

# M5312 OneNET 平台接入流程指导手册 基础通信套件技术规范版本

NB-IoT 系列 R1.0



中移物联网有限公司

2018.08.02



## 重要声明

本文档描述的产品及其附件的某些特性和功能，取决于当地网络的设计和性能。因此，本文档中的描述可能与您购买的产品或其附件并非完全一一对应。本公司保留随时修改本文档中任何信息的权利，无需进行任何提前通知且不承担任何责任。

### 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©中移物联网有限公司，2018，保留所有权利，但注明引用其他方的内容除外。

### 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。

### 操作系统更新声明

操作系统仅支持官方升级；如用户自己刷非官方系统，导致安全风险和损失由用户负责。

### 固件包完整性风险声明

固件仅支持官方升级；如用户自己刷非官方固件，导致安全风险和损失由用户负责。

### 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。

## 修订记录

版本	日期	作者	描述
1.0	2018.08.02	唐建伦	创建

本文档内《规范》一词指《中国移动蜂窝物联网终端基础通信套件技术规范 V1.0.0》



中国移动  
China Mobile

# 目 录

文档说明.....	1
一、对接总体流程.....	2
二、平台侧操作流程说明.....	3
2.1 平台侧产品创建流程.....	3
2.2 平台侧设备注册流程.....	4
三、模组侧操作流程说明.....	5
3.1 模组侧设备实体创建流程.....	6
3.2 声明操作资源.....	6
3.2.1 订阅 Object 组配置流程.....	6
3.2.2 订阅 Resource 参数配置流程（托管模式）.....	7
3.3 模组侧发起注册请求.....	7
3.4 注册结果上报.....	8
3.5 OneNET 数据收发.....	9
3.5.1 向 OneNET 平台发送数据——数据上报流程.....	9
3.5.2 设备管理流程——数据下发流程.....	10
3.6 模组侧设备注销流程.....	17
四、附录.....	18
4.1 M5312 对接 OneNET 平台基本 AT 指令流程.....	18
4.1.1 上电检查流程.....	18
4.1.2 模组侧设备创建及资源订阅，登录流程.....	18
4.1.3 OneNET 数据收发流程.....	19
4.1.4 模组侧设备注销流程.....	19
4.2 AT 指令参考手册.....	19

## 文档说明

本手册描述 M5312 与 OneNET 平台对接流程示例，对接时采用 CoAP+LwM2M 协议，模组软件版本为 **M5312-PBRH1S00** 及其以上版本。本文旨在帮助客户快速完成 M5312 模组与 OneNET 平台的对接工作。

在使用本手册进行 OneNET 平台对接之前，需先完成模组初始化入网流程，模组驻网成功后，方可进行 OneNET 平台业务对接流程。



## 一、对接总体流程

对接流程包括两个方面的准备工作：

- 1) **OneNET 平台侧——产品创建与设备注册流程；**
- 2) **M5312 模组侧——初始化驻网与模组侧设备创建/登录流程；**

对接总流程如下图所示，通过如下流程图，可以明确 M5312 模组与 OneNET 平台对接时的总体步骤：



## 二、平台侧操作流程说明

平台侧需要完成产品创建和设备注册流程，用于在平台上创建产品，并声明产品下挂载的设备；该步骤需要在模组侧进行设备登录操作之前完成。

中移物联网在全国有多个省市均部署有 OneNET 平台，每个 OneNET 平台都有独立的门户网站和接入点信息。其中，重庆 OneNET 平台为主平台，其他省平台均以重庆平台为参考，但其他平台更新进度可能存在不一致，若使用其他平台进行开发和测试，使用前需向 OneNET 平台或 FAE 技术支持确认相关信息。

本文档例程中，使用重庆 OneNET 平台，门户网站为：<https://open.iot.10086.cn/>

### 2.1 平台侧产品创建流程

目前，OneNET 平台已经支持通过配置界面直接创建 LwM2M 产品，创建产品时直接选择联网方式为 NB-IoT，设备接入协议选择 LwM2M。

设备接入方式：

☒ 公开协议      ☐ 私有协议(RGMP)

联网方式：

☐ wifi      ☐ 移动蜂窝网络      ☒ NB-IoT

设备接入协议：

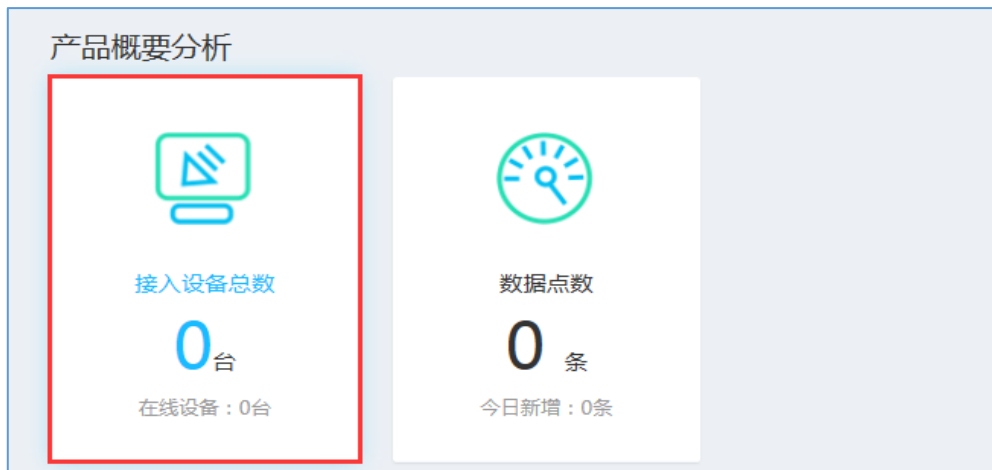
LWM2M

**LWM2M基本功能介绍：**

- 1、低功耗；
- 2、高覆盖；
- 3、低成本；
- 4、强连接；

## 2.2 平台侧设备注册流程

产品创建成功后，同样可以直接在界面直接创建 LwM2M 设备，创建设备之前需要先进入产品详情界面。



创建设备时，IMEI 和 IMSI 为设备登录时的鉴权信息，**必须**使用模组对应的 IMEI 号和 IMSI 号进行设备注册（模组端可通过 AT+EGMR=2,7 和 AT+CIMI 获取），既能满足鉴权信息的唯一性，也能和模组一一对应，便于开发管理。

编辑设备

LWM2M

×

• 设备名称:

P3.1

• IMEI:

111111111111112

• IMSI:

460040314201579

是否开启自动订阅:

☒ 是

☐ 否

编辑

取消

设备添加成功后，点击进入设备管理界面，即可看到注册成功的设备信息，

设备名称前面的圆圈表示设备状态，灰色指示设备处于离线状态，绿色指示设备正处于在线状态。



至此，平台侧操作完成，后续进行 M5312 模组侧操作流程。

若需要使用调用 API 的方式进行平台侧产品及设备注册，请联系中移物联网 FAE 获取技术支持。

### 三、模组侧操作流程说明

模组侧需要完成设备创建工作，声明需操作的资源（包括 Object 组配置和 Resource 配置）；上述流程完成之后，方可执行登录操作。

模组侧设备创建步骤的目的是，在模组上创建一个和平台注册配置相匹配的下级设备。

订阅 Object 组的目的是，在模组侧声明需要使用的通信套件；这些套件均为满足 LwM2M 协议的特定功能的实体，这些实体在平台上按照 LwM2M 协议已经完成了预定义；

