

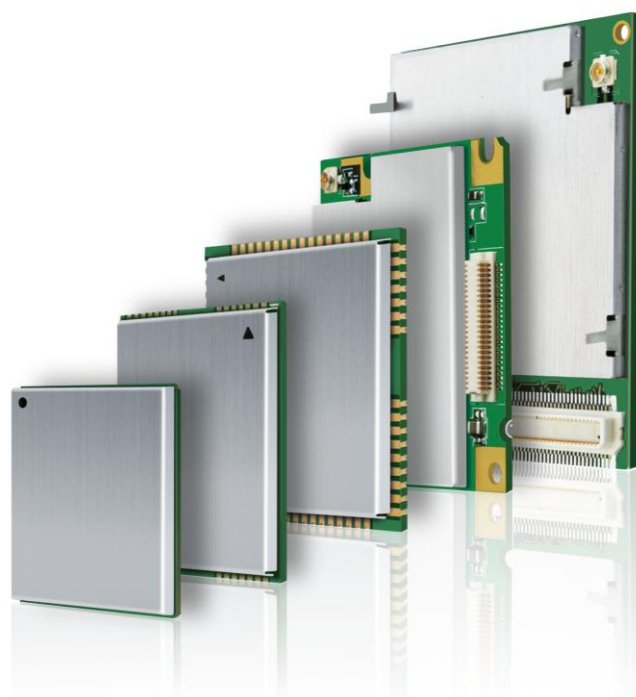


GSM 无线通信模块

GSM 模块 TCPIP

应用流程指导

GSM 模块 TCPIP 应用流程指导_V1.2



文档标题	GSM 模块 TCPIP 应用流程指导
版本	1.2
日期	2015-04-03
状态	正式发布
文档受控号	GSM 模块 TCPIP 应用流程指导_V1.2

前言

移远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，移远公司有权对该文档规范进行更新。

版权申明

本文档手册版权属于移远公司，任何人未经我公司复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术有限公司 2015，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2015.

目录

目录.....	2
0. 修改记录.....	3
1. 概要.....	4
1.1. 参考文档.....	4
2. 初始化工作.....	5
3. 建立 TCP 连接	7
4. 发送数据.....	9
4.1. 不指定长度发送.....	9
4.2. 指定长度发送.....	10
5. 接收数据.....	11
5.1. 数据直接输出到串口.....	11
5.2. 通过命令提取接收到的数据.....	11
6. TCP 连接维持和侦测	13
7. 关闭 TCP 连接	15
8. 透传连接.....	16
8.1. 示例.....	16
8.2. 异常处理.....	16
9. TCPIP 设计参考流程图.....	18

0. 修改记录

版本	日期	作者	修改内容记录
1.0	2010-10-25	胡孔林	初始版本
1.1	2012-07-08	胡志琴	增加 TCPIP 设计参考流程图
1.2	2015-04-03	王淑娟	增加适用模块说明

1. 概要

该文档主要结合实际示例，介绍Quectel标准模块内嵌的TCPIP服务使用方法以及一些异常情况处理方案。

本文档适用于所有Quectel GSM模块。

1.1. 参考文档

表 1: 参考文档表

序号	文档名	备注
[1]	Mxx_ATC.pdf	GSM 模块 AT 命令集简介
[2]	GSM_TCPIP_AN.pdf	GSM 模块 TCPIP 应用指导

2. 初始化工作

```

AT+IPR=115200&W      // 配置固定波特率
OK
AT+CPIN?              //首先要确保SIM卡的PIN码已解
+CPIN:READY            //这表明SIM卡的PIN码已解

OK
AT+CREG?              //确认找网成功
+CREG: 0,1             //这表明找网成功，如果没有成功，继续执行命令AT+CREG?
                        //查询。这里也可以通过在开始时设置命令AT+CREG=1打开网络
                        //状态变换时的自动上报。这样，等待+CREG:1或+CEREG:5的自
                        //动上报就可以了

OK
AT+CGATT?            //查询GPRS附着是否成功
+CGATT: 1              //这表明GPRS附着成功，如果没有成功，需要继续执行命令
                        // AT+CGATT?查询。对于GPRS是否附着的判断也可以通过在开
                        // 机时执行命令AT+CGREG=1，然后等待+CGREG: 1或
                        // +CGREG: 5的自动上报

OK
AT+QIFGCNT=0          //将Context 0设为前台Context。此后的相关操做都是针对Context
                        // 0进行操作

OK
AT+QICSGP=1,"CMNET" //设置GPRS的APN。如果是使用中国移动或中国联通的SIM卡，
                        // 并通过GPRS建立TCP连接，则这条命令可以省掉

OK

