

# BC260Y-CN

# TCP/IP 应用指导

**NB-IoT 模块系列**

版本：1.1

日期：2021-04-13

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](mailto: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](mailto: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。因未能遵守有关操作或设计规范而造成的损害，上海移远通信技术股份有限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。

## 免责声明

上海移远通信技术股份有限公司尽力确保开发中功能的完整性、准确性、及时性或效用，但不排除上述功能错误或遗漏的可能。除非其他有效协议另有规定，否则上海移远通信技术股份有限公司对开发中功能的使用不做任何暗示或明示的保证。在适用法律允许的最大范围内，上海移远通信技术股份有限公司不对任何因使用开发中功能而遭受的损失或损害承担责任，无论此类损失或损害是否可以预见。

## 保密义务

除非上海移远通信技术股份有限公司特别授权，否则我司所提供文档和信息的接收方须对接收的文档和信息保密，不得将其用于除本项目的实施与开展以外的任何其他目的。未经上海移远通信技术股份有限公司书面同意，不得获取、使用或向第三方泄露我司所提供的文档和信息。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，上海移远通信技术股份有限公司有权追究法律责任。

## 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 2021，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2021.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
1.0	2020-09-16	Taber JIANG	初始版本
1.1	2021-04-13	Randy LI/ Albert ZHANG	<ol style="list-style-type: none"> <li>1. 增加 AT+QISEND 以数据模式发送数据，支持定长和不定长发送数据（第 2.3.5 章）。</li> <li>2. 增加 AT+QPING 命令参数&lt;rai_mode&gt;，支持 ping 包业务的快速释放（第 2.3.8 章）；</li> </ol>

# 目录

文档历史.....	2
目录.....	3
表格索引.....	5
<b>1 引言.....</b>	<b>6</b>
<b>2 TCP/IP AT 命令详解.....</b>	<b>7</b>
2.1. AT 命令语句.....	7
2.1.1. 定义.....	7
2.1.2. AT 命令语句.....	7
2.2. AT 示例声明.....	8
2.3. AT 命令详解.....	8
2.3.1. AT+QICFG 配置可选参数.....	8
2.3.2. AT+QIOPEN 打开 Socket.....	9
2.3.3. AT+QICLOSE 关闭 Socket.....	10
2.3.4. AT+QISTATE 查询 Socket 状态.....	11
2.3.5. AT+QISEND 发送十六进制/文本字符串数据.....	13
2.3.6. AT+QIDNSGIP 转换域名为 IP 地址.....	15
2.3.7. AT+QIDNSCFG 配置 DNS 服务器地址.....	16
2.3.8. AT+QPING 远程服务器 Ping 操作.....	17
2.3.9. AT+QNTPT 通过 NTP 服务器同步本地时间.....	18
2.4. URC 详解.....	20
2.4.1. 连接断开 URC.....	20
2.4.2. 下行数据已达 URC.....	20
<b>3 结果码.....</b>	<b>21</b>
<b>4 举例.....</b>	<b>23</b>
4.1. 初始化工作.....	23
4.2. TCP 客户端使用说明.....	23
4.2.1. 建立 TCP 连接.....	23
4.2.2. 配置发送的数据格式为文本字符串.....	24
4.2.3. 配置发送的数据格式为十六进制格式.....	24
4.2.4. 发送文本数据.....	24
4.2.5. 发送十六进制数据.....	25
4.2.6. ACK 响应检查.....	25
4.2.7. 配置接收的数据格式为文本字符串.....	25
4.2.8. 配置接收的数据格式为十六进制格式.....	25
4.2.9. 关闭 TCP 连接.....	26
4.2.10. TCP 长连接的会话保活说明.....	26
4.3. UDP 客户端使用说明.....	27
4.3.1. 建立 UDP 连接.....	27
4.3.2. 配置发送的数据格式为文本字符串.....	27

