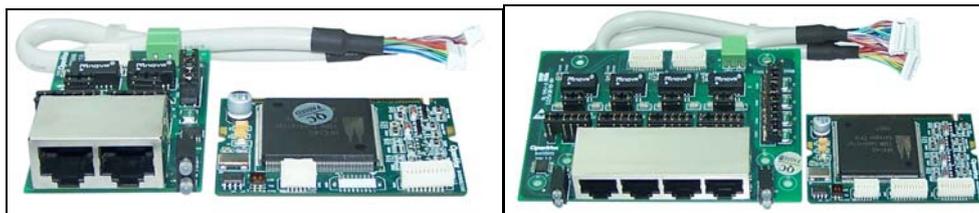


OpenVox

深圳开源通信有限公司

OpenVox-Best Cost Effective Asterisk Cards

OpenVox B200M/B400M User Manual for mISDN



Written by: James. zhu

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

Date: 23/04/2008

Version: 0. 01



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 Technical Support: support@openvox.com.cn, zhulizhong@gmail.com, james.zhu@openvox.cn

Business Hours: 9:00AM-18:00PM from Monday-Friday

URL: www.openvox.com.cn

Thank You for Choosing OpenVox Products!

Table of Contents

Chapter 1	Overview.....	4
Chapter 2	Card Installation and Configuration.....	6
Chapter 3	Hardware Setting	12

Chapter 1 Overview

1. What is B200M/B400M

B200M/B400M series is a Mini PCI type III BRI card supporting 2/4 BRI S/T interfaces. NT/TE mode can be independently configured on each of 2/4 ports.

B200M/B400M series 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:

Two/four integrated S/T interfaces

ITU-T I.430 and TBR 3 certified and S/T ISDN supporting in TE and NT mode

DTMF detection on all B-channels

Multiparty audio conferences bridge

Support mini PCI type III

Designed for low-power systems

Support AskoziaPBX system, trixbox, Elastix and other asterisk based distributions

Support VIA, PC Engines motherboard and AMD geode based motherboard

Each port can be configured for TE or NT mode

Support Bristuff, ISDN4BSD and mISDN driver.

RoHS compliant

Certificates: CE and 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.

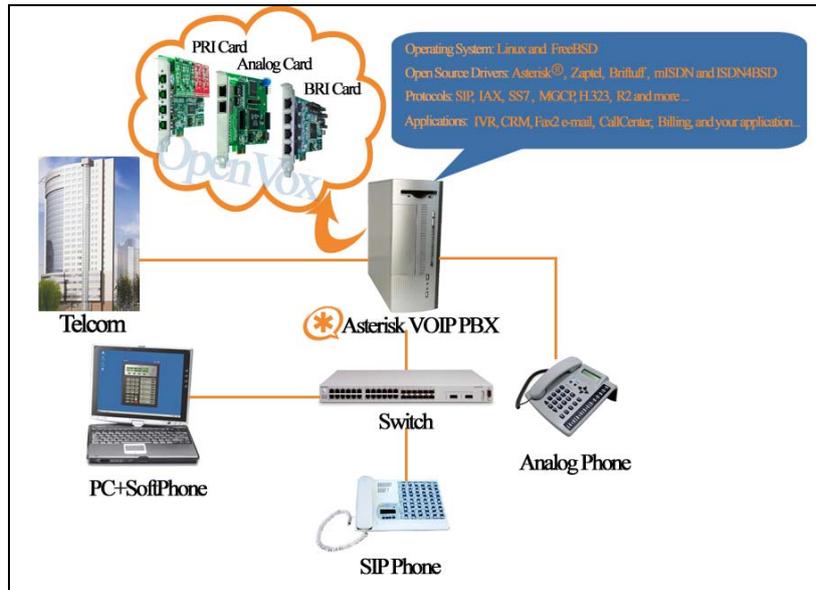


Figure 1: Asterisk_OpenVox Setup

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 B200M/B400M, user should follow the steps.

A. Setting Card ID Switch

If user wants to install more than one cards of B200M/B400M in one pc, you should take care of the card id switch. It has two rules, which user must follow:

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.