订阅 Resource 资源的目的是，向平台声明需要在资源列表中显示的 Resource 信息。

具体的 Object 编码规范可以参照 IPSO 规范或 OMA 模型规范。具体文档请从如下地址获取：

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

### 3.1 模组侧设备实体创建流程

模组侧设备创建使用 AT+MIPLCREATE 指令，其使用格式为：

AT+MIPLCREATE=<totalsize>,<config>,<index>,<currentsize>,<flag>

其中，

<totalsize>	指示<config>部分总数据长度，按照ASCII计数；
<config>	具体的设备配置数据，满足配置结构体规范；
<index>	配置文件的序号，从前到后按照降序依次分配序号为N-1~0，按照从大到小序号的顺序每段调用一次AT指令，如此当index为0时意味着该条指令为最后一条配置消息
<currentsize>	当前指令所包含的配置文件大小
<flag>	消息标识
	1: 第一条消息
	2: 中间消息
	0: 最后一条消息

AT 指令如下：

```
AT+MIPLCREATE=49,130031F10003F200220400110005434D494F540000000000
0D31383332E3233302E34302E33390000F30008E200C8000000,0,49,0（自维护模式
指令）
```

```
AT+MIPLCREATE=49,130031F10003F200220400110005434D494F540000000000
0D31383332E3233302E34302E33390000F30008E200C8000001,0,49,0（托管模式指
令）
```

### 3.2 声明操作资源

#### 3.2.1 订阅 Object 组配置流程

订阅 Object 组配置添加，直接使用 AT+MIPLADDOBJ 指令，在进行组添加之前，需要确认业务流程中需要使用到的 Object 信息；

例程中，使用 Object 为 3303，instance 为 0；AT 指令如下：

```
AT+MIPLADDOBJ=0,3303,1,"1",6,1
```

```
OK
```

在本步骤中，添加了订阅 Object 组配置；在登录平台时，订阅的 Object 信息

将会上传到 OneNET 平台，注册鉴权通过后，平台会对所有被订阅的 Object 下发 Observer 消息及 Discover 消息；所以，通信流程中需要使用的所有 Object 都应在模组发起登录前配置完毕。

### 3.2.2 订阅 Resource 参数配置流程（托管模式）

订阅 Resource 参数配置，使用 MIPLDISCOVERRSP 指令；订阅后的 Resource 资源，在登录成功后，将在 OneNET 平台的资源管理页面中呈现。

需要注意的是，一旦订阅 Resource 参数上报之后，只能向上报过的 Resource 资源实体中上传数据，务必确认所有通信中需使用的 Resource 参数都完成订阅上报设置。

例程中，仅使用 Object 3303，instance 0 下的 5700;5601;5602;5603;5604;5701 ;5605 Resource；AT 指令如下：

```
AT+MIPLDISCOVERRSP=0,3303,1,34,"5700;5601;5602;5603;5604;5701;5605"
```

上述指令代表订阅 Object 3303，instance 0 下的 5500 Resource 为需要使用到的资源实体。

该步骤完成后，模组登录到平台之后即可在平台侧的资源列表中看到实例。

若为自维护模式，执行该步骤将返回 ERROR。

### 3.3 模组侧发起注册请求

注册请求为异步事件，其登录是否成功，可以从 3.4 节的返回数据中看到。

登录时，使用如下 AT 指令：

```
AT+MIPLOPEN=0,3000,30
```

上述指令中，3000 代表本次注册成功后的生存周期；30 代表注册超时时间；如果超过该时间，模组还未接收到平台的登录响应，模组则会上报登录失败的通

知。

### 3.4 注册结果上报

模组上报注册请求，服务器收到注册请求数据之后，会根据数据内容，返回本次注册结果；如果注册失败，或者注册超时，模组都会上报注册失败的响应。

注册成功上报打印如下：

`+MIPLEVENT:0,2` //bootstrap 流程成功（启用 bootstrap 服务方返回）

`+MIPLEVENT:0,6` //注册服务器时成功

注册失败上报打印如下：

`+MIPLEVENT:0,3` //bootstrap 流程失败

`+MIPLEVENT:0,7` //注册服务器时失败

模组注册成功后，平台会下发 Observer 消息和 Discover 消息；

`+MIPLOBSERVE:0,64289,1,3303,0,-1`

`+MIPLDISCOVER:0,64290,3303`

Observer 消息是平台传递的观测请求消息，若为**托管模式**模组收到该消息后模组会维护相关观测记录，用户**无需处理**；若为**自维护模式**，用户则**5S 内必须返回**以下指令：

`AT+MIPLOBSERVERSP=0,64289,1`

Discover 消息是上报通知模组需要获取指定 Object 的属性，若为**托管模式**模组收到该消息后模组将自动反馈相关属性，用户无需处理；若为**自维护模式**，**必须返回**以下指令，该流程对应托管模式的 3.2.2 订阅 Resource 参数配置流程：

`AT+MIPLDISCOVERRSP=0,64290,1,34,"5700;5601;5602;5603;5604;5701;5605"`

登陆成功后，可以在 OneNET 平台对应设备的资源列表中看到订阅的 Object 实体。

全部

部

操作

P3.1

设备ID:28386706

创建时间:2018-04-09 15:47:39

资源列表

进入资源列表，可以看到登录时订阅的 Object/Resource 信息：

序号	对象名称	实例个数	属性个数
1	Temperature	1	7

序号	实例名称	属性名	属性类型	属性值	时间	操作
1	Temperature_0	Sensor Value	float	null	null	读 写 执行
		Min Measured Value	float	null	null	读 写 执行
		Max Measured Value	float	null	null	读 写 执行
		Min Range Value	float	null	null	读 写 执行
		Max Range Value	float	null	null	读 写 执行
		Sensor Units	string	null	null	读 写 执行
		Reset Min and Max	string	null	null	读 写 执行

### 3.5 OneNET 数据收发

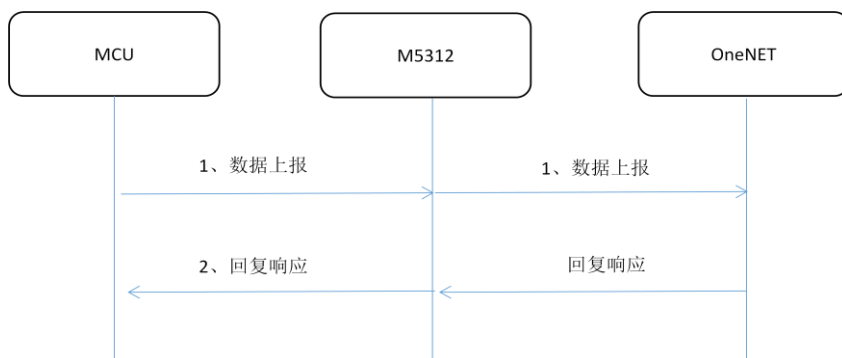
#### 3.5.1 向 OneNET 平台发送数据——数据上报流程

用户数据上报使用 MIPLNOTIFY 指令，

AT+MIPLNOTIFY=<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<index>,<flag> [,<ackid>]

