

M5310 AT 使用流程示例

NB-IoT 系列

版 本：V2.1

日 期：2017-09-12



中国移动
China Mobile

中移物联网有限公司

iot.10086.cn



重要声明

版权声明

本文档中的任何内容受《中华人民共和国著作权法》的保护，版权所有 © 2017, 中移物联网有限公司，保留所有权利，但注明引用其他方的内容除外。

商标声明

中移物联网有限公司和中移物联网有限公司的产品是中移物联网有限公司专有。在提及及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。

不作保证声明

中移物联网有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。

保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

关于文档

修订记录

版本	日期	作者	描述
1.0	2017.6.10	林紫微	首次创建
1.0	2017.6.23	林紫微	修改模块登录 OneNET 的相关描述
2.0	2017.8.28	周俨	更新 OneNET 流程
2.1	2017.9.12	周俨	更新 OneNET 管理流程中数据长度限制
2.2	2017.9.29	李彦论	添加长数据发送，修改 OneNET 设备创建流程



注：文档中涉及的相关指令参数意义详见中移物联网公司《M5310 AT Command Set》文档

目录

关于文档	2
1. 上电初始化	4
2. UDP 数据收发	5
2.1. 创建 UDP SOCKET	5
2.2. 发送 UDP 数据	5
2.3. 接收 UDP 数据	5
2.4. 关闭 UDP	6
3. ONENET 数据收发	7
3.1. ONENET 平台端创建设备	7
3.2. 设备注册流程	13
3.3. 数据上报流程	14
3.4. 设备管理流程	15
3.5. 设备注销流程	18
3.6. 长数据收发	18
附录 A IPSO 资源模型示例	21

1. 上电初始化

注意：每个 AT 命令之间应该留有一定时间间隔

M5310

OK //开机启动信息

AT //开机之后循环发送 AT 直到返回 OK，证明模块初始化正常

OK

AT+COPS=1,2,"46000" //设置手动注册移动运营商 MNC

OK

AT+NEARFCN=0,3555 //锁定频点为 3555，锁频可以有效减小搜网时间，但是频点设置错误会导致搜网失败，建议通常情况下不要设置锁频

OK

AT+CSCON=1 //打开信号提示自动上报

OK

AT+CEREG=1 //打开注册信息自动上报

OK

+CSCON:1 //自动上报的网络信号提示——已连接

+CEREG:1,19E6,94,7 //自动上报的网络注册信息——1-本地网络已注册入网，5-漫游已注册，其它情况为注册异常，详细请参考 AT 命令手册

//如果未使能自动上报，则用户需要使用 AT+CEREG? 查询注册状态

AT+CGDCONT?

+CGDCONT:0,"IP","nbiot.MNC002.MCC460.GPRS",,0,0

//查询当前 APN，次步骤可省略

OK

注：需要确认入网状态为已注册才能进行后续数据收发操作，如果不使用自动上报功能，可使用 AT+CEREG? 命令主动查询当前注册状态直到变为已注册，目前测试开机注册时间范围为 20s-120s

2. UDP 数据收发

2.1. 创建 UDP Socket

AT+NSOCR=<type>,<protocol>,<listen prt>[,<receive control>]

例如:

```
AT+NSOCR="DGRAM",17,2334,1    //创建本地UDP监听端口，开启数据到达自动上报
0                                //创建成功返回socket编号，数值0-6，最多监听7个端口
OK                               //创建UDP成功
```

2.2. 发送 UDP 数据

向目的UDP地址发送数据可使用AT+NSOST或AT+NSOSTF命令，如下，

AT+NSOST=<socket>,<remote_addr>,<remote_port>,<length>,<data>

AT+NSOSTF=<socket>,<remote_addr>,<remote_port>,<flag>,<length>,<data>

例如:

```
AT+NSOST=0,183.230.40.150,36000,10,30313233343536373839
0,10                                //第0号Socket成功发送10 Bytes UDP数据
OK
+NSONMI:0,30                        //第0号Socket接收到30 Bytes数据
```

2.3. 接收 UDP 数据

当接收到UDP数据时，可以使用AT+NSORF读取，当读取长度大于实际接收长度时，返回缓冲区实际接收数据长度

```
AT+NSORF=<socket>,<req_length>
+NSONMI:0,30                        //提示第 0 号 Socket 接收到 30 Bytes 数据
AT+NSORF=0,30                        //读取接收到的 30 Bytes 数据，
0,183.230.40.150,36000,30,5B3131372E3136392E31353A323137395D30313233343536373839,0
//读取到183.230.40.150:36000发过来的30 Bytes数据
```

注：本示例的测试服务器为中移物联网公司内部测试服务器，会自动回复“[远程IP:端口]接收数据”，只做测试用途，不保证服务器功能。

2.4. 关闭 UDP

AT+NSOCL=<socket>

//<socket>为 2.1 创建时系统分配的 id

OK



3. OneNET 数据收发

3.1. OneNET 平台端创建设备

访问OneNET门户网站：<http://open.iot.10086.cn/>，其它省市自建平台请访问对应门户，方法类似，目前仅支持API方式创建NBCOAP产品及设备

