深圳开源通信有限公司

OpenVox-Best Cost Effective Asterisk Cards

OpenVox B800P User Manual for mISDN

Written by: James.zhu

Email:james.zhu@openvox.cn,zhulizhong@gmail.com

Date:15/10/2007

Version: 0.01

OpenVox

深圳开源通信有限公司

OpenVox-Best Cost Effective Asterisk Cards

OpenVox Communication Co. Ltd.
Address: F/2,Building No.14,Shangsha Science & Technology Park,
No.9283,Binhe Road,Futian District,ShenZhen ,Guangdong 518048,China
Tel:+86-755-82535362,82535095,Fax:+86-755-82535174
E-Mail：sales@openvox.com.cn
IM for Sales: betty_yljiang@hotmail.com rubyzhang1217@hotmail.com
IM for Technical Support: support@openvox.com.cn,zhulizhong@gmail.com,zhulizhongum@hotmail.com
Business Hours: 9:30AM-17:30PM from Monday-Friday
URL：www.openvox.com.cn

Thank You for Choosing OpenVox Products!

Table of Contents

Chapter 1 Overview

1. What is B800P

B800P is a PCI 2.2 compliant card supporting 8 BRI S/T interfaces, with an onboard multi NT power feeding circuits. NT/TE mode can be individually configured on each of 8 ports.

B800P can be implemented for building Open Source Asterisk based systems such as ISDN PBX and VoIP gateway.

Target Applications:

High Performance ISDN PC Cards
ISDN PABX for BRI
VoIP Gateways
ISDN LAN Routers for BRI
ISDN Least Cost Routers for BRI
ISDN Test Equipment for BRI

Main Features:

Eight integrated S/T interfaces
ITU-T I.430 and TBR 3 certified and S/T ISDN supporting in TE and NT mode
Integrated PCI bus interface (Spec.2.2) for 3.3V and 5V signal environments
DTMF detection on all B-channels
Multiparty audio conferences bridge
Onboard power feeding for 8 NT Ports
PCM bus connectors daisy chaining
Each of the 8 ports can be individually configured for TE or NE mode
Full software and hardware compatible with Junghanns.NET™ Bristuff driver and mISDN© driver
Application ready: use Asterisk® to build your IP-PBX/Voicemail system
RoHS compliant
Certificates: CE, FCC

2. What is Asterisk:

The Definition of Asterisk is described as follow:
Asterisk is a complete PBX in software. It runs on Linux, BSD, Windows (emulated) and provides all of the features you would expect from a PBX and more. Asterisk does voice over IP in four protocols, and can interoperate with almost all standards-based

telephony equipment using relatively inexpensive hardware.

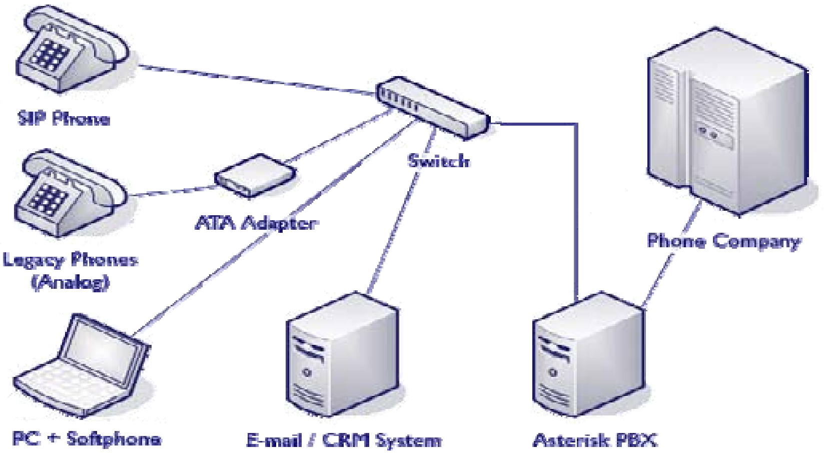


Figure 1: Asterisk Setup

Source (http://www.siriusit.co.uk/uploads/images/consulting/asteriskSetup.gif)

Asterisk provides Voicemail services with Directory, Call Conferencing, Interactive Voice Response, Call Queuing. It has support for three-way calling, caller ID services, ADSI, IAX, SIP, H.323 (as both client and gateway), MGCP (call manager only) and SCCP/Skinny(voip-info.org).

Chapter 2 Card Installation and Configuration

1. Hardware Installation and Setup

   1) Configure the Jumper Settings

      Please refer jumper setting for details. To install B800P, user should follow these steps.