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

1. If a port will work on NT mode, you should set jumper to CONNECT (ON).
2. If a port will 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).
3. Use the connection cable to connect B400MM and B400MS. On the B400MM, there are two slots for RJ45 expansion board; the two slots are used to connect the slots on B400MS. The connection should be:

Con 1 on B400MS <-> Con 2 on B400MM

Con 2 on B400MS <-> Con 4 on B400MM

C. Power Feeding Connector

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

✧ If the port will work 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 four power feeding connectors is CONNECT(ON), plug in the DC input power supply that can provide 34-42 V power supply. User can buy OpenVox **PFM100** for the power supply.

E. PCM IN/PCM OUT

These are for future usages.

- 2) Power off PC, remembering unplug the AC power cable
- 3) Insert B200M/B400M into a mini PCI slot
- 4) Plug the power supply cable to power feeding input jack if need providing power to external equipment, please refer jumper setting section for the detail
- 5) Plug back the AC power cable, and power on PC

2. Software Installation and Setup

B200M/B400M supports original mISDN driver form http://www.misdn.org/index.php/Installing_mISDN. User can download it from that URL. There are few steps to install the driver drivers.

1) Checking the B400M hardware by command: `lspci -vvvvvvvvv`

```
02:08.0 ISDN controller: Cologne Chip Designs GmbH ISDN network Controller [HFC-4S] (rev 01)
Subsystem: Cologne Chip Designs GmbH Unknown device e888
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 db00 [size=8]
Region 1: Memory at d7ff9000 (32-bit, non-prefetchable) [disabled] [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
```

Here, we are using mISDN-1_1_7_2 and mISDNuser_1_1_7_2. Users can get more details from mISDN.org.

- B. 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/.

- C. Compiling mISDN, mISDNuser, chan_mISDN and asterisk

```
cd /usr/src/mISDN-1_1_7_2
```

```
make
```

```
make install
```

```
std2kern
```

```
cd /usr/src/mISDNuser-1_1_7_2
```

```
make
```

```
make install
```

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

```
cd /usr/src/asterisk-1.4.8  
./configure  
make menuselect
```

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

After that procedure you should have the current mISDN releases installed and the current chan_misdn with asterisk.

- D. Scan mISDN and configure mISDN by:

```
cd /usr/sbin  
mISDN scan // it will detect the isdn card  
mISDN config // it will configure the card  
mISDN start // it will start the card
```

- E. Check the mISDN.conf and misdn.conf

```
vi /etc/mISDN.conf, and edit the mISDN.conf like this:
```

```

<?xml version="1.0"?>
<!--
-----
Card Type: BN2S0, BN4S0, BN8S0
Card Attributes: ulaw=(yes|no), dtmf=(yes|no), pcm_slave=(yes|no),
                 ignore_pcm_frameclock=(yes|no), rxclock=(yes|no),
                 crystalclock=(yes|no), watchdog=(yes|no)
Port Attributes: mode=(te|nt), link=(ptp|ptmp), master-clock=(yes|no),
                 capi=(yes|no)
-----
Card Type: BN2E1
Card Attributes: ulaw=(yes|no), dtmf=(yes|no), pcm_slave=(yes|no),
                 ignore_pcm_frameclock=(yes|no), rxclock=(yes|no),
                 crystalclock=(yes|no), watchdog=(yes|no)
Port Attributes: mode=(te|nt), link=(ptp|ptmp), optical=(yes|no), los=(yes|no),
                 ais=(yes|no), slip=(yes|no), nocrc4=(yes|no), capi=(yes|no)
-----
Card Type: hfcmulti, avmfritz, w6692pci
Port Attributes: mode=(te|nt), link=(ptp|ptmp), capi=(yes|no)
-----
Module: hfcmulti
Options: poll=<number>, pcm=<number>, debug=<number>, timer=(yes|no)
-----
Module: mISDN_dsp
Options: debug=<number>, options=<number>, poll=<number>,
        dtmfthreshold=<number>
-----
-->
<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="BN4S0">
    <port mode="te" link="ptp">1</port>
    <port mode="te" link="ptp">2</port>
    <port mode="te" link="ptp">3</port>
    <port mode="te" link="ptp">4</port>
  </card>
</mISDNconf>