```

/******

其他可选设置：

- 一、设置服务器的地址格式。默认情况下，模块会认为服务器地址是一个IP地址。但是，如果服务器的地址是一个域名，则需要执行命令"**AT+QIDNSIP=1**"将服务器的地址格式改为域名格式。
- 二、设置数据接收方式。目前模块支持两种方式来处理接收到的TCP/UDP数据：直接通过串口输出所有接收到的TCP/UDP数据（默认情况下，使用的就是这种方式）；接收到数据后，不是立即输出，而是输出一个提示，然后需要通过命令提取刚刚收到的数据

如果希望使用这种方式，则需要执行下面的命令。

AT+QINDI=1 //接收到数据后，输出提示:" +QIRDI: <id>,<sc>,<sid>"。然后就可以通过命令" **AT+QIRD=<id>,<sc>,<sid>,<len>**"提取数据了

OK

三、设置接收到的数据显示格式。当选择通过串口输出接收到的数据时。可能需要通过以下命令设置接收数据的显示格式。下面这些命令都可以根据实际需要取舍。

AT+QIHEAD=1 //在接收到的数据之前增加头信息"**IPD<len>:**"

OK

AT+QISHOWRA=1 //在接收到的数据头位置增加数据来源的地址和端口号。具体的格式为: **RCV FROM:<IP ADDRESS>:<PORT>**

OK

AT+QISHOWPT=1 //在接收到的数据之前增加传输层的协议类型，TCP或者UDP。这个应用不是很多

OK

四、设置透传模式。默认情况下，模块采用非透传方式，下面的2、3、4、5、6几个章节就是介绍这种方式下的连接建立，数据发送和数据接收的方法。如果希望采用透传方式则需要执行下面的命令。关于透传方式的具体应用请参考章节7。

AT+QIMODE=1 //使用透传方式

OK

AT+QITCFG=3,2,512,1 //这里重点要关注中间两个参数。第二个参数2表示模块在收到串口输入的数据后（如果全部输入的数据长度小于512个字节）会等待200ms，然后发送已经输入的全部数据。第三个参数512表示当模块从串口收到的数据长度超过512个字节之后，则立即以512个字节的数据为一组发送数据，直到缓冲区中的数据长度小于512

OK

*****/

3. 建立 TCP 连接

```
AT+QIOPEN="TCP","116.226.39.202","7007" //连接一个TCP服务器。地址是
116.226.39.202:7007
OK //命令的语法格式正确并且当前的状态
可以建立TCP连接
```

其他错误返回的分析及处理：

一、"ERROR"，这里返回ERROR可能有两个原因：

- 1、命令格式不对，如果所有数据格式都对，需要看看QIMUX是不是为0（通过命令"AT+QIMUX?"查询，如果查询结果为1，则需要通过命令AT+QIMUX=0将其重新设为0）。
- 2、当前的TCPIP服务的状态不是IP INITIAL或IP STATUS或IP CLOSE（通过命令AT+QISTAT查询）。如果当前状态为TCP CONNECTING，则需要执行AT+QICLOSE关闭当前失败的TCP连接。如果是其他状态，则需要执行命令AT+QIDEACT断开当前失败的GPRS场景。

二、"ALREADY CONNECT"，这表明已经存在一个TCP连接或UDP连接。如果确认需要建立新的连接，则需要命令AT+QICLOSE关闭当前连接。

```
CONNECT OK //TCP连接成功
```

其他错误返回的分析及处理：

一、"CONNECT FAIL"，这表明TCP连接建立失败。正确的处理方法是：首先，执行命令AT+QISTAT查询当前的状态，如果当前状态为TCP CONNECTING，建议通过AT+QICLOSE关闭当前失败的连接，这样可以保持GPRS仍然处于激活状态，不用重启GPRS，以提高程序运行的速度。如果当前状态不是TCP CONNECTING，建议执行命令AT+QIDEACT直接关闭当前的GPRS场景（因为在这些状态下，通常是因为GPRS场景激活失败所致），关于命令AT+QIDEACT的返回处理，可以参考第6章的说明。这条命令理论上最长的等待时间大概在2分半到3分半。客户在实际应用中也可以根据实际情况设置一个小于2分钟的超时时间，超时后的处理方法和收到CONNECT FAIL之后的处理方法一样。