      A. Setting Card ID Switch

         User should take care of the card id switch, if the user wants to install more than one cards of B800P in one pc. The user should follow these produces:

         1. The card id of the first card must be set to 0, and the second card should be set to 1, and so on.
         2. The first card is a card that will be initialized (i.e installing driver) first when system is booting.
         3. At most of cases, Linux will initialize PCI devices according to PCI slot order. The slot is nearest to the CPU will be initialized first; the slot at the far end from CPU will be initialized at last. That is to say that if user has more than one cards of B800P in pc, the one is nearest to CPU should be set to card id 0.
         4. For more details about Card ID, please check F&Q section.

      B. Adjusting Termination of S/T Interface (100 ohm)

         1. If a port works on NT mode, user should set jumper to CONNECT (ON).
         2. If a port work on TE mode, Theoretically it should be to OPEN(OFF), but user might connect to some non-standard isdn terminal equipments that do not have terminal resistors, for such equipments, you should set it to CONNECT(ON).

      C. Power Feeding Connector

         These jumpers control whether the card will feed power to the external isdn terminal. User should adjust accordingly from the following:

         ✧ If the port works on TE mode, user MUST set the jumper to OPEN(OFF)

         ✧ If this port will work on NT mode, the ISDN terminal requires ISDN power supply, user should set the jumper to CONNECT(ON). ***ISDN terminal does not require ISDN power supply, user should set the jumper to OPEN(OFF).***

D. Power Feeding Input

If one of the eight power feeding connectors is CONNECT(ON), user should connect a D-type connecter from pc power supply to this jack, the D-type connecter is used to provide power to CDROM ahd 3.5" HDD.

E. Power Supply Selection

Some latest model PCs do not provide +5V on the PCI slots. If user uses those kinds of PCs, user has to set the jumper to 3.3v.

F. PCM IN/PCM OUT

These are for future usages.

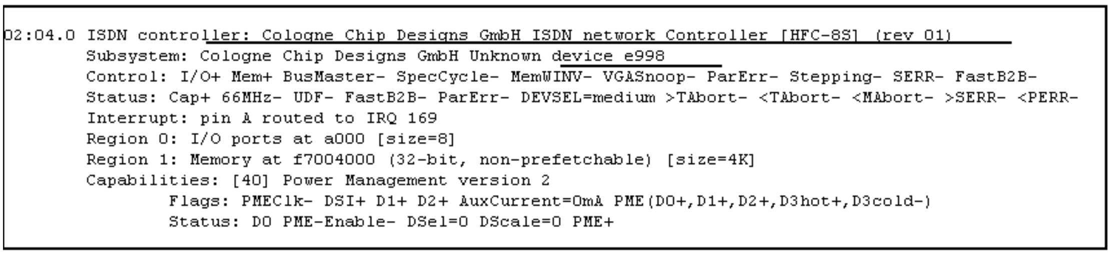
2) Power off PC, remembering unplug the AC power cable
3) Insert B800P card into a 3.3v or 5.0v PCI slot
4) Plug the hard disk power supply cable(D style) to power feeding input jack if needing providing power to external equipments, please refer jumper setting section for the detail
5) Plug back the AC power cable, and power on PC

If user wants to know the details about hardware configuration of B800P, please refer hardware setting section.

2. Software Installation and Setup

B800P supports original mISDN driver form http://www.misdn.org/index.php/Installing_mISDN. Customers can download it from that URL. There are few steps to install the driver.

1) Checking the B800P hardware by command: **lspci -vvvvvvvvv**



```
02:04.0 ISDN controller: Cologne Chip Designs GmbH ISDN network Controller [HFC-8S] (rev 01)
        Subsystem: Cologne Chip Designs GmbH Unknown device e998
        Control: I/O+ Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR-
        Interrupt: pin A routed to IRQ 169
        Region 0: I/O ports at a000 [size=8]
        Region 1: Memory at f7004000 (32-bit, non-prefetchable) [size=4K]
        Capabilities: [40] Power Management version 2
                Flags: PMEClk- DSI+ D1+ D2+ AuxCurrent=0mA PME(D0+,D1+,D2+,D3hot+,D3cold-)
                Status: D0 PME-Enable- DSel=0 DScale=0 PME+
```

2) Checking the supporting packages

Note that if there is no kernel source in the system, user should install them. User can run **yum** again: ***yum install kernel-devel***`. If user uses this command **yum** will install the sources for your current version of the kernel. It is time to check for the availability of some other packages:

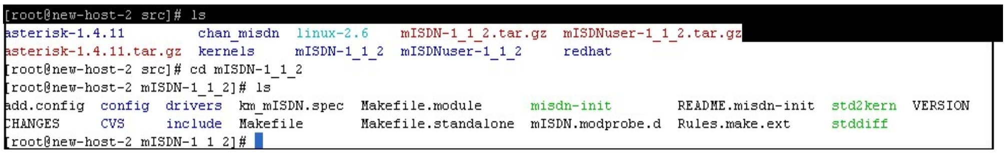
```
rpm -q bison
rpm -q bison-devel
rpm -q ncurses
rpm -q ncurses-devel
rpm -q zlib
rpm -q zlib-devel
rpm -q openssl
rpm -q openssl-devel
rpm -q gnutls-devel
rpm -q gcc
rpm -q gcc-c++
```

If any of those packages are not installed install them by using yum

```
yum install bison
yum install bison-devel
yum install ncurses
yum install ncurses-devel
yum install zlib
yum install zlib-devel
yum install openssl
yum install openssl-devel
yum install gnutls-devel
yum install gcc
yum install gcc-c++
```

3) Downloading, unzipping and compiling driver

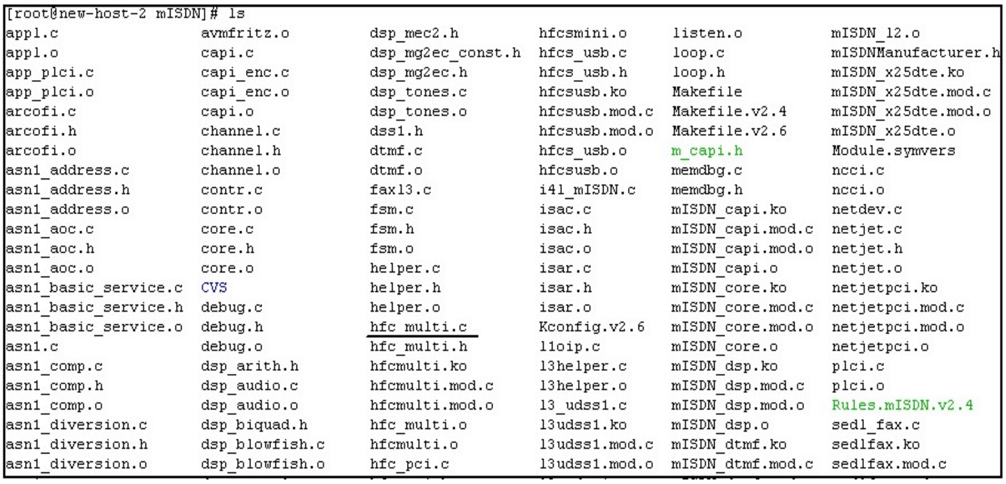
A. Download the stable version of mISDN, mISDNuser, chan_mISDN and asterisk drivers from http://www.misdn.org/index.php/Installing_mISDN, and copy the tar file to /usr/src/:



```
[root@new-host-2 src]# ls
asterisk-1.4.11         chan_misdn  linux-2.6    mISDN-1_1_2.tar.gz  mISDNuser-1_1_2.tar.gz
asterisk-1.4.11.tar.gz  kernels     mISDN-1_1_2  mISDNuser-1_1_2     redhat
[root@new-host-2 src]# cd mISDN-1_1_2
[root@new-host-2 mISDN-1_1_2]# ls
add.config  config  drivers  km_mISDN.spec  Makefile.module      misdn-init         README.misdn-init  std2kern  VERSION
CHANGES     CVS     include  Makefile       Makefile.standalone  mISDN.modprobe.d   Rules.make.ext     stddiff
[root@new-host-2 mISDN-1 1 2]#
```

Here, we are using mISDN-1_1_2 and mISDNuser_1_1_2. User can get more details from mISDN.org.