4.3.3.	配置发送的数据格式为十六进制格式 .....	28
4.3.4.	发送文本数据 .....	28
4.3.5.	发送十六进制数据 .....	28
4.3.6.	配置接收的数据格式为文本字符串 .....	29
4.3.7.	配置接收的数据格式为十六进制格式 .....	29
4.3.8.	关闭 UDP 连接 .....	29
4.3.9.	UDP 连接的深休眠唤醒数据收发 .....	30
4.4.	转换域名为 IP 地址 .....	31
4.5.	配置 DNS 服务器地址 .....	31
4.6.	远程服务器 Ping 操作 .....	31
4.7.	通过 NTP 服务器同步本地时间 .....	31
5	附录 A 参考文档及术语缩写 .....	33

表格索引

表 1: AT 命令及响应类型..... 7

表 2: 结果码列表 ..... 21

表 3: 参考文档..... 33

表 4: 术语缩写..... 33

# 1 引言

移远通信 BC260Y-CN 模块内置 TCP/IP 协议栈，使终端设备可以直接通过 AT 命令访问网络。因此，在很大程度上降低了终端设备对 PPP 协议和 TCP/IP 协议栈的依赖性，从而降低成本。

BC260Y-CN 模块可提供 TCP 客户端、UDP 客户端等 Socket（套接字）服务。

## 2 TCP/IP AT 命令详解

本章节主要描述和 TCP/IP 相关的 AT 命令。通过 TCP/IP AT 命令，可以配置相关参数、打开/关闭 Socket，并通过 Socket 接收和发送数据。

### 2.1. AT 命令语句

#### 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。在本文档中表现命令和响应的表格中，省略了 **<CR><LF>**，仅显示命令和响应。

表 1：AT 命令及响应类型

AT 命令类型	语句	描述
测试命令	<b>AT+&lt;cmd&gt;=?</b>	测试是否存在相应的设置命令，并返回有关其参数的类型、值或范围的信息。
查询命令	<b>AT+&lt;cmd&gt;?</b>	查询相应设置命令的当前参数值。
设置命令	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	设置用户可定义的参数值。
执行命令	<b>AT+&lt;cmd&gt;</b>	返回特定的参数信息或执行特定的操作。



2.2. AT 示例声明

本文中的示例仅为方便用户了解 AT 命令的使用方法，不构成移远通信对终端流程设计的建议或意见，也不代表模块应被设置成相应示例中的状态。某些 AT 命令存在多个示例，这些示例之间不存在承接关系或连续性。

2.3. AT 命令详解

2.3.1. 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)  OK
设置命令 AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]	响应 若省略可选参数，则查询当前配置： +QICFG: "dataformat",<send_data_format>,<recv_data_format>  OK  若指定可选参数，设置发送数据格式和接收数据格式： OK  若有任何错误： ERROR
最大响应时间	5 秒
特性说明	该命令立即生效。 深休眠唤醒后仍有效；自动保存至 NVRAM。

参数

<send_data_format>	整型。发送数据的格式。 0 文本字符串格式 1 十六进制格式
<recv_data_format>	整型。接收数据的格式。

0	文本字符串格式
1	十六进制格式

2.3.2. AT+QIOPEN 打开 Socket

该命令用于打开 Socket 以创建 TCP/UDP 连接，可通过参数<service\_type>指定服务类型，并通过参数<access\_mode>指定数据访问模式。命令执行成功以后，模块会上报 URC +QIOPEN: <connectID>, <result>通知是否成功打开 Socket。

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",<host>,<remote_port>,<local_port>,(支持的<access_mode>列表)  OK
设置命令 AT+QIOPEN=<contextID>,<connectID>,<service_type>,<host>,<remote_port>,<local_port>,<access_mode>]]	响应 OK  +QIOPEN: <connectID>,<result>  若有任何错误: ERROR
最大响应时间	5 秒
特性说明	该命令立即生效。 参数配置在 UDP 下深休眠唤醒后有效；在 TCP 下深休眠唤醒后无效。不保存至 NVRAM。

