.....	33
8.3 上报版本信息 +IOTAVER.....	33
8.4 接收固件新版本信息 +IOTANEWVER.....	33
8.5 上报升级进度 +IOTAPGS.....	34
8.6 读取块数据 +IOTABLOCK.....	34
8.7 终止固件升级 +IOTABORT.....	35
9 其它指令.....	35
9.1 命令总览.....	35
9.2 设置日志等级 +ILOGLVL.....	36
9.3 重启模组 +IREBOOT.....	36
9.4 恢复出厂设置 +IDEFAULT.....	36
9.5 标准指令.....	37

1 范围

本规范规定了物联网领域端内设备通信的AT指令集。

2 术语、定义和缩略语

2.1 缩略语

下列缩略语适用于本文件。

缩略语	英文全称	中文全称
TE	Terminal Equipment	终端设备
TA	Terminal Adaptor	终端适配器
CoAP	Constrained Application Protocol	资源受限设备应用协议
DTLS	Datagram Transport Layer Security	数据包传输层安全性协议
HTTP	Hyper Text Transfer Protocol	超文本传输协议
ICA	IoT Connectivity Alliance	物联网合作伙伴计划联盟
IoT	Internet Of Things	物联网
JSON	JavaScript Object Notation	JavaScript 对象标记格式
MQTT	Message Queuing Telemetry Transport	消息队列遥测传输协议
MCU	Micro Controller Unit	微控制器
MT	Mobile Terminal	移动终端
TLS	Transport Layer Security Protocol	安全传输层协议
URL	Uniform Resource Locator	统一资源定位符
OTA	Over The Air	无线下载

3 概述

终端设备 (TE, Terminal Equipment) 可通过发送本标准所述的AT命令控制移动终端 (MT, Mobile Terminal) 功能及相关网络业务。终端适配器 (TA, Terminal Adaptor) 完成终端设备与移动设备之间的命令及消息适配功能。终端设备 (TE)、终端适配器 (TA) 及移动终端 (MT) 的物理实现可以是下列情况:

TE、TA和MT是三个独立的实体;

TE是独立的实体, TA集成在MT内部;

MT是独立的实体, TA集成在TE内部;

TE、TA和MT集成为一个实体。

在本规范中, TE认为是物联网设备的MCU模块, TA集成在MT内部, 认为是通信模块。通信模块包括但不限于GPRS通信模块、Wi-Fi通信模、LoRa通信模块。

终端设备 (TE)、终端适配器 (TA) 及移动终端 (MT) 的系统结构及之间建立关联的基本流程如图1所示。终端设备及适配器间接口可使用串口线缆、红外等方式。

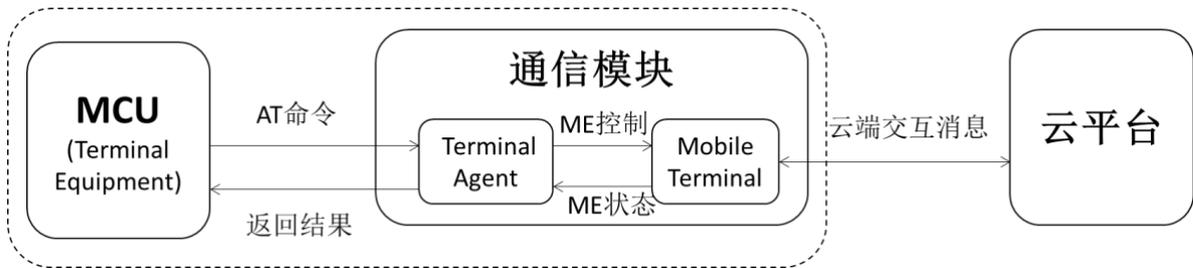


图 1 系统结构图

根据图1所示，MCU模块与通信模块共同集成于物联网设备，MCU通过AT命令与TA实现通信，从而控制MT实现物联网设备与云端的交互。

由于物联网设备与云端的交互是通过HTTP协议、MQTT协议等来进行，因此在本规范中，通过对标准的AT指令进行扩展，实现支持MQTT指令、HTTP指令、OTA指令等，从而实现物联网设备与云端的消息交互。

4 AT 指令语法

AT指令采用基于ASCII码的命令，命令格式如下：

请求消息格式为：AT+<CMD>[OP][para-1, para-2, ……para-n]<\r>

表格 1 AT 请求消息格式

域	说明
AT+	命令消息前缀
CMD	指令字符串
op	指令操作符。可以是以下内容： ✓ “=”：表示参数设置。 ✓ “?”：表示查询参数的当前值。 ✓ “”：表示执行指令。 ✓ “=?”：表示查询设置指令的参数。
para-1, para-2, …… para-n	表示设置的参数值，或者是指定要查询的参数
\r	回车结束符，ASCII码为0x0D

回应消息格式为：<\r\n>[+CMD:] [para-1, para-2, ……para-n]<\r\n>

或者：<\r\n><STATUS><\r\n>

或者上述两者都有。

表格 2 AT 回应消息格式

域	说明
\n	换行符，ASCII码为0x0A
+CMD	相应的指令字符串
para-1, para-2, …… para-n	相应的参数字符串
STATUS	指令执行状态。可以是以下内容： ✓ “OK”：表示指令执行成功。 ✓ “ERROR”：表示指令执行失败。

	✓ “+CME ERROR:<err>”: 表示指令执行失败，并返回相应的错误代码。
--	--

注：

- <>：表示必须包含的内容。
- []：表示可选的内容。
- \r：回车结束符，ASCII码为0x0D
- \n：换行符，ASCII码为0x0A。

例如，查询MQTT的连接模式，发送指令为：

AT+IMQTTMODE?\r

回复的消息为：

\r\n+IMQTTMODE:1\r\n

\r\nOK\r\n

下文中为了便于阅读，将\r\n隐藏。

5 MQTT 指令

5.1 命令总览

通信模组提供设备端使用的AT指令如下：

表格 3 MQTT 的 AT 指令集

命令	描述	实现方式
AT+IMQTTMODE	设置连接方式：是否采用TLS	可选
AT+IMQTTOPEN	设置服务器地址和端口	可选
AT+IMQTTAUTH	设备鉴权	强制
AT+IMQTTPARA	设置参数	强制
AT+IMQTTCONN	MQTT连接	强制
AT+IMQTT PUB	发布消息	强制
AT+IMQTT PUBIN	发布消息（二进制格式）	可选
AT+IMQTT SUB	订阅消息	强制
AT+IMQTT UNSUB	取消订阅	强制
+IMQTTTRCV PUB	收到publish消息	强制
+IMQTTTRCV PUBIN	收到publish消息（二进制格式）	可选
AT+IMQTT STATE	查询MQTT连接状态	强制
AT+IMQTT DISCONN	断开MQTT连接	强制
AT+IMQTT DBG	设置调试参数，如心跳等	可选
+IMQTT PINGRSP	收到ping response消息	可选

5.2 设置连接方式 +IMQTTMODE

表格 4 设置 MQTT 连接方式

命令类型	命令格式	响应
测试命令	AT+IMQTTMODE=?	+IMQTTMODE:“mode” OK
查询命令	AT+IMQTTMODE?	+IMQTTMODE:<mode> OK

执行命令	AT+IMQTTMODE=<mode>	OK
参数说明	<mode>: MQTT连接方式	
返回值说明	0: 不采用TLS 1: 采用TLS	
示例	AT+IMQTTMODE=1 OK	
注意事项	默认采用普通的TCP连接方式; 如果需要采用TLS连接方式, 请在鉴权之前使用该指令设置。	

5.3 设置服务器地址和端口 +IMQTTOPEN

表格 5 设置服务器地址和端口

命令类型	命令格式	响应
测试命令	AT+IMQTTOPEN=?	+IMQTTOPEN:"hostname", "port" OK
查询命令	AT+IMQTTOPEN?	+IMQTTOPEN:<hostname>, <port> OK
执行命令	AT+IMQTTOPEN=<hostname>, <port>	OK
参数说明	<hostname>: 服务器的域名或者IP地址;	
返回值说明	<port>: 服务器的端口, 默认1883。	
示例	AT+IMQTTOPEN="productxxx.aliyuncs.com", 1883 OK AT+IMQTTOPEN="139.196.135.135", 1883 OK	
注意事项		

5.4 设备鉴权 +IMQTTAUTH

表格 6 设备鉴权

命令类型	命令格式	响应
测试命令	AT+IMQTTAUTH=?	+IMQTTAUTH:<ParaTag1>, <ParaTag2>, ...<ParaTagn> OK
查询命令	AT+IMQTTAUTH?	+IMQTTAUTH:<ParaValue1>, <ParaValue2>, ...<ParaValuen> OK
执行命令	AT+IMQTTAUTH=<ParaValue1>, <ParaValue2>, ...<ParaValuen>	OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后可以启动自动鉴权, 返回鉴权结果。 +IMQTTAUTH:OK 鉴权成功 +IMQTTAUTH:FAIL 鉴权失败
参数说明	<ParaTag1>, <ParaTag2>,<ParaTagn>: 鉴权参数1, 2,n的名称;	
返回值说明	<ParaValue1>, <ParaValue2>,<ParaValuen>: 鉴权参数1, 2,n的参数值;	

