
M6311 模块 onenet 串行接口

通讯指令

版本：V1.0

编制人：

时间：2016.3.21

版本信息

日期	版本	撰写人	审核人	修改说明
2016.3.21	V1.0	续芳		初稿
2016.4.20	V1.1	周俨		修改部分流程

1 概述

描述 WIFI 模组的扩展指令集功能以及使用方法。

扩展指令集主要应用在和中国移动物联网 OneNet 开放平台实现设备数据上传和命令控制功能，模块原有 AT 命令、Wifi 功能 AT 命令、TCP/IP 工具箱 AT 命令等保持不变。

2 模组扩展指令功能需求

2.1 模组扩展指令设计简介

模组扩展指令的设计旨在实现终端设备和平台服务器之间数据通讯，其中终端设备主要由 WIFI 模组及 MCU 主控模块组成，模组扩展指令的功能主要是将平台接入协议（Http 协议和 EDP 协议）植入 WIFI 模组，然后进行封装，给用户提供一个开放的指令接口，用户使用这些指令就可以轻松实现终端设备和平台服务器之间的数据通讯，平台服务器、WIFI 模组及 MCU 主控模块三者之间的关系如下图所示。



图 1 平台服务器、模组及 MCU 主控模块关系图

2.1 模组扩展指令设计思路

模组扩展指令的设计主要围绕终端设备和平台之间的数据通讯进行设计，主要包括以下过程和步骤。

步骤 1：主 Apikey 参数固化——终端设备开发商在 OneNET 平台注册公司管理用户账户，并在该账号下新建一个项目，一个项目对应平台上的一个 MasterKey（也称主 Apikey），将平台申请到的主 Apikey 通过串口工具或者 MCU 上的程序代码发送给 WIFI 模组，WIFI 模组将该参数烧录在固定地址空间。

步骤 2：平台设备定义——WIFI 模组利用在主 Apikey 在平台申请该设备的设备 ID 和设备 Apikey，同时也将申请成功的参数进行保存，后续 WIFI 模组就可以利用自身保存的设备 ID 及设备 Apikey 同平台服务器完成数据通讯。

步骤 3：建立设备连接——WIFI 模组的数据通讯是采用的 EDP 协议，EDP 协议是建立在

TCP/IP 基础上，所以要完成设备连接，先得确保 WIFI 模组连上可用的 AP，然后 WIFI 模组和平台服务器建立 TCP 连接，最后利用设备 ID 及设备 Apikey 封装成 EDP 连接报文建立设备连接。