参数

<contextID>	整型。上下文 ID。范围：0~11。目前仅支持<contextID>=0。 0 自动适配当前默认的 PDP 上下文 1~11 指定特定的 PDP 上下文
<connectID>	整型。Socket ID。范围：0~4。
<service_type>	字符串型。Socket 服务类型。 "TCP" 客户端创建 TCP 连接 "UDP" 客户端创建 UDP 连接
<host>	字符串型。远程服务器地址，格式为带双引号（" "）的域名或者 IP 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，建议配置 5 位以上的端口且不使用特殊协议默认的端口
<access_mode>	整型。Socket 的数据访问模式。	
	1	直吐模式
<result>	整型。结果码，详情请参考第 3 章。	

## 备注

1. 模块开机后，请参考文档 [1] 检查是否成功入网后再执行 **AT+QIOPEN** 命令。
2. **URC +QIOPEN: <connectID>,<result>** 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** 关闭当前 Socket，然后再次尝试打开 Socket 以创建 TCP/UDP 连接。
4. 如果使用<local\_port>指定了本地端口号，当执行 **AT+QICLOSE** 命令后，本地端口号需要释放，才能确保下一次正常使用本地端口号。因此，建议在关闭 Socket 后等待 70 秒再重新执行 **AT+QIOPEN** 命令。如果在回收端口之前使用 **AT+QIOPEN** 创建新连接，则会导致连接创建失败。
5. Socket 的数据访问模式目前仅支持直吐模式，暂不支持缓存模式。
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 可以在唤醒恢复完成后直接收发数据。在 TCP 模式下，当模块从深休眠状态唤醒时无法保存 TCP 连接，需通过 **AT+QIOPEN** 重新开启 Socket。

## 2.3.3. 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 <b>AT+QICLOSE=?</b>	响应 <b>+QICLOSE: (支持的&lt;connectID&gt;范围)</b>  <b>OK</b>
设置命令 <b>AT+QICLOSE=&lt;connectID&gt;</b>	响应 若成功关闭： <b>OK</b>  <b>CLOSE OK</b>  若有任何错误： <b>ERROR</b>
最大响应时间	5 秒
特性说明	/

参数

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

2.3.4. 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 +QISTATE: (支持的<query_type>范围),(支持的<connectID>范围)  OK
查询命令 AT+QISTATE?	响应 返回所有现存连接的状态: [+QISTATE: <connectID>,<service_type>,<host>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>] [...]  OK  若有任何错误: ERROR
设置命令 若<query_type>=0, 查询指定上下文的连接状态 AT+QISTATE=<query_type>,<contextID>	响应 返回指定上下文里所有现存连接的状态: [+QISTATE: <connectID>,<service_type>,<host>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>] [...]  OK  若有任何错误: ERROR
设置命令 若<query_type>=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>,<host>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>]  OK

	若有任何错误： <b>ERROR</b>
最大响应时间	5 秒
特性说明	/

### 参数

<b>&lt;query_type&gt;</b>	整型。查询类型。 0 查询指定 <b>&lt;contextID&gt;</b> 的连接状态 1 查询指定 <b>&lt;connectID&gt;</b> 的连接状态
<b>&lt;contextID&gt;</b>	整型。上下文 ID。范围：0~11。目前仅支持 <b>&lt;contextID&gt;=0</b> 。 0 自动适配当前默认的 PDP 上下文 1~11 用于指定特定的 PDP 上下文
<b>&lt;connectID&gt;</b>	整型。Socket ID。范围：0~4。
<b>&lt;service_type&gt;</b>	字符串型。Socket 服务类型。 "TCP" 客户端创建 TCP 连接 "UDP" 客户端创建 UDP 连接
<b>&lt;host&gt;</b>	字符串型。远程服务器地址，格式为带双引号（" "）的域名或者 IP 地址。最大长度：150 字节。
<b>&lt;remote_port&gt;</b>	整型。远程服务器的端口号。范围：1~65535。
<b>&lt;local_port&gt;</b>	整型。分配的本地端口号。 0 自动分配本地端口号 1~65535 指定的本地端口号，建议配置 5 位以上的端口且不使用特殊协议默认的端口
<b>&lt;socket_state&gt;</b>	整型。Socket 状态。 0 "Initial"（初始状态） 1 "Connecting"（正在连接） 2 "Connected"（已连接） 3 "Closing"（正在关闭） 4 "Remote Closing"（正在远程关闭）
<b>&lt;access_mode&gt;</b>	整型。Socket 的数据访问模式。 1 直吐模式