B. Download OpenVox patch called hfc_multi.c from www.openvox.com.cn. It contains a patched hfc_multi.c file; user overwrites the original hfc_multi.c file with the new hfc_multi.c under /usr/src//usr/src/mISDN-1_1_2/drivers/isdn/hardware/mISDN:

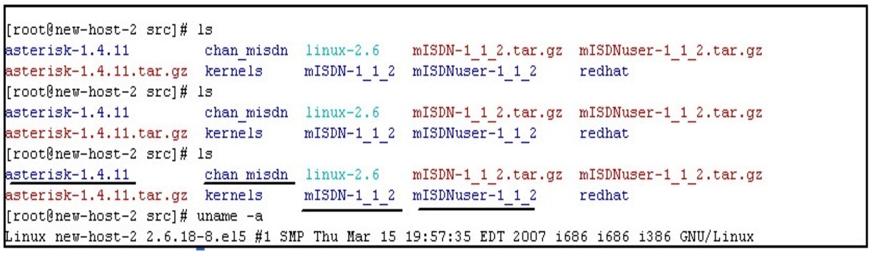


```
[root@new-host-2 mISDN]# ls
appl.c                avmfritz.o         dsp_mec2.h         hfcsmini.o     listen.o          mISDN_l2.o
appl.o                capi.c             dsp_mg2ec_const.h  hfcs_usb.c     loop.c            mISDNManufacturer.h
app_plci.c            capi_enc.c         dsp_mg2ec.h        hfcs_usb.h     loop.h            mISDN_x25dte.ko
app_plci.o            capi_enc.o         dsp_tones.c        hfcsusb.ko     Makefile          mISDN_x25dte.mod.c
arcofi.c              capi.o             dsp_tones.o        hfcsusb.mod.c  Makefile.v2.4     mISDN_x25dte.mod.o
arcofi.h              channel.c          dss1.h             hfcsusb.mod.o  Makefile.v2.6     mISDN_x25dte.o
arcofi.o              channel.h          dtmf.c             hfcs_usb.o     m_capi.h          Module.symvers
asn1_address.c        channel.o          dtmf.o             hfcsusb.o      memdbg.c          ncci.c
asn1_address.h        contr.c            faxl3.c            i4l_mISDN.c    memdbg.h          ncci.o
asn1_address.o        contr.o            fsm.c              isac.c         mISDN_capi.ko     netdev.c
asn1_aoc.c            core.c             fsm.h              isac.h         mISDN_capi.mod.c  netjet.c
asn1_aoc.h            core.h             fsm.o              isac.o         mISDN_capi.mod.o  netjet.h
asn1_aoc.o            core.o             helper.c           isar.c         mISDN_capi.o      netjet.o
asn1_basic_service.c  CVS                helper.h           isar.h         mISDN_core.ko     netjetpci.ko
asn1_basic_service.h  debug.c            helper.o           isar.o         mISDN_core.mod.c  netjetpci.mod.c
asn1_basic_service.o  debug.h            hfc_multi.c        Kconfig.v2.6   mISDN_core.mod.o  netjetpci.mod.o
asn1.c                debug.o            hfc_multi.h        l1oip.c        mISDN_core.o      netjetpci.o
asn1_comp.c           dsp_arith.h        hfcmulti.ko        l3helper.c     mISDN_dsp.ko      plci.c
asn1_comp.h           dsp_audio.c        hfcmulti.mod.c     l3helper.o     mISDN_dsp.mod.c   plci.o
asn1_comp.o           dsp_audio.o        hfcmulti.mod.o     l3_udss1.c     mISDN_dsp.mod.o   Rules.mISDN.v2.4
asn1_diversion.c      dsp_biquad.h       hfc_multi.o        l3udss1.ko     mISDN_dsp.o       sedl_fax.c
asn1_diversion.h      dsp_blowfish.c     hfcmulti.o         l3udss1.mod.c  mISDN_dtmf.ko     sedlfax.ko
asn1_diversion.o      dsp_blowfish.o     hfc_pci.c          l3udss1.mod.o  mISDN_dtmf.mod.c  sedlfax.mod.c
```

C. Make links with kernel source:

ln -s /usr/src/kernels/2.6.18-8.el5-i686/ /usr/src/linux-2.6

Here, under /usr/src there is kernel source, user must create link linux-2.6 under /usr/src/. There are many files under /usr/src/, please check:



```
[root@new-host-2 src]# ls
asterisk-1.4.11         chan_misdn  linux-2.6    mISDN-1_1_2.tar.gz  mISDNuser-1_1_2.tar.gz
asterisk-1.4.11.tar.gz  kernels     mISDN-1_1_2  mISDNuser-1_1_2     redhat
[root@new-host-2 src]# ls
asterisk-1.4.11         chan_misdn  linux-2.6    mISDN-1_1_2.tar.gz  mISDNuser-1_1_2.tar.gz
asterisk-1.4.11.tar.gz  kernels     mISDN-1_1_2  mISDNuser-1_1_2     redhat
[root@new-host-2 src]# ls
asterisk-1.4.11         chan_misdn  linux-2.6    mISDN-1_1_2.tar.gz  mISDNuser-1_1_2.tar.gz
asterisk-1.4.11.tar.gz  kernels     mISDN-1_1_2  mISDNuser-1_1_2     redhat
[root@new-host-2 src]# uname -a
Linux new-host-2 2.6.18-8.el5 #1 SMP Thu Mar 15 19:57:35 EDT 2007 i686 i686 i386 GNU/Linux
```