步骤 4：数据上传---WIFI 模组接收主控 MCU 的数据，然后封装成 EDP 报文，发送至平台服务器。

步骤 5：命令控制---WIFI 模组接收到平台服务器的 EDP 命令控制报文，解析报文内容，然后发送给主控 MCU。

平台服务器、模组及 MCU 主控模块三者之间的数据流程如下图所示。

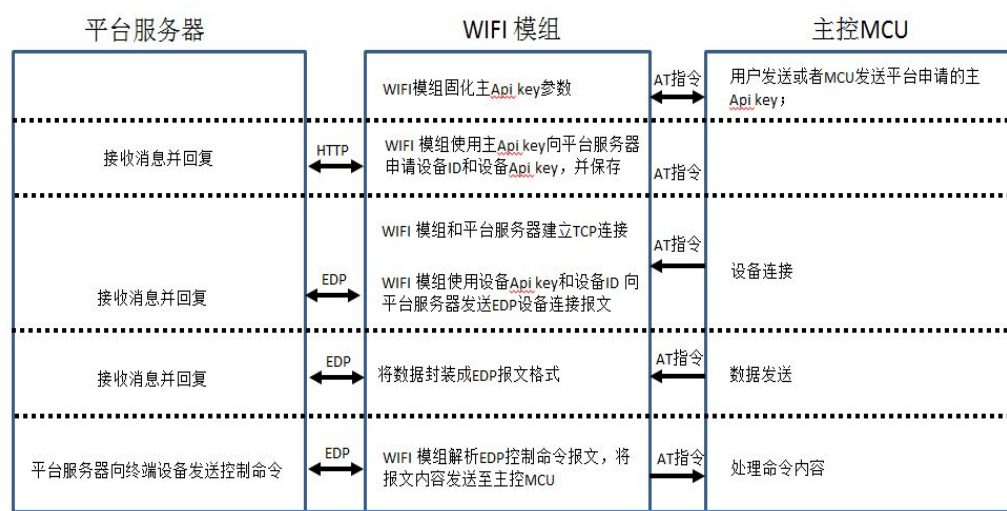


图 2 平台服务器、模组及 MCU 主控模块数据流程图

2.2 模组扩展指令命名规则及使用说明

扩展指令保持原 AT 指令命名风格，采用 AT+CIOT***的方式。

2.3 模组扩展指令所实现的主要功能及分类

模组扩展指令暂时设计为 18 条，后续可扩展更丰富的功能。

- 基本参数查询设置功能
- 平台业务定义功能（利用 HTTP 协议的 RESTful API）
- 平台设备连接功能（利用平台提供的 EDPSDK）
- 平台数据上传功能（利用平台提供的 EDPSDK）
- 平台命令控制功能（利用平台提供的 EDPSDK）

3 M6311 模组 onenet 大众版扩展指令详细说明

3.1 M6311 模组 onenet 大众版扩展指令一览表

WIFI 模组扩展 AT 指令	
命令	描述
AT+CIOT	查询固化参数;
AT+CIOTMKEY	设置并查询主 Apikey;
AT+CIOTINIT	启动参数创建;
AT+CIOTID	查询设备操作命令;
AT+CIOTIDC	删除设备操作命令;
AT+CIOTKEY	查询设备 Apikey 操作命令;
AT+CIOTKEYC	删除设备 Apikey 操作命令;
AT+CIOTSTATUS	查询 WIFI 模组和物联网平台的连接状态;
AT+CIOTSTART	启动向物联网平台数据发送;
AT+CIOTDAT	向物联网平台传送小数据;
AT+CIOTBINSET	设置二进制文件上传的信息;
AT+CIOTBINSEND	发送二进制文件内容;
AT+CIOTBINEND	结束二进制文件;
AT+CIOTQUIT	停止向物联网平台数据传送;
AT+CIOTSEND	单次向物联网平台传送小数据;
AT+CIOTPING	心跳指令;
+IOTCMD	物联网平台发送的控制命令;

3.2 错误码定义

错误码说明	错误码
WLAN 错误	-100
网络错误	-101
没有 master key	-102
没有设备 ID	-103
没有设备 key	-104
设备不在线	-105
无效的操作	-106
文件发送错误	-107
参数错误	-108
设备连接失败	-109
数据发送失败	-110
设备连接断开失败	-111
设备参数初始化失败	-112

3.3 指令格式说明

(1)、格式说明

<>:表示必须包含的部分

[]:表示可选的部分

(2)、命令消息格式

AT+<CMD>[op][para-1,para-2,para-3,.....]<CR>

a) AT+:命令消息前缀

b) CMD:命令字符串

c) [op]:命令操作符，指定是参数设置或查询。可以是：

1、"=":表示参数设置

2、"无":表示参数查询

d) [para-n]:参数设置时的输入，查询命令时为"?"

e) <CR>:结束符，回车，ASCII 码 0x0d

(3)、回应消息格式

+<RSP>=[para-1,para-2,para-3,.....]<CR><LF>

a) +: 响应消息前缀

b) RSP: 相应字符串，包括：

1. "OK": 表示成功

2. "ERR": 表示失败

c) [para-n]:查询时返回的参数，或者出错时的错误码

d) <CR>: ASCII 码 0x0d

e) <LF>: ASCII 码 0x0a

3.3 AT 指令

3.3.1 AT+CIOT

说明：查询 WIFI 模块的主 APIKEY，设备 ID 及设备 APIKEY 是否已经固化存在。

功能：查询参数是否存在	
查询指令格式： AT+CIOT?<CR><LF>	正确响应： +OK<CR><LF> 错误响应分三种情况：

	+ERR=-102<CR><LF> +ERR=-103<CR><LF> +ERR=-104<CR><LF>
	参数：无

注：模块需要固化主 APIKEY，设备 ID 及设备 APIKEY 三个参数，目前 APIKEY 的长度为 28 个字节长度，但后续有扩展的可能；设备 ID 长度也是扩展的。