3.1.1 获取用户ID及secret用于创建NB通讯产品

用户ID可在已创建产品中获取，如没有，可先任意创建一个产品，例如



The screenshot shows the '产品信息' (Product Information) section of the OneNET platform. It includes fields for '产品名称' (Product Name), '产品行业' (Product Industry), '产品类别' (Product Category), '当前产品进度' (Current Product Progress), '产品简介' (Product Introduction), '技术参数' (Technical Parameters), '操作系统' (Operating System), '网络运营商' (Network Operator), '联网方式' (Networking Method), '设备接入方式' (Device Access Method), and '设备接入协议' (Device Access Protocol). The form is partially filled out, with 'Linux' selected for the operating system, '移动' (China Mobile) for the network operator, 'wifi' for the networking method, and '公开协议' (Public Protocol) for the device access method. The '设备接入协议' dropdown is set to 'HTTP', and a tooltip for 'HTTP 协议基本功能介绍' (HTTP Protocol Basic Function Introduction) is visible, listing '1、上报传感器数据点' (1. Upload sensor data points).

产品信息

产品名称：
最多输入16个字符

产品行业：
智能家居

产品类别：
请选择

当前产品进度：
☒ 有创意想法 ☐ 有产品原型 ☐ 有工程样机
☐ 已量产

产品简介：

技术参数

操作系统：
☒ Linux ☐ Android ☐ VxWorks
☐ µC/OS ☐ 无 ☐ 其他

网络运营商：
☒ 移动 ☐ 电信 ☐ 联通
☐ 其他

联网方式：
☒ wifi ☐ 移动蜂窝网络

设备接入方式：
☒ 公开协议 ☐ 私有协议(RGMP)

设备接入协议：
HTTP

HTTP 协议基本功能介绍：
1、上报传感器数据点

确定 取消

至此，可在开发者中心->公开产品主页面看到有HTTP产品
如果已有产品，可忽略上述步骤，直接通过下图中页面获取用户ID (user_id)



secret的获取在开发者中心->私有协议产品页面



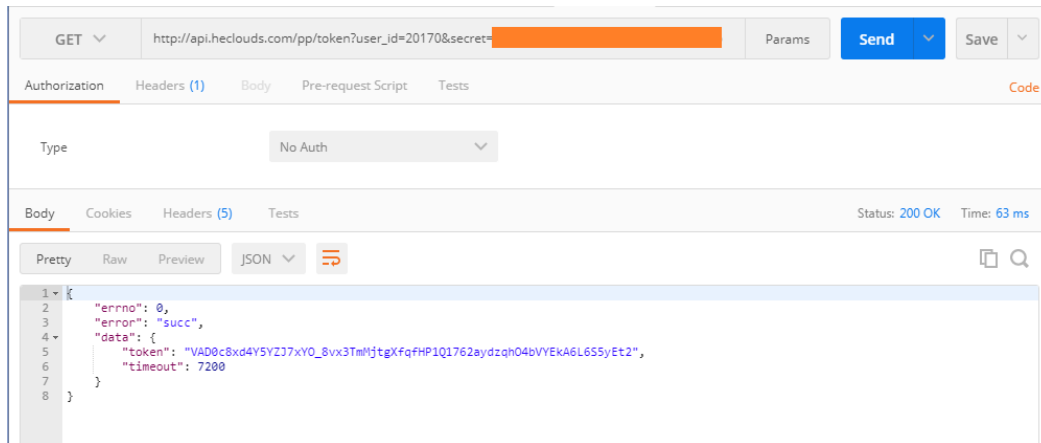
3.1.2 获取通信token

API格式如下表，user_id与secret为上述3.1.1中获取

HTTP 方法	GET
URL	http://<API_ADDRESS>/pp/token
URL 参数	<div>user_id=123 //用户 id</div> <div>secret=xx //用户密钥</div>
成功返回	<pre>{ "errno": 0, "error": "succ", "data": { "token": "xxxx", //用于后面 API 的 token 调用 "timeout": 7200 //单位: 秒, token 过期时间, 过期后要重新获取 token } }</pre>

