# **minicom 使用总结**

[功能]

串口通信程序。

[描述]

\*简介

在嵌入式开发中，Linux主机上安装好minicom程序之后，将linux主机和开发板通过串口相连接，这样可以直接在pc上面通过启动的minicom,"登陆"到开发板上，和开发板进行交互，效果类似远程登陆telnet不过使用的是串口连接。

\*配置：

启动minicom的时候，会检测MINICOM环境变量，将它们作为命令参数，如果命令中指定了某个参数，那么这个MINICOM中的相应的某个参数会被替换。注意必须对这个环境变量进行export才行。例如：

MINICOM='-m -c on'

export MINICOM

这样，启动minicom的时候，会自动地将这个环境变量指定的参数作为minicom的启动参数。如果命令行中又指定了这个环境变量中的某个参数，那么命令行中指定的参数值将会覆盖这个环境变量的参数值。

或者其它等效的设置，然后启动minicom，minicom 会假定你的终端有Meat键或Alt键，并且支持彩色。如果你从一个不支持彩色的终端登录，并在你的启动文件(.profile或等效文件)中设置了 MINICOM，而且你又不想重置你的环境变量，那么你就可以键入‘minicom -c off’,来运行这次没有色彩支持的会话。

The configuration 参数更有趣。通常，minicom从文件“minirc.dfl”中获取其缺省值。不过，若你给minicom一个参数，它将尝试从文件“minirc.configuration”中获取缺省值。因此，为不同端口、不同用户等创建多个配置文件是可能的。最好使用设备名，如：tty1， tty64，sio2等。如果用户创建了自己的配置文件，那么该文件将以“.minirc.dfl”为名出现在他的home目录中。

minicom的全局配置文件通常在/etc/minirc.dfl. minicom将其配置文件保存在一个目录中，通常是：/var/lib/minicom, /usr/local/etc或者/etc。要想知道minicom编译时内定的缺省目录，可用命令“minicom -h”，在那儿你可能还会找到runscript(1) 的demo文件。

配置文件的文件名格式：minirc.user。minicom默认加载配置文件minirc.df1，但你也可以仿照minirc.df1编辑自己的配置文件minirc.user，输入minicom user启动时则加载minirc.user。如在本机上，我做了一个minirc.root，输入minicom root 启动minicom，则系统自动加载/etc/minirc.root中的配置。

$HOME/.minirc.\*

$HOME/.dialdir

$HOME/minicom.log

/usr/share/locale/\*/LC\_MESSAGES/minicom.mo

\*命令行启动minicom的参数

更多的参数，参见"man minicom"的输出。

-s

设置。 root 使用此选项在/etc/minirc.dfl中编辑系统范围的缺省值。使用此参数后，minicom 将 不 进行初始化， 而是直接进入配置菜单。如果因为你的系统被改变，或者第一次运行minicom时，minicom不能启动，这个参数就会很有用。对于多数系统，已经内定了比较合适的缺省值。

-o

不进行初始化。minicom 将跳过初始化代码。如果你未复位(reset)就退出了minicom，又想重启一次会话(session)，那么用这个选项就比较爽(不会再有错误提示：modem is locked ---- 注)。但是也有潜在的危险：由于未对lock文件等进行检查，因此一般用户可能会与uucp之类的东东发生冲突......也许以后这个参数会被去掉。现在姑且假定，使用modem的用户对自己的行为足够负责。

-m

用Meta或Alt键重载命令键。在1.80版中这是缺省值，也可以在minicom 菜单中配置这个选项。不过若你一直使用不同的终端，其中有些没有Meta或Alt键，那么方便的做法还是把缺省的命令键设置为Ctrl-A，当你有了支持 Meta或Alt键的键盘时再使用此选项。Minicom假定你的Meta键发送ESC前缀，而不是设置字符最高位的那一种(见下)。

-M

跟“-m”一样，但是假定你的Meta键设置字符高端的第八位(发送128+字符代码)。

-z

使用终端状态行。仅当终端支持，并且在其termcap或terminfo数据库入口中有相关信息时才可用。

-l

逐字翻译 高位被置位的字符。使用此标志， minicom将不再尝试将IBM行字符翻译为ASCII码，而是将其直接传送。许多PC-Unix克隆不经翻译也能正确显示它们(Linux使用专门的模式：Coherent和Sco)。