3.3.2 AT+CIOTMKEY

说明：设置 WIFI 模块的主 APIKEY 参数。如果本地已经存在主 APIKEY，执行此条指令会覆盖已有的主 APIKEY，同时删除原有主 APIKEY 创建的设备 ID 和设备 APIKEY。

功能：设置并固化 master key	
查询指令格式： AT+CIOTMKEY?<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR=-102<CR><LF> 参数：无
指令格式： AT+CIOTMKEY=[mkey]<CR><LF>	正确响应： +OK=[mkey]<CR><LF> 错误响应： +ERR<CR><LF> 参数：mkey 为主 APIKEY

3.3.5 AT+CIOTINIT

说明：AT+CIOTINIT 用来启动设备 ID 和设备 Apikey 参数的创建或者参数的有效性验证，创建参数前，必须已经固化了主 Apikey，WIFI 模块使用主 Apikey 向平台服务器申请设备 ID 的创建，然后再进行设备 Apikey 的创建。参数创建成功后，WIFI 模组都需要将创建成功的参数固化在自身的参数存储区。

如果本地主 Apikey，设备 ID 和设备 Apikey 都存在，则只需要去平台服务器查询设备 ID 和设备 Apikey 是否存在。（平台查询参数是确保平台上对应的设备没有被删除），如果平台查询失败，则必须清除设备 ID 和设备 APIKEY，然后使用主 APIKEY 在平台重新创建设备，获取新的设备 ID 和设备 APIKEY，再进行固化。

功能：启动参数创建	
指令格式： AT+CIOTINIT<CR><LF>	正确响应： +OK<CR><LF>

	错误响应： +ERR=-100<CR><LF> +ERR=-101<CR><LF> +ERR=-102<CR><LF> +ERR=-112<CR><LF>
	参数：无

3.3.6 AT+CIOTID

说明：查询设备 ID

功能：查询设备 ID 是否存在	
查询指令格式： AT+CIOTID?<CR><LF>	正确响应： +OK=[dev_id]<CR><LF> 错误响应： +ERR=-103<CR><LF>
	参数：dev_id

3.3.7 AT+CIOTIDC

说明：删除设备 ID，必须删除对应的设备 APIKEY。

功能：删除设备 ID	
指令格式： AT+CIOTIDC<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR<CR><LF>
	参数：无

3.3.8 AT+CIOTKEY

说明：查询设备 APIKEY

功能：查询设备 apikey	
查询指令格式： AT+CIOTKEY?<CR><LF>	正确响应： +OK=[apikey]<CR><LF> 正确响应： +ERR=-104<CR><LF>
	参数：apikey 设备 APIKEY

3.3.9 AT+CIOTKEYC

说明：删除设备 APIKEY，不必删除设备 ID。

功能：删除设备 APIKEY	
指令格式： AT+CIOTKEYC<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR<CR><LF>
	参数：无

3.3.10 AT+CIOTSTATUS

说明：指令功能是查询设备的连接状态。

功能：查询设备连接状态	
指令格式： AT+CIOTSTATUS<CR><LF>	响应： +OK=state<CR><LF>
	参数：state ◇ device_up, 设备登录 ◇ Device_down, 设备未登录

3.3.11 AT+CIOTSTART

说明：指令功能是建立设备连接，启动数据发送。

功能：建立设备连接，启动数据发送	
指令格式： AT+CIOTSTART=[encrypt]<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR=-108<CR><LF> +ERR=-109<CR><LF>
	参数：encrypt ◇ 0, 不使用加密 ◇ 1, 使用加密