	<err>: error代码, 详见《AT command set for User Equipment (UE)》。
示例	AT+MQTTAUTH="productKey001", "deviceName001", "deviceSecret123" OK +MQTTAUTH:OK
注意事项	终端设备和通信模组需要根据实际应用场景, 事先约定参数的个数及具体含义; 使用参数鉴权成功后, 建议将参数保存在通信模组的非易失存储器中, 下次可直接使用AT+MQTTAUTH进行鉴权; 如果该指令仅用于鉴权参数设置, 而不启动鉴权, 无须返回鉴权结果。 鉴权模式的配置使用AT+MQTTPARA指令中的“AUTHMODE”参数。

5.5 设置 MQTT 参数 +MQTTPARA

表格 7 设置 MQTT 参数

命令类型	命令格式	响应										
测试命令	AT+MQTTPARA=?	+MQTTPARA:“ParaTag”, “ParaValue” OK										
查询命令	AT+MQTTPARA?	+MQTTPARA:<ParaTag>, <ParaValue> OK										
执行命令	AT+MQTTPARA=<ParaTag>, <ParaValue> >	OK 或者 +CME ERROR:<err>										
参数说明	<ParaTag>: MQTT参数名称, 包含以下参数项;											
返回值说明	<table border="1"> <tr> <td>“TIMEOUT”</td> <td>MQTT请求超时时间, 单位秒。</td> </tr> <tr> <td>“CLEAN”</td> <td>是否清除session, 0: 不清除; 1: 清除。</td> </tr> <tr> <td>“KEEPALIVE”</td> <td>保活时间, 单位秒。</td> </tr> <tr> <td>“VERSION”</td> <td>MQTT版本, “3.1”或者“3.1.1”。</td> </tr> <tr> <td>“AUTHMODE”</td> <td>MQTTAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。</td> </tr> </table> <ParaValue>: MQTT参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。		“TIMEOUT”	MQTT请求超时时间, 单位秒。	“CLEAN”	是否清除session, 0: 不清除; 1: 清除。	“KEEPALIVE”	保活时间, 单位秒。	“VERSION”	MQTT版本, “3.1”或者“3.1.1”。	“AUTHMODE”	MQTTAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。
“TIMEOUT”	MQTT请求超时时间, 单位秒。											
“CLEAN”	是否清除session, 0: 不清除; 1: 清除。											
“KEEPALIVE”	保活时间, 单位秒。											
“VERSION”	MQTT版本, “3.1”或者“3.1.1”。											
“AUTHMODE”	MQTTAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。											
示例	AT+MQTTPARA="TIMEOUT", 2 OK AT+MQTTPARA="TIMEOUT", 2, "CLEAN", 1, "KEEPALIVE", 60, "VERSION", "3.1.1" OK											
注意事项	查询命令可以查询所有参数的名称和相应的值; 执行命令可以设置一个或多个参数。											

5.6 MQTT 连接 +MQTTCONN

表格 8 MQTT 连接

命令类型	命令格式	响应
测试命令	AT+MQTTCONN=?	OK
执行命令	AT+MQTTCONN	OK 或者 +CME ERROR:<err>
参数说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。	
返回值说明		
示例	AT+MQTTCONN	

	OK
注意事项	调用该指令前，请先用AT+MQTTPARA指令设置MQTT参数。

5.7 发布消息 +MQTTPUB

表格 9 发布消息

命令类型	命令格式	响应						
测试命令	AT+MQTTPUB=?	+MQTTPUB:“topic”,“qos”,“message”, “format”,“fragment_id” OK						
查询命令	AT+MQTTPUB?	+MQTTPUB:<topic>,<qos>,<message>,[forma t],[fragment_id] OK						
执行命令	AT+MQTTPUB=<topic>,<qos>,<messag e>,[format],[fragment_id]	+MQTTPUB:<packet_id> OK 或者 +CME ERROR:<err> 如果输入合法，首先返回OK，然后主动上报结果。 +MQTTPUB:<packet_id>,<status>						
参数说明	<topic>: 发布的主题; <qos>: QoS值, 范围0~2; <message>: 发布的消息体内容; <packet_id>: 发布消息时返回的id; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。							
返回值说明	如果该字段为1时, 通信模组需要将HEX字符串转换成二进制数据格式, 再执行发布操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 终端模组发布长消息前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。 如果通信模组收到的数据包里面分包计数器大于0, 它需要继续接收MQTTPUB命令, 直到收到分包计数器为0, 然后将收到的消息按顺序组包, 再执行发布操作。 <status>: 发布消息返回状态值: <table border="1" data-bbox="300 1496 699 1626"> <tr> <td>0:</td> <td>成功。</td> </tr> <tr> <td>1:</td> <td>超时。</td> </tr> <tr> <td>2:</td> <td>其它错误。</td> </tr> </table> <err>: error代码, 详见《AT command set for User Equipment (UE)》。		0:	成功。	1:	超时。	2:	其它错误。
0:	成功。							
1:	超时。							
2:	其它错误。							
示例	发布短消息示例: AT+MQTTPUB="/test_topic_001",1,"hellosim" +MQTTPUB:24761 OK +MQTTPUB:24761,0 发布长消息示例:							

	<pre> AT+MQTTPUB="/test_topic_002",1,{"\name\":"hellosim123",0,1 OK AT+MQTTPUB="/test_topic_002",1,"456789end\}"}",0,0 +MQTTPUB:24762 OK +MQTTPUB:24762,0 发布HEX字符串消息示例: AT+MQTTPUB="/test_topic_003",1,"0001AC",1 +MQTTPUB:24763 OK +MQTTPUB:24763,0 </pre>
注意事项	<p>对于一些复杂的消息，消息体内容带引号(“)或其它特殊字符，则需要用转义符(\)转义。如果仅存在一个可选的字段，则该字段必须是format字段；如果存在fragment_id字段，则format字段也必须存在。</p> <p>对于长消息体需要分包的情况，仅最后一包需要回packet_id，其它的分包只需要回OK。</p>

5.8 发布消息（二进制格式） +MQTTPUBIN

表格 10 发布消息（二进制格式）

命令类型	命令格式	响应
测试命令	AT+MQTTPUBIN=?	+MQTTPUBIN:"topic","qos","msg_len" OK "bin_message"
查询命令	AT+MQTTPUBIN?	+MQTTPUBIN:<topic>,<qos>,<msg_len> OK <bin_message>
执行命令	AT+MQTTPUBIN=<topic>,<qos>,<msg_len>	当msg_len大于0时, OK 当msg_len等于0时, >
	<bin_message>[+++]	+MQTTPUBIN:<packet_id> OK 或者 +CME ERROR:<err> 如果输入合法，首先返回OK，然后主动上报结果。 +MQTTPUBIN:<packet_id>,<status>

参数说明	<p><topic>: 发布的主题;</p> <p><qos>: QoS值, 范围0~2;</p> <p><msg_len>: 发布的消息长度;</p> <p><bin_message>: 发布的消息体内容, 二进制格式;</p> <p>[+++]: 从二进制数据传输模式返回命令模式, 仅当msg_len为0时存在;</p>						
返回值说明	<p><packet_id>: 发布消息时返回的id;</p> <p><status>: 发布消息返回状态值:</p> <table border="1" style="margin-left: 20px;"> <tr> <td>0:</td> <td>成功。</td> </tr> <tr> <td>1:</td> <td>超时。</td> </tr> <tr> <td>2:</td> <td>其它错误。</td> </tr> </table> <p><err>: error代码, 详见《AT command set for User Equipment (UE)》。</p>	0:	成功。	1:	超时。	2:	其它错误。
0:	成功。						
1:	超时。						
2:	其它错误。						
示例	<pre>AT+IMQTTPUBIN="/test_topic_001",1,5 OK \x11\x22\x33\xAA\xBA +IMQTTPUBIN:24761 OK +IMQTTPUBIN:24761,0</pre>						
注意事项	<p>该指令用于发送二进制格式的消息, 共分为两步:</p> <ol style="list-style-type: none"> 1. 终端设备发送指令AT+IMQTTPUBIN, 指定topic, qos和二进制消息长度; 通信模组回复OK。 2. 终端设备发送指定长度的二进制数据流; 通信模组回复消息ID和OK后, 转发该数据包。转发完成后, 主动上报发送结果给终端设备。 <p>如果数据长度为0, 则终端设备先发送二进制数据, 最后发送+++ , 以返回命令模式; 通信模组回复消息ID和OK后, 转发该数据包。</p>						

5.9 订阅消息 +IMQTTSUB

表格 11 订阅消息

命令类型	命令格式	响应
测试命令	AT+IMQTTSUB=?	+IMQTTSUB:"topic","qos" OK
查询命令	AT+IMQTTSUB?	+IMQTTSUB:<topic>,<qos> OK
执行命令	AT+IMQTTSUB=<topic>,<qos>	+IMQTTSUB:<packet_id> OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后主动上报结果。 +IMQTTSUB:<packet_id>,<status>