-w

在启动的时候就自动换行，这样启动之后不用运行<C-a> w进行设置了，详细参考后面。

-a

特性使用。 有些终端，特别是televideo终端， 有个很讨厌的特性处理(串行而非并行)。minicom缺省使用‘-a on’，但若你在用这样的终端，你就可以(必须!)加上选项‘-a off’。尾字‘on’或‘off’需要加上。

-t

终端类型。使用此标志， 你可以重载环境变量TERM， 这在环境变量MINICOM 中使用很方便； 你可以创建一个专门的 termcap 入口以备minicom 在控制台上使用，它将屏幕初始化为raw模式， 这样，连同‘-l’标志一起，就可以不经翻译而显示IBM行字符。

-c

颜色使用。些终端(如Linux控制台)支持标准ANSI转义序列色彩。由于termcap显然没有对于色彩的支持，因而minicom 硬性内置了这些转义序列的代码。所以此选项缺省为off。使用‘-c on’可以打开此项。把这个标志，还有‘-m’放入MINICOM环境变量中是个不错的选择。

-S

脚本。 启动时执行给定名字的脚本。到目前为止，还不支持将用户名和口令传送给启动脚本。如果你还使用了‘-d’选项，以在启动时开始拨号，此脚本将在拨号之前运行，拨号项目入口由‘-d’指明。

-p

要使用的伪终端。 它超载配置文件中定义的终端端口，但仅当其为伪tty设备。提供的文件名必须采用这样的形式：(/dev/)tty[p-z][0-f]

-C

文件名。 启动时打开捕获文件。

-8

不经修改地传送8位字符。“连续”意指未对地点/特性进行真正改变，就不插入地点/特性控制序列。此模式用于显示8位多字节字符，比如日本字(TMD！应该是中国字!!)。不是8位字符的语言都需要(例如显示芬兰文字就不需要这个)。

[举例]

\*启动minicom的时候，开启自动换行：

$minicom -w

如果不加这个项，那么在minicom和pc交互的时候中键入命令超过一行时候会被截断，（这时候可以通过<C-a> w来开和关切换截断行功能）.

\*启动minicom的时候，显示颜色：

$minicom -c on

这样，启动之后我们会发现显示的内容不是黑白的了。

\*启动一个图形的minicom:

$xminicom

\*启动minicom的时候，自动创建日志捕捉文件my\_capturefile：

$minicom -C my\_capturefile

这样，启动之后，所在minicom的输出都会在my\_capturefile中保留一份,如果原来文件存在，则追加，不存在则创建一个。

\*启动minicom的时候，将命令键修改成Meta键（<C-a>变成[Alt]或者[ESC]等）:

$minicom -m

这样，我们可以取代用<C-a> \*发送命令的方式，将<C-a>替换成[Alt]或者[ESC].

\*启动minicom的时候，指定运行的脚本：

$minicom -S <filename>

这里，<filename>是你的脚本文件的名字，应该指定绝对路径，否则就会在你启动minicom的路径下寻找。关于minicom的脚本，可以参考man手册"man runscript".交互命令中可以运行"<C-a>G"来运行脚本。

关于runscript脚本：

注释是'#'开始的。

send命令中的字符串，有\c就不会发送默认的\r了。例:send "ls \c"就不会自动回车，这里'\c'也可以在前面。

目前发现，脚本功能比较弱。在minicom原代码中有简单的例子：scriptdemo和unixlogin,可以直接在linux运行runscript.

可以指定minicom的脚本运行程序，

D - Script program

    作为脚本解释器的程序。缺省是“runscript”，也可用其它的东东(如: /bin/sh 或 "expect"，这样就可以用shell或者其他的脚本了^\_^)。Stdin和Stdout连接到modem，Stderr连接到屏幕。

\*\*minicom的交互使用

Minicom是基于窗口的。要弹出所需功能的窗口，可按下Ctrl-A (以下使用C-A来表示Ctrl-A),然后再按各功能键(a-z或A-Z)。先按C-A，再按'z'，将出现一个帮助窗口，提供了所有命令的简述。配置 minicom(-s 选项，或者C-A、O)时，可以改变这个转义键，不过现在我们还是用Ctrl-A吧。

这里，只给出很少的命令，更多的交互命令参见"<C-a> z"的帮助输出。

\*查看帮助：

输入"<C-a> z".

这样会显示所命令列表，便于查找。

\*回到shell:

输入"<C-a> j".

这样minicom会在后台stop,直到fg将它调到前台。

\*开/关捕捉minicom中的输出到一个文件：

输入"<C-a> l".

这样会打开/关闭捕捉输出功能，捕捉的输出会存放在一个你选择的文件中。

\*退出minicom:

输入"<C-a> x".