3.3.12 AT+CIOTDAT

说明：指令功能是长传小数据

功能：发送数据	
指令格式：	正确响应：

<p>AT+CIOTDAT=[msg_id,num,data]<CR><LF></p>	<p>+OK<CR><LF></p> <p>错误响应:</p> <p>+ERR=-108<CR><LF></p> <p>+ERR=-110<CR><LF></p> <p>参数:</p> <p>msg_id: 该数据的消息编号, 范围为(0-65535)。msg_id=0时, 服务器端不返回确认信息。msg_id>0时, 服务器返回包含该编号的确认信息</p> <p>num: 数据记录数量</p> <p>data: 数据内容, 其内容格式为:</p> <p style="padding-left: 20px;">key_1,time_1,value_1;key_2, ... Value_n-1;key_n,time_n,value_n</p> <p>key: 数据采样名称</p> <p>time: 数据采样时间</p> <p>value: 数据采样值</p>
<p>说明:</p> <p>时间格式: 如 2016-01-01 00:00:00 年月日用'-'分隔 时分秒用':'分隔</p> <p>如没有采样时间, 则时间项不填, 格式为 key,,value;</p> <p>数据记录内分隔符为逗号 数据记录分隔符为分号</p>	

3.3.13 AT+CIOTBINSET

说明: 设置二进制文件发送的头部信息, 由于数据缓冲大小的限制, 一般在进行二进制文件上传的时候, 先将 EDP 头部传给服务器, 然后将二进制文件数据分片上传的方式进行二进制文件上传。(AT+CIOTBINSET, AT+CIOTBINSEND 及 AT+CIOTBINEND 组合使用)

功能: 设置二进制文件	
<p>指令格式:</p> <p>AT+CIOTDATBINSET=[msg_id,des,bin_lens]<CR><LF></p>	<p>正确响应:</p> <p>+OK<CR><LF></p> <p>错误响应:</p> <p>+ERR=-108<CR><LF></p> <p>参数:</p> <p>msg_id: 该数据的消息编号, 范围为(0-65535)。msg_id=0时, 服务器端不返回确认信息。msg_id>0时, 服务器返回包含该编号的确认信息</p> <p>des: 二进制文件的数据流 ID</p> <p>bin_lens: 文件总长度</p>
<p>说明: 该命令的 msg_id 若被设置, 在连续使用 AT+CIOTBINSEND 发送完成后方返回</p>	

3.3.14 AT+CIOTBINSEND

说明：发送二进制文件的数据，在参数检查的时候，只需要判断当前长度和数据个数相等即可。

功能：发送二进制文件	
指令格式： AT+CIOTBINSEND=[bin_len,bin_data]<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR=-108<CR><LF>
F>	参数： bin_len：当次发送数据长度 bin_data：数据（数据形式为 16 进制格式，如 01AA55 表示数据 0X01，0XAA，0X55）
说明：正确响应+OK 表示 WIFI 将本次的数据网络发送成功。 当连续发送的数据长度已满足 AT+CIOTBINSET 所设置的长度后，该次二进制上传完毕，打印 BIN SEND OVER!	

3.3.15 AT+CIOTBINEND

说明：设置二进制文件发送的结束标记，返回文件发送的状态。

功能：结束二进制文件	
指令格式： AT+CIOTBINEND<CR><LF>	正确响应： +OK=<CR><LF> 错误响应： +ERR=-107<CR><LF>
	参数：无

3.3.16 AT+CIOTQUIT

说明：指令实现的功能为断开设备连接，停止数据发送。

功能：断开设备连接	
指令格式： AT+CIOTQUIT<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR=-111<CR><LF>
	参数：无