参数说明	<p><topic>: 订阅的主题;</p> <p><qos>: QoS值, 范围0~2;</p> <p><packet_id>: 订阅消息时返回的id;</p> <p><status>: 订阅消息返回状态值:</p> <table border="1"> <tr> <td>0:</td> <td>成功, QoS0。</td> </tr> <tr> <td>1:</td> <td>成功, QoS1。</td> </tr> <tr> <td>2:</td> <td>成功, QoS2。</td> </tr> <tr> <td>128:</td> <td>失败。</td> </tr> </table>	0:	成功, QoS0。	1:	成功, QoS1。	2:	成功, QoS2。	128:	失败。
0:	成功, QoS0。								
1:	成功, QoS1。								
2:	成功, QoS2。								
128:	失败。								
返回值说明	<p><err>: error代码, 详见《AT command set for User Equipment (UE)》。</p>								
示例	<p>AT+MQTTSUB="/test_topic_001",2</p> <p>+MQTTSUB:24761</p> <p>OK</p> <p>+MQTTSUB:24761,1</p>								
注意事项	查询指令只返回最后一个订阅的topic和对应的QoS。								

5.10 取消订阅 +MQTTUNSUB

表格 12 取消订阅

命令类型	命令格式	响应						
测试命令	AT+MQTTUNSUB=?	+MQTTUNSUB:"topic" OK						
查询命令	AT+MQTTUNSUB?	+MQTTUNSUB:<topic> OK						
执行命令	AT+MQTTUNSUB=<topic>	+MQTTUNSUB:<packet_id> OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后主动上报结果。 +MQTTUNSUB:<packet_id>,<status>						
参数说明	<p><topic>: 取消订阅的主题;</p> <p><packet_id>: 取消订阅消息时返回的id;</p> <p><status>: 取消订阅消息返回状态值:</p> <table border="1"> <tr> <td>0:</td> <td>成功。</td> </tr> <tr> <td>1:</td> <td>超时。</td> </tr> <tr> <td>2:</td> <td>其它错误。</td> </tr> </table>	0:	成功。	1:	超时。	2:	其它错误。	
0:	成功。							
1:	超时。							
2:	其它错误。							
返回值说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。							
示例	<p>AT+MQTTUNSUB="/test_topic_001"</p> <p>+MQTTUNSUB:24761</p> <p>OK</p>							

	+IMQTTUNSUB:24761,0
注意事项	查询指令只返回最后一个取消订阅的topic。

5.11 收到 publish 消息 +IMQTTTRCV PUB

表格 13 收到 publish 消息

命令类型	命令格式	响应
测试命令	AT+IMQTTTRCV PUB=?	OK
查询命令	AT+IMQTTTRCV PUB?	+IMQTTTRCV PUB:<packet_id>,<topic>,<msg_len>,<message>,[format],[fragment_id] OK
		主动上报收到的消息: +IMQTTTRCV PUB:<packet_id>,<topic>,<msg_len>,<message>,[format],[fragment_id] OK
参数说明	<topic>: 收到消息的主题; <packet_id>: 收到消息的id;	
返回值说明	<msg_len>: 收到消息体的长度; <message>: 收到消息体的内容; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。 如果该字段为1时, 终端模组需要将HEX字符串转换成二进制数据格式, 再执行后续操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 通信模组将收到的长消息发给终端模组前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。 如果终端模组收到的数据包里面分包计数器大于0, 它需要继续接收IMQTTTRCV PUB命令, 直到收到分包计数器为0, 然后将收到的消息按顺序组包, 再执行后续操作。	
示例	+IMQTTTRCV PUB=24761,“/test_topic_001”,5,“hello” OK +IMQTTTRCV PUB=24762,“/test_topic_002”,6,“0001AC”,1 OK	
注意事项	通信模组在收到消息后, 可以主动上报收到的消息; 如果通信模组不支持主动上报模式, 终端设备可以采用查询命令去读取收到的消息。 推荐使用主动上报模式。 对于一些复杂的消息, 消息体内容带引号(”)或其它特殊字符, 则需要用转义符(\)转义。 如果仅存在一个可选的字段, 则该字段必须是format字段; 如果存在fragment_id字段, 则format字段也必须存在。	

5.12 收到 publish 消息（二进制格式） +IMQTTTRCV PUBIN

表格 14 收到 publish 消息（二进制格式）

命令类型	命令格式	响应
测试命令	AT+IMQTTTRCV PUBIN=?	OK
查询命令	AT+IMQTTTRCV PUBIN?	+IMQTTTRCV PUBIN:<packet_id>,<topic>,<msg_len> OK <bin_message>

		主动上报收到的消息： +MQTTTRCVBIN:<packet_id>,<topic>,<msg_len> OK <bin_message>
参数说明	<topic>: 收到消息的主题； <packet_id>: 收到消息的id；	
返回值说明	<msg_len>: 收到消息体的长度； <bin_message>: 收到消息体的内容，二进制格式。	
示例	+MQTTTRCVBIN=24761,“/test_topic_001”,5 OK \x11\x22\x33\xAA\xBA	
注意事项	通信模组在收到消息后，可以主动上报收到的消息； 如果通信模组不支持主动上报模式，终端设备可以采用查询命令去读取收到的消息。 推荐使用主动上报模式。	

5.13 查询 MQTT 连接状态 +MQTTSTATE

表格 15 查询 MQTT 连接状态

命令类型	命令格式	响应
测试命令	AT+MQTTSTATE=?	OK
查询命令	AT+MQTTSTATE?	+MQTTSTATE:<state> OK 当网络状态发生变化时，模组也可以主动上报： +MQTTSTATE:<state> OK
参数说明	<state>: MQTT连接状态	
返回值说明	0: MQTT断开状态。 1: 保留。 2: MQTT连接状态。	
示例	查询网络状态示例： AT+MQTTSTATE? +MQTTSTATE:2 OK 网络状态变化时，主动上报网络状态示例： +MQTTSTATE:0 OK	
注意事项		

5.14 断开 MQTT 连接 +MQTTDISCONN

表格 16 断开 MQTT 连接

命令类型	命令格式	响应
测试命令	AT+MQTTDISCONN=?	OK
执行命令	AT+MQTTDISCONN	OK 或者

		+CME ERROR:<err>
参数说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。	
返回值说明		
示例	AT+IMQTTDISCONN OK	
注意事项		

5.15 设置调试参数 +IMQTTDBGPARA

表格 17 设置调试参数

命令类型	命令格式	响应						
测试命令	AT+IMQTTDBGPARA=?	+IMQTTDBGPARA:“ParaTag”,“ParaValue” OK						
查询命令	AT+IMQTTDBGPARA?	+IMQTTDBGPARA:<ParaTag>,<ParaValue> OK						
执行命令	AT+IMQTTDBGPARA=<ParaTag>,<ParaValue>	OK 或者 +CME ERROR:<err>						
参数说明	<ParaTag>: 调试参数名称, 包含以下参数项;							
返回值说明	<table border="1" style="width: 100%;"> <tr> <td>“PINGRSP”</td> <td>通信模组上报ping response设置:</td> </tr> <tr> <td>0:</td> <td>禁止主动上报, 默认值;</td> </tr> <tr> <td>1:</td> <td>使能主动上报。</td> </tr> </table> <ParaValue>: 调试参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。		“PINGRSP”	通信模组上报ping response设置:	0:	禁止主动上报, 默认值;	1:	使能主动上报。
“PINGRSP”	通信模组上报ping response设置:							
0:	禁止主动上报, 默认值;							
1:	使能主动上报。							
示例	AT+IMQTTDBGPARA=“PINGRSP”, 1 OK							
注意事项	通信模组根据KEEPALIVE设定的时间, 定期发送ping消息, 也会收到response消息。默认情况下, response消息不会上报给终端设备。 如果通过该AT指令, 设置PINGRSP为1, 通信模组收到ping消息的response后, 会主动上报给终端设备。							

5.16 收到 ping response 消息 +IMQTTPINGRSP

表格 18 收到 ping response 消息

命令类型	命令格式	响应
测试命令	AT+IMQTTPINGRSP=?	OK
		主动上报收到的消息: +IMQTTPINGRSP OK
参数说明		
返回值说明		
示例	+IMQTTPINGRSP OK	
注意事项	只有在通过AT+IMQTTDBGPARA命令使能ping response主动上报后, 通信模组才会主动上报收到的ping response消息。	

6 CoAP 指令

6.1 命令总览

通信模组提供设备端使用的AT指令如下：

表格 19 COAP 的 AT 指令集

命令	描述	实现方式
AT+ICOAPARA	设置参数	可选
AT+ICOAPOPEN	设置服务器地址和端口	可选
AT+ICOAPAUTH	设备鉴权	强制
AT+ICOAPSENDRQ	发送request消息	强制
AT+ICOAPSENCBIN	发送request消息（二进制格式）	可选
AT+ICOAPSENDRQCN	发送不需要被确认的request消息	强制
AT+ICOAPSENDCBIN	发送不需要被确认的request消息（二进制格式）	可选
+ICOAPRCVRSP	接收response消息	强制
+ICOAPRCVCBIN	接收response消息（二进制格式）	可选

6.2 设置参数 +ICOAPARA

表格 20 设置参数

命令类型	命令格式	响应
测试命令	AT+ICOAPARA=?	+ICOAPARA:“ParaTag”, “ParaValue” OK
查询命令	AT+ICOAPARA?	+ICOAPARA:<ParaTag>, <ParaValue> OK
执行命令	AT+ICOAPARA=<ParaTag>, <ParaValue>	OK 或者 +CME ERROR:<err>
参数说明	<ParaTag>: 参数名称, 包含以下参数项;	
返回值说明	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> “TIMEOUT” COAP请求超时时间, 单位秒。 </div> <ParaValue>: 参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+ICOAPARA=“TIMEOUT”, 2 OK	
注意事项	查询命令可以查询所有参数的名称和相应的值; 执行命令可以设置一个或多个参数。	

