

短信猫二次开发包

Linux 接口说明

V2.5

目录

1	概述	3
2	流程说明	3
3	函数列表	4
3.1	MonInitModem	4
3.2	MonSendMsg	4
3.3	MonGetMsg	5
3.4	MonCloseModem	6
3.5	MonGetStatus	6
3.6	MonGetSendCount	6
3.7	MonGetRecvCount	6
3.8	MonClearSendBuff	6
3.9	MonClearRecvBuff	7
3.10	MonSetNationalMode	7
3.11	MonSetValPeriod	7
4	例子程序	8
5	JAVA 编程	8
6	版本记录	8

图目录

图 1	接口调用示意图	3
-----	---------------	---

1 概述

梦网短信猫 linux 接口提供函数调用接口，接口完全底层调用，没有任何界面，适合于广大自主品牌的开发用户。函数接口简单，只需几个简单的函数调用，就可以实现信息的发送和接收功能。调用结构图如下：

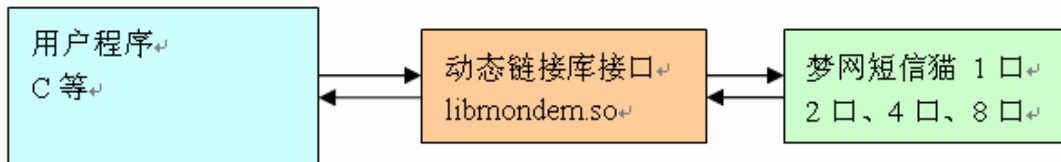


图 1 接口调用示意图

短信猫通过串行口（COM）和计算机相连，我们这里编程都是对串行口进行的，一般在 linux 下串口设备名可能是/dev/ttyS0、/dev/ttyS1 等等。

本文适合 libmondem.so 版本 2.5。

2 流程说明

用户使用 libmondem.so 编程的基本流程如下：

- 1) 把 libmondem.so 文件置于当前目录（执行程序）下或系统的/usr/lib 目录下。
- 2) 执行初始化函数 MonInitModem,可以指定打开指定的端口，参数为端口名，第二个参数为该端口短信猫的通道个数，如 MonInitModem("/dev/ttyS0",1)。
- 3) 调用 MonSendMsg 函数进行信息提交，同样可以指定用某个端口发送，或者-1表示用所有可能的端口进行发送。
- 4) 以查询方式调用 MonGetMsg 接收信息，本函数也可以指定单独端口或所有端口。本函数可以接收 SIM 卡接收到的信息和接收由 MonSendMsg 发送信息的发送结果。具体请看函数说明部分。注意每次查询时间间隔不要太快，Sleep 1 秒钟即可。
- 5) 退出程序前调用 MonCloseModem 关闭端口。

注：在发送之前可先用 MonGetStatus 函数得到端口状态，状态编号大于等于 8，表示该端口设备连接正常，可以发送。

3 函数列表

3.1 MonInitModem

说明：初始化函数

定义：int MonInitModem(char *dev, int num)

参数：dev 串口设备名称，如” /dev/ttyS0”，

num 该设备的短信猫的通道的个数。

如用 MonInitModem (“/dev/ttyS0” ,1)初始化。

返回：

0 - 成功

-1-端口已经被打开

-2-设备错

-3-设备错

1- 打开端口错

2- 设置串口参数错

3.2 MonSendMsg

说明：提交发送函数

定义：int MonSendMsg(int Chno,char * szHeader, char * szMsg)

参数：Chno 0~7 代表通道号码，-1 表示所有可能通道

szHeader 目标手机号码+序列号+已发送次数+发送者 ID+发送类型+报告类型
(共六项)

例如： 138123456789,1,0,1388888888,0,1

其中目标手机号码是毕添项目，其他可添可不添

其中手机号码和序列号之间用逗号隔开，序列号由用户自己定义，用于唯一标识这条短信，当 MonDem.DLL 返回状态时，同时返回该序列号，这样便于用户区分是哪一条短信的返回状态。参见 fnReadMsgEx。另外：序列号只是为了用户便于识别返回状态使用的，它不影响发送过程。已发次数代表该条短信已经发送的次数，在返回函数里，该次数将被累加 1 并返回。

序列号为字符型，最长支持 20 位。

发送者 ID 是一个 11 位的字符串，方便用户记录发送者帐号信息

发送类型：0-普通信息，1-PUSH 信息

报告类型：0-无信息达到报告，1-有信息达到报告

szMsg 发送信息内容（当信息类型是 PUSH 信息时，填写 16 进制的字符串，如：C5B894231AE6F8，该字符串的格式内容由用户自己定义。）

(应该在 70 个字符以内,包含标点符号,1 个汉字算 1 个字符)

例如：恭喜发财,测试成功! 长度为 10

返回：0 - 提交成功

深

- 1 - 端口还未打开
- 2 - 通道号指定错误
- 3 - 接收号码错误
- 4 - 信息长度错误
- 5 - 缓冲区满

备注: 提交成功并不代表发送成功, 发送结果信息用 MonGetSms 来读取

发送缓冲区中最多容纳 500 条记录,超过 500 提交失败,函数返回-1

3.3 MonGetMsg

说明: 接收函数

定义: `int MonGetMsg(int chno, char * szHeader, char * szReadMsg)`

参数: chno 0~7 代表通道号码, -1 表示所有端口

szHeader 包含: 信息类型, 端口号, 接收时间, 手机号码, 序列号, 已发次数六部分

例如:

0,5,2004-05-28 12:30:52,1382523xxxx,001,1

信息类别的说明:

- 0- 接收信息,
- 1- 发送成功信息
- 2- 发送失败
- 3- 发送超时 1
- 4- 发送超时 2

端口号码 5 代表第 5 个端口

接收时间 2004-05-28 12:30:52

手机号码: 1382523xxxx

序列号: 发送结果与提交信息时的序列号相同, 接收信息为 0

已发次数: 尝试发送信息的次数 Msg 接收信息内容

返回: 0-成功

- 1- 端口未打开
- 2- 通道号错
- 3- 无信息
- 4- 未找到指定通道的信息(其他通道有信息, 请用-1 查询)

注意: java 调用该函数时有些区别, 返回的是字符串数组, 调用方式如下:

```
public static native String[] MongateGetSms(int clientsocket);
```

备注: 函数每次最多只返回 10 条信息, 如果收到信息数为 10 时注意接着接收下面的信息(属于没有发送成功正好又正在重试发送的信息不会被接收, 最大重试发送次数不会大于 100), 接收缓冲区中最多保留 500 条记录, 多余部分将被丢弃, 所以请您编程时一定要及时读取, 以免信息丢失

3.4 MonCloseModem

说明：关闭函数

定义：int MonCloseModem ()

参数：

返回：0 - 成功

XXX -错误代码，参见代码表

3.5 MonGetStatus

说明：取端口状态函数

定义：int MonGetStatus (int PortNo)

参数：nPortNo 0~7 代表端口号码

返回：端口状态，大于 8 表示设备连接正常，可以接收和发送。

小于 1，可能连线没有接好或者电源没有打开，注意复位短信猫。

大于 1 小于 8，可能是注册网络错误，请检查卡是否正常

3.6 MonGetSendCount

说明：取发送缓冲区条数

定义：int MonGetSendCount (int nPortNo)

参数：PortNo 0~7 代表端口号码， -1 表示所有端口

返回：发送缓冲区信息条数。

-1 端口号指定错误

3.7 MonGetRecvCount

说明：取接收缓冲区条数

定义：int MonGetRecvCount (int PortNo)

参数：PortNo 0~7 代表端口号码， -1 表示所有端口

返回：发送缓冲区信息条数。

-1 端口号指定错误

3.8 MonClearSendBuff

说明：清除发送缓冲区

深

定义: `int MonClearSendBuff (int PortNo)`

参数: PortNo 0~7 代表端口号码, -1 表示所有端口

返回: 0-成功。

-1 端口号指定错误

3.9 MonClearRecvBuff

说明: 清除接收缓冲区

定义: `int MonClearRecvBuff (int PortNo)`

参数: PortNo 0~7 代表端口号码, -1 表示所有端口

返回: 0-成功。

-1 端口号指定错误

3.10 MonSetNationalMode

说明: 设置国际格式(支持国际格式发送)

定义: `int MonSetNationalMode (int nMode)`

本函数用来设置采用国际模式发送。缺省情况下为非国际模式。

参数: nMode - 1 开启国际模式, 此时发送号码前需要加国别码, 如:

香港加: 852 (如 852912345678)

国内加: 86 (如 8613612345678)

nMode - 0 关闭国际模式 (缺省)

注意:一定在的 `MonSendMsg` 函数调用之前调用这个才有效, 否则采用缺省的模式(0-国内)发送。每次启动动态库后, 只需调用一次设置功能, 无需在每次发送前都调用该函数。

另外, 只有在 0 模式下, 才可以发送 SP 端口号码。

返回: 0-成功

3.11 MonSetValPeriod

说明: 设置短信发送有效期

定义: `int MonSetValPeriod (int nPeriod)`

本函数用来设置短信发送的有效期限。

参数: nPeriod: 0~255, 缺省为 255

0~143 -- nPeriod*5 分钟, 范围 5 分钟到 12 小时

144~167 -- 12 小时+(nPeriod-143)*30 分钟

168~196 -- (nPeriod-167)*1 天

197~255 -- (nPeriod-196)*1 周

4 例子程序

我们为该接口提供了 C/C++和 java 的调用例程，请参照。
C 语言编译的时候请加-ldl 参数例： gcc -g -o test test.c -ldl

5 JAVA 编程

动态库 libmondem.so 中支持有 11 个函数，供 JAVA 调用。它们是：

```
public native int MonInitModem(String strDev,int num);
public native int MonSendMsg(int Chno,String strHeader,String strMsg);
public native int MonCloseModem();
public native String[] MonGetMsg(int Chno);
public native int MonGetStatus(int Chno);
public native int MonGetSendCount(int Chno);
public native int MonGetRecvCount(int Chno);
public native int MonClearSendBuff(int Chno);
public native int MonClearRecvBuff(int Chno);
public native int MonSetNationalMode (int nMode);
public native int MonSetValPeriod (int nPeriod);
```

前三个的调用方法与 C 语言的通用，最后一个是将信息返回到一个字符串组中：其中第一个字符串保存返回值，第二个为 szHeader，第三个为 szMsg， 格式和 C 语言的相同。
mondem.java 是包含动态库的类
testj.java 是测试程序例子

运行前先用 javac testj.java 编译
执行方法 java -Djava.library.path=' .' testj

6 版本记录

版本号	更改者	时间	说明
2.5	CX	2007-7-26	1. 支持国际模式 2. 支持有效期 3. 支持缓冲区操作函数 4. 修正 2.4 版本中的错误