

BC26&BC20

TCP/IP 应用指导

NB-IoT 模块系列

版本: BC26&BC20_TCP/IP_应用指导_V1.1

日期: 2020-05-28

状态: 受控文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://www.quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://www.quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2020，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2020.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2019-07-26	蒋涛	初始版本
1.1	2020-05-28	蒋涛	并入《Quectel_BC26&BC20_TCP(IP)_AT 命令手册_V1.0》文档中有关 TCP/IP AT 命令和 URC 的详情。

目录

文档历史	2
目录	3
表格索引	4
1 引言	5
1.1. TCP/IP AT 命令简述	5
1.2. 数据访问模式说明	5
2 TCP/IP AT 命令详解	6
2.1. AT 命令类型	6
2.1.1. 定义	6
2.1.2. AT 命令语句	6
2.2. AT 命令详解	7
2.2.1. AT+QIOPEN 打开 Socket	7
2.2.2. AT+QICLOSE 关闭 Socket	8
2.2.3. AT+QISTATE 查询 Socket 状态	9
2.2.4. AT+QISEND 发送十六进制/文本字符串数据	10
2.2.5. AT+QIRD 读取 TCP/IP 数据	12
2.2.6. AT+QISENDEX 发送十六进制字符串数据	14
2.2.7. AT+QISWTMD 切换数据访问模式	14
2.2.8. AT+QPING Ping 远程服务器	15
2.2.9. AT+QNTTP 通过 NTP 服务器同步本地时间	16
2.2.10. AT+QIDNSGIP 转换域名为 IP 地址	17
2.2.11. AT+QIDNSCFG 配置 DNS 服务器地址	18
2.2.12. AT+QICFG 配置可选参数	19
2.2.13. AT+QIGETERROR 查询最新一次错误码	22
2.3. URC 详解	22
2.3.1. 连接断开 URC	22
2.3.2. 下行数据已达 URC	23
2.3.3. 下行数据缓存已满 URC	23
3 错误码	25
4 举例	27
4.1. TCP 客户端使用说明	27
4.1.1. 初始化工作	27
4.1.2. 建立 TCP 连接	27
4.1.3. 配置发送的数据格式（配置为文本字符串格式）	28
4.1.4. 配置发送的数据格式（配置为十六进制格式）	28
4.1.5. 配置在数据模式下是否回显	28
4.1.6. 发送文本数据（命令模式）	29
4.1.7. 发送文本数据（定长数据模式）	29
4.1.8. 发送文本数据（不定长数据模式）	30

4.1.9.	发送十六进制数据（命令模式）	30
4.1.10.	发送十六进制数据（定长数据模式）	31
4.1.11.	发送十六进制数据（不定长数据模式）	31
4.1.12.	ACK 响应检查	32
4.1.13.	配置接收的数据格式（配置为文本字符串格式）	32
4.1.14.	配置接收的数据格式（配置为十六进制格式）	32
4.1.15.	配置所接收数据的输出格式（配置为换行格式）	33
4.1.16.	配置所接收数据的输出格式（配置为逗号分隔格式）	33
4.1.17.	所接收数据的存取方式（缓存模式）	34
4.1.18.	所接收数据的存取方式（缓存模式下显示长度）	34
4.1.19.	所接收数据的存取方式（直吐模式）	36
4.1.20.	关闭连接	36
4.2.	TCP 长连接的会话保活说明	37
4.3.	Ping 远程服务器	37
4.4.	同步本地时间	38
4.5.	配置 DNS 服务器	38
4.6.	查询最新一次错误码	38
5	附录 A 术语缩写	39

表格索引

表 1: AT 命令及响应类型 6

表 2: 错误码列表 25

表 3: 术语缩写 39

1 引言

移远通信 BC26 和 BC20 模块均内置 TCP/IP 协议栈，可直接通过 AT 命令访问网络；这大大降低模块对 PPP 以及外部 TCP/IP 协议栈的依赖，从而降低终端设计的成本。

BC26 和 BC20 模块可提供 TCP 客户端、UDP 客户端等套接字（Socket）服务。

1.1. TCP/IP AT 命令简述

通过 TCP/IP AT 命令，模块可以打开/关闭 Socket，并通过 Socket 收发数据。

1.2. 数据访问模式说明

BC26 和 BC20 模块支持如下两种数据访问模式：

- 缓存模式
- 直吐模式

当通过 **AT+QIOPEN** 打开 Socket 时，可以通过 **<access_mode>** 指定数据访问模式。成功打开 Socket 后，可以通过 **AT+QISWTMD** 切换数据访问模式。两种数据访问模式下，均可通过 **AT+QISEND** 或 **AT+QISENDEX** 发送数据。

1. 缓存模式下，模块接收到数据时，会先缓存所接收的数据并上报 **URC +QIURC: "recv",<connectID>[,<current_recv_length>]**，之后可通过 **AT+QIRD** 读取缓存数据。
2. 直吐模式下，模块接收到数据时，会通过上报的 **URC +QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>** 直接输出数据。

备注

缓存模式下，在缓存数据被全部读取完之前，模块不会再上报新的 URC 通知 MCU 收到下行数据。

2 TCP/IP AT 命令详解

本章节主要描述和 TCP/IP 相关的 AT 命令。

2.1. AT 命令类型

2.1.1. 定义

- **<CR>** 回车符。
- **<LF>** 换行符。
- **<...>** 参数名称。实际命令行中不包含尖括号。
- **[...]** 可选参数或 TA 信息响应的可选部分。实际命令行中不包含方括号。若无特别说明，配置命令中的可选参数被省略时，将默认使用其之前已设置的值或其默认值。
- **下划线** 参数的默认设置。

2.1.2. AT 命令语句

前缀 **AT** 或 **at** 必须加在每个命令行的开头。输入**<CR>**将终止命令行。通常，命令后面跟随形式为**<CR><LF><response><CR><LF>**的响应。在本文档中，仅显示响应 **<response>**，省略**<CR><LF>**。

表 1：AT 命令及响应类型

测试命令	AT+<cmd>=?	返回相应设置命令或内部程序可支持的参数取值列表或范围。
查询命令	AT+<cmd>?	返回相应设置命令的当前参数设置值。
设置命令	AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]	设置用户可自定义的参数值。
执行命令	AT+<cmd>	主动执行内部程序实现的功能集。

2.2. AT 命令详解

2.2.1. AT+QIOPEN 打开 Socket

该命令用于打开 Socket 以创建 TCP/UDP 连接；可通过<service_type>指定服务类型，并通过<access_mode>指定数据访问模式。命令执行成功以后，会上报 URC +QIOPEN: <connectID>,<err>通知 Socket 是否成功打开。