注：上述API每天最多执行2000次

例如使用Postman工具发送上述HTTP请求，获取token字段如下图所示



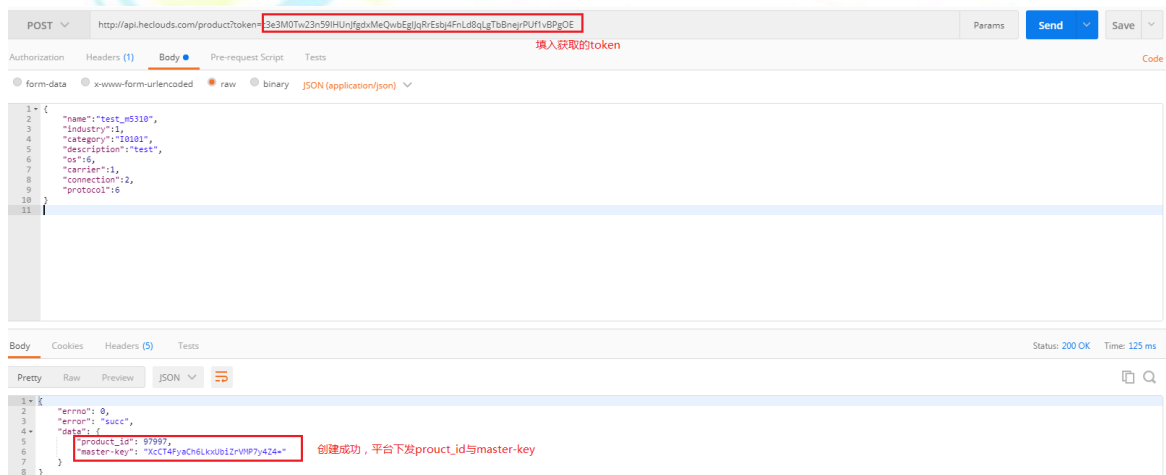
3.1.3 创建NBCOAP协议产品

API数据格式参考下表，token字段为上述3.1.2已获取字段

HTTP 方法	POST
URL	http://<API_ADDRESS>/product
URL 参数	token = xxxxxxxxx //填入 3.1.2 获取的 token
请求 body	<pre>{ "name": "test", //产品名字, String, 必填; "industry": 1, //产品行业, int, 必填, 详情看说明; "category": "xxx", //产品类型, string, 必填, 详情看说明; "description": "test", //简介, string, 必填, ; "os": 1, //操作系统, int, 必填, 详情看说明; "carrier": 1, //运营商, int, 必填, 详情看说明; "connection": 1, //联网方式, int, 必填, 详情看说明; "protocol": 0 //接入协议, int, 必填, 详情看说明; NBCoAP 协议填 6 }</pre>
成功返回	<pre>{ "errno": 0, "error": "succ", "data": { "product_id": 344, //所创建的产品 id "master-key": "xxxxx" } }</pre>
说明	<p>1、os, 操作系统, 其中 1: 'Linux', 2: 'Android', 3: 'VxWorks', 4: 'µC/OS', 5: '无', 6: '其他';</p> <p>2、carrier, 网络运营商, 其中</p>

	<p>1: '移动', 2: '电信', 3: '联通', 4 : '其他';</p> <p>3、connection, 联网方式, 其中 1: 'wifi', 2: '移动蜂窝网络';</p> <p>4、protocol: 设备接入方式, 其中 1 : 'HTTP', 2: 'EDP', 3: 'MQTT', 4 : 'Modbus', 5: 'JT/T808', 6: 'NBCoAP';</p> <p>5、industry, 行业, 其中 1: '智能家居', 2: '车载设备', 3: '可穿戴设备', 4: '医疗保健', 5: '智能玩具', 6: '新能源', 7: '运动监控', 8: '智能教育', 9: '环境监控', 10: '办公设备', 11: '其他';</p> <p>6、category, 类别, 其中编码对应为 A0114: '大家电', A0218: '生活电器', B0103: '办公外设产品', B0208: '办公网络产品', C0104: '穿戴钟表', D0110: '母婴童床童车', D0211: '母婴童装童鞋', E0112: '汽车车载设备', E0210: '汽车安全自驾', F0105: '智能玩具遥控/电动', F0203: '智能玩具娃娃玩具', G0110: '数码摄影摄像', G0201: 'MP3/MP4', G0202: '电视盒子', G0203: '耳机/耳麦', G0204: '音响音箱', H0108: '骑行运动', H0205: '垂钓用品', H0305: '背包', H0306: '户外照明', I0101: '其他'</p>
--	---

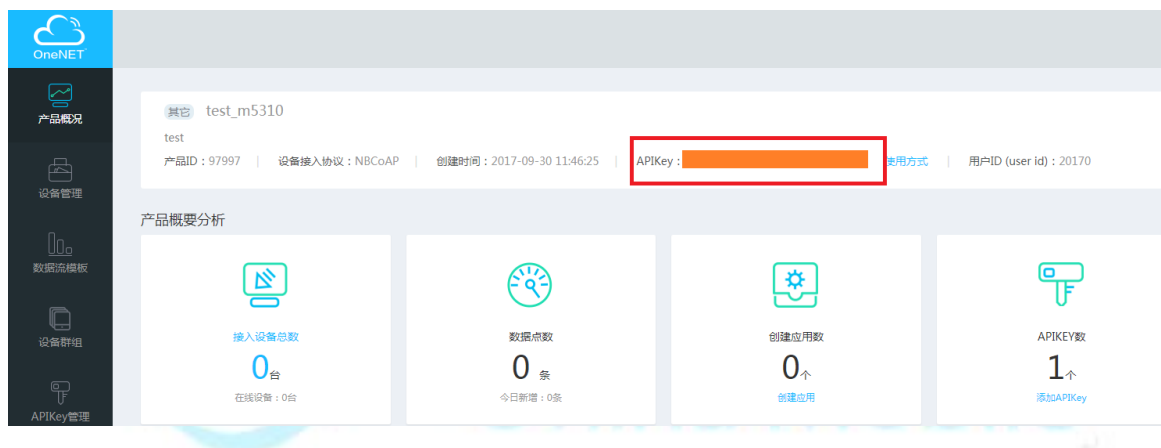
例如使用 postman 工具下发 api 请求, 结果如下



之后可在开发者中心-公开协议产品中看到刚才创建的产品



- 3.1.4 目前仅支持平台API方式创建NB-CoAP设备，在特定产品下创建设备，创建流程需要使用产品apiKey，查看方式如下
开发者中心-产品选择-产品概况找到产品APIkey（MasterKey）