6.3 设置服务器地址和端口 +ICOAPOPEN

表格 21 设置服务器地址和端口

命令类型	命令格式	响应
测试命令	AT+ICOAPOPEN=?	+ICOAPOPEN:“hostname”, “port” OK
查询命令	AT+ICOAPOPEN?	+ICOAPOPEN:<hostname>, <port> OK

执行命令	AT+ICOAPOPEN=<hostname>, <port>	OK
参数说明	<hostname>: 服务器的域名或者IP地址; <port>: 服务器的端口, 默认5684。	
返回值说明		
示例	AT+ICOAPOPEN="productxxx.aliyuncs.com", 5684 OK AT+ICOAPOPEN="139.196.135.101", 5684 OK	
注意事项		

6.4 设备鉴权 +ICOAPAUTH

表格 22 设备鉴权

命令类型	命令格式	响应
测试命令	AT+ICOAPAUTH=?	+ICOAPAUTH:<ParaTag1>, <ParaTag2>, ...<ParaTagn> OK
查询命令	AT+ICOAPAUTH?	+ICOAPAUTH:<ParaValue1>, <ParaValue2>, ...<ParaValuen> OK
执行命令	AT+ICOAPAUTH=<ParaValue1>, <ParaValue2>, ...<ParaValuen>	OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后可以启动自动鉴权, 返回鉴权结果。 +ICOAPAUTH:OK 鉴权成功 +ICOAPAUTH:FAIL 鉴权失败
参数说明	<ParaTag1>, <ParaTag2>,<ParaTagn>: 鉴权参数1, 2,n的名称;	
返回值说明	<ParaValue1>, <ParaValue2>,<ParaValuen>: 鉴权参数1, 2,n的参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+ICOAPAUTH="productKey001", "deviceName001", "deviceSecret123" OK +ICOAPAUTH:OK	
注意事项	终端设备和通信模组需要根据实际应用场景, 事先约定参数的个数及具体含义。	

6.5 发送 request 消息 +ICOAPSENDREQ

表格 23 发送 request 消息

命令类型	命令格式	响应
测试命令	AT+ICOAPSENDREQ=?	+ICOAPSENDREQ:"method", "path", "message", "format", "fragment_id" OK
查询命令	AT+ICOAPSENDREQ?	+ICOAPSENDREQ:<method>, <path>, <message>, [format], [fragment_id] OK
执行命令	AT+ICOAPSENDREQ=<method>, <path>, <message>, [format], [fragment_id]	+ICOAPSENDREQ:<packet_token> OK

		<p>或者 +CME ERROR:<err></p> <p>如果输入合法，首先返回OK，然后主动上报结果。 +ICOAPSENDREQ:<packet_token>,<status></p>								
<p>参数说明</p>	<p><method>: 发送消息的方法:</p> <table border="1" data-bbox="296 432 703 600"> <tr><td>1:</td><td>GET</td></tr> <tr><td>2:</td><td>POST</td></tr> <tr><td>3:</td><td>PUT</td></tr> <tr><td>4:</td><td>DELETE</td></tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
<p>返回值说明</p>	<p><path>: 消息对应的path; <message>: 发送的消息体内容; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。 如果该字段为1时, 通信模组需要将HEX字符串转换成二进制数据格式, 再执行发送操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 终端模组发送长消息前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。 如果通信模组收到的数据包里面分包计数器大于0, 它需要继续接收ICOAPSENDREQ命令, 直到收到分包计数器为0, 然后将收到的消息按顺序组包, 再执行发送操作。 <packet_token>: 发送消息时返回的token, 用于和response消息匹配; <status>: 发布消息返回状态值:</p> <table border="1" data-bbox="296 1059 703 1182"> <tr><td>0:</td><td>成功。</td></tr> <tr><td>1:</td><td>超时。</td></tr> <tr><td>2:</td><td>其它错误。</td></tr> </table> <p><err>: error代码, 详见《AT command set for User Equipment (UE)》。</p>		0:	成功。	1:	超时。	2:	其它错误。		
0:	成功。									
1:	超时。									
2:	其它错误。									
<p>示例</p>	<p>发送短消息示例: AT+ICOAPSENDREQ=2, "/livingroom/light", "{\ "action\":\ "turnon\"}"</p> <p>+ICOAPSENDREQ:24761 OK</p> <p>+ICOAPSENDREQ:24761,0</p> <p>发送长消息示例: AT+ICOAPSENDREQ=2, "/livingroom/light", "{\ "name\":\ "hellosim123",0,1</p> <p>OK</p> <p>AT+ICOAPSENDREQ=2, "/livingroom/light", "456789end\}",0,0</p> <p>+ICOAPSENDREQ:24762 OK</p> <p>+ICOAPSENDREQ:24762,0</p> <p>发送HEX字符串消息示例: AT+ICOAPSENDREQ=2, "/livingroom/switch", "0001AC",1</p>									

	+ICOAPSENDREQ:24763 OK +ICOAPSENDREQ:24763,0
注意事项	查询命令只返回最后一个被发送的request消息。 如果仅存在一个可选的字段，则该字段必须是format字段；如果存在fragment_id字段，则format字段也必须存在。 对于长消息体需要分包的情况，仅最后一包需要回packet_token，其它的分包只需要回OK。

6.6 发送 request 消息（二进制格式） +ICOAPSENBIN

表格 24 发送 request 消息（二进制格式）

命令类型	命令格式	响应								
测试命令	AT+ICOAPSENBIN=?	+ICOAPSENBIN:"method", "path", "msg_len" OK "bin_message"								
查询命令	AT+ICOAPSENBIN?	+ICOAPSENBIN:<method>, <path>, <msg_len> OK <bin_message>								
执行命令	AT+ICOAPSENBIN=<method>, <path>, <msg_len>	当msg_len大于0时, OK 当msg_len等于0时, >								
	<bin_message>[+++]	+ICOAPSENBIN:<packet_token> OK 或者 +CME ERROR:<err> 如果输入合法，首先返回OK，然后主动上报结果。 +ICOAPSENBIN:<packet_token>, <status>								
参数说明	<method>: 发送消息的方法: <table border="1" data-bbox="295 1541 702 1711"> <tr> <td>1:</td> <td>GET</td> </tr> <tr> <td>2:</td> <td>POST</td> </tr> <tr> <td>3:</td> <td>PUT</td> </tr> <tr> <td>4:</td> <td>DELETE</td> </tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
返回值说明	<path>: 消息对应的path; <msg_len>: 发送的消息长度; <bin_message>: 发送的消息体内容，二进制格式; [+++]: 从二进制数据传输模式返回命令模式，仅当msg_len时为0存在; <packet_token>: 发送消息时返回的token，用于和response消息匹配; <status>: 发布消息返回状态值: <table border="1" data-bbox="295 1962 702 2002"> <tr> <td>0:</td> <td>成功。</td> </tr> </table>		0:	成功。						
0:	成功。									

	<table border="1"> <tr> <td>1:</td> <td>超时。</td> </tr> <tr> <td>2:</td> <td>其它错误。</td> </tr> </table> <p><err>: error代码, 详见《AT command set for User Equipment (UE)》。</p>	1:	超时。	2:	其它错误。
1:	超时。				
2:	其它错误。				
示例	<pre>AT+ICOAPSENDBIN=2, "/livingroom/light", 5 OK \x11\x22\x33\xAA\xBA +ICOAPSENDBIN:24761 OK +ICOAPSENDBIN:24761, 0</pre>				
注意事项	<p>该指令用于发送二进制格式的消息, 共分为两步:</p> <ol style="list-style-type: none"> 1. 终端设备发送指令AT+ICOAPSENDBIN, 指定method、path和二进制消息长度; 通信模组回复OK。 2. 终端设备发送指定长度的二进制数据流; 通信模组回复消息token和OK后, 转发该数据包。转发完成后, 主动上报发送结果给终端设备。 <p>如果数据长度为0, 则终端设备先发送二进制数据, 最后发送+++ , 以返回命令模式; 通信模组回复消息token和OK后, 转发该数据包。</p>				

6.7 发送不需要被确认的 request 消息 +ICOAPSENDREQN

表格 25 发送 request 消息

命令类型	命令格式	响应								
测试命令	AT+ICOAPSENDREQN=?	+ICOAPSENDREQN:"method", "path", "message", "format", "fragment_id" OK								
查询命令	AT+ICOAPSENDREQN?	+ICOAPSENDREQN:<method>, <path>, <message> , [format], [fragment_id] OK								
执行命令	AT+ICOAPSENDREQN=<method>, <path>, <message>, [format], [fragment_id]	+ICOAPSENDREQN:<packet_token> OK 或者 +CME ERROR:<err>								
参数说明	<p><method>: 发送消息的方法:</p> <table border="1"> <tr> <td>1:</td> <td>GET</td> </tr> <tr> <td>2:</td> <td>POST</td> </tr> <tr> <td>3:</td> <td>PUT</td> </tr> <tr> <td>4:</td> <td>DELETE</td> </tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
返回值说明	<p><path>: 消息对应的path; <message>: 发送的消息体内容; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。 如果该字段为1时, 通信模组需要将HEX字符串转换成二进制数据格式, 再执行发送操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 终端模组发送长消息前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。</p>									