其中 flag 标示数据是否上报(为 0 时为最后一包，上报)；ackid 为可选参数，配置平台响应标示。

数据上报流程如下图所示：



例程中，向 3303 Object/5700Resource 中上传用户数据。

`AT+MIPLNOTIFY=0,0,3303,0,5700,4,3,1,1,0,0`

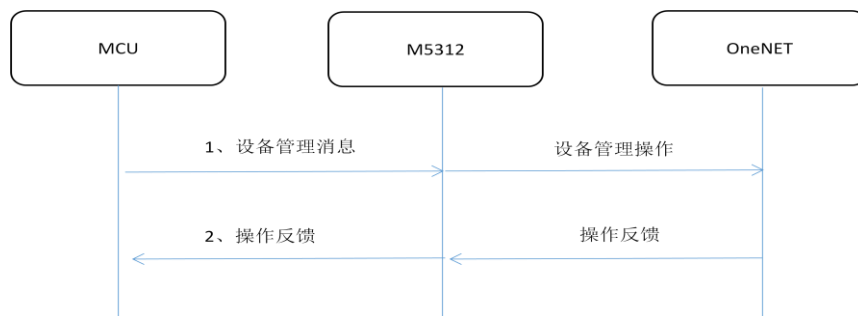
### 3.5.2 设备管理流程——数据下发流程

设备管理流程，即平台下发 **Read/Write/Execute** 操作指令，模组接收到指令后通过串口转发给 **MCU**，**MCU** 实现相应操作并响应的流程。本流程需要传感器，**MCU**，模组，平台相结合进行操作。

目前，数据下发流程，可以通过 **OneNET** 平台直接测试；在登录状态下，资源管理界面中可以看到读、写、执行等操作；同时，也支持通过 **API** 调用相应接口的方式，实现数据下发流程。

并且，如果下发数据后，平台未收到回复，平台会尝试多次下发，在超过门限次数后，将会反馈失败。所以，**MCU** 应在收到模组转发的平台操作指示后数秒钟内（推荐 3s）上报对应操作结果，否则可能导致操作超时失败。

设备管理流程如下图所示：



例程中，采用调用 **API** 的方式进行数据下发流程测试；并且通过手动回复操作响应结果的方式，模拟 **MCU** 的操作响应。

## (1) Read 操作流程

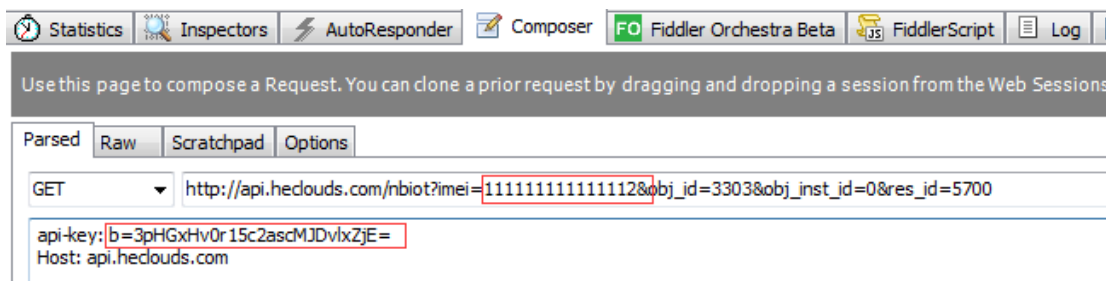
控制平台下发 Read 操作报文格式如下示例如下：

```
GET http://api.heclouds.com/nbiot?imei=11111111111112&obj_id=3303 HTTP/1.1
```

```
Host: api.heclouds.com
```

```
api-key: <api-key>
```

测试中，使用调用 API 方式（此处使用 Fiddler 4 工具），列举读取指定 Object 下指定 instance 下的所有 resource 资源请求操作：



调用 API 中需要提供对应设备注册时的 IMEI 和产品 API-KEY，其中，IMEI 可以在设备详情中查看：



其中，API-KEY 可以在产品概况中查看：



调用 API 指令下发成功后，模组会收到对应的 Read 操作指令：

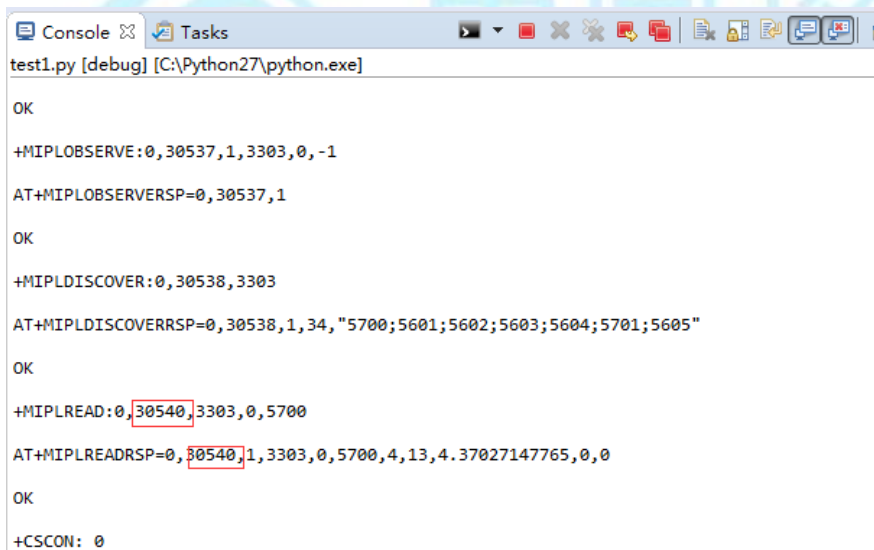
```
[2018-05-03_11:59:28:908]AT+MIPLNOTIFY=0,64290,3303,0,5700,4,1,3,"1.1",0,15
[2018-05-03_11:59:28:908]OK
[2018-05-03_11:59:29:723]
[2018-05-03_11:59:29:723]+CSCON: 1
[2018-05-03_11:59:35:802]
[2018-05-03_11:59:35:802]+MIPLREAD:0,25915,3303,0,5700
[2018-05-03_12:00:02:742]
[2018-05-03_12:00:02:742]+CSCON: 0
```

模组收到 Read 指令并转发到 MCU 后,MCU 需要在响应时间窗内做出 Read 响应操作(平台侧现在推荐 3s 内执行操作回复),否则平台会在 API 调用回复中,上报 time\_out 报错。

Read 操作回复的格式为:

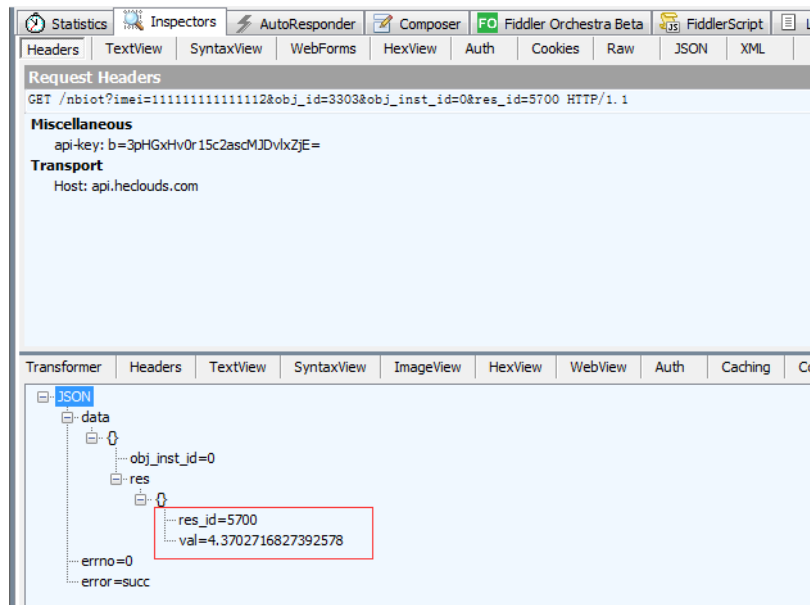
AT+MIPLREADRSP=<ref>,<msgid>,<result>[,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<index>,<flag>]

平台侧通过 msgid 识别不同 Read 操作对应的响应;所以 MCU 执行 Read 操作回复指令中的 msgid 必须和对应平台下发的 Read 操作中的 msgid 相同。按照格式,执行操作回复如下:



```
test1.py [debug] [C:\Python27\python.exe]
OK
+MIPLOBSERVE:0,30537,1,3303,0,-1
AT+MIPLOBSERVERSP=0,30537,1
OK
+MIPLDISCOVER:0,30538,3303
AT+MIPLDISCOVERRSP=0,30538,1,34,"5700;5601;5602;5603;5604;5701;5605"
OK
+MIPLREAD:0,30540,3303,0,5700
AT+MIPLREADRSP=0,30540,1,3303,0,5700,4,13,4.37027147765,0,0
OK
+CSCON: 0
```

在响应时间内做出操作回复后,查询报文回复中即可看到上一步中回复的数据值:



通过 API 参数的不同设置，还能实现读取指定 Object 下指定 instance 下的指定 resource 的数据；读取指定 Object 下所有 instance 的所有 resource 的数据等作为；具体 API 报文格式参照 OneNET 给出的开发文档。

## (2) Write 操作流程

控制平台下发 Write 操作报文格式如下示例如下：

POST

http://api.heclouds.com/nbiot?imei=111111111111112&obj\_id=3303&obj\_inst\_id=0&mode=2 HTTP/1.1

Host: api.heclouds.com

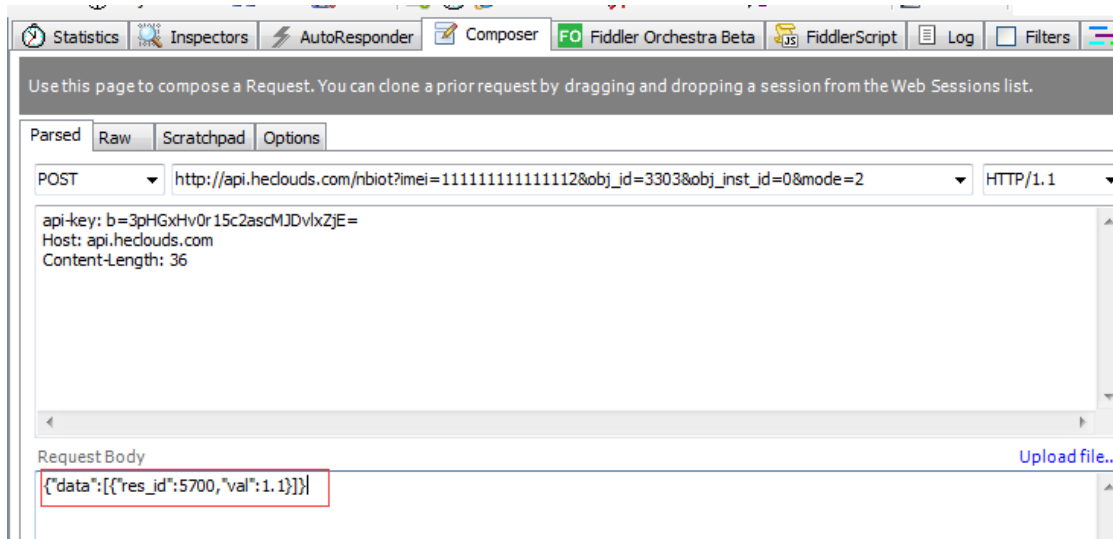
api-key: <api-key>

Body 部分报文如下：

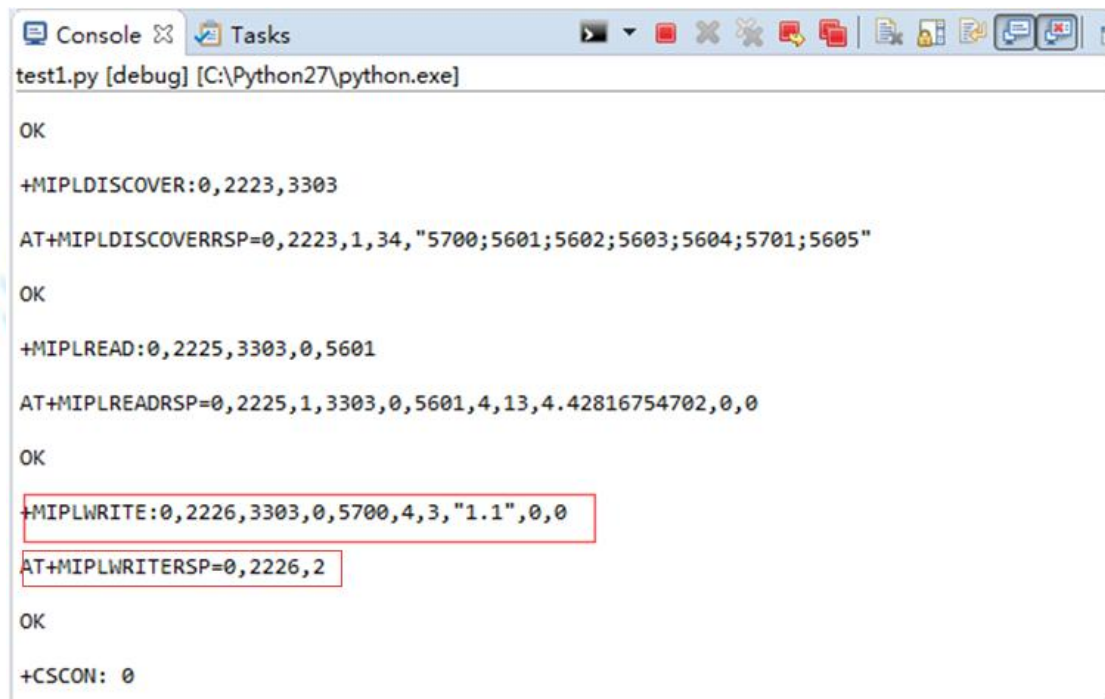
```
{
  "data": [{
    "val": 0,
    "res_id": 5700
  }]
```

测试中，使用调用 API 方式（此处使用 Fiddler 4 工具）；向 Object 3303 的

5700Res 中写入浮点型变量值"1.1"。



API 调用成功后，模组将会接收到对应的 Write 指令：



模组收到 Write 指令转发给 MCU 后，MCU 同样需要在响应时间窗内做出相应 Write 响应操作，否则平台会执行命令重传，重传次数为 2 次。两次重传后，若平台还未收到模组的回复响应，则会判断模组回复超时失败。