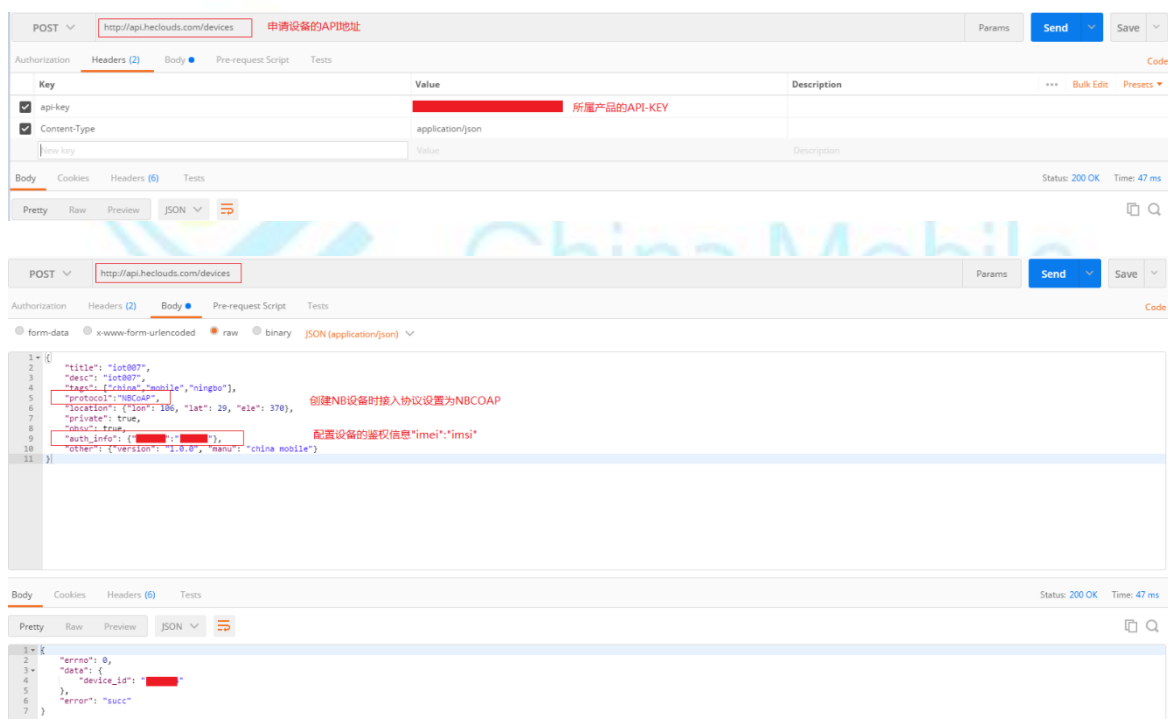


创建设备的API方法如下表

HTTP 方法	POST
URL	http://api.heclouds.com/devices
HTTP 头部	api-key:xxxx-ffff-zzzzz, 必须为 MasterKey
HTTP 内容	<pre>{ "title": "mydevice", //设备名 "desc": "some description", //设备描述（可选） "tags": ["china","mobile"], //设备标签（可选，可为一个或多个） "protocol": "NB-CoAP", //接入协议 "location": {"lon": 106, "lat": 29, "ele": 370}, //设备位置{"纬度", "精度", "高度"}（可选） "private": true false, //设备私密性（可选，默认为ture） "auth_info": {"imei": "imsi"}, //设备 IMEI 及 SIM 卡 IMSI, IMEI 必须保证唯一性，不然会导致创建设备失败 }</pre>

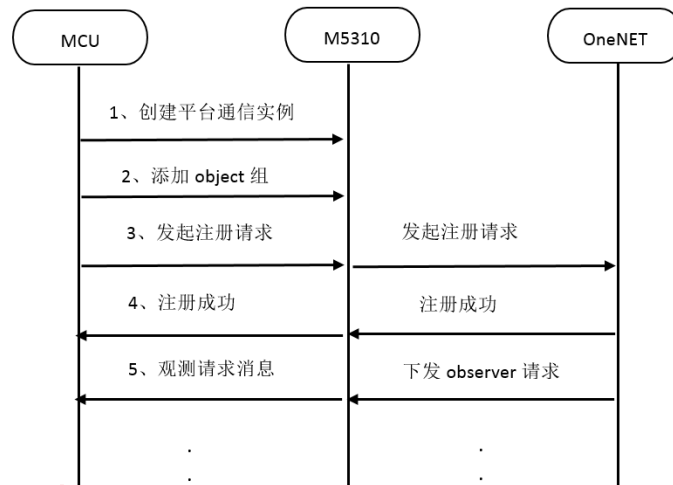
	<pre> "obsv":true false, //是否订阅设备资源，默认为 true "other": {"version": "1.0.0", "manu": "china mobile"} //其他信息（可选，JSON 格式，可自定义） } </pre>
HTTP 响应响应消息内容	<pre> { "errno": 0, "error": "succ", "data": { "device_id": "233444" //平台分配的唯一设备 ID } } </pre>
说明	<p>1、 other 字段如果有可填写，如果也不影响设备的创建；</p> <p>2、 响应消息中 errno 表示错误码，error 表示错误原因，如果创建设备失败，则不会返回 device_id 字段；</p> <p>4、 NBCoAP 设备 auth_info 中 imei 和 imsi 均为小于 32 位字符</p>