注：EDP 协议没有断开设备连接请求的报文，AT+CIOTQUIT 要实现断开设备连接的功能，直接断开和服务器的 TCP 连接，可以达到断开设备连接的目的。

3.3.17 AT+CIOTSEND

说明：发送小数据，功能等同于指令集（AT+CIOTSTART； AT+CIOTDAT； AT+CIOTQUIT）指令

内部执行流程如下：

功能：发送采样数据	
<p>指令格式： AT+CIOTDAT=[encrypt,msg_id,num,data]<CR><LF></p>	<p>正确响应： +OK<CR><LF></p> <p>错误响应： +ERR=-108<CR><LF> +ERR=-109<CR><LF> +ERR=-110<CR><LF> +ERR=-111<CR><LF></p>
	<p>参数：</p> <p>encrypt: 是否加密 ◇ 0, 不加密 ◇ 1, 加密</p> <p>msg_id: 该数据的消息编号，范围为（0-65534）。msg_id=0 时，服务器端不返回确认信息。msg_id>0，服务器返回包含该编号的确认信息。65535 此处不可被设置</p> <p>num: 数据记录数量</p> <p>data: 数据内容，其内容格式为： key_1,time_1,value_1;key_2, ... Value_n-1;key_n,time_n,value_n</p> <p>key: 数据采样名称</p> <p>time: 数据采样时间</p> <p>value: 数据采样值</p>
<p>说明：</p> <ol style="list-style-type: none"> 1、时间格式：如 2016-01-01 00:00:00 年月日用'-'分隔 时分秒用':'分隔 如没有采样时间，则时间项不填，格式为 key,,value; 2、数据记录内分隔符为逗号 数据记录分隔符为分号 3、若在 5 秒钟内数据发送不成功，发送超时，模组断开连接 	

3.3.18 AT+CIOTPING

功能：心跳	
<p>指令格式： AT+CIOTPING<CR><LF></p>	<p>正确响应： +OK PING succ!</p> <p>错误响应： +ERR<CR><LF></p>
	<p>参数：无</p>

3.3.19 +IOTCMD

说明：此指令是 WIFI 模组主动发送给 MCU，当 WIFI 模组接收到物联网平台发送的控制命令时，向串口发送“+IOTCMD=[datalen,data]”其中“+IOTCMD:”仅仅是给 MCU 标识符，表示这是平台下发的控制命令。

平台服务器向设备发送控制命令	
事件格式： +IOTCMD=[datalen,data]<CR><LF>	参数： datalen: 控制命令的数据长度 data: 控制命令内容

3.3.20 AT+CIOTCONNECTPARA

说明：服务器登录参数设置

功能：设置服务器连接参数	
指令格式： AT+CIOTCONNECTPARA=[mode, devid_temp/projectid, apikey_temp/authinfo_key, authinfo_value]<CR><LF>	正确响应： +OK<CR><LF> 错误响应： +ERR=-108<CR><LF> 参数： mode: 连接模式 devid_temp: 设备 ID projectid_temp: 项目 ID apikey_temp: 鉴权信息 (api-key) authinfo_key_temp: 鉴权信息 (auth_info) authinfo_value_temp: 鉴权信息 (auth_info)
<p>说明：该命令可设置、查询当前服务器登录参数</p> <ol style="list-style-type: none">当 mode == 0 时，1 参数，设置命令为： AT+CIOTCONNECTPARA=0 当 mode == 1 时，3 参数，设置命令为： AT+CIOTCONNECTPARA=1,"1088826","syp2wjH2rowlyX1H7bqwynVVnK8="当 mode == 2 时，4 参数，设置命令为： AT+CIOTCONNECTPARA=2,"40440","SYS","DcvGb46zAagazMtLuBcVNnk=12"temp 表示被设置的参数是临时的，重启后不被保存当被设置为 mode 0 时，登录服务器时采用 devid 与 apikey 参数。devid 与 apikey 参数应被初始化，初始化后固化在模组内部。重启不丢失，无需再次初始化（服务器端未作改变或删除的情况下）。设置相关参考 AT+CIOTINIT, AT+CIOTIDC, AT+CIOTMKEYA 命令devid_temp 与 apikey_temp 在 devid 与 apikey 未被初始化的情况下使用。如被初始化，在一般情况下，devid 与 apikey 的值应与后两者一致。如不运行此命令，将默认使用 mode 1	