AT+QIOPEN 打开 Socket	
测试命令 AT+QIOPEN=?	响应 +QIOPEN: (支持的<contextID>范围),(支持的<connectID>范围),"TCP/UDP",<IP_address>/<domain_name>,<remote_port>,<local_port>,(支持的<access_mode>列表)[,(支持的<protocol_type>列表)] OK
设置命令 AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>[,<access_mode>][,<protocol_type>]]	响应 OK +QIOPEN: <connectID>,<err> 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	该命令立即生效。 "<IP_address>/<domain_name>"、<remote_port>、<local_port>、<access_mode>和<protocol_type>深休眠唤醒后仍有效；参数配置不保存至 NVRAM。

参数

<contextID>	整型。场景 ID。范围：1~3。
<connectID>	整型。Socket ID。范围：0~4。
<service_type>	字符串类型。Socket 服务类型。 "TCP" 客户端创建 TCP 连接 "UDP" 客户端创建 UDP 连接
<IP_address>	字符串类型。远程服务器的 IP 地址，例如：220.18.23.22。最大长度：150 字节。
<domain_name>	字符串类型。远程服务器的域名地址。最大长度：150 字节。
<remote_port>	整型。远程服务器的端口号。范围：1~65535。
<local_port>	整型。本地端口号。 0 自动分配本地端口号

	1~65535	指定的本地端口号
<access_mode>	整型。Socket 的数据访问模式。	
	0	缓存模式
	1	直吐模式
<protocol_type>	整型。互联网协议类型。	
	0	IPv4
	1	IPv6
<err>	整型。错误码，详情请参考第 3 章。	

备注

1. 目前仅支持<contextID>=1。
 2. URC +QIOPEN: <connectID>,<err>上报的超时时间为 60 秒，请在 URC 上报完成后再进行其他操作。
 3. 如果 TCP/UDP 连接创建失败，需执行 AT+QICLOSE=<connectID>以关闭当前 Socket。
 4. 模块开机后，需等 IP 地址 URC（例如：+IP: 10.18.237.42，表明模块注网成功）上报完成后方可执行该命令。
 5. TCP 会话创建后，模块将无法进入深休眠（Deep Sleep）模式。TCP 连接关闭后，模块方可如期进入深休眠模式。
 6. UDP 会话创建后，模块可正常进入深休眠模式，并且唤醒后 UDP 配置信息不会丢失，MCU 可以在唤醒后直接收发数据。

2.2.2. AT+QICLOSE 关闭 Socket

该命令用于关闭指定的 Socket。

AT+QICLOSE 关闭 Socket	
测试命令 AT+QICLOSE=?	响应 +QICLOSE: (支持的<connectID>范围) OK
设置命令 AT+QICLOSE=<connectID>	响应 若成功关闭： OK CLOSE OK 若有任何错误： ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<connectID>	整型。Socket ID。范围：0~4。
-------------	----------------------

2.2.3. AT+QISTATE 查询 Socket 状态

该命令用于查询 Socket 状态。

AT+QISTATE 查询 Socket 状态	
测试命令 AT+QISTATE=?	响应 OK
查询命令 AT+QISTATE?	响应 返回所有现存连接的状态： [+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>] [...] OK 若有任何错误： ERROR
设置命令 若<query_type>为 0，查询指定场景的连接状态 AT+QISTATE=<query_type>,<context ID>	响应 返回指定场景里所有现存连接的状态： [+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>] [...] OK 若有任何错误： ERROR
设置命令 若<query_type>为 1，查询指定 Socket 的连接状态 AT+QISTATE=<query_type>,<connectID>	响应 返回指定 Socket 的连接状态： [+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>] OK 若有任何错误： ERROR

最大响应时间	300 毫秒
特性说明	/

参数

<query_type>	整型。查询类型。 0 查询指定<contextID>的连接状态 1 查询指定<connectID>的连接状态
<contextID>	整型。场景 ID。范围：1~3。
<connectID>	整型。Socket ID。范围：0~4。
<service_type>	字符串类型。Socket 服务类型。 "TCP" 客户端创建 TCP 连接 "UDP" 客户端创建 UDP 连接
<IP_address>	字符串类型。远程客户端的 IP 地址。
<domain_name>	字符串类型。远程服务器的域名地址。
<remote_port>	整型。远程服务器的端口号。
<local_port>	整型。分配的本地端口号。
<socket_state>	整型。Socket 状态。 0 "Initial": 客户端连接尚未建立 1 "Connecting": 客户端正在连接 2 "Connected": 客户端连接已经创建成功 3 "Closing": 客户端连接正在关闭 4 "Remote Closing": 远程服务器正在关闭客户端连接
<access_mode>	整型。数据访问模式。 0 缓存模式 1 直吐模式

备注

1. 目前仅支持<contextID>=1。
2. 若响应中无+QISTATE: 列表，则说明当前没有连接。

2.2.4. AT+QISEND 发送十六进制/文本字符串数据

该命令用于通过指定连接发送十六进制或文本字符串格式的 Socket 数据。

AT+QISEND 发送十六进制/文本字符串数据	
测试命令 AT+QISEND=?	响应 +QISEND: (支持的<connectID>范围),(支持的<send_length>范围),<data>

	<p>OK</p>
<p>设置命令 AT+QISEND=<connectID>,<send_length>,<data></p>	<p>响应 若数据发送成功: OK</p> <p>SEND OK</p> <p>若数据发送失败: OK</p> <p>SEND FAIL</p> <p>若有任何错误: ERROR</p>
<p>设置命令 数据模式下发送不定长数据 AT+QISEND=<connectID></p>	<p>响应 > 响应 >后, 模块将在 500 毫秒内进入数据模式; 之后可直接输入待发数据, 按 Ctrl+Z 发送数据, 按 Esc 取消发送。</p> <p>若数据发送成功: OK</p> <p>SEND OK</p> <p>若数据发送失败: OK</p> <p>SEND FAIL</p> <p>若有任何错误: ERROR</p>
<p>设置命令 数据模式下发送定长数据 AT+QISEND=<connectID>,<send_length></p>	<p>响应 > 响应 >后, 模块将在 500 毫秒内进入数据模式; 之后即可输入长度等于<send_length>的待发数据。</p> <p>若数据发送成功: OK</p> <p>SEND OK</p> <p>若数据发送失败: OK</p>