使用 Postman 工具向 api.heclouds.com 申请创建 NBCoAP 设备如下：



创建设备所使用的鉴权信息 imei 及 imsi 需要记录并在终端登录时使用，返回的 device_id 为平台端创建的设备接入标识，如果需要查询设备信息，需要提供设备 id

3.2. 设备注册流程



3.2.1 创建一个通信实例设备，该指令目前只允许拥有 1 个实例

AT+MIPLCONF=49,0000B80B00003A161900636F61703A2F2F3138332E3233302E3430
2E34303A353638330C0074657374323B303030303031,1,1

0
OK

配置格式可使用我司提供的生成工具转换，具体请参考 AT 手册中+MIPLCONF 说明，如需程序自动生成配置数据，请联系 FAE 获取源码；目前仅支持一个实例，需要销毁创建的实例才能再次创建

3.2.2 向通信套件添加 object 组

AT+MIPLADDOBJ=0,3200,0

OK

AT+MIPLADDOBJ=0,3201,0

OK

LwM2M 规范定义了每个对象对应客户端的某个特定功能实体 object，instance 代表着这个 object 的不同实例。通过 objectid 和 instanceid 可以确认到一个指定的 object 实例；而每个 object 下可以有多个 resource 属性数据。例如一个设备有芯片温度传感器与外界温度传感器的值须上报，温度传感器便是一种 object 对象，而具体到某个温度传感器则需要 instanceid 来区分；某个温度传感器所对应的单位、温度等数值可视为其 resource 属性，具体的编码规范可参照 IPSO 规范或 OMA 模型规范

<http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html>。

在注册前添加的 objects 在注册时会通过注册信息发给 OneNET 平台，注册成功后平台会对添加的所有 object 下发 observer 请求；当前版本传输的 object 皆应在注册前添加。

