

M5310-A TCP/IP 应用指导

NB-IoT 系列

版 本：V1.0

日 期：2018-06-15



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 © 2018, 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在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	2018.6.15	林紫微	首次创建



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-A 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
2.5 UDP 数据收发格式控制	6
3 TCP 数据收发.....	7
3.1. 创建 TCP SOCKET	7
3.2. 连接远程服务器	7
3.3. 发送 TCP 数据	7
3.4. 接收 TCP 数据	8
3.5. 关闭 TCP	8
3.6. TCP 数据收发格式控制	8
4 注意事项	9

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","CMNBIOT",,0,0,,,,,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 port>[,<receive control>]`

例如:

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

<code>AT+NSOST=0,183.230.40.150,36000,10,30313233343536373839</code>	向183.230.40.150:36000发送10个UDP数据
<code>0,10</code>	0# Socket成功发送10 Bytes UDP数据
<code>OK</code>	
<code>+NSONMI:0,30</code>	0# 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

例如

AT+NSOCL=0

删除socket 0

OK

2.5 UDP 数据收发格式控制

AT+NSOCFG=<socket>,<in_mode>,<out_mode>

以上2.1-2.4节操作均为系统默认配置下例程，当需要直接输入/输出数据而不进行16进制转换时，需要使用上述+NSOCFG命令配置输入输出数据格式；

请注意在字符串模式下，由于 " 字符为AT命令字符串分隔符，当输入双引号时需要使用十六进制转义字符\x22或八进制转义字符\042替换，否则会引起AT命令解析错误；同时字符串模式建议输入数据加双引号，如"data"

例如：

AT+NSOCFG=0

读取当前socket格式控制配置

+NSOCFG:0,1,1

Socket 0 输入模式为HEX，输出模式为HEX（系统默认值）

OK

AT+NSOCFG=0,0,0

配置为输入字符串模式(支持转义字符)，输出原始数据模式

AT+NSOST=0,47.93.217.230,2008,10,30313233343536373839,1

在字符串输入模式下上报数据，此时length字段被忽略

0,20

发送输入的字符串数据30313233343536373839，长度20Bytes

OK

+NSOSTR:0,1,1

已发送成功

AT+NSOST=0,47.93.217.230,2008,,"AbC\r\npfE",1

以字符串模式上报数据（支持转义字符处理），length参数可忽略

0,8

发送AbC\r\npfE，其中\r\n将被转义为0x0D 0x0A，总共8Bytes

OK

+NSOSTR:0,1,1

已发送成功

+NSONMI:1,11

接收到11字节数据

AT+NSORF=0,512

读取数据

+NSORF:1,47.93.217.230,2008,11,Hello CMCC!,0

此时将直接输出原始接收数据，不经过16进制转换

OK

3 TCP 数据收发

3.1. 创建 TCP Socket

`AT+NSOCR=<type>,<protocol>[,<listen port>[,<receive control>,
[<af_type>[,<ipaddr>]]]]`

例如:

<code>AT+NSOCR="STREAM",6,6003,1</code>	创建TCP Socket, 并绑定本地端口6003, 开启数据到达自动上报
	+NSONMI, 需要+NSORF命令读取数据
<code>1</code>	创建成功返回socket编号, 数值0-6, 最多监听7个端口
<code>OK</code>	创建TCP成功
<code>AT+NSOCR="STREAM",6,0,2</code>	创建TCP Socket, 并绑定本地端口6003, 开启自动上报接收数据, 无需通
	过AT+NSORF读取
<code>1</code>	创建成功返回socket编号, 数值0-6, 最多监听7个端口
<code>OK</code>	创建TCP成功

3.2. 连接远程服务器

`AT+NSOCO=<socket>,<remote_addr>,<remote_port>`

例如:

<code>AT+NSOCO=1,47.93.217.230,2000</code>	连接远程地址47.93.217.230,2000
<code>OK</code>	命令解析正确, 开始连接远程服务器
<code>CONNECT OK</code>	连接成功, 如果连接出现错误或者超时, 将返回socket关闭提示+NSOCLI: <socket>, 需要重新开始TCP流程

3.3. 发送 TCP 数据

向目的TCP地址发送数据使用+NSOSD命令, 发送之前需要首先保证与远程服务器的连接成功, AT命令如下,

`AT+NSOSD=,<length>,<data>[,<flag>[,sequence]]`

例如:

<code>AT+NSOSD=1,10,30313233343536373839,,4</code>	//发送数据0123456789,发送序号为4
<code>1,10</code>	1# Socket成功发送10 Bytes TCP数据
<code>OK</code>	

`+NSOSTR:1,4,1`

Socket 1 发送序号为4的TCP包发送成功并收到对方ACK

3.4. 接收 TCP 数据

当接收到TCP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:1,30`

提示第0号Socket接收到30 Bytes数据

`AT+NSORF=1,30`

读取接收到的30 Bytes数据，

`1,47.93.217.230,2000,30,5B3131372E3136392E33362E31353A323137395D30313233343536373839,0`

读取到47.93.217.230:2000发过来的30 Bytes数据

3.5. 关闭 TCP

`AT+NSOCL=<socket>`

<socket>为 3.1 创建时系统分配的 id

`AT+NSOCL=1`

关闭socket 1

OK

3.6. TCP 数据收发格式控制

与 UDP 相同，请参考 2.5 节

4 注意事项

- 由于内部 Socket 数量有限，仅支持 7 个同时工作，且内部释放 Socket 资源需要一定时间，建议在使用 UDP/TCP Socket 做完业务之后再执行关闭操作，避免短时间内重复创建/收发/关闭 Socket，以免 Socket 资源短时间内耗尽导致 +NSOCR 再次创建时失败
- 当连续快速输入待发送数据且速度高于底层发送速度时，可能引起发送缓冲区满而引起发送错误，可通过配置 AT+CME=1 使能错误码显示，此时如果错误码为 159，证明底层发送缓冲区满，请间隔数秒后再次发送数据
- 模组内部可缓存一定接收数据量，当有数据到达且为手动接收模式时，需及时读取数据，否则可能引起后续到达数据丢失；当需要连续接收大量数据时，建议将波特率调高以加快接收速度