	SEND FAIL 若有任何错误： ERROR
设置命令 查询已应答和已发送但未应答数据的总长度（包含通过 AT+QISENDEX 和 AT+QISEND 发送的全部数据） AT+QISEND=<connectID>,0	响应 +QISEND: <sent>,<acked>,<nAked> OK 若有任何错误： ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<connectID>	整型。Socket ID。范围：0~4。
<send_length>	整型。待发数据长度。单位：字节。 在非数据模式（命令模式）下，文本格式数据的最大长度 1024 字节，十六进制格式数据的最大长度是 512 字节。 在数据模式下，文本格式数据的最大长度是 1460 字节，十六进制格式数据的最大长度是 730 字节。
<data>	字符串类型。待发送的十六进制或文本格式数据。
<sent>	整型。通过会话已发送数据的总长度。单位：字节。
<acked>	整型。远程服务器已应答数据的总长度；仅 TCP 会话时可用。
<nAked>	整型。已发送但是还未被远程服务器应答的数据总长度；仅 TCP 会话时可用。

备注

1. **SEND OK** 仅表示数据已经发送至协议栈。
 2. **<send_length>**的值和**<data>**的实际长度必须一致。
 3. **<data>**为特殊格式（如 JSON）字符时，必须为带双引号的字符串。目前，**<data>**不支持特殊命令字符，如分号（;）。
 4. MCU 应在收到 **SEND OK** 或者 **SEND FAIL** 响应后再继续发送下一条数据。
 5. 在数据模式下发送数据时，建议返回 **>** 后等待 500 毫秒再发送数据。

2.2.5. AT+QIRD 读取 TCP/IP 数据

该命令用于读取从指定连接接收的 Socket 数据。

在缓存模式下，模块接收到数据后会先缓存数据，并向外部 MCU 上报 URC **+QIURC:** "recv",<connectID>,<current_rcv_length>]以告知接收到新的下行数据。

AT+QIRD 读取 TCP/IP 数据	
测试命令 AT+QIRD=?	响应 +QIRD: (支持的<connectID>范围),(支持的<read_length>范围) OK
设置命令 AT+QIRD=<connectID>,<read_length>	响应 若指定连接收到数据： +QIRD: <actual_read_length>,<remaining_length>] <data> OK 若没有收到数据： +QIRD: 0 OK 若有任何错误： ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<connectID>	整型。Socket ID。范围：0~4。
<read_length>	整型。待读取数据的最大长度。范围 1~512；单位：字节。
<actual_read_length>	整型。已读取数据的长度。单位：字节。
<remaining_length>	整型。剩余数据的长度。单位：字节。
<data>	读取的数据。

备注

- 接收缓存池仍有数据未读取时，模块若再接收到下行数据将不会上报新的 URC；只有当缓存池中所有数据均被读取完成后，才会上报新的 URC。
- 配置 **AT+QICFG="showlength",1**（详情请参考第 2.1.12 章节）后，模块才会在缓存模式下显示该命令返回值中的可选参数<remaining_length>。
- 剩余数据长度并非缓存池中已接收数据的总长度，仅表示某一个节点中保存的当前剩余数据长度。

2.2.6. AT+QISENDEX 发送十六进制字符串数据

该命令用于通过指定连接发送十六进制字符串 Socket 数据。

AT+QISENDEX 发送十六进制字符串数据	
测试命令 AT+QISENDEX=?	响应 +QISENDEX: (支持的<connectID>范围),(支持的<send_length>范围),<hex_string> OK
设置命令 AT+QISENDEX=<connectID>,<send_length>,<hex_string>	响应 若数据发送成功: OK SEND OK 若数据发送失败: OK SEND FAIL 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<connectID>	整型。Socket ID。范围：0~4。
<send_length>	整型。待发数据的长度。最大长度为 512 字节。
<hex_string>	字符串类型。待发送数据。

备注

1. **SEND OK** 仅表示数据已经发送至协议栈。
 2. MCU 应在收到 **SEND OK** 或者 **SEND FAIL** 响应后再继续发送下一条数据。

2.2.7. AT+QISWTMD 切换数据访问模式

打开新 Socket 时，模块可通过 **AT+QIOPEN** 的<access_mode>指定数据访问模式；之后可通过此命令切换数据访问模式。

AT+QISWTMD 切换数据访问模式	
测试命令 AT+QISWTMD=?	响应 +QISWTMD: (支持的<connectID>范围),(支持的<access_mode>列表) OK
设置命令 AT+QISWTMD=<connectID>,<access_mode>	响应 OK 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	该命令立即生效。 深休眠唤醒后仍有效; 参数配置不保存至 NVRAM。

参数

<connectID>	整型。Socket ID。范围: 0~4。
<access_mode>	整型。Socket 的数据访问模式。 0 缓存模式 1 直吐模式

2.2.8. AT+QPING Ping 远程服务器

该命令用于测试主机设备的 IP 地址是否可达。

AT+QPING Ping 远程服务器	
测试命令 AT+QPING=?	响应 +QPING: (支持的<contextID>范围),"<host>"[(支持的<timeout>范围)][(支持的<ping_num>范围)][(支持的<ping_size>范围)]]] OK
设置命令 AT+QPING=<contextID>,"<host>"[,<time_out>[,<ping_num>[,<ping_size>]]]	响应 若远程服务器 Ping 成功: OK +QPING: <result>[,<IP_address>,<bytes>,<time>,<TTL>] [...]

	<p>+QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]</p> <p>若有任何错误: ERROR</p>
最大响应时间	300 毫秒
特性说明	/

参数

<contextID>	整型。场景 ID。范围：1~3。
<host>	字符串类型。远程服务器地址，格式为域名或者 IP 地址。最大长度：150 字节。
<time_out>	整型。每次 Ping 请求后等待响应的最大时长。范围：1~255，默认值：4，单位：秒。
<ping_num>	整型。发送 Ping 请求最大次数。范围：1~10，默认值：4。
<ping_size>	整型。每次 Ping 请求的大小。范围：32~200，默认值：32，单位：字节。
<result>	<p>整型。每次 Ping 请求的结果。</p> <p>0 接收到远程服务器的 Ping 响应</p> <p>其他 错误码；请参考第 3 章获取详细错误码信息</p>
<IP_address>	字符串类型。远程服务器的 IP 地址。
<bytes>	整型。每次发送的 Ping 请求长度。单位：字节。
<time>	整型。发送 Ping 请求花费的时间。单位：毫秒。
<TTL>	整型。Ping 请求的 TTL 值。
<finresult>	<p>整型。Ping 操作的最终结果。</p> <p>0 Ping 成功</p> <p>其他 错误码；请参考第 3 章获取详细错误码信息</p>
<sent>	整型。Ping 请求发送的字节总长度。
<rcvd>	整型。Ping 响应中接收到的字节总长度。
<lost>	整型。Ping 请求中丢失的字节总长度。
<min>	整型。最小响应时间。单位：毫秒。
<max>	整型。最大响应时间。单位：毫秒。
<avg>	整型。平均响应时间。单位：毫秒。