	<p>如果通信模组收到的数据包里面分包计数器大于0，它需要继续接收ICOAPSENDREQN命令，直到收到分包计数器为0，然后将收到的消息按顺序组包，再执行发送操作。</p> <p><packet_token>: 发送消息时返回的token，用于和response消息匹配；</p> <p><err>: error代码，详见《AT command set for User Equipment (UE)》。</p>
示例	<p>AT+ICOAPSENDREQN=2, "/livingroom/light", "{\“action\”:\“turnon\”}"</p> <p>+ICOAPSENDREQN:24761</p> <p>OK</p>
注意事项	<p>查询命令只返回最后一个被发送的不需要被确认的request消息。</p> <p>如果仅存在一个可选的字段，则该字段必须是format字段；如果存在fragment_id字段，则format字段也必须存在。</p> <p>对于长消息体需要分包的情况，仅最后一包需要回packet_token，其它的分包只需要回OK。</p>

6.8 发送不需要被确认的 request 消息（二进制格式） +ICOAPSENDNBIN

表格 26 发送 request 消息（二进制格式）

命令类型	命令格式	响应								
测试命令	AT+ICOAPSENDNBIN=?	+ICOAPSENDNBIN:"method", "path", "msg_len" OK "bin_message"								
查询命令	AT+ICOAPSENDNBIN?	+ICOAPSENDNBIN:<method>, <path>, <msg_len> OK <bin_message>								
执行命令	AT+ICOAPSENDNBIN=<method>, <path>, <msg_len>	当msg_len大于0时, OK 当msg_len等于0时, >								
	<bin_message>[+++]	+ICOAPSENDNBIN:<packet_token> OK 或者 +CME ERROR:<err>								
参数说明	<p><method>: 发送消息的方法:</p> <table border="1" style="margin-left: 20px;"> <tr><td>1:</td><td>GET</td></tr> <tr><td>2:</td><td>POST</td></tr> <tr><td>3:</td><td>PUT</td></tr> <tr><td>4:</td><td>DELETE</td></tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
返回值说明	<p><path>: 消息对应的path;</p> <p><msg_len>: 发送的消息长度;</p> <p><bin_message>: 发送的消息体内容，二进制格式;</p> <p>[+++]: 从二进制数据传输模式返回命令模式，仅当msg_len为0时存在;</p> <p><packet_token>: 发送消息时返回的token，用于和response消息匹配;</p> <p><err>: error代码，详见《AT command set for User Equipment (UE)》。</p>									
示例	<p>AT+ICOAPSENDNBIN=2, "/livingroom/light", 5</p> <p>OK</p>									

	<pre>\x11\x22\x33\xAA\xBA</pre> <pre>+ICoAPSENDNBIN:24761</pre> <pre>OK</pre>
注意事项	<p>查询命令只返回最后一个被发送的不需要被确认的request消息。</p> <p>该指令用于发送二进制格式的消息，共分为两步：</p> <ol style="list-style-type: none"> 1. 终端设备发送指令AT+ICoAPSENDNBIN，指定method、path和二进制消息长度；通信模组回复OK。 2. 终端设备发送指定长度的二进制数据流；通信模组回复消息token和OK后，转发该数据包。 <p>如果数据长度为0，则终端设备先发送二进制数据，最后发送+++，以返回命令模式；通信模组回复消息token和OK后，转发该数据包。</p>

6.9 收到 response 消息 +ICoAPRCVRSP

表格 27 收到 response 消息

命令类型	命令格式	响应
测试命令	AT+ICoAPRCVRSP=?	OK
查询命令	AT+ICoAPRCVRSP?	+ICoAPRCVRSP:<packet_token>,<msg_len>,<message>,[format],[fragment_id] OK
		主动上报收到的消息： +ICoAPRCVRSP:<packet_token>,<msg_len>,<message>,[format],[fragment_id] OK
参数说明	<p><packet_token>：收到消息的token，与request消息返回的值匹配；</p> <p><msg_len>：收到消息体的长度；</p>	
返回值说明	<p><message>：收到消息体的内容；</p> <p>[format]：消息体格式，0：一般字符串；1：HEX字符串。可选字段，默认为0。</p> <p>如果该字段为1时，终端模组需要将HEX字符串转换成二进制数据格式，再执行后续操作；</p> <p>[fragment_id]：长消息体分包计数器。可选字段，默认为0。</p> <p>通信模组将收到的长消息发给终端模组前，需要先设置分包计数器为最大分包数-1；然后每发送一个新的分包，计数器-1，0表示这是最后一个分包。如果重发一个分包，分包计数器保持不变。</p> <p>如果终端模组收到的数据包里面分包计数器大于0，它需要继续接收ICoAPRCVPUB命令，直到收到分包计数器为0，然后将收到的消息按顺序组包，再执行后续操作。</p>	
示例	<pre>+ICoAPRCVRSP=24761,"/livingroom/light",15,"{\\"status\\":\\"on\\"}"</pre> <pre>OK</pre>	
注意事项	<p>通信模组在收到消息后，可以主动上报收到的消息；</p> <p>如果通信模组不支持主动上报模式，终端设备可以采用查询命令去读取收到的消息。</p> <p>如果仅存在一个可选的字段，则该字段必须是format字段；如果存在fragment_id字段，则format字段也必须存在。</p>	

6.10 收到 response 消息（二进制格式） +ICoAPRCVBIN

表格 28 收到 response 消息

命令类型	命令格式	响应
------	------	----

测试命令	AT+ICOAPRCVBIN=?	OK
查询命令	AT+ICOAPRCVBIN?	+ICOAPRCVBIN:<packet_token>, [path],<msg_len> OK <bin_message>
		主动上报收到的消息: +ICOAPRCVBIN:<packet_token>,[path],<msg_ len> OK <bin_message>
参数说明	[path]: 收到消息的path, 该字段可省略; <packet_token>: 收到消息的token, 与request消息返回的值匹配;	
返回值说明	<msg_len>: 收到消息体的长度; <bin_message>: 收到消息体的内容, 二进制格式。	
示例	+ICOAPRCVBIN=24761,“/livingroom/light”,5 OK \x11\x22\x33\xAA\xBA	
注意事项	通信模组在收到消息后, 可以主动上报收到的消息; 如果通信模组不支持主动上报模式, 终端设备可以采用查询命令去读取收到的消息。	

7 HTTP 指令

7.1 命令总览

通信模组提供设备端使用的AT指令如下:

表格 29 HTTP 的 AT 指令集

命令	描述	实现方式
AT+IHTTPOPEN	设置服务器地址和端口	可选
AT+IHTTPARA	设置参数	可选
AT+IHTTPAUTH	设备鉴权	强制
AT+IHTTPSEND	发送消息并同步接收响应	强制
AT+IHTTPSENDREQ	发送request消息	可选
AT+IHTTPSENCBIN	发送request消息, 二进制格式	可选
+IHTTPRCVRSP	接收response消息	可选
+IHTTPRCVBIN	接收response消息, 二进制格式	可选
AT+IHTTPDISCONN	断开连接	强制

7.2 设置服务器地址和端口 +IHTTPOPEN

表格 30 设置服务器地址和端口

命令类型	命令格式	响应
测试命令	AT+IHTTPOPEN=?	+IHTTPOPEN: “hostname”, “port” OK
查询命令	AT+IHTTPOPEN?	+IHTTPOPEN:<hostname>,<port>

		OK
执行命令	AT+IHTTPOPEN=<hostname>, <port>	OK
参数说明	<hostname>: 服务器的域名或者IP地址;	
返回值说明	<port>: 服务器的端口, 默认443。	
示例	AT+IHTTPOPEN="productxxx.aliyuncs.com", 443 OK	
注意事项		

7.3 设置参数 +IHTTPARA

表格 31 设置参数

命令类型	命令格式	响应						
测试命令	AT+IHTTPARA=?	+IHTTPARA:"ParaTag", "ParaValue" OK						
查询命令	AT+IHTTPARA?	+IHTTPARA:<ParaTag>, <ParaValue> OK						
执行命令	AT+IHTTPARA=<ParaTag>, <ParaValue>	OK 或者 +CME ERROR:<err>						
参数说明	<ParaTag>: 参数名称, 包含以下参数项;							
返回值说明	<table border="1"> <tr> <td>"KEEPALIVE"</td> <td>HTTP保活模式设置: 0: 关闭保活功能, 默认值; 1: 使能保活功能。</td> </tr> <tr> <td>"TIMEOUT"</td> <td>同步接收方式的等待超时时间, 单位秒, 默认值5。</td> </tr> <tr> <td>"AUTHMODE"</td> <td>IHTTPAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。</td> </tr> </table> <ParaValue>: 参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。		"KEEPALIVE"	HTTP保活模式设置: 0: 关闭保活功能, 默认值; 1: 使能保活功能。	"TIMEOUT"	同步接收方式的等待超时时间, 单位秒, 默认值5。	"AUTHMODE"	IHTTPAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。
"KEEPALIVE"	HTTP保活模式设置: 0: 关闭保活功能, 默认值; 1: 使能保活功能。							
"TIMEOUT"	同步接收方式的等待超时时间, 单位秒, 默认值5。							
"AUTHMODE"	IHTTPAUTH的鉴权模式, 0: 仅设置参数, 而不启动鉴权; 1: 设置参数并启动鉴权。默认值为1。							
示例	AT+IHTTPARA="KEEPALIVE", 1, "TIMEOUT", 6 OK							
注意事项	查询命令可以查询所有参数的名称和相应的值; 执行命令可以设置一个或多个参数。							