### 备注

- 若响应中无 **+QISTATE:** 列表，则说明当前未建立 TCP/UDP 连接，请使用 **AT+QIOPEN** 建立 TCP/UDP 连接。
- Socket 的数据访问模式目前仅支持直吐模式，暂不支持缓存模式。

### 2.3.5. 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 <b>AT+QISEND=?</b>	响应 <b>+QISEND: (支持的&lt;connectID&gt;范围),(支持的&lt;send_length&gt;范围),&lt;data&gt;,&lt;rai_mode&gt;</b>  <b>OK</b>
设置命令 命令模式发送数据 <b>AT+QISEND=&lt;connectID&gt;,&lt;send_length&gt;,&lt;data&gt;[,&lt;rai_mode&gt;]</b>	响应 若连接建立成功并且数据成功发送到协议栈: <b>OK</b>  <b>SEND OK</b>  若命令执行成功但数据发送失败: <b>OK</b>  <b>SEND FAIL</b>  若有任何错误: <b>ERROR</b>
设置命令 数据模式下发送不定长数据 <b>AT+QISEND=&lt;connectID&gt;</b>	响应 <b>&gt;</b> 响应 <b>&gt;</b> 后, 模块进入数据模式, 输入数据并键入 <b>Ctrl + Z</b> 发送数据; 当输入的数据长度达到最大长度 (1024 字节) 时将自动发送数据; 或键入 <b>Esc</b> 取消发送。  若连接建立成功并且数据成功发送到协议栈: <b>OK</b>  <b>SEND OK</b>  若命令执行成功但数据发送失败: <b>OK</b>  <b>SEND FAIL</b>  若有任何错误: <b>ERROR</b>
设置命令 数据模式下发送定长数据 <b>AT+QISEND=&lt;connectID&gt;,&lt;send_len</b>	响应 <b>&gt;</b> 响应 <b>&gt;</b> 后, 模块进入数据模式, 之后可输入长度等于

gth>	<p>&lt;send_length&gt;的待发数据。</p> <p>若连接建立成功并且数据成功发送到协议栈： <b>OK</b></p> <p><b>SEND OK</b></p> <p>若命令执行成功但数据发送失败： <b>OK</b></p> <p><b>SEND FAIL</b></p> <p>若有任何错误： <b>ERROR</b></p>
<p>设置命令</p> <p>查询已发送、已应答，以及已发送但未应答数据的总长度</p> <p><b>AT+QISEND=&lt;connectID&gt;,0</b></p>	<p>响应</p> <p><b>+QISEND: &lt;sent&gt;,&lt;acked&gt;,&lt;nAked&gt;</b></p> <p><b>OK</b></p> <p>若有任何错误： <b>ERROR</b></p>
最大响应时间	5 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。范围：0~1024；单位：字节。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。
<data>	字符串型。待发送的十六进制或文本格式数据（数据格式由 <b>AT+QICFG</b> 配置）。
<rai_mode>	<p>整型。消息传输携带的快速释放标记 RAI，该标记用于指示核心网如何释放与模块的 RRC 连接。范围：0~2。</p> <p><b>0</b> 无标记，不使用释放辅助指示</p> <p><b>1</b> 该包数据上行后不期望有进一步的上行或下行数据，核心网可立即释放连接</p> <p><b>2</b> 该包数据上行后期望有对应的单个下行数据包回复，核心网在回复后立即释放连接</p>
<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 会话时有效。单位：字节。
<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 会话时有效。单位：字节。

备注

1. **SEND OK** 仅表示数据已经发送至协议栈。

2. 参数<send\_length>的值和<data>的长度必须一致。即：发送文本字符串数据时，<send\_length>的值等于<data>的实际长度；发送十六进制格式数据时，<send\_length>的值等于<data>实际长度的二分之一。