Write 操作回复格式为：

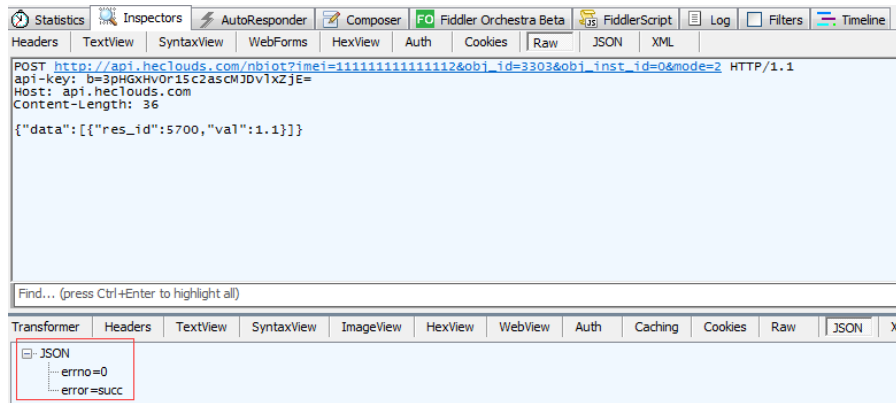
AT+MIPLWRITERSP=<ref>,<msgid>,<result> //result=2 表示写入操作

成功

和 Read 操作相同，执行 Write 操作回复指令中的 msgid 必须和对应平台下

发的 Write 操作中的 msgid 相同。

在回复响应时间内做出回复操作后，查询报文回复即可看 Write 指令操作成功的提示。



具体 API 报文格式，见 OneNET 平台提供的开发文档。

### (3) Execute 操作流程

除 Read 和 Write 操作之外，平台还支持 Execute 操作；平台控制下发 Execute 操作报文格式如下示例如下：

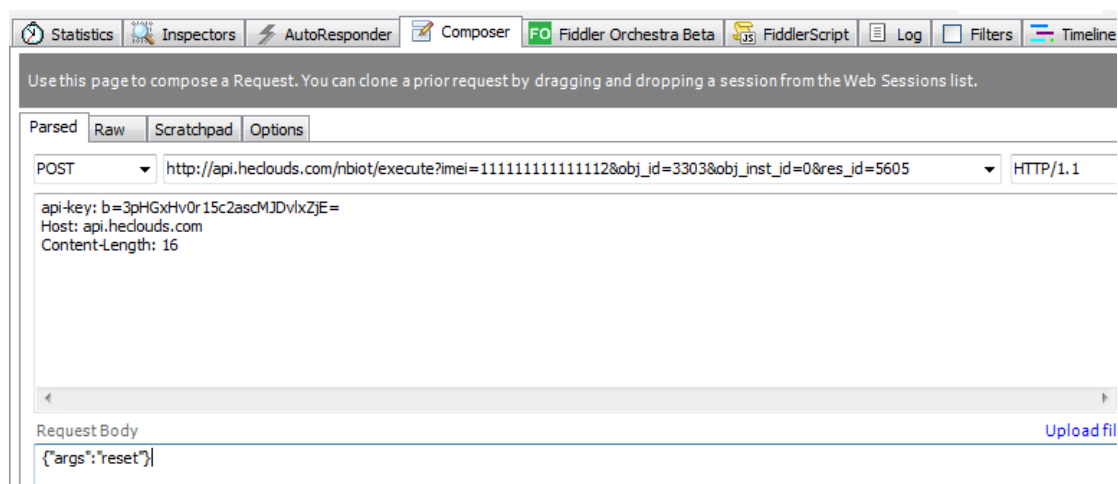
POST

```
http://api.heclouds.com/nbiot/execute?imei=111111111111112&obj_id=3303&obj_inst_id=0&res_id=5605 HTTP/1.1
Host: api.heclouds.com
api-key: <api-key>
```

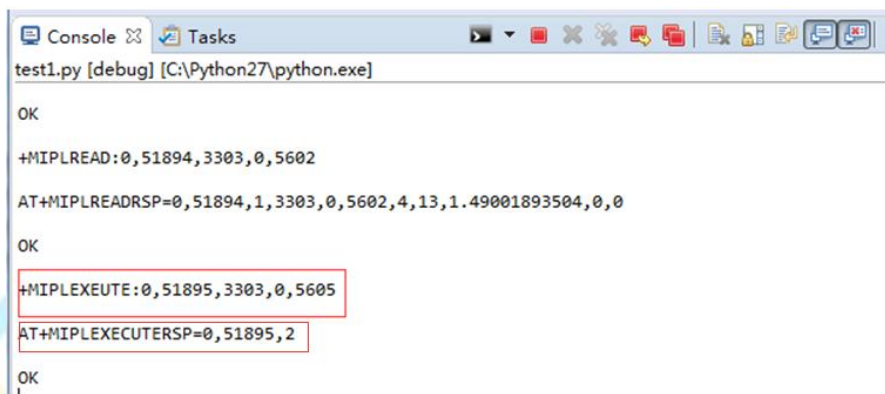
Body 部分报文如下：

```
{
  "args": "reset"
}
```

测试中，使用调用 API 方式（此处使用 Fiddler 4 工具）；向 Object 3303 的 5605Res 中下发 Execute 操作，其值为字符型变量值 “reset”。



API 调用成功后，模组将会接收到对应的 Execute 指令：



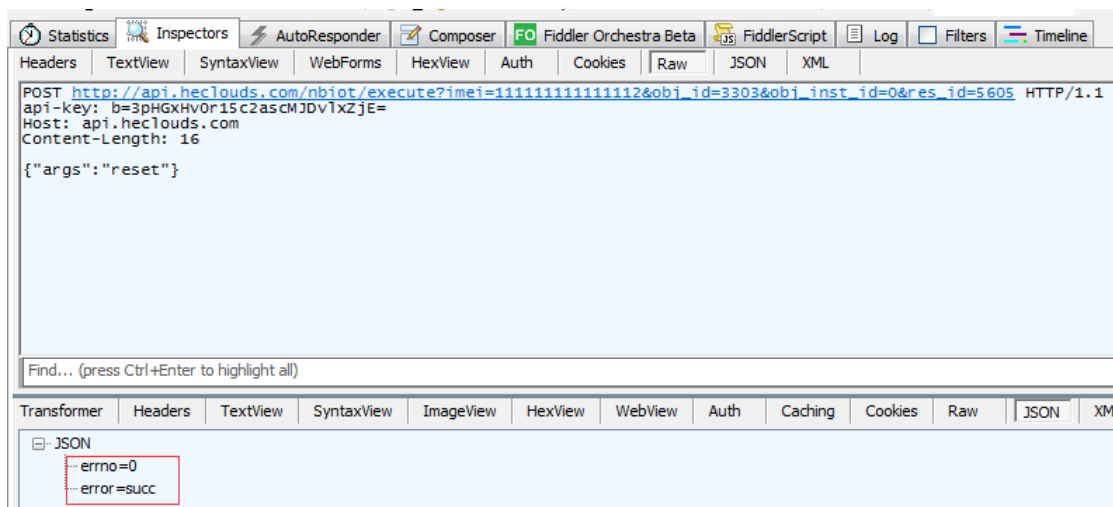
和前面的 Read、Write 操作相同，模组收到 Execute 指令转发 MCU 后，MCU 同样需要在响应时间窗内做出相应 Execute 响应操作，否则平台会执行命令重传，重传次数为 2 次。两次重传后，若平台还未收到模组的回复响应，则会判断模组回复超时失败。