这样会退出minicom.

\*\*

[其他]

\*配置minicom：

配置minicom的目的是让pc机可以通过minicom连接开发板。

下面是一个配置的例子：

(1)#minicom -s

(2)选择"Serial port setup"

配置其中的如下配置项：

A - Serial Device : /dev/ttyS0

E - Bps/Par/Bits : 115200 8N1

F - Hardware Flow Control : No

G - Software Flow Control : No

这里，也可以在启动minicom之后运行"<C-a> o"来进行配置。

\*使用minicom在开发板和主机之间传输文件

这里详细讲述如何在minicom下面，开发板子和pc机器直接通过串口传输文件。

主要内容：

1)准备需要的文件

假设我的开发板是arm体系的cpu。那么需要如下文件：

a)lrzsz-0.12.20.tar.gz: 传输文件的工具，需要编译成开发板子可运行的，也要再编译一份在pc上面运行的。

b)bin\_forarm/\* :这里是随便建立的一个目录，里面存放编译好的在开发板子运行的文件传输工具(也就是前面那个lrzsz的arm运行版本,编译它们需要借助交叉编译工具)，需要拷贝到板子上的/usr/bin目录下。

c)bin\_forpc\_maybe/\* :这里是随便建立的一个目录，里面存放编译好的在pc运行的文件传输工具(也就是前面那个lrzsz的pc运行版本)，因为可能pc上面原来就有lrzsz工具，所以可以先只把bin\_forarm的内容拷贝到开发板上面尝试一下看能否传输，如果只拷贝bin\_forarm/\*不好用的话就把这里的内容拷贝到pc的/usr/bin下。

2)工具(lrzsz,minicom)的获取

a)下载lrzsz-0.12.20

http://download.chinaunix.net/download/0007000/6293.shtml

b)编译

交叉编译lrzsz拷贝到开发板上.

编译lrzsz并拷贝到pc机器上.

c)下载minicom

#apt-get install minicom

这里假设我的系统是ubuntu.

3)配置

a)打开minicom,配置环境。

b)配置文件所在目录

--------------------------------------------------

Ctrl + a     o

         +-----[configuration]------+

         | Filenames and paths      |

         | File transfer protocols -|

         | Serial port setup        |

         | Modem and dialing        |

         | Screen and keyboard      |

         | Save setup as dfl        |

         | Save setup as..          |

         | Exit                     |

         +--------------------------+

+-----------------------------------------------------------------------+

| A - Download directory : /tmp                                         |

| B - Upload directory   : /home/quietheart/tmpTrans                    |

| C - Script directory   :                                              |

| D - Script program     : runscript                                    |

| E - Kermit program     :                                              |

| F - Logging options                                                   |

|                                                                       |

|    Change which setting?                                              |

+-----------------------------------------------------------------------+

    A - download 下载的文件的存放位置

    B - upload   从此处读取上传的文件

    download     开发板 ---> PC

    upload       PC     ---> 开发板

    B - Upload directory   : /home/quietheart/tmpTrans

    PC机向开发板发送文件，需要发送的文件在/home/quietheart/tmpTrans目录下(PC机上的目录)。

做了此项配置后，每次向开发板发送文件时，只需输入文件名即可>，无需输入文件所在目录的绝对路径

4)传输文件

实际前面的upload directory表示pc机器的上传目录，将会把这个目录中的内容传到板子上面.上面的配置不是必须的，但在后面选择的时候使用上面的目录做为默认。

方法：

假设我在pc上面，使用minicom,在minicom中，

PC-->开发板

a）首先minicom中，运行开发板子上的rz,准备接受数据：

#lrz

b）在minicom中，输入

[Ctrl]a S

这里的Send表示从pc发送到开发板子.然后会进入选择文件的目录(默认为upload目录),选择PC机上面准备传输的文件，然后确认即可。

这样会开始传输文件，传输的文件最终存放在开发板子上运行lrz命令的地方。

开发板-->PC

a)在开发板子上面运行:

#lsz filename

b)在开发板子上minicom下运行(有时候不需要这一步)

[Ctrl]a R

这样,会自动把要传送的文件传送到前面设置的Download directory上面(/tmp)

\*参考资料：

http://blog.csdn.net/sikinzen/archive/2010/04/05/5451758.aspx

minicom手册

其他网上收集的资料

以上，是我对Minicom使用的一些经验的总结。

作者：QuietHeart

Email：quiet\_heart000@126.com

日期：2010年12月8日