/*同样，模块也支持UDP方式，在使用UDP方式之前，先要建立UDP连接。方式和建立TCP连接一样。*/

```
AT+QIOPEN="UDP","116.226.39.202","7007" //连接一个UDP服务器。地址是
116.226.39.202:7007
OK
```

```
CONNECT OK //UDP连接成功。实际上UDP方式并不用
建立连接，这里其实只是设置了接下来需
```


要发送UDP数据的目标地址和端口

4. 发送数据

4.1. 不指定长度发送

AT+QISEND > TEST<Ctrl+Z>	//准备发送数据。 //发送数据TEST, <Ctrl+Z>表明请求发送已输入的数据。 记住,每次最多可以输入1460个字节的数据。在这种方式下,有几个特殊字符不能发送: 0x08 (back space), 模块收到这个字符时,将会删除上一个输入的字符。0x1A (<Ctrl+Z>), 模块收到这个字符,会停止接收后面的字符,并将之前收到的数据发送出去。0x1B (ESC), 模块收到这个字符后,会退出此次发送操作
SEND OK	//数据已成功交给TCP协议层
AT+QISACK +QISACK: 4, 4, 0	//检查数据是否已经发送成功。 //前面两个参数分别是已发送的数据长度和已确认发送成功的数据长度。最后一个参数是尚未确认发送成功的数据长度,如果这个参数为0,表明所有数据都已发送成功。该命令在UDP方式下没有实际意义,因为UDP本身是无连接连接,不能确定数据有没有成功发送,所以在UDP方式,执行该命令,会发现中间的那个参数一直为0。
OK	

/******

注:

- 一、因为底层socket的发送窗口有7300,因此,不一定要等到所有数据都已经确认(就是命令**AT+QISACK**返回的最后一个参数为0),再发下一包。考虑到一个TCP的最大长度为1448,建议可以设置一个阈值3000。如果没有确认的数据长度(命令**AT+QISACK**返回的最后一个参数)小于3000,就继续使用命令**AT+QISEND**发送下一段报文,如果超过3000,则停止发送,然后每隔5秒执行命令**AT+QISACK**查询一次,直到返回的最后一个参数小于3000,就继续发送数据。如果查询到一定次数(例如20次,相当于超时100秒)后,返回的最后一个参数一直大于3000,则可以认为TCP连接出现异常。关闭此次连接,重新建立TCP连接,继续发送之前一直没有确认的数据和接下来需要发送的数据。
- 二、假如想通过非透传的方式发送十六进制的字符时,尤其是0x08, 0x1A, 0x1B这三个字符时,建议使用指定长度的方式发送数据。请看下一节。

*****/

4.2. 指定长度发送

AT+QISEND=3

>0x080x1A0x1B

//指定待发送数据的长度为3。

//依次发送0x08, 0x1A, 0x1B这几个特殊字符。实际输入时, 一定要直接输入0x08, 0x1A, 0x1B这几个具体数据, 而不是像例子中的样子输入“0x080x1A0x1B”的字符串, 例子中是为了看的清楚一点

SEND OK

AT+QISACK

+QISACK: 17, 17, 0

OK

5. 接收数据

5.1. 数据直接输出到串口

默认情况下，模块接收到TCP数据都是直接通过串口输出，不带任何头和尾。为了将TCP数据与AT命令返回或URC区分开来，建议增加头信息，至于怎么增加头信息，请参考第1章节。接收到的数据到底以什么样的格式显示，最好在TCP连接建立之前就设置好。下面是一个接收到的数据的示例。

```

RECV FROM:116.228.146.250:7070<CR><LF> //模块收到来自远端
116.228.146.250:7070的数据。如果在初始
化时没有执行命令AT+QISHOWRA=1，
则不会有这一行提示

IPD36TCP:1234567890abcdefghijklmnopqrstuvwxy
//接收到长度为36的TCP数据。具体的数
据是冒号之后的36个字符。如果在初始化
时没有执行命令AT+QIHEAD=1，则不
会有头部: "IPD36TCP:"。如果在初始
化时没有执行命令AT+QISHOWPT=1，
则不会有头部的"TCP"
  
```

注：在这种数据接收方式下，因为数据是以收到就立即通过串口输出。这有时难免会出现接收到的数据插入AT命令中间。在目前的设计中，做了一些超时处理，以避免接收到的数据打断一个完整的AT数据（比如说，直接将AT分开）。但是还是不能避免将命令和该命令的返回隔开。