7.4 设备鉴权 +IHTTPAUTH

表格 32 设备鉴权

命令类型	命令格式	响应
测试命令	AT+IHTTPAUTH=?	+IHTTPAUTH:<ParaTag1>, <ParaTag2>, ...<ParaTagn> OK
查询命令	AT+IHTTPAUTH?	+IHTTPAUTH:<ParaValue1>, <ParaValue2>, ...<ParaValuen> OK
执行命令	AT+IHTTPAUTH=<ParaValue1>, <ParaValue2>, ...<ParaValuen>	OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后可以启动自动鉴权, 返回鉴权结果。

		+IHTTPAUTH:OK 鉴权成功 +IHTTPAUTH:FAIL 鉴权失败
参数说明	<ParaTag1>, <ParaTag2>, ……<ParaTagn>: 鉴权参数1, 2, ……n的名称;	
返回值说明	<ParaValue1>, <ParaValue2>, ……<ParaValuen>: 鉴权参数1, 2, ……n的参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+IHTTPAUTH="productKey001", "deviceName001", "deviceSecret123" OK +IHTTPAUTH:OK	
注意事项	终端设备和通信模组需要根据实际应用场景, 事先约定参数的个数及具体含义; 使用参数鉴权成功后, 建议将参数保存在通信模组的非易失存储器中, 下次可直接使用AT+IHTTPAUTH进行鉴权; 如果该指令仅用于鉴权参数设置, 而不启动鉴权, 无须返回鉴权结果。	

7.5 发送消息并同步接收响应 +IHTTSEND

表格 33 发送消息并同步接收响应

命令类型	命令格式	响应														
测试命令	AT+IHTTSEND=?	+IHTTSEND:"method", "path", "message" OK														
查询命令	AT+IHTTSEND?	+IHTTSEND:<method>, <path>, <message> OK														
执行命令	AT+IHTTSEND=<method>, <path>, <message>	OK 或者 +CME ERROR:<err> 如果输入合法, 首先返回OK, 然后主动上报结果。 +IHTTSEND:<status>, [resp_len], [response]														
参数说明	<method>: 发送消息的方法: <table border="1" data-bbox="296 1375 702 1550"> <tr><td>1:</td><td>GET</td></tr> <tr><td>2:</td><td>POST</td></tr> <tr><td>3:</td><td>PUT</td></tr> <tr><td>4:</td><td>DELETE</td></tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE						
1:	GET															
2:	POST															
3:	PUT															
4:	DELETE															
返回值说明	<path>: 消息对应的path; <message>: 发送的消息体内容; <status>: 发送消息返回状态值: <table border="1" data-bbox="296 1675 1046 1966"> <tr><td>0:</td><td>成功。</td></tr> <tr><td>1:</td><td>token过期, 需要重新执行设备鉴权。</td></tr> <tr><td>2:</td><td>token为空, 需要执行设备鉴权。</td></tr> <tr><td>3:</td><td>token错误, 需要重新执行设备鉴权。</td></tr> <tr><td>4:</td><td>发送数据失败。</td></tr> <tr><td>5:</td><td>发送次数过多。</td></tr> <tr><td>255:</td><td>其它错误。</td></tr> </table> [resp_len]: 接收到的响应消息长度;		0:	成功。	1:	token过期, 需要重新执行设备鉴权。	2:	token为空, 需要执行设备鉴权。	3:	token错误, 需要重新执行设备鉴权。	4:	发送数据失败。	5:	发送次数过多。	255:	其它错误。
0:	成功。															
1:	token过期, 需要重新执行设备鉴权。															
2:	token为空, 需要执行设备鉴权。															
3:	token错误, 需要重新执行设备鉴权。															
4:	发送数据失败。															
5:	发送次数过多。															
255:	其它错误。															

	[response]: 接收到的响应消息体内容; <err>: error代码, 详见《AT command set for User Equipment (UE)》。
示例	AT+IHTTPSEND=2, "/livingroom/light", "{\“action\”:\“turnon\”}" OK +IHTTPSEND:0, 15, "{\“status\”:\“on\”}"
注意事项	查询命令只返回最后一个被发送的request消息。 仅当接收消息的状态为成功时, status=0, 才有回复的消息长度和内容; 其它状态时, 上述两项内容省略。 同步接收等待的超时时间由AT+IHTTPPARA指令的“TIMEOUT”参数指定。

7.6 发送 request 消息 +IHTTPSENDREQ

表格 34 发送 request 消息

命令类型	命令格式	响应								
测试命令	AT+IHTTPSENDREQ=?	+IHTTPSENDREQ:“method”, “path”, “message”, “format”, “fragment_id” OK								
查询命令	AT+IHTTPSENDREQ?	+IHTTPSENDREQ:<method>, <path>, <message>, [format], [fragment_id] OK								
执行命令	AT+IHTTPSENDREQ=<method>, <path>, <message>, [format], [fragment_id]	+IHTTPSENDREQ:<packet_token> OK 或者 +CME ERROR:<err>								
参数说明	<method>: 发送消息的方法: <table border="1" style="margin-left: 20px;"> <tr> <td>1:</td> <td>GET</td> </tr> <tr> <td>2:</td> <td>POST</td> </tr> <tr> <td>3:</td> <td>PUT</td> </tr> <tr> <td>4:</td> <td>DELETE</td> </tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
返回值说明	<path>: 消息对应的path; <message>: 发送的消息体内容; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。 如果该字段为1时, 通信模组需要将HEX字符串转换成二进制数据格式, 再执行发送操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 终端模组发送长消息前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。 如果通信模组收到的数据包里面分包计数器大于0, 它需要继续接收IHTTPSENDREQ命令, 直到收到分包计数器为0, 然后将收到的消息按顺序组包, 再执行发送操作。 <packet_token>: 发送消息时返回的token, 用于和response消息匹配; <err>: error代码, 详见《AT command set for User Equipment (UE)》。									
示例	AT+IHTTPSENDREQ=2, "/livingroom/light", "{\“action\”:\“turnon\”}" +IHTTPSENDREQ:24761									

	OK
注意事项	<p>查询命令只返回最后一个被发送的request消息。</p> <p>如果仅存在一个可选的字段，则该字段必须是format字段；如果存在fragment_id字段，则format字段也必须存在。</p> <p>对于长消息体需要分包的情况，仅最后一包需要回packet_token，其它的分包只需要回OK。</p>

7.7 发送 request 消息（二进制格式） +IHTTPSENBIN

表格 35 发送 request 消息（二进制格式）

命令类型	命令格式	响应								
测试命令	AT+IHTTPSENBIN=?	+IHTTPSENBIN:“method”,“path”,“msg_len” OK “bin_message”								
查询命令	AT+IHTTPSENBIN?	+IHTTPSENBIN:<method>, <path>, <msg_len> OK <bin_message>								
执行命令	AT+IHTTPSENBIN=<method>, <path>, <msg_len>	当msg_len大于0时, OK 当msg_len等于0时, >								
	<bin_message>[+++]	+IHTTPSENBIN:<packet_token> OK 或者 +CME ERROR:<err>								
参数说明	<p><method>: 发送消息的方法:</p> <table border="1" style="margin-left: 20px;"> <tr><td>1:</td><td>GET</td></tr> <tr><td>2:</td><td>POST</td></tr> <tr><td>3:</td><td>PUT</td></tr> <tr><td>4:</td><td>DELETE</td></tr> </table>		1:	GET	2:	POST	3:	PUT	4:	DELETE
1:	GET									
2:	POST									
3:	PUT									
4:	DELETE									
返回值说明	<p><path>: 消息对应的path;</p> <p><msg_len>: 发送的消息长度;</p> <p><bin_message>: 发送的消息体内容，二进制格式;</p> <p>[+++]: 从二进制数据传输模式返回命令模式，仅当msg_len时为0存在;</p> <p><packet_token>: 发送消息时返回的token，用于和response消息匹配;</p> <p><err>: error代码，详见《AT command set for User Equipment (UE)》。</p>									
示例	<pre>AT+IHTTPSENDREQ=2, "/livingroom/light", 5 OK \x11\x22\x33\xAA\xBA +IHTTPSENBIN:24761 OK</pre>									
注意事项	<p>查询命令只返回最后一个被发送的request消息。</p> <p>该指令用于发送二进制格式的消息，共分为两步:</p>									

	<ol style="list-style-type: none"> 1. 终端设备发送指令AT+IHTTPSENBIN, 指定method、path和二进制消息长度; 通信模组回复OK。 2. 终端设备发送指定长度的二进制数据流; 通信模组回复消息token和OK后, 转发该数据包。 如果数据长度为0, 则终端设备先发送二进制数据, 最后发送+++, 以返回命令模式; 通信模组回复消息token和OK后, 转发该数据包。
--	---