```

Copy the misdn.conf under chan_misdn to /etc/asterisk directory

Edit the setting for your system:

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

```

;
; change this threshold to enable dejitter functionality
;
jitterbuffer_upper_threshold=0

;
; change this to yes, if you want to bridge a mISDN data channel to
; another channel type or to an application.
;
hdlc=no

[call-isdn]
; define your ports, e.g. 1,2 (depends on mISDN-driver loading order)
ports=1,2,3,4
; context where to go to when incoming Call on one of the above ports
context=demo

```

vi /etc/asterisk/extensions.conf, the dialplan shows as the follow:

```
[demo]
;
; We start with what to do when a call first comes in.
;
exten => s,1,Wait(1) ; Wait a second, just for fun
exten => s,n,Answer ; Answer the line
exten => s,n,Set(TIMEOUT(digit)=5) ; Set Digit Timeout to 5 seconds
exten => s,n,Set(TIMEOUT(response)=10) ; Set Response Timeout to 10 seconds
exten => s,n(restart),BackGround(demo-congrats) ; Play a congratulatory message
exten => s,n(instruct),BackGround(demo-instruct) ; Play some instructions
exten => s,n,WaitExten ; Wait for an extension to be dialed.

[from-internal]
exten => 100,1,Dial(mISDN/1/135[REDACTED])
exten => 100,2,Hangup
```

The dialplan is linked with misdn.conf. The example shows that the port 1 is used for inbound and outbound calls.

F. Running mISDN and asterisk:

Execute: mISDN start // if it has started, you do not need run it again

Execute: asterisk -vvvvvvc // start asterisk server

The screens show the asterisk loading chan_misdn.so and making outbound call, inbounds:

```
*CLI>
*CLI> misdn show stacks
BEGIN STACK LIST:
 * Port 1 Type TE Prot. PMP L2Link UP L1Link:UP Blocked:0 Debug:0
 * Port 2 Type TE Prot. PMP L2Link UP L1Link:UP Blocked:0 Debug:0
 * Port 3 Type TE Prot. PMP L2Link UP L1Link:UP Blocked:0 Debug:0
 * Port 4 Type TE Prot. PMP L2Link UP L1Link:UP Blocked:0 Debug:0
*CLI>
```

```
== Spawn extension (from-internal, 100, 1) exited non-zero on 'SIP/500-0979e540'
== Starting mISDN/2-u15 at demo,,1 failed so falling back to exten 's'
rnxbox1*CLI> == Starting mISDN/2-u15 at demo,,1 failed so falling back to exten 's'
-- Executing [s@demo:1] Wait("mISDN/2-u15", "1") in new stack
rnxbox1*CLI> -- Executing [s@demo:1] Wait("mISDN/2-u15", "1") in new stack
-- Executing [s@demo:2] Answer("mISDN/2-u15", "") in new stack
-- Executing [s@demo:3] BackGround("mISDN/2-u15", "demo-instruct") in new stack
-- Executing [s@demo:2] Answer("mISDN/2-u15", "") in new stack
-- Executing [s@demo:3] BackGround("mISDN/2-u15", "demo-instruct") in new stack
-- <mISDN/2-u15> Playing 'demo-instruct' (language 'en')
rnxbox1*CLI> -- <mISDN/2-u15> Playing 'demo-instruct' (language 'en')
== Spawn extension (demo, s, 3) exited non-zero on 'mISDN/2-u15'
== Spawn extension (demo, s, 3) exited non-zero on 'mISDN/2-u15'
```

```
-- Registered extension context 'trunkint'
rnxbox1*CLI> -- Executing [100@from-internal:1] Dial("SIP/500-0979e540", "mISDN/1/135[REDACTED]")
-- Called 1/135[REDACTED]
-- Executing [100@from-internal:1] Dial("SIP/500-0979e540", "mISDN/1/135[REDACTED]") in new stack
-- Called 1/135[REDACTED]
rnxbox1*CLI> -- mISDN/1-u13 is proceeding passing it to SIP/500-0979e540
-- mISDN/1-u13 is proceeding passing it to SIP/500-0979e540
rnxbox1*CLI> -- mISDN/1-u14 is ringing
-- mISDN/1-u14 is ringing
rnxbox1*CLI> == Spawn extension (from-internal, 100, 1) exited non-zero on 'SIP/500-0979e540'
== Spawn extension (from-internal, 100, 1) exited non-zero on 'SIP/500-0979e540'
```