3.2.3 发起注册请求

由于注册结果为异步事件，注册是否成功应以步骤 3.2.4 为准

AT+MIPLOPEN=0,15

OK

3.2.4 注册结果上报

上报的注册请求服务器收到后会返回本次注册结果，如在超时时间还未收到服务器回复，则也会注册失败，如下

+MIPLOPEN:0,1

//注册成功

+MIPLOPEN:0,0

//注册失败或超时

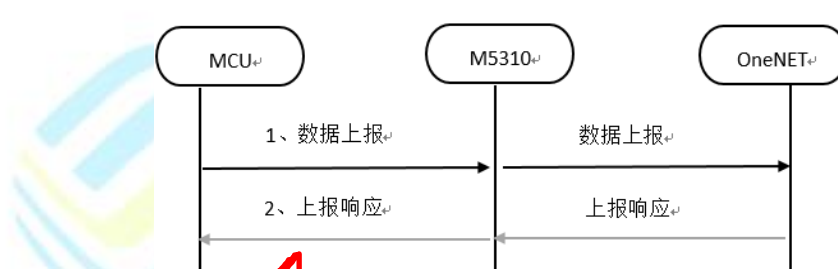
3.2.5 Observe 请求消息

该消息为平台下发，模块会自动处理 Observe 请求，用户可不必处理

+MIPLOBERVE:0,2657,3200,0,-1,0

+MIPLOBERVE:0,2658,3202,0,-1,0

3.3. 数据上报流程



3.3.1 用户上报数据

AT+MIPLNOTIFY=0,3200,0,5500,5,"1",1

//flag 为 1 时添加数据并上报，<ackid>缺省平台侧将无 ACK

OK

3.3.2 平台响应上报

AT+MIPLNOTIFY=0,3200,0,5500,5,"1",0

//flag 为 0 时只添加数据不上传数据

OK

AT+MIPLNOTIFY=0,3200,0,5750,1,"juidl",1,278

//flag 为 1 时添加数据并触发上报

OK

+MIPLNOTIFY:0,278

//设置<ackid>情况下收到平台的上传信息回复

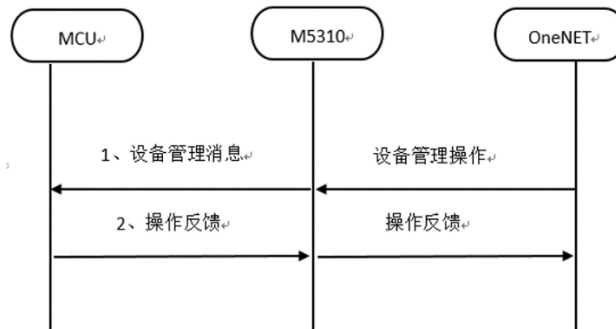
注：

<ackid>被设置为大于 1 的情况下平台侧有响应，如果缺省上传数据平台侧不会有 ACK 回复

<ackid>仅在<flag>参数为 1 时设置有效

每次 notify 上报数据长度应小于 500Bytes

3.4. 设备管理流程



设备管理交互流程

设备管理目前版本提供 3 种操作：write、read、execute

MCU 应在执行平台下发的规定操作后数秒内（建议 3s）上报对应操作结果，否则传输信息可能丢失

3.4.1 read 操作

1)、读取指定 object 下指定 instance 下的所有 resource 请求

```

+MIPLREAD:0,289,3200,0,-1 //收到平台侧下发的读取 object,instance 下的所有 resource 请求
AT+MIPLREAD=0,289,3200,0,5500,5,"1",0 //终端读取某个 resource 值，但不触发上报
OK
AT+MIPLREAD=0,289,3200,0,5750,1,"juidl",1 //读取 resource 值，并触发上报数据
OK
    
```

控制平台下发读操作的 API 报文示例：

```

GET http://api.heclouds.com/nbiot?imei=test2&obj_id=3200 &obj_inst_id=0 HTTP/1.1
Host: api.heclouds.com
api-key: //<APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY
Content-Length: 0
    
```

2)、读取指定 object 下所有 instance 下的所有 resource 请求

```

+MIPLREAD:0,289,3200,-1,-1 //收到平台侧下发的读取 object 下所有 instance 下所有 resource 请求
AT+MIPLREAD=0,289,3200,1,5500,5,"1",0 //终端读取某个 resource 值，但不触发上报
OK
AT+MIPLREAD=0,289,3200,0,5500,5,"1",0 //终端读取某个 resource 值，但不触发上报
OK
AT+MIPLREAD=0,289,3200,0,5750,1,"juidl",1 //读取 resource 值，并触发上报数据
OK
    
```

控制平台下发读操作的 API 报文示例：

```

GET http://api.heclouds.com/nbiot?imei=test2&obj_id=3200 HTTP/1.1
Host: api.heclouds.com
api-key: //<APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY
    
```

Content-Length: 0

3)、读取指定 object 下指定 instance 下的指定 resource 请求

+MIPLREAD:0,289,3200,0,5500 //收到平台侧下发的读取 object 下指定 instance 下指定 resource 请求

AT+MIPLREAD=0,289,3200,1,5500,5,"1",1 //终端读取指定值，上报

OK

控制平台下发读操作的 API 报文示例:

GET http://api.heclouds.com/nbiot?imei=test2&obj_id=3200&res_id=5500 HTTP/1.1

Host: api.heclouds.com

api-key: <APK-KEY>

//<APK-KEY>参数请替换成产品或者设备 API-KEY

Content-Length: 0

资源读取 API 报文协议如下:

操作	Read (资源读取)
lwm2m-URI	address /{Object ID}/{Object Instance ID}/{Resource ID}
参数说明	{ Object ID }: 必选, 如设备上的传感器类型; { Object Instance ID }: 可选, 该类型传感器的编号; { Resource ID }: 可选, 该传感器的某种类型的数据, 如温度的当前值, 最大值等
CoAP-Method	GET
CoAP-Option	Option 1: Uri-Path (11): {Object ID} Option 2: Uri-Path (11): {Object Instance ID} Option 3: Uri-Path (11): {Resource ID} Option 4: Accept (17): 第 1 节中的 Content Format 表中的类型, 如 application/vnd.oma.lwm2m+tlv 这个参数是指明读取数据的格式。
CoAP-payload	
Success	2.05 Content
Failure	4.00 Bad Request, 4.01 Unauthorized, 4.04 Not Found, 4.05 Method Not Allowed

注: 每次 read 操作后模块响应上报 COAP 报文长度应小于 512Bytes

3.4.2 write 操作

+MIPLWRITE:0,290,3200,0,5500,0 //平台下发向 3200,0,5500 写入 0

AT+MIPLWRITE=0,290,1 //终端向平台回复写入操作成功

OK

AT+MIPLWRITE=0,290,0 //终端向平台回复写入操作失败

OK

控制平台下发写操作的 API 报文示例:

POST http://api.heclouds.com/nbiot?imei=test2&obj_id=3200&obj_inst_id=0&mode=2 HTTP/1.1

api-key: <APK-KEY>

//<APK-KEY>参数请替换成产品或者设备 API-KEY

Host: api.heclouds.com

Content-Length: 47

//数值根据 HTTP 报文实际 body 大小修改

```
{
  "data": [{
    "val": 0,
    "res_id": 5500}],
}
```

资源写入 API 报文协议如下:

操作	Write (资源写入)
lwm2m-URI	address /{Object ID}/{Object Instance ID}/{Resource ID}/{NewValue}
参数说明	{ Object ID }: 必选, 如设备上的传感器类型; { Object Instance ID }: 必选, 该类型传感器的编号;