7.8 收到 response 消息 +IHTTPCRVRSP

表格 36 收到 response 消息

命令类型	命令格式	响应														
测试命令	AT+IHTTPCRVRSP=?	OK														
查询命令	AT+IHTTPCRVRSP?	+IHTTPCRVRSP:<packet_token>,<status>,[path],[msg_len],[message],[format],[fragment_id] OK														
		主动上报收到的消息: +IHTTPCRVRSP:<packet_token>,<status>,[path],[msg_len],[message],[format],[fragment_id] OK														
参数说明	<p><status>: 返回消息的状态值:</p> <table border="1" style="width: 100%;"> <tr> <td>0:</td> <td>成功。</td> </tr> <tr> <td>1:</td> <td>token过期, 需要重新执行设备鉴权。</td> </tr> <tr> <td>2:</td> <td>token为空, 需要执行设备鉴权。</td> </tr> <tr> <td>3:</td> <td>token错误, 需要重新执行设备鉴权。</td> </tr> <tr> <td>4:</td> <td>发送数据失败。</td> </tr> <tr> <td>5:</td> <td>发送次数过多。</td> </tr> <tr> <td>255:</td> <td>其它错误。</td> </tr> </table>		0:	成功。	1:	token过期, 需要重新执行设备鉴权。	2:	token为空, 需要执行设备鉴权。	3:	token错误, 需要重新执行设备鉴权。	4:	发送数据失败。	5:	发送次数过多。	255:	其它错误。
0:	成功。															
1:	token过期, 需要重新执行设备鉴权。															
2:	token为空, 需要执行设备鉴权。															
3:	token错误, 需要重新执行设备鉴权。															
4:	发送数据失败。															
5:	发送次数过多。															
255:	其它错误。															
返回值说明	<p><packet_token>: 收到消息的token, 与request消息返回的值匹配; [path]: 收到消息的path; [msg_len]: 收到消息体的长度; [message]: 收到消息体的内容; [format]: 消息体格式, 0: 一般字符串; 1: HEX字符串。可选字段, 默认为0。 如果该字段为1时, 终端模组需要将HEX字符串转换成二进制数据格式, 再执行后续操作; [fragment_id]: 长消息体分包计数器。可选字段, 默认为0。 通信模组将收到的长消息发给终端模组前, 需要先设置分包计数器为最大分包数-1; 然后每发送一个新的分包, 计数器-1, 0表示这是最后一个分包。如果重发一个分包, 分包计数器保持不变。 如果终端模组收到的数据包里面分包计数器大于0, 它需要继续接收IHTTPCRVRSP命令, 直到收到分包计数器为0, 然后将收到的消息按顺序组包, 再执行后续操作。</p>															
示例	+IHTTPCRVRSP=24761,0,"/livingroom/light",15,"{\\"status\\":\\"on\\"}" OK															
注意事项	通信模组在收到消息后, 可以主动上报收到的消息; 如果通信模组不支持主动上报模式, 终端设备可以采用查询命令去读取收到的消息。 推荐使用主动上报模式。															

	仅当接收消息的状态为成功时，status=0，才有消息path，长度，消息体内容，消息格式和消息分包计数器；其它状态时，上述五项内容省略。
--	---

7.9 收到 response 消息（二进制格式） +IHTTPCVBIN

表格 37 收到 response 消息

命令类型	命令格式	响应														
测试命令	AT+IHTTPCVBIN=?	OK														
查询命令	AT+IHTTPCVBIN?	+IHTTPCVBIN:<packet_token>,<status>,[path],[msg_len] OK [bin_message]														
		主动上报收到的消息： +IHTTPCVBIN:<packet_token>,<status>,[path],[msg_len] OK [bin_message]														
参数说明	<status>: 返回消息的状态值： <table border="1" style="margin-left: 20px;"> <tr> <td>0:</td> <td>成功。</td> </tr> <tr> <td>1:</td> <td>token过期，需要重新执行设备鉴权。</td> </tr> <tr> <td>2:</td> <td>token为空，需要执行设备鉴权。</td> </tr> <tr> <td>3:</td> <td>token错误，需要重新执行设备鉴权。</td> </tr> <tr> <td>4:</td> <td>发送数据失败。</td> </tr> <tr> <td>5:</td> <td>发送次数过多。</td> </tr> <tr> <td>255:</td> <td>其它错误。</td> </tr> </table>		0:	成功。	1:	token过期，需要重新执行设备鉴权。	2:	token为空，需要执行设备鉴权。	3:	token错误，需要重新执行设备鉴权。	4:	发送数据失败。	5:	发送次数过多。	255:	其它错误。
0:	成功。															
1:	token过期，需要重新执行设备鉴权。															
2:	token为空，需要执行设备鉴权。															
3:	token错误，需要重新执行设备鉴权。															
4:	发送数据失败。															
5:	发送次数过多。															
255:	其它错误。															
返回值说明	<packet_token>: 收到消息的token，与request消息返回的值匹配； [path]: 收到消息的path； [msg_len]: 收到消息体的长度； [bin_message]: 收到消息体的内容，二进制格式。															
示例	+IHTTPCVBIN=24761,0,“/livingroom/light”,5, OK \x11\x22\x33\xAA\xBA															
注意事项	通信模组在收到消息后，可以主动上报收到的消息； 如果通信模组不支持主动上报模式，终端设备可以采用查询命令去读取收到的消息。 推荐使用主动上报模式。 仅当接收消息的状态为成功时，status=0，才有消息path，长度及消息体内容；其它状态时，上述三项内容省略。															

7.10 断开 HTTP 连接 +IHTTPODISCONN

表格 38 断开 HTTP 连接

命令类型	命令格式	响应
测试命令	AT+IHTTPODISCONN=?	OK
执行命令	AT+IHTTPODISCONN	OK 或者 +CME ERROR:<err>
参数说明	<err>: error代码，详见《AT command set for User Equipment (UE)》。	

返回值说明	
示例	AT+IHTTPODISCONN OK
注意事项	

8 OTA 指令

8.1 命令总览

通信模组提供设备端使用的AT指令如下：

表格 39 OTA 的 AT 指令集

命令	描述	实现方式
AT+IOTAPARA	设置参数	可选
AT+IOTAVER	上报版本信息	强制
+IOTANEWVER	接收固件新版本信息	强制
AT+IOTAPGS	上报升级进度	可选
AT+IOTABLOCK	读取新固件的分块数据	强制
AT+IOTABORT	终止固件升级过程	可选