Notes:

Test environments:

OS: trixbox-2.6.0.7

Kernel version: 2.6.18-53.1.4.el5

Asterisk version: Asterisk-1.4.8

mISDN version: mISDN-1_1_7_2

Hardware: OpenVox B400M, workable for B200M

References:

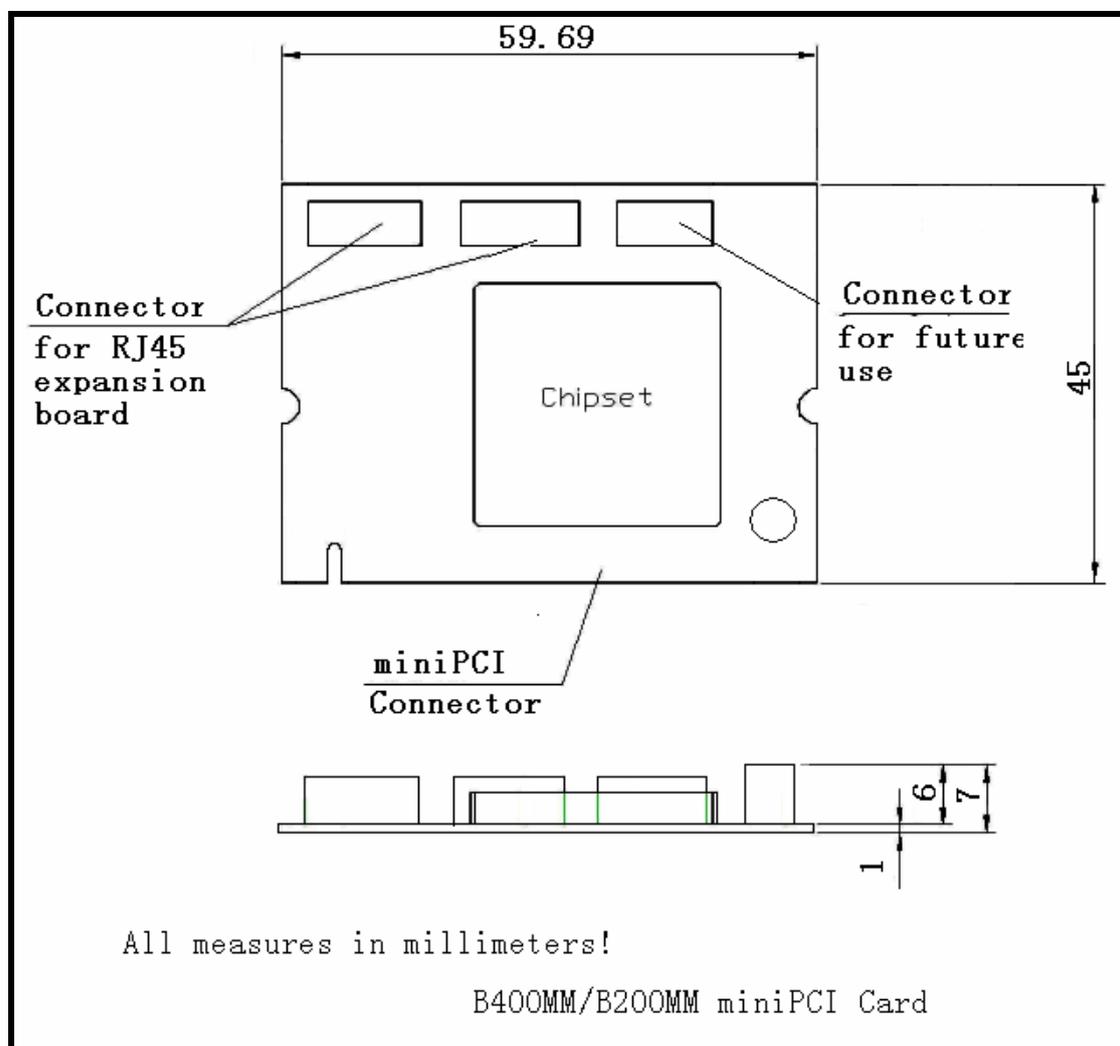
www.openvox.com.cn

