

**Q** 我写了个短信发送程序，使用 PDU 格式发送，程序在广州使用一点问题也没有，在河南却怎么也发不出去。不知道为什么，短信“你好吗”格式如下：

河南：0891683108200005F011000D91683170031618F20008A9064F60597D5417

广州：0891683108301705F011000D91683170031618F20008A9064F60597D5417

**A** 发送短信时要用 SIM 卡属地的 SMSC 号码。如果是在广州办的卡，即使在外地还是要用广州的 SMSC 号码。你的两个短信内 SMSC 号码不同，但用的是同一张 SIM 卡，不知是否是此原因。

**Q** 短信中心的号码可否直接使用 SIM 卡中的号码，而不要用户输入？我用过的短信软件好像都是不用输 SMSC 号码的。

**A** 有一条“AT+CSCA”指令，可用于设置或查询服务中心号码。若手机中已存在此号码，有两种解决办法：

- 用“AT+CSCA?”指令查询出来，然后自动将此号码写到 PDU 的 SCA 中。
- PDU 的 SCA 字段只写一个“00”：“08 91 68 31 ...” -> “00”

可用“AT+CSCA=xxxxxxxx”指令设置服务中心号码。

**Q** 我在超级终端上，用 at+cmgs 发送短消息，格式好像没有错误，但总返回“ERROR”。我输入的就象这样：

```
at+cmgs=30>
```

```
0891683108100005F011000D91683118405057F0000000006C8329BFD0E01 请问是什么原因？
```

**A** “at+cmgs”指令很特殊，回车后还需要输入数据。此处是“CR”，不是“CRLF”，注意在超级终端里直接回车是不是生成了两个字符（查看设置）。象“at+cmgl”指令，即使最后输入“CRLF”也是不要紧的。

你的问题出在长度上。长度不是随便写的，你的例子中，长度应为 21。除去 SMSC 段（0891683108100005F0），从“11”开始算（即“11000D91683118405057F0000000006C8329BFD0E01”），除以 2 即得。

正确的写法应该是

```
at+cmgs=21>
```

```
0891683108100005F011000D91683118405057F0000000006C8329BFD0E01 (“>”是手机提示，不是输入的)
```

**Q** 我最近在编一个关于短消息的程序，在你的“通过串口收发短消息”中提到“Text Mode 是纯文本方式，可使用不同的字符集，从技术上说也可用于发送中文短消息，但国内手机基本上不支持，主要用于欧美地区。”是不是说我用 AT 指令“AT+CMGF=1”或“AT+CMGF=0”对我后来的收发短消息没什么影响啊？

**A** Text mode 写起来简单，直接发原文就行，发送非 ASCII 码内容也能发，但需要手机支持才能正确显示。如法语、德语的很多字符，编码大于 0x80，他们都是用 text mode。Text mode 靠什么区分字符编码方式呢？有专门的字符集设定指令“AT+CSCS”。可以设定为扩充字符集“UCS2”。Siemens TC35/TC37 资料上说，它的“AT+CSCS”支持“UCS2”字符集，但我目前没有机会去亲自试验。正在使用 TC35/TC37 模块的朋友不妨试一下。

据我了解，中文短消息方面，在国内卖的各种手机只支持 PDU mode，这成了事实上的标准。其实 PDU mode 真的挺好用，估计以后 text mode 会萎缩。我们写的程序，我建议只采用 PDU mode，即使是发纯英文信息也这样，编码倒是可以

灵活采取 7bit 或 UCS2, 因为 7bit 能发的长度是 UCS2 的 2 倍 (仅对纯英文而言)。如果发送纯数据, 不需要手机显示, 可用 8bit。

**Q** 你的 smstraffic 类中的发送接收大循环中, 是不是把所有收到的消息都放入消息队列后, 然后执行删除程序啊? 如果我是并发量很大的话, 就是网关有很多短消息等着进入手机, 读完所有短消息后, 进行删除的过程中, 因为短消息的排列顺序, 而导致误删除呢 (比如说我现在手机里有 1-15 条短消息, 然后在我删除第一二条后, 第三条自动填补为第一条, 而新进来的短消息, 16 条排在了第三条, 而被 cancel 掉呢?) 我试过好象短消息的排列不是每次都一样啊? (在接收的时候, 同一条短消息有时是 14 条, 有时是第 15 条) 这个怎么解决啊?

