BC26&BC20

TCP/IP 应用指导

LPWA 模块系列

版本：BC26&BC20_TCP/IP_应用指导_V1.0

日期：2019-07-26

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://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://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。

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

目录

图片索引

1 引言

该文档主要结合实例，介绍如何使用移远通信 BC26 和 BC20 模块内嵌的 TCP/IP 服务，以及使用过程中的重要注意事项。

2 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?                      //查询网成功
+CEREG: 0,1                    //找网已成功，如果未成功，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**) 地址上报之后再进行。

3 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                                      //获取最后一个 Socket 的错误码信息
+QIGETERROR: 0,operate successfully                //建立连接操作成功

OK
```

4 发送数据

4.1. 配置发送的数据格式

4.1.1. 发送的数据为 Text 字符串

```
AT+QICFG="dataformat",0,0  //配置发送的数据格式为 Text 字符串（<send_data_format>=0）
                           //此为默认配置（Text 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 Text 字符串（<send_data_format>=0）

OK
```

4.1.2. 发送的数据转换为 Hex 格式

```
AT+QICFG="dataformat",1,0  //配置发送的数据格式为 Hex 格式（<send_data_format>=1）
                           //此配置默认为<send_data_format>=0（Text 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   //查询结果：发送的数据格式为 Hex 格式（<send_data_format>为=1）

OK
```

4.2. 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 串口
                        （<echo_mode>=0）
                       //此配置默认为<echo_mode>=1（回显数据模式下输入的数据到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    //查询结果：不回显（<echo_mode>=0）
```

```
OK
```

4.3. 发送 Text 字符串数据

4.3.1. 普通模式

```
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
```

4.3.2. 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         //配置在数据模式（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
```

4.3.3. 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<CTRL+Z>                  //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"}<CTRL+Z>           //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               //配置在数据模式（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，数据输入完成后自动退出数据模式并发送

OK

SEND OK
```

4.4. 发送 Hex 字符串数据

4.4.1. 普通模式

```
AT+QISENDEX=0,3,313233              //发送 Hex 字符串数据（模块自动将 Hex 数据 313233 转换为 Text 数据 123 并发送到服务端）
                                    //注意<hex_string>需要满足 Hex 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            //查询结果：发送的数据格式为 Hex 格式（<send_data_format>=1）

OK

AT+QISEND=0,3,313233                /发送 Hex 字符串数据（模块自动将 Hex 数据 313233 转换为 Text 数
                                    据 123 并发送到服务端）
                                    //注意<data >需要满足 Hex 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
```

4.4.2. 定长数据模式

```
AT+QICFG="dataformat"               //查询当前配置
+QICFG: "dataformat",1,0            //查询结果：<send_data_format>为 1（Hex 格式）

OK

AT+QISEND=0,10                      //发送 10 字节 Hex 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 >需要满足 Hex 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               //配置在数据模式（data mode）下将输入的数据不回显到 UART 串口
                                    （<echo_mode>=0）
OK
AT+QISEND=0,5                       //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
```

4.4.3. 不定长数据模式

```
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 Hex 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<CTRL+Z>             //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 Hex 格式，否则会报 ERROR

SEND OK

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
```

4.5. 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
```

5 接收数据

5.1. 配置接收的数据格式

5.1.1. 接收的数据为 Text 字符串

```
AT+QICFG="dataformat",0,0  //配置接收的数据格式为 Text 字符串（<recv_data_format>=0）
                           //此为默认配置（Text 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 Text 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）：hi, i am tcp server

OK
```

5.1.2. 接收的数据转换为 Hex 格式

```
AT+QICFG="dataformat",0,1  //配置接收的数据格式为 Hex 格式（<recv_data_format>=1）
                           //此配置默认为<recv_data_format>=0（Text 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   //查询结果：接收的数据格式为 Hex 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 Text 数据 1234567890 后自动转换成 Hex 格式数据

OK
```

5.2. 配置所接收数据的输出格式

5.2.1. 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             // 查询结果：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
```