备注

目前仅支持<contextID>=1。

2.2.9. AT+QNTPT 通过 NTP 服务器同步本地时间

该命令通过 NTP 服务器同步本地时间为世界标准时间（UTC）。

AT+QNTTP 通过 NTP 服务器同步本地时间

测试命令 AT+QNTTP=?	响应 +QNTTP: (支持的<contextID>范围),"<server>"[,<port>[, (支持的<auto_set_time>列表)]] OK
设置命令 AT+QNTTP=<contextID>,"<server>"[,<port>[,<auto_set_time>]]	响应 若同步成功: OK +QNTTP: <err>,<time> 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<contextID>	整型。场景 ID。范围：1~3。
<server>	字符串类型。NTP 服务器地址，格式为域名或者点分十进制 IP 地址。最大长度：150 字节。
<port>	整型。NTP 服务器端口号。范围：0~65535，默认值：123。
<auto_set_time>	整型。是否自动同步本地时间为 UTC。 0 不自动同步 1 自动同步
<time>	字符串类型。从 NTP 服务器上同步的时间。 格式为："YY/MM/DD, hh:mm:ss±zz"，各字符分别表示年（YY）、月（MM）、日（DD）、时（hh）、分（mm）、秒（ss）、时区（±zz）。±zz 的范围为-47 ~ +48。
<err>	整型。错误码，详情请参考第 3 章。

备注

1. 目前仅支持<contextID>=1。
2. 当<autosettime>=1，RTC 会自动更新到同步时间；随后可使用 **AT+QCCLK?** 查询更新后的时间。
3. 注网成功后，模块会自动更新 RTC 时间。

2.2.10. AT+QIDNSGIP 转换域名为 IP 地址

该命令用于将指定域名转换为 IP 地址。

AT+QIDNSGIP 转换域名为 IP 地址

测试命令 AT+QIDNSGIP=?	响应 +QIDNSGIP: (支持的<contextID>范围),"<hostname>" OK
设置命令 AT+QIDNSGIP=<contextID>,"<hostname>"	响应 OK +QIURC: "dnsgip",<err>,<IP_count>,<DNS_TTL> [+QIURC: "dnsgip",<hostIPAddr>] [...] 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<contextID>	整型。场景 ID。范围：1~3。
<hostname>	字符串类型。域名。最大长度：150 字节。
<IP_count>	整型。参数<hostname>对应的 IP 地址个数。
<DNS_TTL>	整型。DNS 的 TTL 值。
<hostIPAddr>	字符串类型。<hostname>的 IP 地址。
<err>	整型。错误码，详情请参考第 3 章。

备注

目前仅支持<contextID>=1。

2.2.11. AT+QIDNSCFG 配置 DNS 服务器地址

该命令用于配置主要和次要 DNS 服务器地址。

AT+QIDNSCFG 配置 DNS 服务器地址

测试命令 AT+QIDNSCFG=?	响应 +QIDNSCFG: (支持的<contextID>范围),<pridnsaddr>,<secdnsaddr>
------------------------------	--

	OK
设置命令 配置主要和次要 DNS 服务器地址 AT+QIDNSCFG=<contextID>,<pridnsaddr>[,<secdnsaddr>]	响应 OK 若有任何错误: ERROR
设置命令 DNS 服务器地址成功配置后, 查询主要和次要 DNS 服务器地址 AT+QIDNSCFG=<contextID>	响应 [+QIDNSCFG: <contextID>,<pridnsaddr>[,<secdnsaddr>]] OK 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	该命令立即生效。 深休眠唤醒后仍有效; 参数配置不保存至 NVRAM。

参数

<contextID>	整型。场景 ID。范围: 1~3。
<pridnsaddr>	字符串类型。主要 DNS 服务器的 IP 地址。最大长度: 64 字节。
<secdnsaddr>	字符串类型。次要 DNS 服务器的 IP 地址。最大长度: 64 字节。

备注

1.	目前仅支持 <contextID>=1 。
2.	若网络仅支持 IPv4, 则只能设置 IPv4 DNS 地址; 若网络仅支持 IPv6, 则只能设置 IPv6 DNS 地址。
3.	模块开机后, 需等 IP 地址 URC (例如: +IP: 10.18.237.42 , 表明模块注网成功) 上报完成后方可执行设置命令。
4.	AT+QIDNSCFG=<contextID> 可查询模块默认的 DNS 服务器或者网络下发的 DNS 服务器。

2.2.12. AT+QICFG 配置可选参数

该命令可为 TCP/IP 各项功能配置可选参数。

AT+QICFG 配置可选参数	
测试命令 AT+QICFG=?	响应 +QICFG: "dataformat", (支持的 <send_data_format> 列表), (支持的 <recv_data_format> 列表) +QICFG: "viewmode", (支持的 <view_mode> 列表)

	<p>+QICFG: "showlength",(支持的<show_length_mode>列表)</p> <p>+QICFG: "echomode",(支持的<echo_mode>列表)</p> <p>OK</p>
<p>设置命令</p> <p>查询/设置发送/接收数据的格式</p> <p>AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]</p>	<p>响应</p> <p>若省略可选参数，则查询当前配置：</p> <p>+QICFG: "dataformat",<send_data_format>,<recv_data_format></p> <p>OK</p> <p>若指定可选参数，则设置发送数据格式和接收数据格式：</p> <p>OK</p> <p>若有任何错误：</p> <p>ERROR</p>
<p>设置命令</p> <p>查询/设置已接收数据的输出格式</p> <p>AT+QICFG="viewmode"[,<view_mode>]</p>	<p>响应</p> <p>若省略可选参数，则查询当前配置：</p> <p>+QICFG: "viewmode",<view_mode></p> <p>OK</p> <p>若指定可选参数，则设置已接收数据的输出格式：</p> <p>OK</p> <p>若有任何错误：</p> <p>ERROR</p>
<p>设置命令</p> <p>查询/设置在缓存模式下是否显示可选数据长度参数¹⁾</p> <p>AT+QICFG="showlength"[,<show_length_mode>]</p>	<p>响应</p> <p>若省略可选参数，则查询当前配置：</p> <p>+QICFG: "showlength",<show_length_mode></p> <p>OK</p> <p>若指定可选参数，则设置在缓存模式下是否显示可选数据长度参数：</p> <p>OK</p> <p>若有任何错误：</p> <p>ERROR</p>
<p>设置命令</p> <p>查询/设置在数据模式下是否向 UART 回显输入数据</p> <p>AT+QICFG="echomode"[,<echo_mode>]</p>	<p>响应</p> <p>若省略可选参数，则查询当前配置：</p> <p>+QICFG: "echomode",<echo_mode></p>

de>]	<p>OK</p> <p>若指定可选参数，则设置在数据模式下是否向 UART 回显输入数据：</p> <p>OK</p> <p>若有任何错误：</p> <p>ERROR</p>
最大响应时间	300 毫秒
特性说明	<p>该命令立即生效。</p> <p>1. <send_data_format>、<recv_data_format>、<view_mode>和<show_length_mode>参数配置自动保存至 NVRAM；深休眠唤醒后仍有效。</p> <p>2. <echo_mode>参数配置不保存至 NVRAM；深休眠唤醒后无效。</p>