D. Compiling mISDN, mISDNuser,chan_mISDN and asterisk

```
cd /usr/src/mISDN-1_1_2
make
make install
std2kern

cd /usr/src/mISDN-1_1_2
make
make install
```

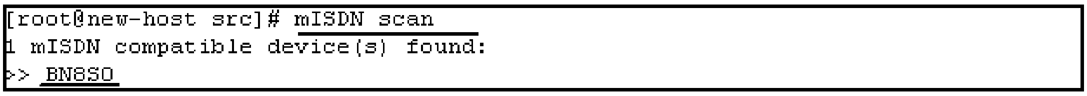
```
cd /usr/src/chan_misdn
make
make install

cd /usr/src/asterisk-1.4.11
./configure
make menuselect
```

Now you should enable chan_misdn in the Channel Driver Section and reinstall asterisk with "make install".

After those procedures you should have the current mISDN releases installed and the current chan_misdn with asterisk.

E. Modifying and loading modules for mISDN

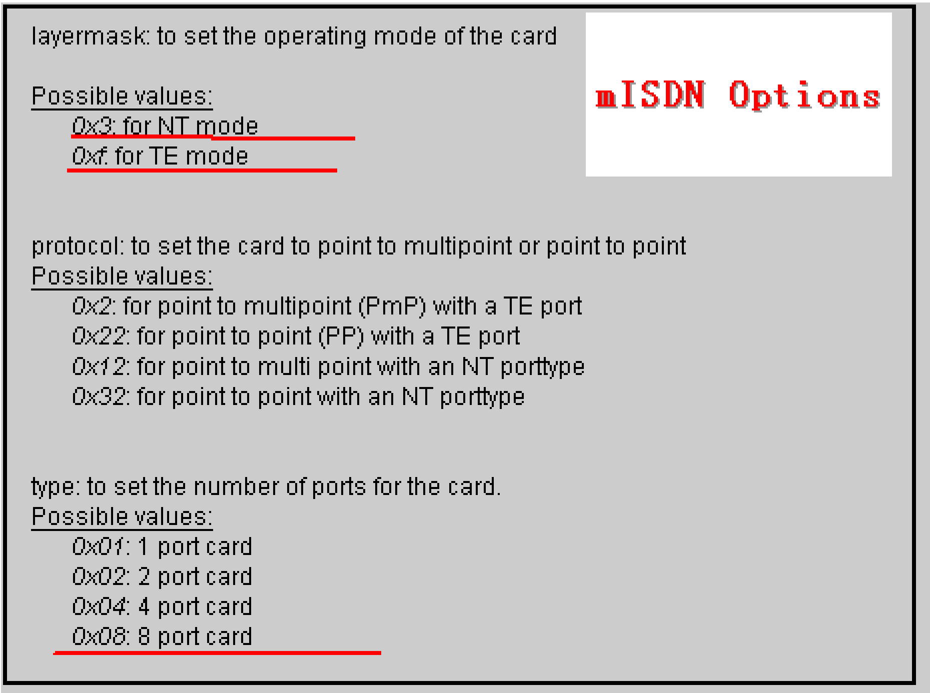
Execute: mISDN scan



```
[root@new-host src]# mISDN scan
1 mISDN compatible device(s) found:
>> BN8S0
```

Please take note here:

1) B800P uses different chip, the setting will be different from B200P/400P. The command mISDN start will generate an incorrect misdn.conf under /etc. To solve that problem, user has to edit the mISDN from /usr/sbin directory. Please make sure the options with different setting modes is meet the requirements.



mISDN Options

layermask: to set the operating mode of the card

Possible values:
    0x3: for NT mode
    0xf: for TE mode

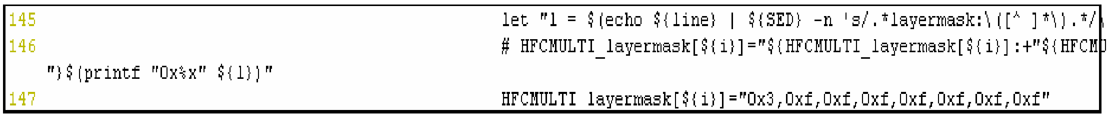
protocol: to set the card to point to multipoint or point to point
Possible values:
    0x2: for point to multipoint (PmP) with a TE port
    0x22: for point to point (PP) with a TE port
    0x12: for point to multi point with an NT porttype
    0x32: for point to point with an NT porttype

type: to set the number of ports for the card.
Possible values:
    0x01: 1 port card
    0x02: 2 port card
    0x04: 4 port card
    0x08: 8 port card