**A** 手机里消息都有一个物理序号, 读出的时候带序号, 删除也要根据序号删。“物理”二字很关键。这个序号相当于 ID, 无论它前面有没有删除、删除了多少消息, 都不会变的。假如原来有 1-15, 删除了 1 和 2, 又来了一条消息, 手机内部的软件有两种处理方式: 有的放在第 1 条, 有的则放在第 16 条, 我都见过。其实, 它愿意放到哪个空闲的地方都行。但无论如何, 不会引起混乱的, 因为读出是什么序号, 就删除什么序号的。在执行删除命令前, 消息还是在原来那个地方, 不会被后来的覆盖。

如果说网关有很多短消息等着进入手机, 量很大, 这种处理方式效率不高, 因为 AT+CMGL 占用很长时间, 这段时间手机无法从 SMSC 接收新消息。采用我说的“实时”接收方法比较好, 消息来了直接传出来, 不经过写入手机的过程。

**Q** 我用 Nokia 8210 串口数据线, 连上电脑的 com1 口, 用 SmsTest 运行提示“没有发现 MODEM”, 跟踪发现 gsmInit() 检测中串口发 AT 指令没有回应“OK”。按您的提示我安装了 Nokia modem 驱动程序, (WIN2000 server 系统) 虚拟出 com3 口的一个 8210 MODEM 设备, 再次调用 smsTest 还是提示“没有发现 MODEM”。但用串口线, 手机能通过 LogoManager 手机管理软件进行相应的图片 LOGO, 短信发送操作。

**A** Nokia 手机本身没有带 modem 功能, 用专业术语讲就是不具备 TA(Terminal Adapter)接口, 需要驱动转换, 不管是真的串口, USB 还是红外接口, 反正它能虚拟出“标准 MODEM”串口来。AT 命令只能用标准异步通信。

在我的印象中, Nokia 8210 需用红外线接口同 PC 通信。估计你装的那个驱动是 IR->COM 转换的, 而不是驱动串口数据线的, 可能你的电脑没有红外接口, 所以 com3 也连不上?

要试(虚拟)串口是否连接正确, 很简单, 用 windows 自带的“超级终端”在特定虚拟端口连上, 敲个“AT”回车, 看有没有反应, 正确回答应该是“OK”。

Nokia 数据线上跑的是“Nokia 语”- Nokia 专有协议的数据, 不是通用/扩展的 AT 命令集。LogoManager 能听、能说“Nokia 语”, 所以不需要安装驱动就能工作。Nokia 有一个免费的“Nokia PC Connectivity SDK”, 可供开发 Nokia 手机使用。至于 LogoManager 是不是用的这个开发包, 那就不得而知了。

**Q** 在 SmsTest 中, 发出 AT 命令, 然后接收应答, 比如  
`WriteComm("AT+CMGF=0\r", 10); ReadComm(ans, 128);` 在 WriteComm 函数后接着就调用 ReadComm, 是不是太急, 这里的 ReadComm 函数是读返回的这个字符串还是其中的单个字符或不完整的字符串? 请问超时控制设多少最合适啊?

A 关于读串口，程序中是这样设定超时控制的：

```
COMMTIMEOUTS timeouts = { // 串口超时控制参数
    100,           // 读字符间隔超时时间：100
    ms            1,           // 读操作时每字符的时间：1 ms (n 个字符总
    共为 n ms)      500,           // 基本的(额外的)读超时时间：
    500 ms         1,           // 写操作时每字符的时间：1 ms (n 个字
    符总共为 n ms) 100};       // 基本的(额外的)写超时时间：100
```

msReadComm 什么时候返回呢？按此 timeout 设定，若 n=128 (ReadComm 的第二个参数)，则

- 若无任何数据，等待 500+1\*128=628 毫秒返回。也就是说，若没有连上手机，根本不存在应答，ReadComm 会持续阻塞 628 毫秒，而后返回。
- 若数据连续传输，且字符间隔也未超过了 100 毫秒，但时间已经到了 628 毫秒，返回已读取的字符(串)。接收到的可能是不完全的字符串。
- 若在 628 毫秒内，字符间隔超过了 100 毫秒（第一个字符之前等待的时间不算），返回已读取的字符(串)。接收到的应该是完整的字符串。

在手机正确连接的情况下，主要是最后一条起作用。一段数据是连续传输的，若波特率是 9600bps，可以算出字符间隔是 0.1 毫秒左右，远小于 100 毫秒，不会读一个字节或部分数据就返回；通常是数据完毕后才可能出现等待 100 毫秒而返回的情况。举个例子，若在执行 ReadComm(ans, 128)后 150 毫秒收到“OK\r\n”，则还需要额外等 100 毫秒，也就是说函数将在 250 毫秒后返回。这里传输 4 个字节数据的时间被忽略不计了。如果觉得读得太急，可将基本的(额外的)读超时时间调大一些。不过 500 毫秒内还没有应答，可能是连接故障造成的。

需要特别注意的是“AT+CMGL”指令及其应答。可能是由于需要扫描所有存储区域的缘故，手机在逐条送出短消息后，还需要延迟好几秒的时间才能送出最后的“OK”。当然可以通过设定上面的基本读超时时间很长（比如 20 秒），并且一次读很长的数据（比如 2000），来达到目的。但这样一来，函数阻塞时间太长，若恰好这时要程序退出，你会赫然发现“该程序无响应”。SmsTest 中解决办法是：循环读取串口数据，将每次读取的数据拼接起来，最后得到完整的应答。

gsmGetResponse() 每次可能读取部分数据，将新数据追加到已读数据后，且检测是否见到“OK”或“ERROR”，以判断是否已经读到完整的数据。