	{Resource ID}: 必选, 该传感器的某种类型的数据, 如温度的当前值, 最大值等; {NewValue}: 必选, 写入的资源属性值。
CoAP-Method	PUT/POST
CoAP-Option	Option 1: Uri-Path (11): {Object ID} Option 2: Uri-Path (11): {Object Instance ID} Option 3: Uri-Path (11): {Resource ID} Option 4: Content-Format (12): 格式为 Content Format 表中的几种格式, 如 application/vnd.oma.lwm2m+tlv
CoAP-payload	{NewValue}, 数据格式为 Option4 中指定的数据格式
Success	2.04 Changed
Failure	4.00 Bad Request, 4.01 Unauthorized, 4.04 Not Found, 4.05 Method Not Allowed

注: 每次 write 操作下发 COAP 报文长度应小于 512Bytes, 故下发数据段长度最大 480Bytes

3.4.3 execute 操作

```
+MIPLEXECUTE=0,291,3200,0,5500,ping //平台下发的执行指令, 内容为 ping
AT+MIPLEXECUTE=0,291,1 //终端向平台上报本次执行操作成功
OK
AT+MIPLEXECUTE=0,291,0 //终端向平台上报本次执行操作失败
OK
```

控制平台下发 EXECUTE 消息的 API 报文示例:

```
POST http://api.heclouds.com/nbiot/execute?imei=test2&obj_id=3200&obj_inst_id=0&res_id=5600 HTTP/1.1
api-key: <APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY
Host: api.heclouds.com
Content-Length: 21 //数值根据 HTTP 报文实际 body 大小修改

{
  "args": "ping7"
}
```

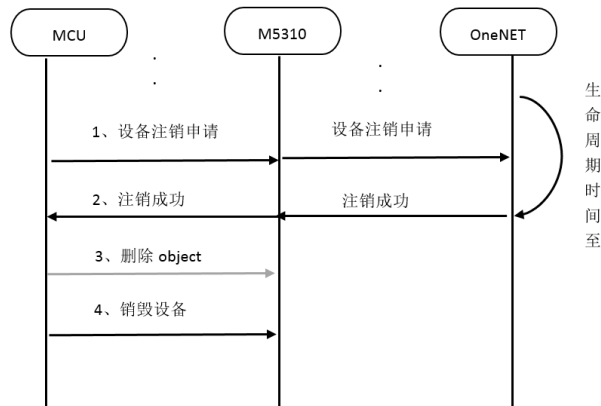
资源执行 API 报文协议如下:

操作	Execute (资源执行)
lwm2m-URI	address/{Object ID}/{Object Instance ID}/{Resource ID}/{Arguments}
参数说明	{Object ID}: 必选, 如设备上的传感器类型; {Object Instance ID}: 必选, 该类型传感器的编号; {Resource ID}: 必选, 该传感器的某种类型的数据, 如温度的当前值, 最大值等; {Arguments}: 可选, 待执行的命令, 如重启, 关机。
CoAP-Method	POST
CoAP-Option	Option 1: Uri-Path (11): {Object ID} Option 2: Uri-Path (11): {Object Instance ID} Option 3: Uri-Path (11): {Resource ID}
CoAP-payload	{Arguments}, 格式为 Content Format 表中的 text/plain 格式
Success	2.05 Changed
Failure	4.00 Bad Request, 4.01 Unauthorized, 4.04 Not Found, 4.05 Method Not Allowed

注:

每次 execute 操作下发 COAP 报文长度应小于 512Bytes, 故下发数据段长度最大 480Bytes

3.5. 设备注销流程



AT+MIPLCLOSE=0 //设备向平台发起注销请求

OK

+ MIPLCLOSE: 0,1 //注销成功，该次登录平台的 resource 信息都将被清理

AT+MIPLDELOBJ=0,3200,0 //删除建立的 3200 object

OK

AT+MIPLDELOBJ=0,3202,0 //删除建立的 3202 object

OK

AT+MIPLDEL=0 //销毁建立的 OneNET 平台通信实例

OK

注：

用户亦可在向平台发送注销请求后执行销毁实例，此时模组会自动删除所有 object
 用户亦可直接执行销毁实例，模组将会在向平台提出注销请求后删除 object 再销毁 Object 以及通信实例。

3.6. 长数据收发

分包大小设置：配置工具中生成配置参数时加上下列参数，参考 AT 指令集中+MIPLCONF 命令说明

-u 设置 BLOCK1 的大小，对 write 和 execute 有效，范围 0-6，大小 $2^{(4+n)}$

-g 设置 BLOCK2 的大小，对 read 有效，范围 0-6，大小 $2^{(4+n)}$

-x 设置主动分包门限，范围 0-2，大小 $2^{(8+n)}$

3.6.1 模组长数据接收（write/execute 接口）

1) write 指令 API 报文示例

POST http://api.heclouds.com/nbiot?imei=test2&obj_id=3200&obj_inst_id=0&mode=2 HTTP/1.1
 api-key: <APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY

```
{
  "data":{
    "val":"1111111111111111.....", //val 长度可变（大于 1K）
    "res_id":5750},
}
```

[illegible]