For example, if the port 1 is set into NT mode, others are TE mode, go to line 145 and add one line in the mISDN and comment the above line. The edition is shown as follow:

The **type** should be **0x08** // for B800P with 1 NT mode and 7 TE mode

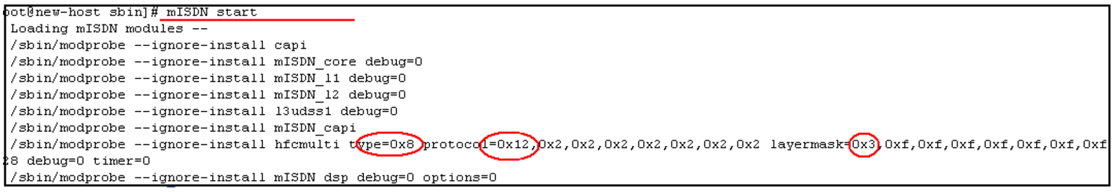
The **layermask** should be: **0x3,0xf,0xf,0xf,0xf,0xf,0xf,0xf**



```
145                                                        let "l = $(echo ${line} | ${SED} -n 's/.*layermask:\([^ ]*\).*/\
146                                                        # HFCMULTI_layermask[${i}]="${HFCMULTI_layermask[${i}]:+"${HFCMU
    ")$(printf "0x%x" ${l})"
147                                                        HFCMULTI layermask[${i}]="0x3,0xf,0xf,0xf,0xf,0xf,0xf,0xf"
```

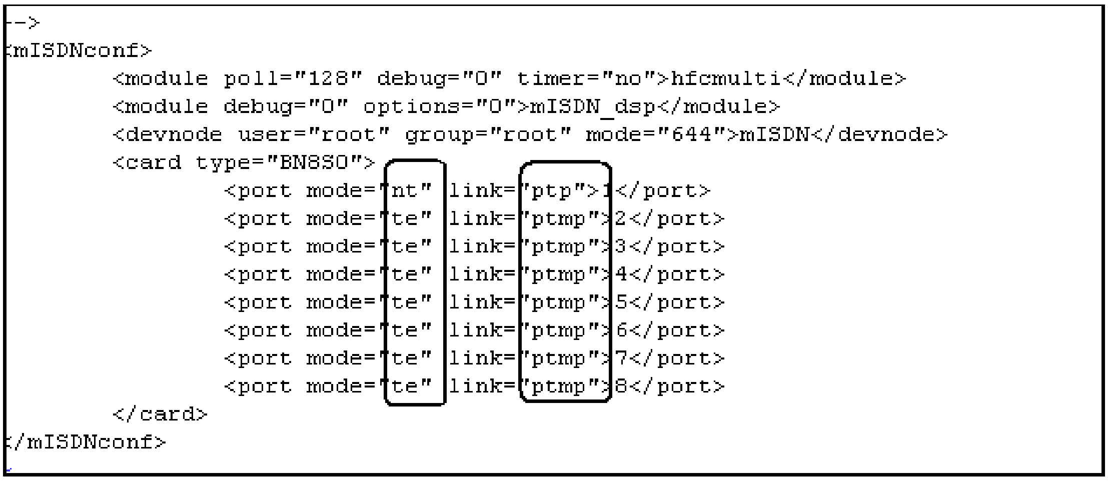
2) Starting mISDN:

Execute: **mSIDN start**



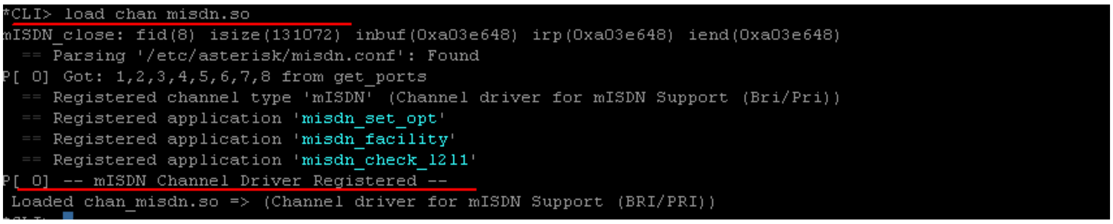
```
oot@new-host sbin]# mISDN start
 Loading mISDN modules --
 /sbin/modprobe --ignore-install capi
 /sbin/modprobe --ignore-install mISDN_core debug=0
 /sbin/modprobe --ignore-install mISDN_l1 debug=0
 /sbin/modprobe --ignore-install mISDN_l2 debug=0
 /sbin/modprobe --ignore-install l3udss1 debug=0
 /sbin/modprobe --ignore-install mISDN_capi
 /sbin/modprobe --ignore-install hfcmulti type=0x8 protocol=0x12,0x2,0x2,0x2,0x2,0x2,0x2,0x2 layermask=0x3,0xf,0xf,0xf,0xf,0xf,0xf,0xf
28 debug=0 timer=0
 /sbin/modprobe --ignore-install mISDN dsp debug=0 options=0
```

vi /etc/mISDN.conf, the part of it looks like this:



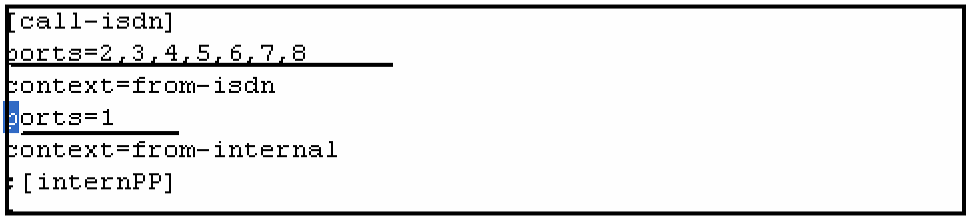
```
-->
<mISDNconf>
        <module poll="128" debug="0" timer="no">hfcmulti</module>
        <module debug="0" options="0">mISDN_dsp</module>
        <devnode user="root" group="root" mode="644">mISDN</devnode>
        <card type="BN8S0">
                <port mode="nt" link="ptp">1</port>
                <port mode="te" link="ptmp">2</port>
                <port mode="te" link="ptmp">3</port>
                <port mode="te" link="ptmp">4</port>
                <port mode="te" link="ptmp">5</port>
                <port mode="te" link="ptmp">6</port>
                <port mode="te" link="ptmp">7</port>
                <port mode="te" link="ptmp">8</port>
        </card>
</mISDNconf>
```

Execute: **load chan_misdn.so** in asterisk console

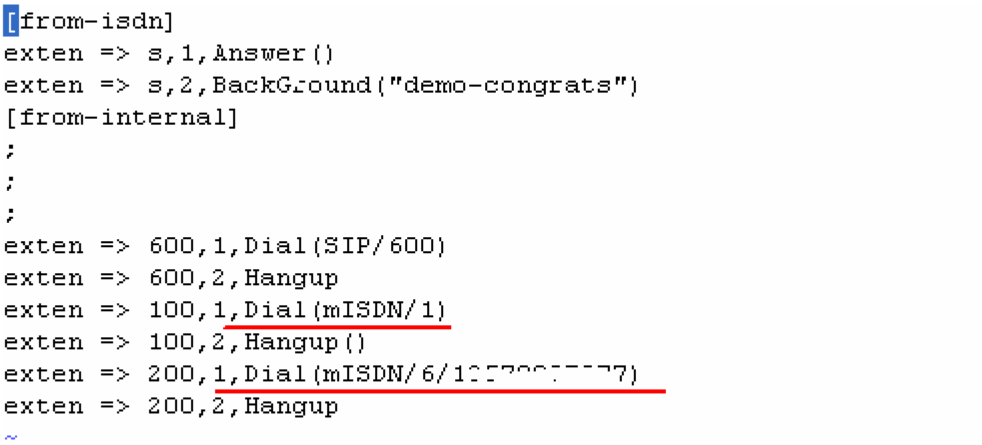


```
*CLI> load chan misdn.so
mISDN_close: fid(8) isize(131072) inbuf(0xa03e648) irp(0xa03e648) iend(0xa03e648)
  == Parsing '/etc/asterisk/misdn.conf': Found
P[ 0] Got: 1,2,3,4,5,6,7,8 from get_ports
  == Registered channel type 'mISDN' (Channel driver for mISDN Support (Bri/Pri))
  == Registered application 'misdn_set_opt'
  == Registered application 'misdn_facility'
  == Registered application 'misdn_check_l2l1'
P[ 0] -- mISDN Channel Driver Registered --
 Loaded chan_misdn.so => (Channel driver for mISDN Support (BRI/PRI))
```