4 使用流程

4.1 平台端注册项目

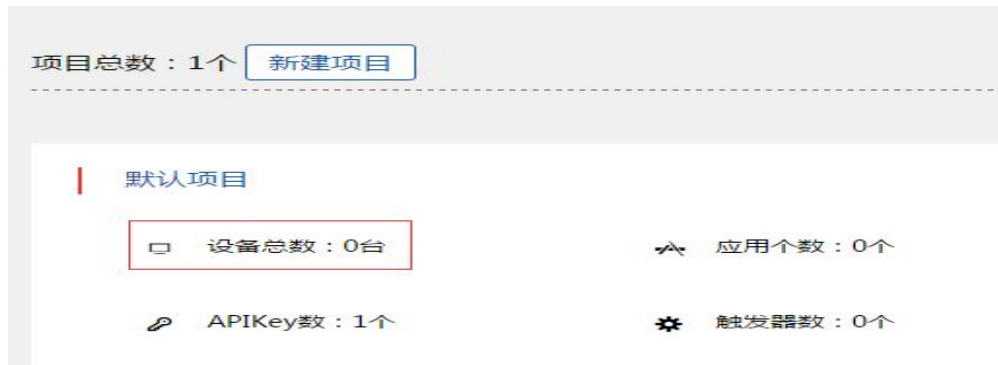
4.1.1、百度搜索 onenet，进入中移物联网 onenet 开放平台，注册用户



4.1.2、注册用户后，点击右上角的项目管理



4.1.3、进入项目管理后，可以直接使用默认项目，也可创建新的项目使用



4.1.4、进入默认项目，获取项目 ID:23219 与项目默认 APIKEY:WRygxraTXCCikfaMnK9S46dfzDYA, 项目 APIKEY 即为其主 APIKEY



4.2 新建设备

4.2.1 方式一：模组端新建设备

- (1) 模组（插入 SIM 卡）上电，正常开机，等待 10 秒
- (2) AT+CIOT? //查询主 APIKEY 存在，返回错误表示未初始化，返回 OK 可跳过 (3)
// (4)、(5)、(6)
- (3) AT+IPSTART=4,"TCP","api.heclouds.com",80 //连接服务器，建立一路 HTTP 连接
- (4) AT+CIOTMKEYA="WRygxraTXCCikfaMnk9S46dfzDYA"
//该参数 **mapikey** 即为 4.1.4 中获取的主 APIKEY
- (5) AT+CIOTINIT //创建设备参数，参考 3.3.5 相关说明
- (6) AT+IPCLOSE=4 //关闭 HTTP 连接。若不执行，无数据交互 60s 后自动断开

说明：唯有此方式创建的设备 ID:dev_id 与设备 APIKEY:apikey（暂不使用）及与之相关的项目 APIKEY: mapikey 被固化,掉电不丢失

4.2.2 方式二：平台端新建设备

4.2.2.1、进入设备管理



4.2.2.2、依次写入设备名称，选择 **EDP 协议**，选定设备位置。可设置鉴权信息写入特定的唯一标识符与设备相关联，如 SIM 卡号等。如不设置，由服务器分配。

所属项目：

设备名称：

设备标签：

设备描述：

添加图片：  [修改默认](#) [本地上传](#)

接入协议：

鉴权信息：

默认路由：

设备权限：

设备位置：

4.2.2.3、进入新建的设备点击设备名，获取设备 ID:1088826(devid_temp),鉴权信息 auth_info: SYS(authinfo_key_temp), DcvGb46zAagazMtLuBcVnNk66tg=(authinfo_value_temp) 由于此方式获得的参数不被固化在模组中，故以 temp 标识。



4.3 模组端上传与接收数据

4.3.1 开机及激活网络

上电 10 秒后执行 AT+CMACCT 命令

说明：相关流程可参考 M6311 手册

4.3.2 设置连接模式及相关参数

方式一：AT+CIOTCONNECTPARA=0

//该方式用于模组端创建设备时

//dev_id 与 mapikey 已固化，故只需输入 mode 参数：0

方式二：AT+CIOTCONNECTPARA=1, devid_temp, apikey_temp

//该方式用在由平台端创建设备时

//参数 devid_temp 为 4.2.2.3 获取的设备 ID

//参数 apikey_temp 为 4.1.4 中获取的主 APIKEY

//此为平台申请设备参数，故不被固化

//此方式为默认设置,用作快速开发测试

示例 AT+CIOTCONNECTPARA=1,"1088826","WRYgxraTXCCikfaMNk9S46dfzDYA"