参数

<send_data_format>	<p>整型。发送数据的格式。</p> <p>0 文本字符串格式</p> <p>1 十六进制格式</p>
<recv_data_format>	<p>整型。接收数据的格式。</p> <p>0 文本字符串格式</p> <p>1 十六进制格式</p>
<view_mode>	<p>整型。已接收数据的输出格式。</p> <p>0 已接收数据输出格式：data header\r\n\data</p> <p>1 已接收数据输出格式：data header,data</p>
<show_length_mode>	<p>整型。在缓存模式下是否显示可选数据长度参数¹⁾。</p> <p>0 不显示</p> <p>1 显示</p>
<echo_mode>	<p>整型。是否在数据模式下向 UART 回显输入数据。</p> <p>0 不回显</p> <p>1 回显</p>

备注

<p>1. ¹⁾ 所述可选数据长度参数包含：</p> <ul style="list-style-type: none">● 下行数据已达 URC +QIURC: "recv",<connectID>[,<current_recv_length>]中的<current_recv_length>;● AT+QIRD 返回值中的<remaining_length>。 <p>2. <echo_mode>仅在数据模式下才有效，且仅 BC26NxR01A07/BC20NxR01A08 及之后的版本中支持该参数。</p>

2.2.13. AT+QIGETERROR 查询最新一次错误码

该命令用于查询最新一次 TCP/IP 命令返回的错误码及其相应描述。

AT+QIGETERROR 查询最新一次错误码	
测试命令 AT+QIGETERROR=?	响应 OK
执行命令 AT+QIGETERROR	响应 +QIGETERROR: <err>,<errcode_description> OK 若有任何错误: ERROR
最大响应时间	300 毫秒
特性说明	/

参数

<errcode_description>	字符串类型。详细的错误码描述，具体请参考第 3 章。
<err>	整型。错误码，详情请参考第 3 章。

2.3. URC 详解

TCP/IP AT 命令的 URC 统一以<CR><LF>+QIURC: <type>[...]<CR><LF>的格式上报。本文中 URC 前后的<CR><LF>均将省略。

备注

1. 当模块在 PSM 模式下，不会上报 URC。
2. 当模块 Modem 在 DRX 或者 eDRX 模式下，URC 上报会有延迟，延迟时间依据寻呼周期而定。
3. 当模块 Modem 在连接状态下，会正常上报 URC。
4. URC 最大长度为 1400 字节。
5. 使用 UDP 直吐模式接收数据时，模块将丢弃总长度大于 1400 字节的 URC。

2.3.1. 连接断开 URC

当 TCP Socket 被远程关闭或者因为网络异常而断开时，模块将上报 URC **+QIURC: "closed",<connectID>**。

连接断开 URC

+QIURC: "closed",<connectID> 表示 Socket 连接被断开

参数

<connectID> 整型。Socket ID。范围：0~4。

2.3.2. 下行数据已达 URC

当模块接收到下行数据时，会上报该 URC：

- 在缓存模式下，URC 格式为+QIURC: "recv",<connectID>[,<current_recv_length>];
- 在直吐模式下，URC 格式为+QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>。

下行数据已达 URC

+QIURC: "recv",<connectID>[,<current_recv_length>] 表示缓存模式下有下行数据

+QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data> 表示直吐模式下有下行数据

参数

<connectID> 整型。Socket ID。范围：0~4。

<current_recv_length> 整型。实际接收到的数据长度。

<data> 接收到的数据。

备注

配置 AT+QICFG="showlength",1 (详情请参考第 2.2.12 章节)后，缓存模式下上报的下行数据已达 URC 才会包含可选参数<current_recv_length>。

2.3.3. 下行数据缓存已满 URC

在缓存模式下，如果已经没有缓存资源可以分配给下行数据，模块会上报该 URC。

下行数据缓存已满 URC

+QIURC: "recv",<connectID>,"buff full" 表示下行数据缓存已满。

参数

<connectID>	整型。Socket ID。范围：0~4。
-------------	----------------------

备注

缓存模式下，数据可分配的最大节点数为 20。

3 错误码

执行 TCP/IP AT 命令后若有返回 **ERROR** 或 **<err>**，则具体的错误码信息可通过 **AT+QIGETERROR** 查询。**AT+QIGETERROR** 仅返回最新一个 TCP/IP AT 命令的错误码。

表 2：错误码列表

错误码	错误码描述	含义
0	Operation successful	操作成功
550	Unknown error	未知原因
551	Operation blocked	操作受阻
552	Invalid parameters	无效参数
553	Memory not enough	内存不足
554	Create socket failed	创建 Socket 失败
555	Operation not supported	操作不支持
556	Socket bind failed	Socket 绑定失败
557	Socket listen failed	Socket 监听失败
558	Socket write failed	Socket 写入失败
559	Socket read failed	Socket 读取失败
560	Socket accept failed	Socket 接受失败
561	Open PDP context failed	打开 PDP 场景失败
562	Close PDP context failed	关闭 PDP 场景失败
563	Socket identity has been used	Socket ID 被占用
564	DNS busy	DNS 繁忙
565	DNS parse failed	NDS 解析失败

566	Socket connection failed	Socket 连接失败
567	Socket has been closed	Socket 已被关闭
568	Operation busy	操作繁忙
569	Operation timeout	操作超时
570	PDP context broken down	PDP 场景发生故障
571	Cancel send	取消发送
572	Operation not allowed	操作不允许
573	APN not configured	未配置 APN
574	Port busy	端口繁忙