5.2. 通过命令提取接收到的数据

在这种方式下，模块接收到TCP数据或UDP数据，不是立即通过串口输出数据，而是输出一个URC通知客户，客户可以随后通过命令提取接收到的TCP数据。默认情况下，是不支持这种数据接收方式的。需要在建立TCP连接之前执行命令**AT+QINDI=1**使能这种方式。下面是在这种情况下，模块收到远端数据时的一个示例。

```

/*假设模块收到远端发过来的TCP数据"1234567890abcdefghijklmnopqrstuvwxy"*/
+QIRDI:0,1,0 //模块收到数据，接收的报文的连接是基于Context 0的，并且是
客户端方式。

/*此时，可以通过下面的命令提取数据*/
AT+QIRD=0,1,0,1024 //从模块的socket缓冲区中提取数据。最大提取的长度为1024。
如果缓冲区中的数据不足1024，则提取所有存于缓冲区中的数据
  
```

```
+QIRD:116.228.146.250:7070,TCP,36<CR><LF>  
1234567890abcdefghijklmnopqrstuvwxyz
```

6. TCP 连接维持和侦测

因为大部分模块连接到Internet上都是通过GPRS网关中转的，GPRS网关在这里的作用相当于我们通常局域网里的路由器。每次，模块试图与Internet上的某个服务器建立连接的时候，首先需要GPRS网关分配一个端口给我们的模块。通过这个端口，我们的模块和Internet建立联系。这样，就引入了一个问题，GPRS网关的端口资源是有限的，因此，它对GPRS网内的终端使用这些端口就有了一些限制，以达到端口资源的合理分配。它会释放长时间没有数据交互的端口资源，以供其他连接使用。这样，模块和Internet上的服务器之间的连接就被不知不觉的给掐断了。

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。我们在上海的测试情况是只要不超过10分钟，基本就可以保持端口资源不会被释放。

为了避免TCP连接经常不知不觉地被GPRS网关断开的问题，建议每隔一定时间就向服务器发送一个短数据包。通过这个方法，可以维持TCP连接和侦测TCP连接的状态。

例如：按照目前我们测试得到的结果，将心跳包的发送的时间间隔定为10分钟。

```

AT+QISACK                                //检查之前数据发送的情况
+QISACK: 1448, 1448, 0                    //假设之前已经发送1448个字节的数据，并且
                                           //都已经发送到服务器

OK

AT+QISEND                                //发送普通数据
> 1234567890abcdefghijklmnopqrstuvwxy<0x1A> //发送数据，<0x1A>表明请求发送已输入
                                           //的数据

SEND OK                                  //数据已成功交给TCP协议层
  
```

假设在这之后很长一段时间都不用发送数据了，则需要利用心跳包数据维持连接。在发送心跳包维持连接之前，可以先通过当前这个报文是否能够发送成功来确定TCP连接是否正常。比如说，过了两分钟（每5秒查询一次，查询24次。），仍然不能确定这个报文发送成功，可以认为目前的TCP连接出现异常，然后执行命令**AT+QICLOSE**关闭当前TCP连接，然后再执行命令**AT+QIOPEN**重新建立TCP连接。连接成功后，再重发上面未发送成功的报文。

*****/

```

AT+QISACK                                //检查数据发送的情况。
+QISACK: 1484,1448,36                    //之前已经发送的数据总长度为1484，但是已
                                           //经确认发送成功的数据长度只有1448，刚刚发
                                           //送的36个字节还不确认是否发送成功。等待5
                                           //秒再次查询
  
```

OK

.....

AT+QISACK

+QISACK: 1484,1484,0

//等待5秒钟

//检查数据发送的情况

//所有已发送的1484个数据都已经确认发送成功。说明到目前为止，TCP连接是正常的

OK

.....

//等待8分钟。一直没有数据发送，也没有收到数据

AT+QISEND

> Heart01<0x1A>

//发送心跳数据。

//发送心跳数据Heart 01。这个心跳数据可以再简单一点，当然，也需要根据实际情况定义心跳数据格式，以避免服务器误解

SEND OK

.....