3. MCU 应在收到 **SEND OK** 或者 **SEND FAIL** 响应后再继续发送下一条数据。

4. <data>为特殊格式（如 JSON 格式）字符时，必须为带双引号的字符串。

2.3.6. 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  +QIDNSGIP: <result>[,<IP_count>,<DNS_ttl>] [+QIDNSGIP: <hostIPaddr>] [...]  若有任何错误： ERROR
最大响应时间	5 秒
特性说明	/

参数

<contextID>	整型。上下文 ID。范围：0~11。目前仅支持<contextID>=0。 0            自动适配当前默认的 PDP 上下文 1~11        用于指定特定的 PDP 上下文
<hostname>	字符串型。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。当前固定为 0。
<hostIPaddr>	字符串型。<hostname>对应的 IP 地址。
<result>	整型。结果码，详情请参考第 3 章。



备注

当前仅支持 DNS 服务器返回的首个 IP 地址。

2.3.7. 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_IPv4>,<secdnsaddr_IPv4>,<pridnsaddr_IPv6>,<secdnsaddr_IPv6>]  OK  若有任何错误: ERROR
最大响应时间	5 秒
特性说明	/

参数

<contextID>	整型。上下文 ID。范围：0~11。目前仅支持<contextID>=0。 0 自动适配当前默认 PDP 上下文 1~11 用于指定特定的 PDP 上下文
<pridnsaddr>	字符串型。主要 DNS 服务器的 IP 地址。最大长度：64 字节。
<secdnsaddr>	字符串型。次要 DNS 服务器的 IP 地址。最大长度：64 字节。
<pridnsaddr_IPv4>	字符串型。IP 格式的 IPv4 主要 DNS 服务器地址。
<secdnsaddr_IPv4>	字符串型。IP 格式的 IPv4 次要 DNS 服务器地址。
<pridnsaddr_IPv6>	字符串型。IP 格式的 IPv6 主要 DNS 服务器地址。
<secdnsaddr_IPv6>	字符串型。IP 格式的 IPv6 次要 DNS 服务器地址。

备注

1.

网络类型取决于模块端和网络侧的各自的配置。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。
2.

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。
3.

执行 **AT+QIDNSCFG=<contextID>** 命令可以查询模块默认的 DNS 服务器地址或者网络下发的 DNS 服务器地址。

2.3.8. AT+QPING 远程服务器 Ping 操作

该命令用于测试主机设备的 IP 地址是否可达。

AT+QPING 远程服务器 Ping 操作	
测试命令 AT+QPING=?	响应 <b>+QPING:</b> (支持的<contextID>范围),<host>,(支持的<time out>范围),(支持的<ping_num>列表),(支持的<ping_size>列表),(支持的<rai_mode>列表)  OK
设置命令 AT+QPING=<contextID>,<host>[,<time _out>[,<ping_num>[,<ping_size>[,<rai _mode>]]]]	响应 若远程服务器 Ping 成功: OK  <b>+QPING:</b> <result>[,<IP_address>,<bytes>,<time>,<ttl>]  [...]  <b>+QPING:</b> <finresult>[,<sent>,<rcvd>,<lost>,<min>,<ma x>,<avg>]  若有任何错误: ERROR
最大响应时间	5 秒
特性说明	/

参数

<contextID>	整型。上下文 ID。范围：0~11。目前仅支持<contextID>=0。 0 自动适配当前默认的 PDP 上下文 1~11 用于指定特定的 PDP 上下文
<host>	字符串型。远程服务器地址，格式为带双引号（" "）的域名或者 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：32~1500；默认值：32；单位：字节。
<rai_mode>	整型。是否使能 RAI 标识。 0 无 RAI 标识 1 使能 RAI 标识（在 ping 的最后一包携带 RAI flag 2）
<result>	整型。结果码。每次 Ping 请求的结果。 0 接收到远程服务器的 Ping 响应 其他 错误码；请参考第 3 章 获取详细结果码信息
<IP_address>	字符串型。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>	整型。结果码。Ping 操作的最终结果。 0 Ping 成功 其他 错误码；请参考第 3 章 获取详细结果码信息
<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