4 举例

4.1. TCP 客户端使用说明

4.1.1. 初始化工作

```

AT+IPR=115200           //设置固定波特率为 115200
OK
AT+CPIN?                //查询 USIM 卡的 PIN 码是否已解
+CPIN: READY           //已解
OK
AT+CEREG?               //查询网络注册状态
+CEREG: 0,1            //网络已注册，如果未注册成功，可以多次查询
OK
AT+CGPADDR?             //找网成功后，可通过此命令获取模块 IP 地址
+CGPADDR: 1,10.57.39.119
OK
    
```

备注

请确保 TCP/UDP 操作在模块获取到 IP（+IP: xxx）地址上报之后再行。

4.1.2. 建立 TCP 连接

```

AT+QIOPEN=1,0,"TCP","220.180.239.212",8164,1234,0 //远程地址：220.180.239.212，端口：8164；
                                                         本地端口为 1234，如果将其指定为 0，模块将
                                                         自动分配一个可用的端口号
OK
+QIOPEN: 0,0                                           //建立连接成功

//更多连接信息可通过如下方式进行查询
    
```

```

AT+QISTATE=1,0 //可查询连接状态信息
+QISTATE: 0,"TCP","220.180.239.212",8164,1234,2,1,0 //连接成功

OK
AT+QIGETERROR //查询最新一次错误码信息
+QIGETERROR: 0,operate successfully //建立连接操作成功

OK
    
```

4.1.3. 配置发送的数据格式（配置为文本字符串格式）

```

AT+QICFG="dataformat",0,0 //配置发送的数据格式为文本字符串（<send_data_format>=0）
//此为默认配置（文本字符串）
OK //配置完成后，会立即生效并自动保存至 NVRAM，无须重复配置
AT+QICFG="dataformat" //查询配置是否成功
+QICFG: "dataformat",0,0 //发送的数据格式为文本字符串（<send_data_format>=0）

OK
    
```

4.1.4. 配置发送的数据格式（配置为十六进制格式）

```

AT+QICFG="dataformat",1,0 //配置发送的数据格式为十六进制格式（<send_data_format>=1）
//此配置默认为<send_data_format>=0（文本字符串）
OK //配置完成后，会立即生效并自动保存至 NVRAM，无须重复配置
AT+QICFG="dataformat" //查询配置是否成功
+QICFG: "dataformat",1,0 //发送的数据格式为十六进制格式（<send_data_format>=1）

OK
    
```

4.1.5. 配置在数据模式下是否回显

```

AT+QICFG="echomode",0 //配置在数据模式（data mode）下将输入的数据不回显到 UART 串
//此配置默认为<echo_mode>=1（回显数据模式下输入的数据到
口（<echo_mode>=0）
UART 串口）
OK //配置完成；重启或退出深休眠后需要重新配置
AT+QICFG="echomode" //查询配置结果
+QICFG: "echomode",0 //不回显（<echo_mode>=0）

OK
    
```

4.1.6. 发送文本数据（命令模式）

```

AT+QISEND=0,10,"1234567890" //发送 10 字节数据 1234567890
                                //命令中<send_length>需要和<data>实际长度保持一致
OK

SEND OK

AT+QISEND=0,17,"{"a":"b","b":"b"}" //发送 17 字节数据{"a":"b","b":"b"}
                                //命令中<send_length>需要和<data>实际长度保持一致
                                //发送 JSON 格式等特殊字符数据时，必须使用双引号将数据包
围
OK

SEND OK

```

4.1.7. 发送文本数据（定长数据模式）

```

AT+QISEND=0,10 //发送 10 字节数据
> //进入数据模式，等待数据输入
1234567890 //必须输入 10 字节才会退出数据模式并发送
OK

SEND OK

AT+QISEND=0,17 //发送 17 字节数据
> //进入数据模式，等待数据输入
{"a":"b","b":"b"} //进入数据模式，任何字符都是数据，无须用双引号包围
//必须输入 17 字节才会退出数据模式并发送
OK

SEND OK

AT+QICFG="echomode",0 //在数据模式（data mode）下将输入的数据不回显到 UART 串口
                        （<echo_mode>=0）
OK

AT+QISEND=0,10 //发送 10 字节数据
> //进入数据模式，等待数据输入
//输入 10 字节数据：1234567890；此处不回显 1234567890，数据输入
完成后自动退出数据模式并发送
OK

SEND OK

```

4.1.8. 发送文本数据（不定长数据模式）

```

AT+QISEND=0           //发送不定长字节数据
>                     //进入数据模式，等待数据输入
1234567890             //输入数据后按 Ctrl+Z 会退出数据模式并发送
OK

SEND OK

AT+QISEND=0           //发送不定长字节数据
>                     //进入数据模式，等待数据输入
{"a":"b","b":"b"}     //进入数据模式，任何字符都是数据，无须用双引号包围；输入数据
                        //后按 Ctrl+Z 会退出数据模式并发送
OK

SEND OK

AT+QICFG="echomode",0 //在数据模式（data mode）下将输入的数据不回显到 UART 串口
                        （<echo_mode>=0）
OK

AT+QISEND=0           //发送不定长字节数据
>                     //进入数据模式，等待数据输入
                        //输入 10 字节数据：1234567890；此处不回显 1234567890，数据
                        //输入完成后按 Ctrl+Z 会退出数据模式并发送
OK

SEND OK

```

4.1.9. 发送十六进制数据（命令模式）

```

AT+QISENDEX=0,3,313233 //发送十六进制字符串数据（模块自动将十六进制数据 313233 转换为
                        文本数据 123 并发送到服务端）
                        //注意<hex_string>需要满足十六进制格式，否则转换会失败返回
                        ERROR
                        //注意命令中<send_length>应该等于<hex_string>长度除以 2
OK

SEND OK

AT+QICFG="dataformat" //查询当前配置
+QICFG: "dataformat",1,0 //发送的数据为十六进制格式（<send_data_format>=1）
OK

AT+QISEND=0,3,313233 //发送十六进制字符串数据（模块自动将十六进制数据 313233 转换为
                        文本数据 123 并发送到服务端）
                        //注意<data>需要满足十六进制格式，否则转换会失败返回 ERROR

```



```

//注意命令中<send_length>应该等于<data>长度除以 2
OK
SEND OK

```

4.1.10. 发送十六进制数据（定长数据模式）

```

AT+QICFG="dataformat" //查询当前配置
+QICFG: "dataformat",1,0 //<send_data_format>=1（发送的数据为十六进制格式）

OK
AT+QISEND=0,10 //发送 10 字节十六进制数据
> //进入数据模式，等待数据输入
31323334353637383930 //必须输入 20 字节才会退出数据模式并发送
OK //注意<data>必须为十六进制格式，否则会返回 ERROR
//注意命令中<send_length>应该等于<data>长度除以 2

SEND OK
AT+QICFG="echomode",0 //数据模式（data mode）下输入的数据不回显到 UART 串口
//(<echo_mode>=0)

OK
AT+QISEND=0,5 //发送 5 字节数据
> //进入数据模式，等待数据输入
//输入数据 3132333435；此处不回显 3132333435，数据输入完成后
//自动退出数据模式并发送

OK

SEND OK