//等待5秒，然后看看心跳数据是否发送成功

AT+QISACK

+QISACK: 1491,1491,0

//检查数据发送的情况

//心跳数据已经成功发送到服务器，说明当前TCP连接正常。如果此时返回的结果是心跳数据还没有确认发送成功(也就是最后一个参数不为0)，则等待5秒后再次查询，如此反复，直到确认心跳数据发送成功或者查询次数超过24次。如果查询次数超过24次(也就是超时2分钟)，则可以认为当前TCP连接出现异常。可以考虑关闭当前TCP连接，重新建立TCP连接

OK

7. 关闭 TCP 连接

AT+QICLOSE //所有数据都已发送完成后，关闭当前的TCP连接
CLOSE OK

AT+QIDEACT //如果此后很长一段时间（比如说一个小时以上）都不会使用这个GPRS场景，则建议使用**AT+QIDEACT**命令关闭GPRS场景。这条命令在正常情况下响应时间大约为2-5秒。但是在网络很差或其他异常情况下，最长会有2分半的超时。建议在这里根据实际应用做一个1分钟或者更短的超时限制，如果超过这个时间仍然没有**DEACT OK**返回，则可以考虑直接使用**EMERG OFF**和**POWER KEY**重启模块

DEACT OK

注：在每条命令相应的返回（或者**ERROR**）收到之后，再继续执行下一条命令。

8. 透传连接

如果在初始化时执行了命令**AT+QIMODE=1**，则模块在建立TCP/UDP连接之后会进入数据模式。在数据模式下，所有从串口输入的数据都会被认为是发送远端的数据，而所有从串口输出的数据都是从远端接收到的数据，除了"**CLOSED**"，"**+PDP DEACT**"等特殊文本。下面是一个TCP透传的示例。

8.1. 示例

```
AT+QIOPEN="TCP","116.226.39.202","7007"
//连接一个TCP服务器。地址是
116.226.39.202:7007
OK
//命令的语法格式正确并且当前的状态可以
建立TCP连接。这里的错误返回可以参考第2
章节的介绍。

CONNECT
//TCP连接成功，并且串口已进入数据模式。
这里的错误返回可以参考第2章节的介绍。

.....
//可以发送数据了，比如说，这里我们输入了
"1234567890abcdefghijklmnopqrstuvwxyz"，这
时输入的数据，是不会通过串口回显的。由于
这段数据的长度小于512（见第1章的设置），
模块会等待200ms（见第1章的设置），然后
才将这段数据包发送出去。

abcdefghijklmnopqrstuvwxyz1234567890
//这一段是从服务器接收到的数据。

.....
//输入"+++"序列退出数据模式。
OK
//"+++"序列的返回，说明已经成功回退到命
令模式。接下来如果想关闭TCP连接，可以参
考第6章节的示例了。
```

8.2. 异常处理

在数据模式下如果收到"**CLOSED**"和"**+PDP DEACT**"字段时，则说明TCP连接可能断开或出现异常了，但是这个也有可能是远端发送过来的TCP数据。此时，建议首先通过输入"**+++**"序列确认当前是否仍在数据模式下。如果输入"**+++**"后，返回OK，则说明刚刚成功从数据模式退回到命令模式，这也就说明此前收到的"**CLOSED**"或"**+PDP DEACT**"字段是正常的TCP数据；否则，如果数据"**+++**"后，只有"**+++**"的回显（回显打开时），没有OK返回，则说明此前已经在命令模式下了，这也就是说此前收到的"**CLOSED**"或"**+PDP DEACT**"字段的意思是TCP连接断开或GPRS场景断开了，这时需要按照下面的步骤重新建立TCP连接。

（当然，如果可以确定远端服务器发送的数据中不会包含"**CLOSED**"和"**+PDP DEACT**"字段时，则可以直接重新建立TCP连接，而不需要使用"+++"序列进行判断。）