相对于 BC260YCNAAR01A02 及之前的版本，BC260YCNAAR01A03 做出如下优化：	
1)	AT+QPING 响应中增加 TTL 值；
2)	AT+QPING 的<ping_size>参数的最大值由 200 字节扩展到 1500 字节。

2.3.9. 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+QNTPT 通过 NTP 服务器同步本地时间	
测试命令 AT+QNTPT=?	响应 +QNTPT: (支持的<contextID>范围),<server>,<port>,(支持的<auto_set_time>列表)  OK
设置命令 AT+QNTPT=<contextID>,<server>[,<p	响应 若同步成功:

ort>[,<auto_set_time>]]	<p>OK</p> <p>+QNTP: &lt;result&gt;,&lt;time&gt;</p> <p>若同步失败:</p> <p>OK</p> <p>+QNTP: &lt;result&gt;</p> <p>若有任何错误:</p> <p>ERROR</p>
最大响应时间	5 秒
特性说明	/

参数

<contextID>	<p>整型。上下文 ID。范围：0~11。目前仅支持&lt;contextID&gt;=0。</p> <p>0        自动适配当前默认的 PDP 上下文</p> <p>1~11    用于指定特定的 PDP 上下文</p>
<server>	<p>字符串型。NTP 服务器地址，格式为带双引号（" "）的域名或者点分十进制 IP 地址。</p> <p>最大长度：150 字节。</p>
<port>	<p>整型。NTP 服务器端口号。范围：1~65535；默认值：123。</p>
<auto_set_time>	<p>整型。是否自动同步本地时间为 UTC。</p> <p>0    不自动同步</p> <p>1    自动同步</p>
<time>	<p>字符串型。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）和时区（±zz）。</p>
<result>	<p>整型。结果码，详情请参考第 3 章。</p>

备注

1.

当<auto\_set\_time>=1，RTC 会自动更新到同步时间；随后可使用 AT+CCLK?查询更新后的时间。关于 AT+CCLK?更多信息，请参考文档 [2]。
2.

注网成功后，模块会自动更新 RTC 时间。

## 2.4. URC 详解

TCP/IP 相关 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. 下行数据最大支持 1024 字节，超过 1024 字节会被分成多条 URC 上报。

### 2.4.1. 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>

Socket 连接被断开

### 参数

<connectID> 整型。Socket ID。范围：0~4。

### 2.4.2. 下行数据已达 URC

#### 下行数据已达 URC

+QIURC: "recv",<connectID>,<current\_recv\_length>,<data>

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。单位：字节。

<data> 接收到的数据。数据格式为十六进制或文本格式。

# 3 结果码

表 2：结果码列表

结果码	结果码描述	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 上下文失败
562	Close PDP context failed	关闭 PDP 上下文失败
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 上下文发生故障
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. 初始化工作

```

AT+QSCLK=0           //禁用休眠模式。
OK

AT+CEREG?             //查询网络注册状态。
+CEREG: 0,1          //注网已成功，若未成功，可多次查询。

OK

AT+CGPADDR?           //找网成功后，可通过此命令获取模块 IP 地址。
+CGPADDR: 5,"10.177.135.245"

OK

```

### 备注

1. 需要使用 **AT+QSCLK=0** 禁用休眠模式；所有业务完成后，请使用 **AT+QSCLK=1** 启用深休眠模式，以便模块进入深休眠。关于 **AT+QSCLK** 命令的详细信息，请参考文档 [2]。
2. 块开机后，请参考文档 [1] 检查是否成功入网后再执行 **AT+QIOPEN** 命令。

## 4.2. TCP 客户端使用说明

### 4.2.1. 建立 TCP 连接

```

AT+QIOPEN=0,0,"TCP","hf.quectel.com",2020 //远程地址: hf.quectel.com, 端口: 2020;
                                              本地端口为自动分配。

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 //查询连接状态。

```



```
+QISTATE: 0,"TCP","hf.quectel.com",2020,0,2,0,1 //连接成功。
```

```
OK
```