```

4.1.11. 发送十六进制数据（不定长数据模式）

```

AT+QICFG="dataformat" //查询当前配置
+QICFG: "dataformat",1,0 //发送的数据为十六进制格式（<send_data_format>=1）

OK
AT+QISEND=0 //发送不定长字节数据
> //进入数据模式，等待数据输入
3132333435 //输入数据后按 Ctrl+Z 会退出数据模式并发送
OK //注意<data>需要满足十六进制格式，否则会报 ERROR

SEND OK

```

4.1.12. ACK 响应检查

```
AT+QISEND=0,0 //<send_length>=0 (查询已发送、已收到 ACK 响应以及未收到 ACK
                响应的数据长度)
+QISEND: 94,94,0 //链路建立连接到断开连接过程中累计发送了 94 字节, 94 个均收到
                  了 ACK 响应, 0 个未响应
                  //连接断开后数据会重置为 0

OK
```

4.1.13. 配置接收的数据格式（配置为文本字符串格式）

```
AT+QICFG="dataformat",0,0 //配置接收的数据格式为文本字符串 (<recv_data_format>=0)
                           //此为默认配置 (文本字符串)
OK                          //配置完成后, 会立即生效并自动保存到 NVRAM, 无须重新配置
AT+QICFG="dataformat"      //查询配置是否成功
+QICFG: "dataformat",0,0    //接收的数据格式为文本字符串 (<recv_data_format>=0)

OK
//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0           //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512              //读取缓存中的数据 (最大 512 字节)
+QIRD: 19
hi, i am tcp server        //读取到了 19 字节数据 (包括空格)

OK
OK
```

4.1.14. 配置接收的数据格式（配置为十六进制格式）

```
AT+QICFG="dataformat",0,1 //配置接收的数据格式为十六进制格式 (<recv_data_format>=1)
                           //此配置默认为<recv_data_format>=0 (文本字符串)
OK                          //配置完成后, 会立即生效并自动保存到 NVRAM, 无须重新配置

AT+QICFG="dataformat"      //查询配置是否成功
+QICFG: "dataformat",0,1    //接收的数据格式为十六进制格式 (<recv_data_format>=1)

OK
//远程服务器向模块发送了"1234567890"消息
+QIURC: "recv",0           //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512              //读取缓存中的数据 (最大 512 字节)
+QIRD: 10
31323334353637383930      //读取到 10 字节文本数据 1234567890 后自动转换成十六进制格式数据
```

OK

4.1.15. 配置所接收数据的输出格式（配置为换行格式）

```

AT+QICFG="viewmode",0 //配置所接收数据的输出格式为：data header\r\ndata (<view_mode>=0)
                        //此为配置默认
OK                      //配置完成后，会立即生效并自动保存到 NVRAM，无须重新配置
AT+QICFG="viewmode"    //查询配置信息
+QICFG: "viewmode",0    //所接收数据的输出格式为 data header\r\ndata (<view_mode>=0)

OK

//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0        //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512           //读取缓存中的数据（最大 512 字节）
+QIRD: 19               //读取到 19 字节数据
hi, i am tcp server      //所读取到的 19 字节数据为： "hi, i am tcp server"

OK
    
```

4.1.16. 配置所接收数据的输出格式（配置为逗号分隔格式）

```

AT+QICFG="viewmode",1 //配置所接收数据的输出格式为：data header,data (<view_mode>=1)
                        //此命令配置默认为<view_mode>=0（data header\r\ndata）
OK                      //配置完成后，会立即生效并自动保存到 NVRAM，无须重新配置
AT+QICFG="viewmode"    //查询配置信息
+QICFG: "viewmode",1    //所接收数据的输出格式为 data header,data (<view_mode>=1)

OK

//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0        //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512           //读取缓存中的数据（最大 512 字节）
+QIRD: 19,hi, i am tcp server //读取到的数据长度和读取到的数据显示在同一行，用逗号分隔；读取到的
                        //19 字节数据为 hi, i am tcp server

OK
    
```

4.1.17. 所接收数据的存取方式（缓存模式）

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8164,1234,0 //<access_mode>=0（缓存模式）
//<access_mode>=0 为默认配置
//缓存模式下，需要用 AT+QIRD=<connectID><read_length>读取数据

OK

+QIOPEN: 0,0 //建立连接成功
AT+QISTATE=1,0 //查询连接状态信息
+QISTATE: 0,"TCP","220.180.239.212",8164,1234,2,1,0 //连接成功，<access_mode>=0（缓存模式）

OK
```

连接建立后，可随时进行切换：

```
AT+QISWTMD=0,0 //更改默认的数据存取方式后，可通过此命令切换回默认的缓存模式
//缓存模式下，需要用 AT+QIRD=<connectID><read_length>读取数据
//配置完成后，将立即生效并自动保存到 NVRAM，无须重新配置

OK //立即生效
```

数据读取方法：

```
//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0 //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512 //读取缓存中的数据（最大 512 字节）
+QIRD: 19 //读取到 19 字节数据
hi, i am tcp server //所读取到的 19 字节数据为：hi, i am tcp server

OK

+QIURC: "recv",0,"buff full" //当缓存队列满了后，会主动上报此 URC，表示队列已满
```

备注

最大缓存队列节点数为 20 个，建议在缓存模式下，及时使用 **AT+QIRD** 命令读取数据，否则容易导致数据丢失。

4.1.18. 所接收数据的存取方式（缓存模式下显示长度）

```
AT+QICFG="showlength",0 //在缓存模式下不显示所接收数据的长度和剩余读取长度
（<show_length_mode>=0）
//此为默认配置

OK //配置完成后，将立即生效并自动保存到 NVRAM，无须重新配置
```

```

AT+QICFG="showlength" //查询配置信息
+QICFG: "showlength",0 //<show_length_mode>=0（不显示所接收数据的长度和剩余读取长度）

OK

//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0 //提示模块已收到远程服务器下发的数据
AT+QIRD=0,512 //读取缓存中的数据（最大 512 字节）
+QIRD: 19 //此处未显示读取剩余长度
hi, i am tcp server //读取到的 19 字节数据为: "hi, i am tcp server"