方式三：AT+CIOTCONNECTPARA=2,projectid,authinfo_key,authinfo_value

//该方式用在由平台端创建设备时

//参数 projectid 为 4.1.4 获取的项目 ID

//参数 authinfo_key 与 authinfo_value 为 4.2.2.3 中获取的鉴权信息

//此平台申请设备参数，故不被固化

//若需根据用户自己的唯一标识符申请对应设备，应采用此方式在 4.2.2.2 中将鉴权信息设为此唯一标识符

示例 AT+CIOTCONNECTPARA=2,"23219","SYS","DcvGb46zAagazMtLuBcVnNk66tg="

4.3.3 短连接单次上传小数据

(1) AT+CIOTSEND=1,361,3,"teet1,,22.5;teet2,,35%;teet3,,89"//加密,响应模式,响应码 361

(2) AT+CIOTSEND=0,0,3,"teet1,,22.5;teet2,,35%;teet3,,89"//非加密,非响应模式

说明：该命令会执行一个完整的数据从建立连接到关闭连接的上传流程

在响应模式下，模组会收到响应消息会打印 SAVE succ!msg:(响应码)

加密模式会耗费更多的流量与时间

4.3.4 长连接循环上传小数据

(1) AT+CIOTSTART=0 //启动数据发送，非加密模式 OR

AT+CIOTSTART=1 //启动数据发送，加密模式

(2) AT+CIOTDAT=0,3,"te,,22.5;hy,,35%;pm,,89" //上传小数据,非响应模式

...

AT+CIOTDAT=65535,3,"te,,10.5;hy,,12%;pm,,9" //上传小数据，响应模式，响应码 65535

(3) AT+CIOTQUIT //停止数据发送，断开连接

说明：在该流程中可持续上传数据，若需保持连接，在不上传数据时应使用 AT+CIOTPING 发送心跳包。否则，若 128s 内无动作，连接将断开

上传成功的数据可登录平台端查询



4.3.5 状态查询与心跳报文

(1) AT+CIOTSTATUS

//查询当前网络状态，返回 CIOTSTATUS:device_down 表明设备未在线

//返回 CIOTSTATUS:device_up 时表明设备在线，可下发命令,且不能建立新的连接

(2) AT+CIOTPING //心跳报文

3084a0f18abcd8ed894da22edd44c3b52c1fb1e8c4eee094c9c5402ee3ae2197a4ed9b869ae14e3
b52505a68e17494217d26f2e8bc5c8e203ae878c37d474743dd6eff989c4e4d04fed5b6687fbad1c
bf42a7ba6278aa1d581ca3013bdd69a6df79568bc48877e5e611c261dbc407ba9cf0dc7cedb5b9a0
6826eed8a6d91a4ba91ab0db37f535db55d2d599b5117aab5b37c90aa8a8da9f52fc65d49050246
adcbba22d813f86edf2e340d867e43bfd4c395d5bb0b6fc778c70a10948fbd77d1c9a014e10e0e8c
fdc9b0e7c746f8fbb3b19cf95d47898189621218d243768e12d66cae737a40408410bb3a6f837e4d
5fbb8fac0923293a5be2f516b256e153403698e672dad3cc350b4d86914375caeb936483886be0b
c669e01830d7c21c65c8a8" //发送二进制文件

(7)AT+CIOTBINSEND=183,"804415ecd0d7061b8ae4cd3294dbc3f738c2cbac711de8d727bb61efdd
bc1efe3361a8a72f3a30a0af906864f70cd484398d46f7bca18359a03837af1be226a684350590d07
0cc6f1d786e040ec5d3ccdc709ddc7ef137e303dd0e221952f036f330c6786c123674ba376b8d5c58
02122abd7d98dda1dd531a9e669b3b3b877d7c9e6d9166a030910f2ff07ad60e8e2f88b609fcac90
c9e9f23cff0031d191d64759124c8eb216f3905d0d721221321916cd7bc933ffd9"

//发送二进制文件

(8) AT+CIOTBINEND //结束二进制文件

(9) AT+CIOTQUIT //停止数据发送，断开连接

发送结束后，登录平台端

数据展示 [添加数据流](#) 05

test

查询时间：

至

获取数据成功

查看最新