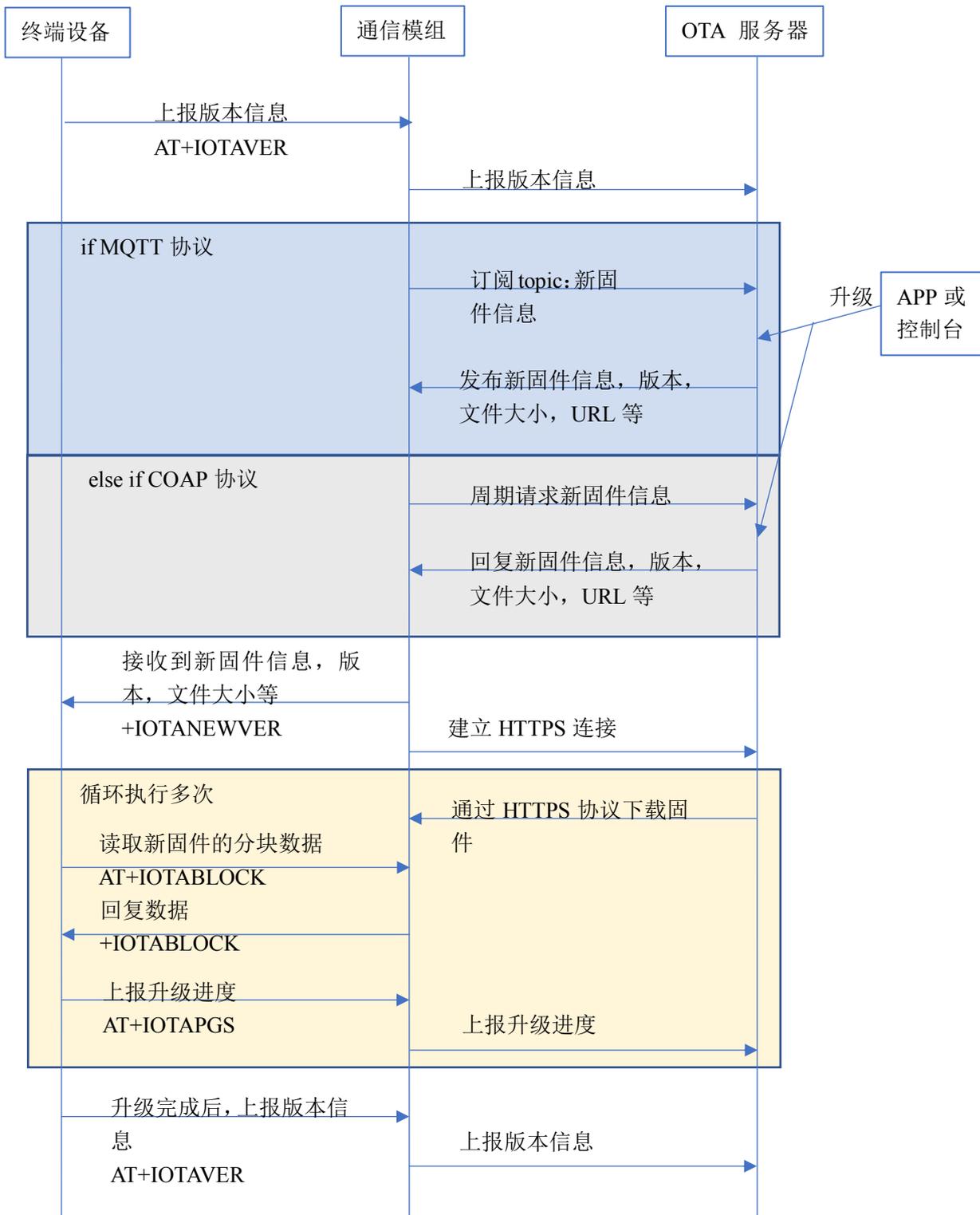


图 2 OTA 流程示意图

8.2 设置参数 +IOTAPARA

表格 40 设置参数

命令类型	命令格式	响应				
测试命令	AT+IOTAPARA=?	+IOTAPARA:“ParaTag”,“ParaValue” OK				
查询命令	AT+IOTAPARA?	+IOTAPARA:<ParaTag>,<ParaValue> OK				
执行命令	AT+IOTAPARA=<ParaTag>,<ParaValue>	OK 或者 +CME ERROR:<err>				
参数说明	<ParaTag>: 参数名称, 包含以下参数项;					
返回值说明	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">“DURATION”</td> <td>COAP协议下, 查询新固件信息的周期, 单位秒。</td> </tr> <tr> <td>“BLOCKSIZE”</td> <td>+IOTABLOCK命令支持的二进制块数据的最大容量, 单位字节。</td> </tr> </table> <ParaValue>: 参数值; <err>: error代码, 详见《AT command set for User Equipment (UE)》。		“DURATION”	COAP协议下, 查询新固件信息的周期, 单位秒。	“BLOCKSIZE”	+IOTABLOCK命令支持的二进制块数据的最大容量, 单位字节。
“DURATION”	COAP协议下, 查询新固件信息的周期, 单位秒。					
“BLOCKSIZE”	+IOTABLOCK命令支持的二进制块数据的最大容量, 单位字节。					
示例	AT+IOTAPARA=“DURATION”, 86400 OK AT+IOTAPARA=“DURATION”, 86400, “BLOCKSIZE”, 512 OK					
注意事项	查询命令可以查询所有参数的名称和相应的值; 执行命令可以设置一个或多个参数。					

8.3 上报版本信息 +IOTAVER

表格 41 上报版本信息

命令类型	命令格式	响应
测试命令	AT+IOTAVER=?	+IOTAVER:“version” OK
查询命令	AT+IOTAVER?	+IOTAVER:<version> OK
执行命令	AT+IOTAVER=<version>	OK 或者 +CME ERROR:<err>
参数说明	<version>: 版本信息, 可包含硬件版本和软件版本;	
返回值说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+IOTAVER=“hw_1.0.0 fw_1.2.0” OK	
注意事项		

8.4 接收固件新版本信息 +IOTANEWVER

表格 42 接收固件新版本信息

命令类型	命令格式	响应
测试命令	AT+IOTANEWVER=?	OK

查询命令	AT+IOTANEWVER?	+IOTANEWVER:<version>,<img_size>,<checksum> OK
		主动上报收到的消息: +IOTANEWVER:<version>,<img_size>,<checksum> OK
参数说明	<version>: 新固件的版本信息; <img_size>: 固件文件大小, 单位字节;	
返回值说明	<checksum>: 固件文件内容的校验值, 建议采用MD5校验。	
示例	+IOTANEWVER="hw_1.0.0 fw_1.2.0",432945,"93230c3bde425a9d7984a594ac55eale" OK	
注意事项	通信模组在收到消息后, 可以主动上报收到的消息; 如果通信模组不支持主动上报模式, 终端设备可以采用查询命令去读取收到的消息。 推荐使用主动上报模式。	

8.5 上报升级进度 +IOTAPGS

表格 43 上报升级进度

命令类型	命令格式	响应
测试命令	AT+IOTAPGS=?	+IOTAPGS:"progress","desc" OK
执行命令	AT+IOTAPGS=<progress>,<desc>	OK 或者 +CME ERROR:<err>
参数说明	<progress>: 升级进度信息:	
返回值说明	<div style="border: 1px solid black; padding: 5px;"> 0~100: 升级进度百分比。 -1: 升级失败。 -2: 下载失败。 -3: 校验失败。 -4: 烧写失败。 </div> <desc>: 当前进度的描述信息; <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+IOTAPGS=80 OK AT+IOTAPGS=-2,"FLASH MEMORY IS NOT ENOUGH" OK	
注意事项	<desc>内容由终端设备自定义, 若无, 可采用空字符串""。	

8.6 读取块数据 +IOTABLOCK

表格 44 接收块数据

命令类型	命令格式	响应
测试命令	AT+IOTABLOCK=?	OK
执行命令	AT+IOTABLOCK=<type>	+IOTABLOCK: <message> OK

参数说明	<type>: 接收新的消息或者要求重发上一条消息: 0: 接收新的消息。 1: 请求重发上一条消息。
返回值说明	<message>: 收到消息体的内容, 十六进制字符串格式数据。
示例	AT+IOTABLOCK=0 +IOTABLOCK= "ABCEEF0123456789AB" OK AT+IOTABLOCK=0 +IOTABLOCK= "1122334455667788CC" +IOTABLOCK= "AABBCCDDEEFF112233" OK AT+IOTABLOCK=1 +IOTABLOCK= "1122334455667788CC" +IOTABLOCK= "AABBCCDDEEFF112233" OK
注意事项	为了节约通信带宽, 允许通信模组收到一条查询指令后, 回复多行数据, 然后回复OK作为该指令的结束。如果终端模组要求重传, 则通信模组会将回复OK前的多行数据都重传。

8.7 终止固件升级 +IOTABORT

表格 45 终止固件升级

命令类型	命令格式	响应
测试命令	AT+IOTABORT=?	OK
执行命令	AT+IOTABORT	OK 或者 +CME ERROR:<err>
参数说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。	
返回值说明		
示例	AT+IOTABORT OK	
注意事项	终止当前固件升级过程。 当APP或者控制台再次发起固件升级, 终端设备必须支持重新开始固件升级。	

9 其它指令

9.1 命令总览

通信模组提供设备端使用的AT指令如下:

表格 46 其它 AT 指令集

命令	描述	实现方式
AT+ILOGLVL	设置日志等级	可选
AT+IREBOOT	重启通信模组	可选
AT+IDEFAULT	通信模组恢复出厂设置	可选

9.2 设置日志等级 +ILOGLVL

表格 47 设置日志等级

命令类型	命令格式	响应
测试命令	AT+ILOGLVL=?	+ILOGLVL:“level” OK
查询命令	AT+ILOGLVL?	+ILOGLVL:<level> OK
执行命令	AT+ILOGLVL=<level>	OK 或者 +CME ERROR:<err>
参数说明	<level>: 日志等级;	
返回值说明	0: 禁止日志信息。 1~5: 使能日志信息, 数字越大, 日志信息越详细。 <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+ILOGLVL=1 OK	
注意事项		

9.3 重启模组 +IREBOOT

表格 48 重启模组

命令类型	命令格式	响应
测试命令	AT+IREBOOT=?	OK
执行命令	AT+IREBOOT=<mode>	OK 或者 +CME ERROR:<err>
参数说明	<mode>: 重启模式;	
返回值说明	0: 立即重启通信模组。 1: 等待通信模组内当前正在发送的帧完成后再重启。 <err>: error代码, 详见《AT command set for User Equipment (UE)》。	
示例	AT+IREBOOT=1 OK	
注意事项	通信模组收到该指令后, 回复OK后, 重启通信模组。重启完成之前, 不再接收任何后续的AT指令。	

9.4 恢复出厂设置 +IDEFAULT

表格 49 恢复出厂设置

命令类型	命令格式	响应
测试命令	AT+IDEFAULT=?	OK
执行命令	AT+IDEFAULT	OK 或者 +CME ERROR:<err>
参数说明	<err>: error代码, 详见《AT command set for User Equipment (UE)》。	
返回值说明		
示例	AT+IDEFAULT	

	OK
注意事项	通信模组收到该指令后，擦除所有保存在非易失性存储器中的内容，恢复至出厂设置的参数；然后回复OK后，重启通信模组。重启完成之前，不再接收任何后续的AT指令。

9.5 标准指令

建议通信模组提供以下3GPP规定的标准AT指令：

表格 50 标准 AT 指令集

命令	描述	实现方式
AT+CGMI	读取模组厂家信息	可选
AT+CGMM	读取模组型号	可选
AT+CGMR	读取模组固件版本信息	可选

3GPP标准文档参见以下链接：

http://www.etsi.org/deliver/etsi_ts/127000_127099/127007/14.04.00_60/ts_127007v140400p.pdf。