The misdn.conf and extensions.conf are shown as follow:

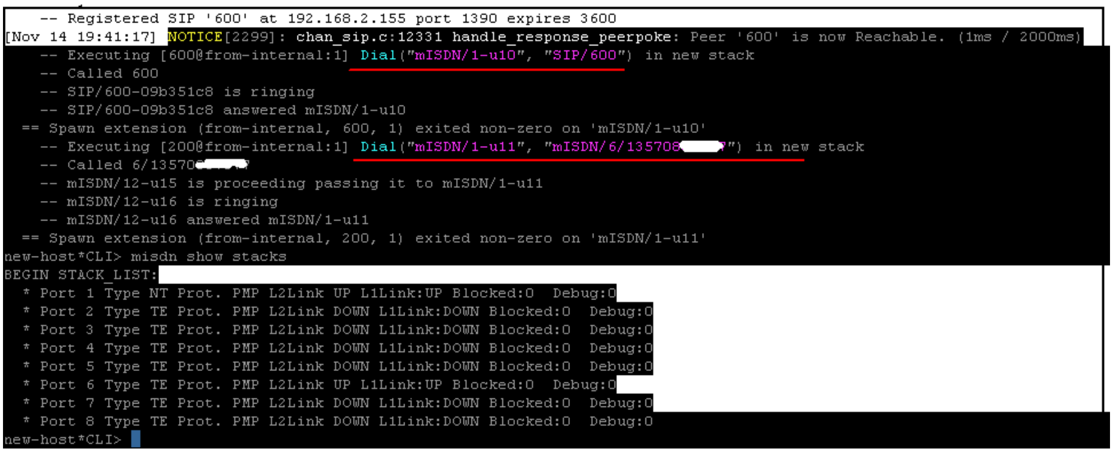


```
[call-isdn]
ports=2,3,4,5,6,7,8
context=from-isdn
ports=1
context=from-internal
;[internPP]
```



```
[from-isdn]
exten => s,1,Answer()
exten => s,2,BackGround("demo-congrats")
[from-internal]
;
;
;
exten => 600,1,Dial(SIP/600)
exten => 600,2,Hangup
exten => 100,1,Dial(mISDN/1)
exten => 100,2,Hangup()
exten => 200,1,Dial(mISDN/6/1[redacted]7)
exten => 200,2,Hangup
~
```

The console shows that user calls to SIP and makes outbound call by using ISDN ISDN phone:



```
    -- Registered SIP '600' at 192.168.2.155 port 1390 expires 3600
[Nov 14 19:41:17] NOTICE[2299]: chan_sip.c:12331 handle_response_peerpoke: Peer '600' is now Reachable. (1ms / 2000ms)
    -- Executing [600@from-internal:1] Dial("mISDN/1-u10", "SIP/600") in new stack
    -- Called 600
    -- SIP/600-09b351c8 is ringing
    -- SIP/600-09b351c8 answered mISDN/1-u10
  == Spawn extension (from-internal, 600, 1) exited non-zero on 'mISDN/1-u10'
    -- Executing [200@from-internal:1] Dial("mISDN/1-u11", "mISDN/6/135708[redacted]7") in new stack
    -- Called 6/13570[redacted]
    -- mISDN/12-u15 is proceeding passing it to mISDN/1-u11
    -- mISDN/12-u16 is ringing
    -- mISDN/12-u16 answered mISDN/1-u11
  == Spawn extension (from-internal, 200, 1) exited non-zero on 'mISDN/1-u11'
new-host*CLI> misdn show stacks
BEGIN STACK_LIST:
  * Port 1 Type NT Prot. PMP L2Link UP L1Link:UP Blocked:0  Debug:0
  * Port 2 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
  * Port 3 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
  * Port 4 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
  * Port 5 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
  * Port 6 Type TE Prot. PMP L2Link UP L1Link:UP Blocked:0  Debug:0
  * Port 7 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
  * Port 8 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0  Debug:0
new-host*CLI>
```

Notes:
Test environments:
OS: Centos 5
Kernel version: 2.6.18-8.15
Asterisk version: Asterisk-1.4.11
mISDN version: mISDN-1_1_2
Hardware：OpenVox B800P

If user wants to check hardware setting, ping assignments and references, please check from: http://www.openvox.com.cn/downloadsFile/1194838660.pdf

Some Concerns:

1) LEDs will turn into green and blink if the drivers are loaded.
2) Misdn show channels commands show nothing.
3) When calls coming, the LED will be turned into green status for very short while