AT+MIPLWRITE=0,16821,1	//终端向平台回复写入操作成功
OK	
AT+MIPLWRITE=0,16821,0	//终端向平台回复写入操作失败
OK	

```
POST http://api.heclouds.com/nbiot/execute?imei=test2&obj_id=3200&obj_inst_id=0&res_id=5600 HTTP/1.1
api-key: <APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY
Host: api.heclouds.com
Content-Length: 21 //数值根据 HTTP 报文实际 body 大小修改
```

```
{
    "args": "666666666666....." //args 长度可继续变大 (>1K)
}
```

[illegible]

19

对 API 控制台回复，数据发送完毕后，只需要对最后一包的包序号进行回复，如上图所示回复如下

```
AT+MIPLEXECUTE=0,53239,1 //终端向平台上报本次执行操作成功
OK
AT+MIPLEXECUTE=0,53239,0 //终端向平台上报本次执行操作失败
OK
```

3.6.2 模组长数据发送（read 接口）

当使用 Read 接口时，超过设置长度会触发模组执行内部 Block 分包发送，整个流程模块内部自动完成。

Read 指令 API 报文示例：

1) 读取指定 object 多个 resource 请求

```
GET http://api.heclouds.com/nbiot?imei=test2&obj_id=3200 HTTP/1.1
Host: api.heclouds.com
api-key: //<APK-KEY> //<APK-KEY>参数请替换成产品或者设备 API-KEY
Content-Length: 0
```

模组端回复

```
+MIPLREAD:0,19500,3200,-1,-1 //收到平台侧下发的读取 object 下所有 instance 下所有 resource 请求
AT+MIPLREAD=0,19500,3200,0,5850,1,"12222.....33333",0 //终端读取某个 resource 值，但不触发上报
OK
AT+MIPLREAD=0,19500,3200,0,5750,1,"juidl....kkkkkk1",1 //读取 resource 值，并触发上报数据
//例子中加长 resid 5850 和 5750 的 payload 长度，使其总和在 1K 至 2K 之间，默认分包阈值 1024
```

附录 A IPSO 资源模型示例

Objects	ObjectID	ResourceIDs	Resource Name
光线传感器	3301	5700	Sensor Val →
光线传感器	3301	5701	Units
光线传感器	3301	5601	Min Measured Val
光线传感器	3301	5602	Max Measured Val
光线传感器	3301	5603	Min Range Val
光线传感器	3301	5604	Max Range Val
光线传感器	3301	5605	Reset Min and Max Measured Vals
温度传感器	3303	5700	Sensor Val →
温度传感器	3303	5701	Units
温度传感器	3303	5601	Min Measured Val
温度传感器	3303	5602	Max Measured Val
温度传感器	3303	5603	Min Range Val
温度传感器	3303	5604	Max Range Val
温度传感器	3303	5605	Reset Min and Max Measured Vals
温度传感器	3303	5750	Applicaton Type
湿度传感器	3304	5700	Sensor Val
湿度传感器	3304	5701	Units
湿度传感器	3304	5601	Min Measured Val
湿度传感器	3304	5602	Max Measured Val
湿度传感器	3304	5603	Min Range Val
湿度传感器	3304	5604	Max Range Val
湿度传感器	3304	5605	Reset Min and Max Measured Vals
湿度传感器	3304	5750	Applicaton Type
噪声传感器	3381	5700	Sensor Val
噪声传感器	3381	5701	Units
噪声传感器	3381	5601	Min Measured Val
噪声传感器	3381	5602	Max Measured Val
噪声传感器	3381	5603	Min Range Val
噪声传感器	3381	5604	Max Range Val
噪声传感器	3381	5605	Reset Min and Max Measured Vals
噪声传感器	3381	5750	Applicaton Type
激光接触传感器	3382	5700	Sensor Val
激光接触传感器	3382	5701	Units

激光接触传感器	3382	5601	Min Measured Val
激光接触传感器	3382	5602	Max Measured Val
激光接触传感器	3382	5603	Min Range Val
激光接触传感器	3382	5604	Max Range Val
激光接触传感器	3382	5605	Reset Min and Max Measured Vals
激光接触传感器	3382	5750	Applicaton Type
灯控制	3322	5850	On/Off
灯控制	3322	5851	Dimmer
灯控制	3322	5706	Colour
灯控制	3322	5701	Units
灯控制	3322	5852	On Time
报警器控制	3383	5850	On/Off
报警器控制	3383	5852	On Time
粉尘传感器	3384	5700	Sensor Val
粉尘传感器	3384	5701	Units
粉尘传感器	3384	5601	Min Measured Val
粉尘传感器	3384	5602	Max Measured Val
粉尘传感器	3384	5603	Min Range Val
粉尘传感器	3384	5604	Max Range Val
粉尘传感器	3384	5605	Reset Min and Max Measured Vals
瓦斯传感器	3385	5700	Sensor Val
瓦斯传感器	3385	5701	Units
瓦斯传感器	3385	5601	Min Measured Val
瓦斯传感器	3385	5602	Max Measured Val
瓦斯传感器	3385	5603	Min Range Val
瓦斯传感器	3385	5604	Max Range Val
瓦斯传感器	3385	5605	Reset Min and Max Measured Vals