#### 4.2.2. 配置发送的数据格式为文本字符串

```
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 //查询结果：发送的数据格式为文本字符串（<send_data_format>=0）。
```

```
OK
```

#### 4.2.3. 配置发送的数据格式为十六进制格式

```
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）。
```

```
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 //查询结果：发送的数据格式为十六进制格式（<send_data_format>=1）。
```

```
OK
```

#### 4.2.4. 发送文本数据

```
AT+QISEND=0,10,"1234567890" //发送 10 字节数据 1234567890（注意命令中<send_length>参数  
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"}（注意命令中<send_length>参  
数需要和<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
```

#### 4.2.5. 发送十六进制数据

```

AT+QICFG="dataformat" //查询当前配置。
+QICFG: "dataformat",1,0 //查询结果：发送的数据格式为十六进制格式（<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

```

#### 4.2.6. 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

```

#### 4.2.7. 配置接收的数据格式为文本字符串

```

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 //查询结果：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, quectel"消息。
+QIURC: "recv",0,10,"hi,quectel" //提示模块已收到远程服务器下发的数据。

```

#### 4.2.8. 配置接收的数据格式为十六进制格式

```

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）。
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 //查询结果：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",0,10,"31323334353637383930" //提示模块已收到远程服务器下发的数据。

```

#### 4.2.9. 关闭 TCP 连接

```

AT+QICLOSE=0 //主动断开。
OK

CLOSE OK //断开连接成功。

AT+QSCLK=1 //启用深休眠模式。
OK

```

#### 备注

1. 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<nAcked>，关闭后重新建立连接重发未收到对端 ACK 的数据。
2. 完成业务后，请使用 **AT+QSCLK=1** 启用休眠模式，以便模块进入深休眠。

#### 4.2.10. TCP 长连接的会话保活说明

一般 TCP 连接建立后，如果一段时间内无数据交互，网络侧可能会主动释放维持连接的必要资源，并且所述释放行为不会通知模块。关于所述无数据交互导致资源释放的时间长短，暂无明确的数值。目前的测试情况是，10 分钟内基本可保持端口资源不会被释放。若需保持长连接，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，以进行会话保活。

基于此，如果长时间不发送数据，在重新发数据时可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 //查询已发送数据是否收到 ACK 响应。
+QISEND: 3,0,3 //发送的心跳字节，均未得到服务器的响应。

OK

```

//等待 30 秒后。

**+QIURC: "closed",0** //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 60 秒后。

**AT+QISEND=0,0** //查询已发送数据是否收到 ACK 响应。

**+QISEND: 3,0,3** //发送的心跳字节，均未得到服务器的响应。

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. UDP 客户端使用说明

### 4.3.1. 建立 UDP 连接

**AT+QIOPEN=0,0,"UDP","hf.quectel.com",2020** //远程地址：hf.quectel.com，端口：2020；  
本地端口为自动分配。

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** //查询连接状态。

**+QISTATE: 0,"UDP","hf.quectel.com",2020,0,2,0,1** //连接成功。

OK

### 4.3.2. 配置发送的数据格式为文本字符串

**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** //查询结果：发送的数据格式为文本字符串（<send\_data\_format>=0）。

OK

### 4.3.3. 配置发送的数据格式为十六进制格式

**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）。

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** //查询结果：发送的数据格式为十六进制格式（<send\_data\_format>=1）。

OK

### 4.3.4. 发送文本数据

**AT+QISEND=0,10,"1234567890"** //发送 10 字节数据 1234567890。

//注意命令中<send\_length>参数需要和<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"}。

//注意命令中<send\_length>参数需要和<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

### 4.3.5. 发送十六进制数据

**AT+QICFG="dataformat"** //查询当前配置。

**+QICFG: "dataformat",1,0** //查询结果：发送的数据格式为十六进制格式（<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>长度的二分之一。

OK

SEND OK

### 4.3.6. 配置接收的数据格式为文本字符串

```
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 //查询结果: 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, quectel"消息。
+QIURC: "recv",0,10,"hi,quectel" //提示模块已收到远程服务器下发的数据。
```