5.2.2. 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             //查询结果：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
```

5.3. 所接收数据的存取方式

5.3.1. 缓存模式

建立连接配置：

```
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。

5.3.2. 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     //查询结果：<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     //查询结果：<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
```

5.3.3. 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
```

6 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**<nAcked>**，关闭后重新建立链路重发未 ACK 的数据。

7 TCP 会话保活

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。如果需要保持长连接，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
```

8 附录

8.1. 术语缩写

表 1：术语缩写

术语	描述	
MCU	Microprogrammed Control Unit	微程序控制器
LWIP	Lightweight TCP/IP Protocol	轻量级 TCP/IP 协议栈
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议
HEX	Hexadecimal	十六进制
NVRAM	Non-Volatile Random Access Memory	非易失性随机访问存储器

8.2. TCP/UDP 应用时序图

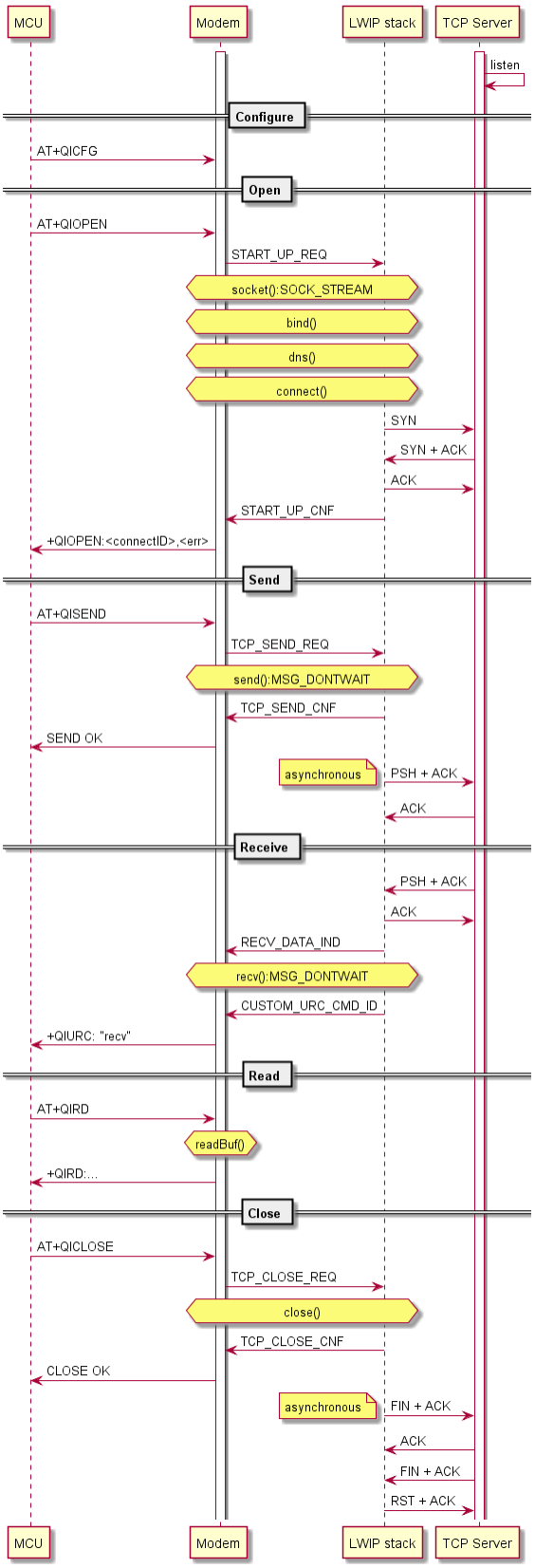

图 1：TCP 应用时序图

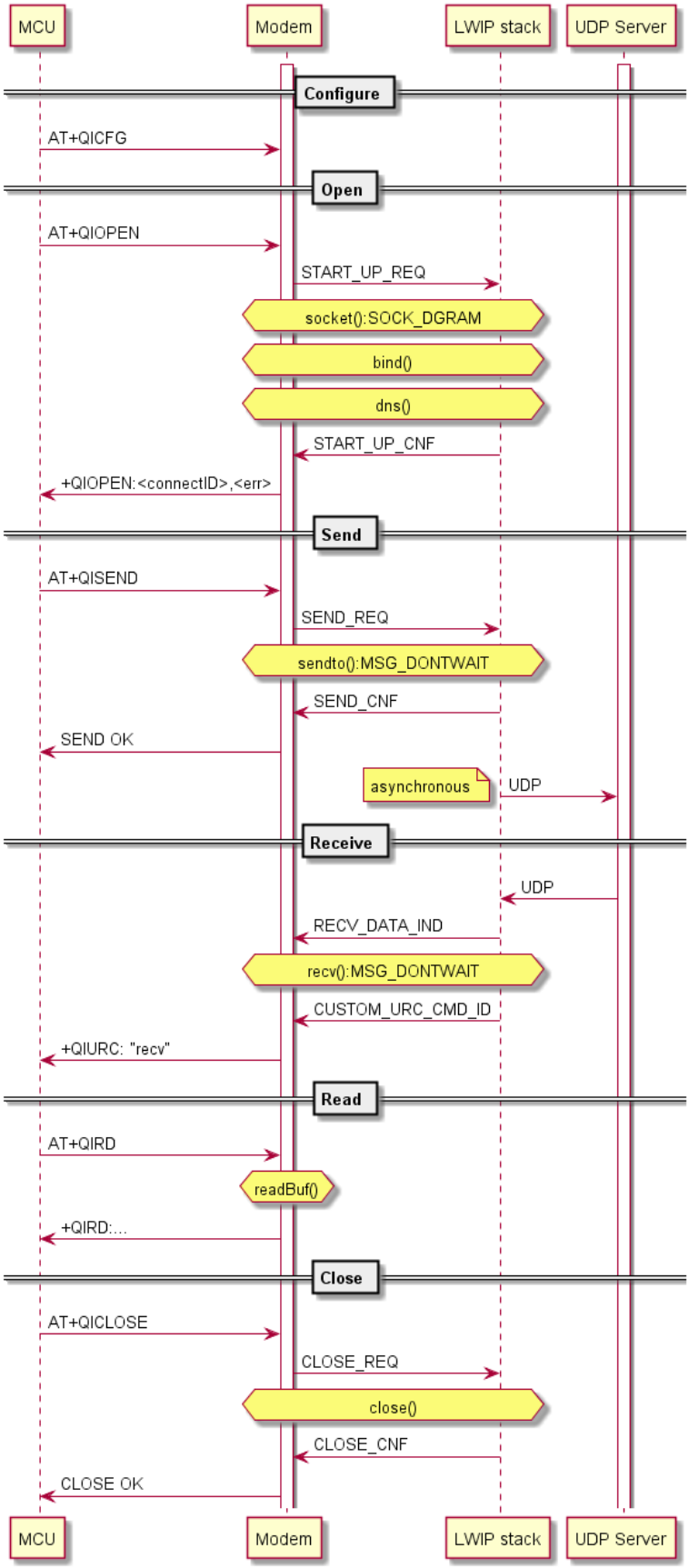
MCU
Modem
LWIP stack
UDP Server
Configure
AT+QICFG
Open
AT+QIOPEN
START_UP_REQ
socket():SOCK_DGRAM
bind()
dns()
START_UP_CNF
+QIOPEN:<connectID>,<err>
Send
AT+QISEND
SEND_REQ
sendto():MSG_DONTWAIT
SEND_CNF
SEND OK
asynchronous
UDP
Receive
UDP
RECV_DATA_IND
recv():MSG_DONTWAIT
CUSTOM_URC_CMD_ID
+QIURC: "recv"
Read
AT+QIRD
readBuf()
+QIRD:...
Close
AT+QICLOSE
CLOSE_REQ
close()
CLOSE_CNF
CLOSE OK
MCU
Modem
LWIP stack
UDP Server

图 2：UDP 应用时序图