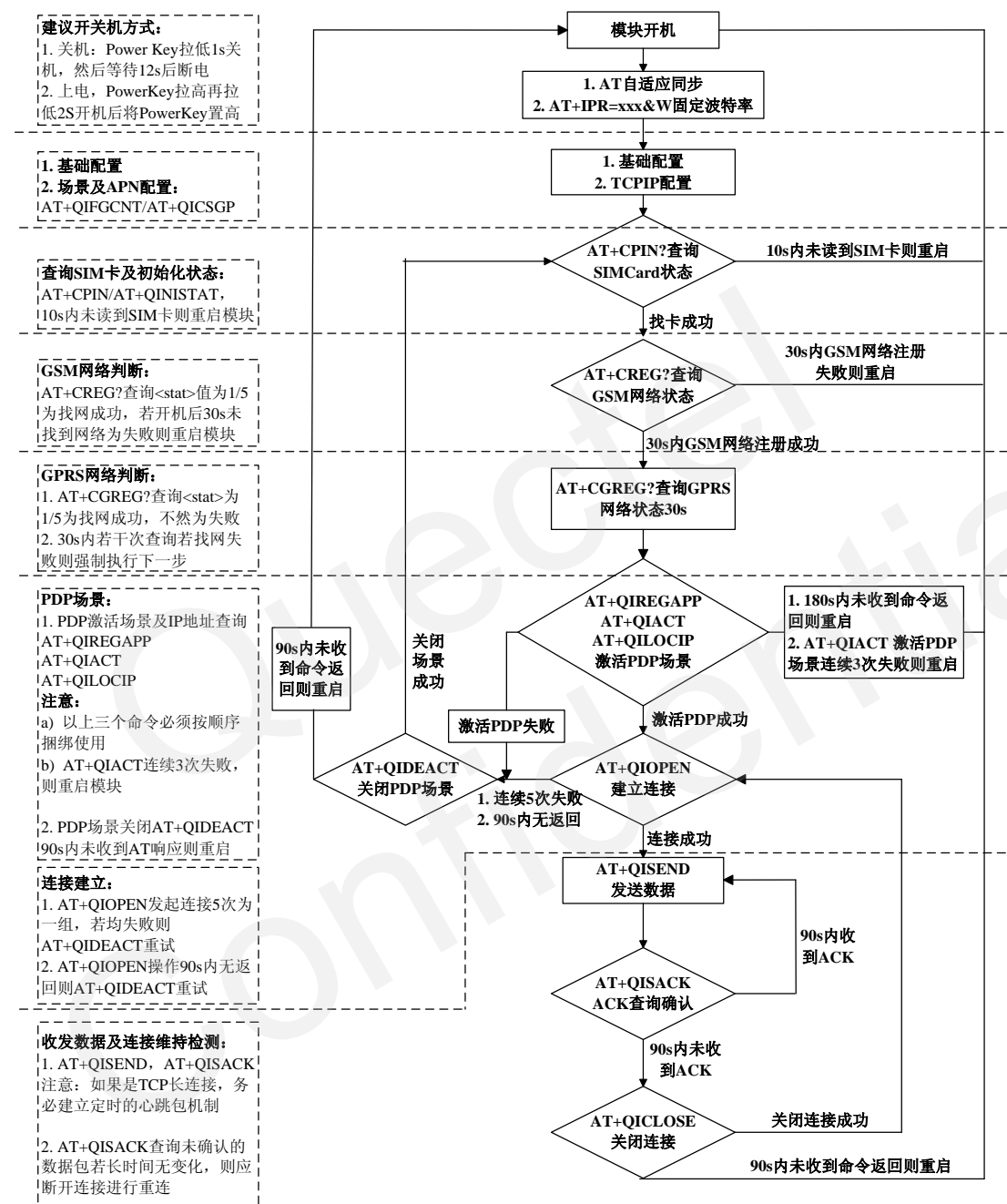
AT+QIDEACT

//关闭当前的GPRS场景。这条命令在正常情况下响应时间大约为2-5秒。但是在网络很差或其他异常情况下，最长会有2分半的超时。建议在这里根据实际应用做一个1分钟或者更短的超时限制，如果超过这个时间仍然没有**DEACT OK**返回，则可以考虑直接使用**EMERG OFF**和**POWER KEY**重启模块

DEACT OK

9. TCP/IP 设计参考流程图

以下为TCP/IP设计参考流程:



TCP/IP流程其他注意事项:

1. 开机后必须有串口自适应AT同步操作处理, 每500ms间隔连续发AT同步(15次), 同步成功后AT+IPR=xxx&W固定波特率并保存
2. 所有AT命令必须有返回后才能执行下一条AT指令, 普通的AT指令可配置60s超时机制, 无响应则重启模块
3. TCP/IP涉及到的部分网络交互命令需要做超时机制, 超时无返回则重启模块, 如: AT+CGATT 超时配置180s, AT+QIACI超时配置180s, AT+QICLOSE超时配置90s, AT+QIDEACT超时配置90s, 超时命令无返回则重启模块
4. **重启机制要求:** 模块不能被连续重启。当出现连续的异常失败(含流程图内的所有重启操作), 第一次可以立即重启, 后续可为10分钟后重启, 30分钟重启, 1小时重启

QUECTEL



quectel
Confidential

上海移远通信技术有限公司

上海市田州路 99 号 13 幢 501 室 200233

电话: +86 21 5108 6236

Mail: info@quectel.com