### 4.3.7. 配置接收的数据格式为十六进制格式

```
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)
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 //查询结果: 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",0,10,"31323334353637383930" //提示模块已收到远程服务器下发的数据。
```

### 4.3.8. 关闭 UDP 连接

```
AT+QICLOSE=0 //主动断开。
OK

CLOSE OK //断开连接成功。

AT+QSCLK=1 //启用深休眠模式。
OK
```

### 备注

完成业务后, 请使用 **AT+QSCLK=1** 启用休眠模式, 以便模块进入深休眠。

#### 4.3.9. UDP 连接的深休眠唤醒数据收发

```

AT+QSCCLK=0                                //开始业务前关闭模块深休眠功能。
OK

AT+QIOPEN=0,0,"UDP","hf.quectel.com",2020    //远程地址: hf.quectel.com, 端口: 2020;
                                                //本地端口为自动分配。
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                              //查询连接状态。
+QISTATE: 0,"UDP","hf.quectel.com",2020,0,2,0,1 //连接成功。
OK

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

SEND OK

AT+QSCCLK=1                                //业务完成后, 需要时开启深休眠功能, 等待模块进入深休眠。
OK

//模块进入深休眠。
+QNBIOEVENT: "ENTER DEEPSLEEP"

//通过下拉 PSM_EINT 引脚或向模块发送一条 AT 命令 (此 AT 命令将会丢失) 唤醒模块。
+QNBIOEVENT: "EXIT DEEPSLEEP" //模块从深休眠中被唤醒。

AT+QSCCLK=0                                //做业务前关闭模块深休眠功能, 业务完成后需要模块深休眠时再开
                                                //启。
OK

//模块会恢复 UDP 相关参数及状态, 此时模块状态与进入深休眠前一致, 可直接发送和接收 UDP 数据。
AT+QISEND=0,10,"1234567890"                //发送 10 字节数据 1234567890。
                                                //注意命令中<send_length>参数需要和<data>实际长度保持一致。
OK

SEND OK

```

#### 4.4. 转换域名为 IP 地址

```
AT+QDNSGIP=0,"baidu.com" //获取域名为 baidu.com 的 IP 地址。
OK

+QIDNSGIP: 0,1,0
+QIDNSGIP: 39.156.69.79
```

#### 4.5. 配置 DNS 服务器地址

```
AT+QIDNSCFG=0,"218.2.2.2","8.8.8.8"
OK

AT+QIDNSCFG=0
+QIDNSCFG: 0,"218.2.2.2","8.8.8.8","240E:5A::6666","240E:5B::6666"
OK
```

#### 4.6. 远程服务器 Ping 操作

```
AT+QPING=0,"iot.quectel.com"
OK

+QPING: 0,47.100.63.174,32,369

+QPING: 0,47.100.63.174,32,479

+QPING: 0,47.100.63.174,32,484

+QPING: 0,47.100.63.174,32,260

+QPING: 0,4,4,0,260,484,398
```

#### 4.7. 通过 NTP 服务器同步本地时间

```
AT+QNTP=0,"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,"19/06/11,11:08:20:35+32"

# 5 附录 A 参考文档及术语缩写

表 3: 参考文档

序号	文档名称	备注
[1]	Quectel_BC260Y-CN_终端应用设计指导	BC260Y-CN 终端应用设计指导
[2]	Quectel_BC260Y-CN_AT 命令手册	BC260Y-CN AT 命令手册

表 4: 术语缩写

术语	英文全称	中文全称
ACK	Acknowledge character	确认字符
DNS	Domain Name System	域名系统
DRX	Discontinuous Reception	非连续接收
eDRX	extended Discontinuous Reception	扩展不连续接收
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
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
PPP	Point to Point Protocol	点对点协议

PSM	Power Saving Mode	省电模式
TCP	Transmission Control Protocol	传输控制协议
TTL	Time to Live	生存时间
UDP	User Datagram Protocol	用户数据报协议
URC	Unsolicited Result Code	非请求结果码
UTC	Universal Time Coordinated	世界标准时间