Execute 操作回复格式为：

AT+MIPLEXECUTE=<ref>,<msgid>,<result> //result=2 表示下发操作成功

与之前相同，执行 Execute 操作回复指令中的 msgid 必须和对应平台下发的 Execute 操作中的 msgid 相同。

在回复响应时间内做出回复操作后，查询报文回复即可看 Execute 指令操作成功的提示。



具体 API 报文格式，见 OneNET 平台提供的开发文档。

注：每次从平台使用 **Execute** 下发一般情况下不带参数。

### 3.6 模组侧设备注销流程

在设备存活时间内，模组可以主动发起设备注销流程，注销流程如下：

```
AT+MIPLCLOSE=0
```

```
OK
```

上述 MIPLCLOSE 指令指示模组主动发起设备登录注销流程，注销成功后，平台侧设备资源列表中的订阅信息将被清理。

```
AT+MIPLDELOBJ=0,3303,0
```

```
OK
```

上述 MIPLDELOBJ 指示删除模组订阅的本地 3303 Object。

```
AT+MIPLDELETE=0
```

```
OK
```

上述 MIPLDELETE 指令指示销毁模组现存的通信实例；在现存通信实例删除之前，不能在模组上新建通信实例。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，再删除 Object 以及通信实例。

## 四、附录

### 4.1 M5312 对接 OneNET 平台基本 AT 指令流程

M5312 对接 OneNET 平台完整的 AT 指令流程，请按顺序参照 4.1.1-4.1.4 节例程。

#### 4.1.1 上电检查流程

- |               |                                  |
|---------------|----------------------------------|
| (1) AT        | //判断模组是否上电开机成功                   |
| (2) AT+CSQ    | //信号质量检查                         |
| (3) AT+CEREG? | //判断 PS 域附着状态，标识位返回 1 或 5 表示附着正常 |
| (4) AT+CGATT? | //检查模组 PS 附着状态                   |
| (5) AT+CGACT? | //检查模组 PDP 激活状态                  |

#### 4.1.2 模组侧设备创建及资源订阅，登录流程

##### 4.1.2.1 托管模式

- |  |                   |
|--|-------------------|
| (1) AT+MIPLCREATE=49,130031F10003F200220400110005434D494F5400000000<br>000D3138332E3233302E34302E33390000F30008E200C8000001,0,49,0 | //设置模组侧设备注册码      |
| (2) AT+MIPLADDOBJ=0,3303,1,"1",6,1   | //添加 Object       |
| (3) AT+MIPLDISCOVERRSP=0,3303,1,34,"5700;5601;5602;5603;5604;5701;5605"  | //订阅 Object 资源设置  |
| (4) AT+MIPLOPEN=0,3600,30  | //注册登录到 OneNET 平台 |

##### 4.1.2.2 自维护模式

- |  |                    |
|--|--------------------|
| (1) AT+MIPLCREATE=49,130031F10003F200220400110005434D494F5400000000<br>000D3138332E3233302E34302E33390000F30008E200C8000000,0,49,0 | //设置模组侧设备注册码       |
| (2) AT+MIPLADDOBJ=0,3303,1,"1",6,1   | //添加 Object        |
| (3) AT+MIPLOPEN=0,3600,30  | //注册登录到 OneNET 平台  |
| (4) +MIPLOBSEVER:0,64289,1,3303,0,-1   | //平台下发的 observe 消息 |
| AT+MIPLOBSEVERSP=0,64289,1   | //返回 observersp 消息 |

- (5) +MIPLDISCOVER:0,64290,3303 //平台下发的 discover 消息  
AT+MIPLDISCOVERRSP=0,64290,1,34,"5700;5601;5602;5603;5604;5701;5605"  
//返回 discoverrsp 消息

### 4.1.3 OneNET 数据收发流程

- (1) AT+MIPLNOTIFY=0,0,3303,0,5700,4,4,1,23,0,0 //数据上传  
(2) AT+MIPLREADRSP=0,20931,1,3303,0,5700,4,3,2,1,0,0 //Read 操作回复流程  
(3) AT+MIPLWRITERSP=0,24536,2 //Write 操作回复流程  
(4) AT+MIPLXECUTERSP=0,59968,2 //EXECUTE 操作回复流程

### 4.1.4 模组侧设备注销流程

- (1) AT+MIPLCLOSE=0 //登录注销流程  
(2) AT+MIPLDELOBJ=0,3303,0 //模组侧订阅资源列表释放  
(3) AT+MIPLDELETE=0 //模组侧通信实例删除

## 4.2 AT 指令参考手册

AT 指令集是一组命令集合，终端使用这组命令集合来和模组或者芯片进行通信。AT 指令由 ASCII 字符组成，使用前缀“AT”。每个 AT 请求都必须要求从模组或者芯片获得回复代码，用于回复终端当前请求的执行状态。基础通信套件提供如下统一的接口给应用和 URC 事件给 MCU：

[需求编号]：TS-UE-SUITE-ATRQ-00001

[需求描述]：

指 令 名 称 :

AT+MIPLCREATE=<totalsize>,<config>,<index>,<currentsize>,<flag><CR>

功能：该指令创建一个基础通信套件的实例

参数：

<totalsize>：config 文件的总长度

<config>：配置文件，参见配置文件格式

<index>：配置文件的序号，考虑到 AT 指令长度有限，一个完整的配置文件未必能在一条 AT 指令中发送完成，可以将内容切分成多段，比如分为 N 段，则从前到后按照降序依次分配序号为 N-1~0，按照从大到小序号的顺序每段调用一次 AT 指令，如此当 index 为 0 时意味着该条指令为最后一条配置消息

<currentsize>：当前指令所包含的配置文件长度

<flag>：消息标识

1：第一条消息

2：中间消息

0: 最后一条消息

返回值:

<CR><LF>OK<CR><LF>: 如果发送的是 index 不为 0 的消息, 该回复代表消息正确接收

<CR><LF><ref><CR><LF>OK<CR><LF>:

只有接到 index=0 的消息才会返回<ref>参数, 该回复代表消息正确接收, 且返回一个创建完成的基础通信套件的一个实例标识, 类型为一个无符号整数。

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。

errid 可包含:

601 参数错误

602 状态错误

100 未知错误

等错误返回, 可扩展

[需求澄清]: 如果只有一条配置消息, 则 index 和 flag 都为 0。

M5312 目前长度仅支持在一条 AT 指令中发送完成。

[需求编号]: TS-UE-SUITE-ATRQ-00002

[需求描述]:

指令名称: AT+MIPLDELETE=<ref><CR>

功能: 该指令删除一个基础通信套件的实例

参数:

<ref>: 基础通信套件的一个实例标识, 类型为一个无符号整数

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00003

[需求描述]:

指令名称: AT+MIPLOPEN=<ref>,<lifetime>[,<timeout>]<CR>

功能: 该指令向平台发起注册请求

参数:

<ref>: 基础通信套件的一个实例标识, 类型为一个无符号整数

<lifetime>: 生命周期, 单位为 s

<timeout>: 注册的超时时长, 单位为 s

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00004

[需求描述]:

指令名称: AT+MIPLCLOSE=<ref><CR>

功能: 该指令向平台发起注销请求

参数:

<ref>: 基础通信套件的一个实例标识, 类型为一个无符号整数

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00005

[需求描述]:

指令名称:

AT+MIPLADDOBJ=<ref>,<objectid>,<instancecount>,<instancebitmap>,<attributecount>,<actioncount><CR>

功能: 该指令添加一个 object

参数:

<ref>: 基础通信套件的一个实例标识, 类型为一个无符号整数

<objectid>: 对象 id

<instancecount>: 实例个数

<instancebitmap>: 实例位图, 字符串格式, 每一个字符表示为一个实例, 其中 1 表示可用, 0 表示不可用。例如当前添加的 object 有 5 个实例, 其中, 1, 3 可用, 则实例位图为 00101

<attributecount>: 可读取的资源个数

<actioncount>: 可执行的资源个数

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00006

[需求描述]:

指令名称: AT+MIPLDELOBJ=<ref>,<objectid><CR>

功能: 该指令删除一个 object

参数:

<ref>: 基础通信套件的一个实例标识, 类型为一个无符号整数

<objectid>: 对象 id

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00007

[需求描述]:

指令名称:

AT+MIPLNOTIFY=<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>

>,<len>,<value>,<index>,<flag>[,<ackid>]<CR>

功能：该指令通知基础通信套件一个数值变化

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id

<objectid>：对象 id

<instanceid>：实例 id

<resourceid>：资源 id

<valuetype>：发送的数据类型

<len>：上报的数据长度

<value>：发送的数据值

<index>：指令序号。若某个 Notify 操作需要 N 条消息组合为一完整指令，则 index 从 N-1 至 0 降序编号，当 index 编号为 0 时表示本次 Notify 指令结束

<flag>：消息标识

1：第一条消息

2：中间消息

0：最后一条消息

发送的数据类型要求如下：

string=1, //字符串型，支持最大字符串长度为 1024 字节；

opaque=2, //不透明类型，支持最大长度为 1024 字节；

integer=3, //整型，取值范围 0-2147483647；

float=4, //浮点型，有效位数为 6 位；

bool=5, //布尔型，取值范围 0-1；

<ackid>：MCU 指定该消息以 CON 形式上报，如果该条 Notify 消息成功，则基础通信套件会将收到的返回用+MIPLEVENT 返回。

返回值：

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>：返回错误。errid 定义同前面。

[需求澄清]：如果对于同一条消息，只有一条 notify 指令，则 index 和 flag 都为 0

[需求编号]：TS-UE-SUITE-ATRQ-00008

[需求描述]：

指令名称：

AT+MIPLREADRSP=<ref>,<msgid>,<result>[,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<index>,<flag>]<CR>

功能：用户接到+MIPLREAD 消息后，需要去读取需要的资源的值，读取到需要的值，用户使用该指令将读取到的值一条一条发给基础通信套件，msgid 值为收到的+MIPLREAD 的消息携带的 msgid。

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id

<result>：读取操作的结果；可返回代码如下：

1            2.05 Content            读取操作正确完成

- 11 4.00 Bad Request
- 12 4.01 Unauthorized
- 13 4.04 Not Found
- 14 4.05 Method Not Allowed
- 15 4.06 Not Acceptable

<objectid>: 对象 id

<instanceid>: 实例 id

<resourceid>: 资源 id

<valuetype>: 发送的数据类型

发送的数据类型要求如下:

string=1, //字符串型, 支持最大字符串长度为 1024 字节;

opaque=2, //不透明类型, 支持最大长度为 1024 字节;

integer=3, //整型, 取值范围 0- 2147483647;

float=4, //浮点型, 有效位数为 6 位;

bool=5, //布尔型, 取值范围 0-1;

<len>: 读取的数据长度

<value>: 发送的数据值

<index>: 指令序号。若某个 Read 操作需要 N 条消息组合为一完整指令, 则 index 从 N-1 至 0 降序编号, 当 index 编号为 0 时表示本次 Read 指令结束

<flag>: 消息标识

1: 第一条消息

2: 中间消息

0: 最后一条消息

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 如果对于同一条消息, 只有一条 notify 指令, 则 index 和 flag 都为 0; 如果 result 不是 1, 则<objectid>, <instanceid>, <resourceid>, <valuetype>, <len>, <value>, <index>, <flag>参数省略。result 不为 1 代表该条读取消息错误, 则上报读取错误结果给平台并结束本次读取操作。

[需求编号]: TS-UE-SUITE-ATRQ-00009

[需求描述]:

指令名称: AT+MIPLWRITERSP=<ref>, <msgid>, <result><CR>

功能: 该指令通知基础通信套件写入的消息结果。用户接到+MIPLWRITE 消息后, 需要去写入需要的资源的值, 同时使用该消息通知基础通信套件写入的结果。调用该命令时 msgid 值为对应+MIPLWRITE 消息携带的 msgid。

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<result>: 写入操作的结果; 可返回代码如下:

2 2.04 Changed 写操作正确完成

11 4.00 Bad Request

- 12 4.01 Unauthorized
- 13 4.04 Not Found
- 14 4.05 Method Not Allowed
- 19 4.15 Unsupported content format

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 同一个消息 id 的 write 返回只需要一次结果返回。

注意:

字符串型, 支持最大字符串长度为 1024 字节;

不透明类型, 支持最大长度为 1024 字节;

整型, 取值范围 0-2147483647;

浮点型, 有效位数为 6 位;

布尔型, 取值范围 0-1;

[需求编号]: TS-UE-SUITE-ATRQ-00010

[需求描述]:

指令名称: AT+MIPLEXECUTERSP=<ref>,<msgid>,<result><CR>

功能: 该指令通知基础通信套件执行操作的结果。用户接到+MIPLEXECUTE 消息后, 需要去执行请求的动作, 同时使用该消息通知基础通信套件执行的结果。调用该命令时将 msgid 置为对应+MIPLEXECUTE 消息携带的 msgid。

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<result>: 执行操作的结果; 可返回代码如下:

- 2 2.04 Changed 执行操作正确完成
- 11 4.00 Bad Request
- 12 4.01 Unauthorized
- 13 4.04 Not Found
- 14 4.05 Method Not Allowed

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

注意:

字符串型, 支持最大字符串长度为 1024 字节;

不透明类型, 支持最大长度为 1024 字节;

整型, 取值范围 0-2147483647;

浮点型, 有效位数为 6 位;

布尔型, 取值范围 0-1;

[需求编号]: TS-UE-SUITE-ATRQ-00011

[需求描述]:

指令名称: AT+MIPLOBSEVERSP=<ref>,<msgid>,<result><CR>

功能: 该指令通知基础通信套件观测指令是否有效。当应用程序收到+MIPLOBSEVER 消息后, 需要去验证该请求是否有效。调用该命令时将 msgid 置为对应+MIPLOBSEVER 消息携带的 msgid。该功能也包括 cancel observe 的回复

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<result>: 观测消息或者取消观察消息是否成功: 可返回代码如下:

1	2.05 Content	操作正确完成
11	4.00 Bad Request	
12	4.01 Unauthorized	
13	4.04 Not Found	
14	4.05 Method Not Allowed	
15	4.06 Not Acceptable	

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00012

[需求描述]:

指令名称:

AT+MIPLDISCOVERRSP=<ref>,<msgid>,<result>,<length>,<valuestring><CR>

功能: 该指令回复基础通信套件获取到的指定 object 的所有属性。

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<result>: 发现操作的结果; 可返回代码如下:

1	2.05 Content	操作正确完成
11	4.00 Bad Request	
12	4.01 Unauthorized	
13	4.04 Not Found	
14	4.05 Method Not Allowed	
15	4.06 Not Acceptable	

<length>: 返回 valuestring 的长度

<valuestring>: object 的属性要求, 多个属性之间使用逗号“;”隔开“1101;1102;1103”

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00013

[需求描述]:

指令名称: AT+MIPLPARAMETERESP=<ref>,<msgid>,<result><CR>

功能: 该指令通知基础通信套件执行操作的结果。用户接到+MIPLPARAMETER 消息后, 需要去执行请求的动作, 同时使用该消息通知基础通信套件执行的结果。调用该命令时 msgid 值为对应+MIPLPARAMETER 消息携带的 msgid。

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<result>: 设置订阅参数操作的结果; 可返回代码如下:

2            2.04 Changed            操作正确完成

11           4.00 Bad Request

12           4.01 Unauthorized

13           4.04 Not Found

14           4.05 Method Not Allowed

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00014

[需求描述]: 该指令通知基础通信套件发送主动更新注册信息

指令名称: AT+MIPLUPDATE=<ref>,<lifetime>,<withObjectFlag><CR>

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<lifetime>: 更新的 lifetime 值, 单位为 s

<withObjectFlag>: 是否需要同时更新注册的 Object 对象

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00015

[需求描述]:

指令名称: AT+MIPLVER? <CR>

功能: 该指令获取基础通信套件版本

参数: 无

返回值:

<CR><LF>OK<CR><LF>

<CR><LF>+CIS ERROR:<errid><CR><LF>: 返回错误。errid 定义同前面。

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00016

[需求描述]:

指令名称:

+MIPLREAD:<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>[,<count>] <CR>

功能：该指令是一个读取请求消息，如果 resourceid = ‘-1’，则该指令会携带 count 参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id，标识该读取请求消息

<objectid>：对象 id

<instanceid>：实例 id

<resourceid>：资源 id，如果为 ‘-1’，则代表需要读取该 instance 下的所有资源

<count>：当前需要读取的资源的个数

[需求澄清]：无

[需求编号]：TS-UE-SUITE-ATRQ-00017

[需求描述]：

指令名称：

+MIPLWRITE:<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len>,<value>,<flag><CR>

功能：该指令是一个写操作请求消息

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id，标识该写入请求消息

<objectid>：对象 id

<instanceid>：实例 id

<resourceid>：资源 id

<valuetype>：待写入的数据类型

待写入的数据类型要求如下：

string =1, //字符串型

opaque=2, //不透明类型

integer=3, //整型

float=4, //浮点型

bool=5, //布尔型

<len>：待写入的数据长度

<value>：待写入的数据值

<flag>：消息标识

1：第一条消息

2：中间消息

0：最后一条消息

[需求澄清]：无

[需求编号]：TS-UE-SUITE-ATRQ-00018

[需求描述]：

指令名称：

+MIPLEXECUTE:<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>[,<len>,<arguments>,<flag>]<CR>

功能：该指令是一个执行操作请求消息

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id，标识该执行请求消息

<objectid>：对象 id

<instanceid>：实例 id

<resourceid>：资源 id

<len>：参数的长度

<arguments>：执行的参数

<flag>：消息标识

1：第一条消息

2：中间消息

0：最后一条消息

[需求澄清]：无

[需求编号]：TS-UE-SUITE-ATRQ-00019

[需求描述]：

指令名称：

+MIPOBSERVE:<ref>,<msgid>,<flag>,<objectid>,<instanceid>,<resourceid><CR>

功能：该指令是一个观测请求消息

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id，标识该观测请求消息

<flag>：1 为添加观测，0 为取消观测

<objectid>：对象 id

<instanceid>：实例 id，如果为 ‘-1’，则代表观测该 object 下所有 instance 下的所有资源

<resourceid>：资源 id，如果为 ‘-1’，则代表观测该 instance 下的所有资源

[需求澄清]：无

[需求编号]：TS-UE-SUITE-ATRQ-00020

[需求描述]：

指令名称：

+MIPLPARAMETER:<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<len>,<parameter><CR>

功能：该指令是一个设置策略参数请求消息

参数：

<ref>：基础通信套件的一个标识，类型为一个无符号整数

<msgid>：消息 id，标识该观测请求消息

<objectid>：对象 id

<instanceid>：实例 id，如果为 ‘-1’，则代表观测该 object 下所有 instance 下的所有资源

<resourceid>：资源 id，如果为 ‘-1’，则代表观测该 instance 下的所有资源

<len>: 参数长度

<parameter>: 策略参数, 格式为字符串形式

包括如下策略:

pmin=xxx; pmax=xxx; gt=xxx; lt=xxx; stp=xxx

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00021

[需求描述]:

指令名称: +MIPLDISCOVER:<ref>,<msgid>,<objectid><CR>

功能: 该指令通知 mcu 需要获取指定 object 的属性。

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<msgid>: 消息 id

<object>: 指定的 object 对象 id

[需求澄清]: 无

[需求编号]: TS-UE-SUITE-ATRQ-00022

指令名称: +MIPLEVENT:<ref>,<evtid>[,<ackid>]<CR>

功能: 该指令上报一个状态事件

参数:

<ref>: 基础通信套件的一个标识, 类型为一个无符号整数

<evtid>: 事件 id

事件类型 id, 可扩展:

#define	CIS_EVENT_BASE	(cis_evt_t)0x00
#define	CIS_EVENT_BOOTSTRAP_START	CIS_EVENT_BASE + 1
#define	CIS_EVENT_BOOTSTRAP_SUCCESS	CIS_EVENT_BASE + 2
#define	CIS_EVENT_BOOTSTRAP_FAILED	CIS_EVENT_BASE + 3
#define	CIS_EVENT_CONNECT_SUCCESS	CIS_EVENT_BASE + 4
#define	CIS_EVENT_CONNECT_FAILED	CIS_EVENT_BASE + 5
#define	CIS_EVENT_REG_SUCCESS	CIS_EVENT_BASE + 6
#define	CIS_EVENT_REG_FAILED	CIS_EVENT_BASE + 7
#define	CIS_EVENT_LIFETIME_TIMEOUT	CIS_EVENT_BASE + 8
#define	CIS_EVENT_STATUS_HALT	CIS_EVENT_BASE + 9
#define	CIS_EVENT_UPDATE_SUCCESS	CIS_EVENT_BASE + 10
#define	CIS_EVENT_NOTIFY_SUCCESS	CIS_EVENT_BASE + 11

<ackid>: 当返回为 CIS\_EVENT\_NOTIFY\_SUCCESS 时, 携带 Notify 时下发的对应 ackid。

[需求澄清]: 无