OK
AT+QICFG="showlength",1 //在缓存模式下显示接收数据长度和剩余读取长度
                           (<show_length_mode>=1)
                           //此配置默认为<show_length_mode>=0（不显示所接收数据的长度和剩
                           余读取长度）
OK //配置完成后，将立即生效并自动保存到 NVRAM，无须重新配置
AT+QICFG="showlength" //查询配置信息
+QICFG: "showlength",1 //<show_length_mode>=1（显示接收数据长度和剩余读取长度）

OK

//远程服务器向模块发送了"hi, i am tcp server"消息
+QIURC: "recv",0,19 //提示模块已收到远程服务器下发的数据，所接收数据长度为 19 字节
AT+QIRD=0,10 //读取缓存中的数据（一次最大读取 10 字节）
+QIRD: 10,9 //此处显示已经读取到 10 字节数据，还剩余 9 字节未读取
hi, i am t //所读取的 10 字节数据为 hi, i am t

OK
AT+QIRD=0,10
+QIRD: 9,0 //此处显示已经读取到剩余的 9 字节，全部数据已经读完
cp server //所读取的 9 字节数据为 cp server

OK
AT+QIRD=0,10
+QIRD: 0 //此处显示无数据未读取

OK

```

4.1.19. 所接收数据的存取方式（直吐模式）

建立连接配置：

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8164,1234,1 //<access_mode>=1（直吐模式）
//默认配置为<access_mode>=0（缓存模式）
//后续收到数据时，直接通过+QIURC: "recv"将数据推送出来

OK

+QIOPEN: 0,0 //建立连接成功
AT+QISTATE=1,0 //查询连接状态信息
+QISTATE: 0,"TCP","220.180.239.212",8164,1234,2,1,1 //连接成功，<access_mode>=1（直吐模式）

OK
```

连接建立后，可随时进行切换：

```
AT+QISWTMD=0,1 //切换所接收数据的存取方式为直吐模式
//后续收到数据时，直接通过+QIURC: "recv"将数据推送出来

OK //立即生效
```

数据读取方法：无须主动使用命令读取，直接以 URC 形式显示

```
+QIURC: "recv",0,19 //提示模块已收到远程服务器下发的数据，所接收数据长度为 19 字节
hi, i am tcp server //自动推送了所接收到的 19 字节数据 hi, i am tcp server

OK
```

4.1.20. 关闭连接

```
AT+QICLOSE=0 //主动断开

OK

CLOSE OK //断开连接成功
```

备注

考虑网络异常等因素，建议关闭连接前，先进行发送数据的 ACK 响应检查，检查无异后再关闭连接，如果存在<nAked>，关闭后重新建立链路重发未 ACK 的数据。

4.2. TCP 长连接的会话保活说明

关于多长时间没有数据交互，就会释放资源，目前来说，没有一个明确的值，目前测试情况是 10 分钟内，基本就可以保持端口资源不会被释放。如果保持长连接，建议每隔一定时间（10 分钟）以内就向服务器发送一个短数据包，通过这个方法可以进行会话保活。

基于此，如果很长时间不用发送数据，在重新发数据时可以通过如下方法进行会话校验：

```

AT+QISEND=0,3,"ACK"           //先发一个短包（发送的数据可以根据需求自由定制）
OK

SEND OK

AT+QISEND=0,0                   //再发送一个校验包
+QISEND: 3,0,3                 //发送的心跳字节，均未得到服务器的响应

SEND OK

//等待 30 秒后

+QIURC: "closed",0             //会收到提示连接已断开的 URC，此时会话已经被关闭，需要重连

//如果一直未收到提示连接已断开的 URC，需要等待 120 秒后
AT+QISEND=0,0                   //再发送一个校验包
+QISEND: 3,0,3                 //发送的心跳字节，均未得到服务器的响应

SEND OK

//如果此时仍未收到服务器响应，需要主动关闭连接并重连
AT+QICLOSE=0                     //主动断开
OK

CLOSE OK                       //断开连接成功
    
```

4.3. Ping 远程服务器

```

AT+QPING=1,"iot.quectel.com"
OK

+QPING: 0,"47.100.63.174",32,560,88

+QPING: 0,"47.100.63.174",32,220,88
    
```

```
+QPING: 0,"47.100.63.174",32,230,88
+QPING: 0,"47.100.63.174",32,280,88
+QPING: 0,4,4,0,220,560,322
```

4.4. 同步本地时间

```
AT+QNTTP=1,"ntp5.aliyun.com" //使用域名为 ntp5.aliyun.com 的 NTP 服务器同步本地时间。
OK
+QNTTP: 0,"19/06/11,11:08:20+32"
```

4.5. 配置 DNS 服务器

```
AT+QIDNSCFG=1,"218.2.2.2","8.8.8.8"
OK
AT+QIDNSCFG=1
+QIDNSCFG: 1,"218.2.2.2","8.8.8.8"
OK
```

4.6. 查询最新一次错误码

```
AT+QIOPEN=1,"UDP","220.180.239.212",8063,0,1 //打开任意一个 Socket。
ERROR
AT+QIGETERROR
+QIGETERROR: 552,invalid parameters
OK
```


5 附录 A 术语缩写

表 3：术语缩写

术语	英文全称	中文全称
ACK	Acknowledge character	确认字符
APN	Access Point Name	接入点名称
eDRX	extended Discontinuous Reception	扩展不连续接收
DNS	Domain Name System	域名系统
DRX	Discontinuous Reception	非连续接收
ID	Identifier	标识符
IP	Internet Protocol	互联网协议
IPv4	Internet Protocol version 4	互联网协议版本 4
IPv6	Internet Protocol version 6	互联网协议版本 6
JSON	JavaScript Object Notation	JS 对象简谱
NVRAM	Non-Volatile Random Access Memory	非易失性随机访问存储器
MCU	Micro Controller Unit	微控制器
ME	Mobile Equipment	移动设备
NB-IoT	Narrow Band Internet of Things	窄带物联网
NTP	Network Time Protocol	网络时间协议
PDP	Packet Data Protocol	分组数据协议
PIN	Personal Identification Number	个人身份识别码
PPP	Point to Point Protocol	点对点协议
PSM	Power Saving Mode	省电模式

RTC	Real Time Clock	实时时钟
TCP	Transmission Control Protocol	传输控制协议
TTL	Time to Live	保活时间
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
UDP	User Datagram Protocol	用户数据报协议
URC	Unsolicited Result Code	非请求结果码
USIM	Universal Subscriber Identity Module	全球用户识别卡
UTC	Universal Time Coordinated	世界标准时间