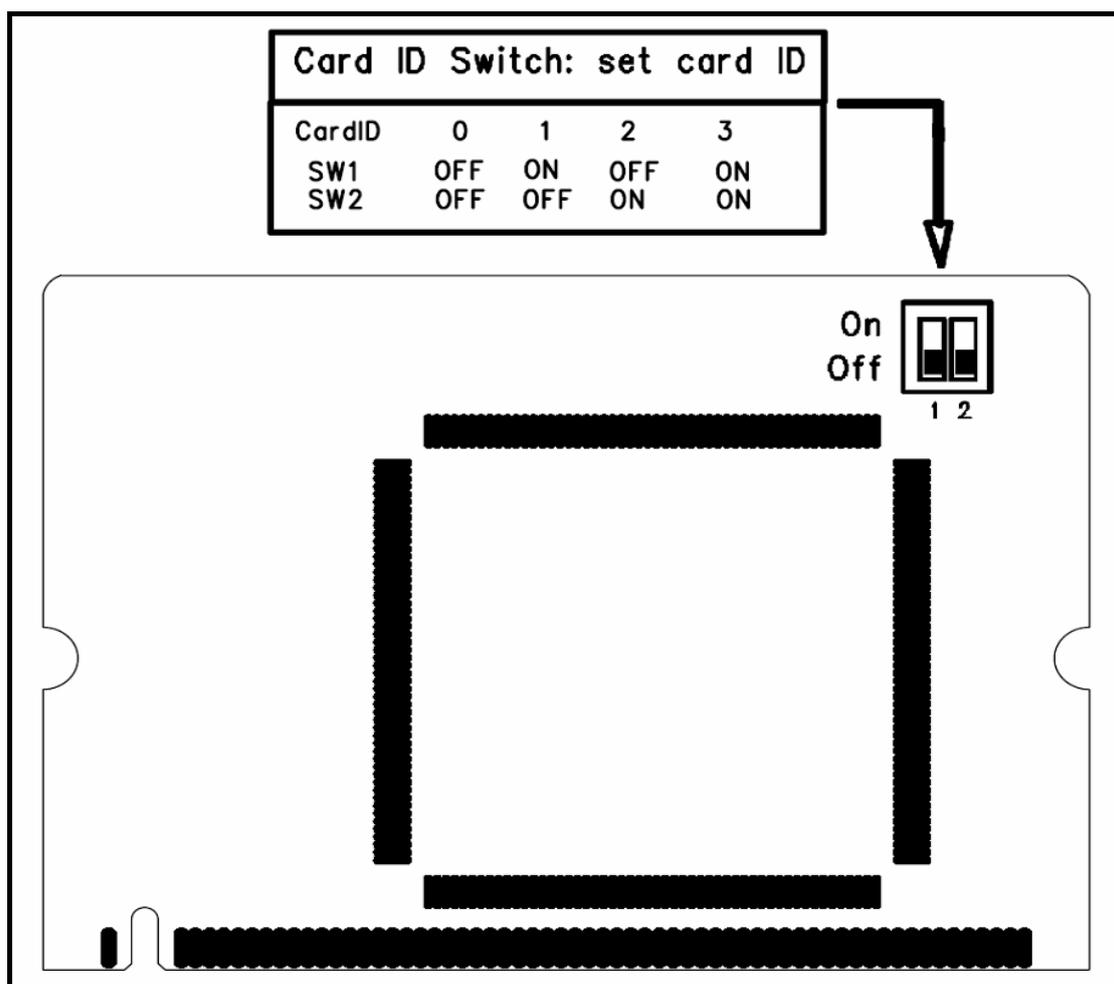
www.asterisk.org

www.misdn.org

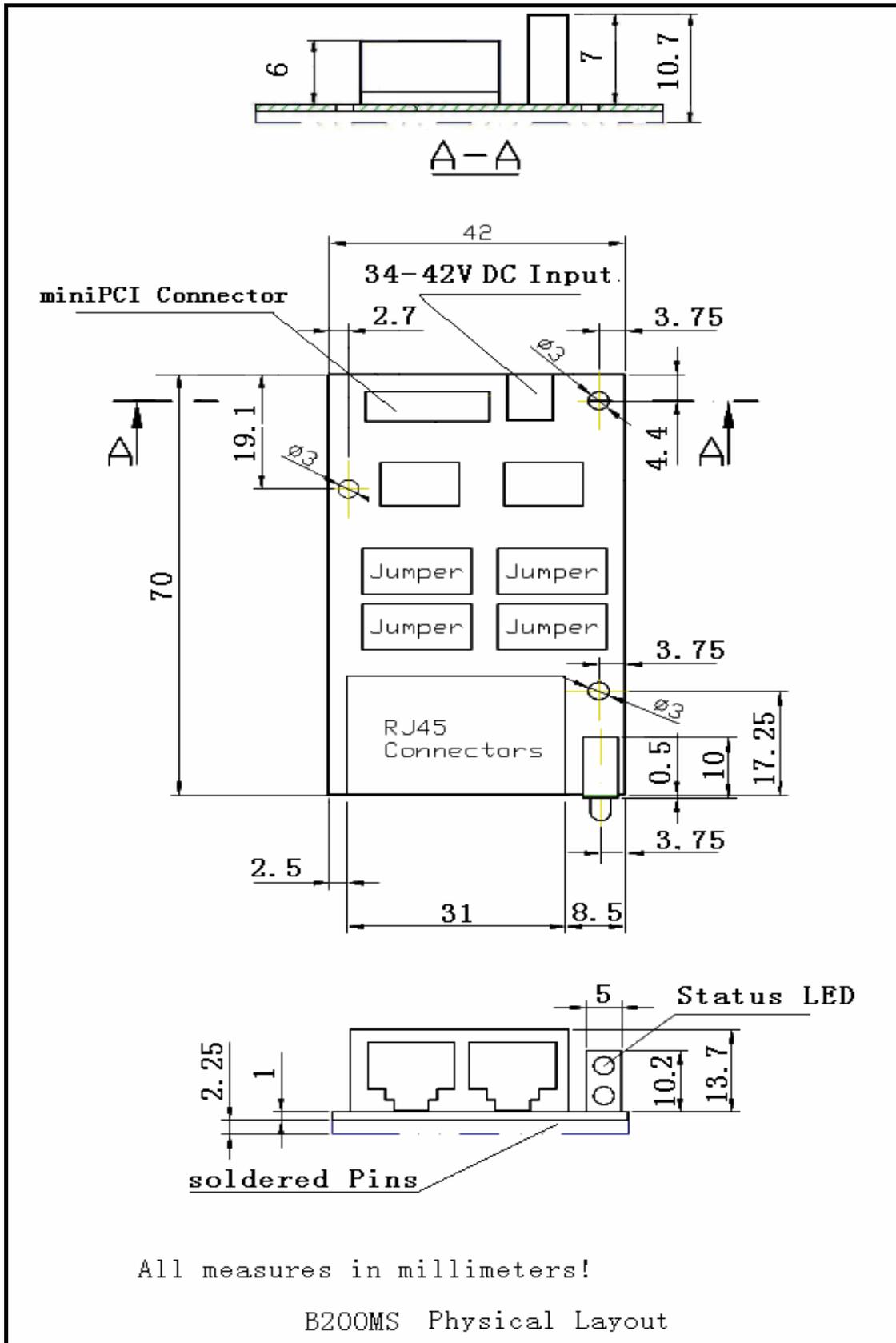
www.voip-info.org

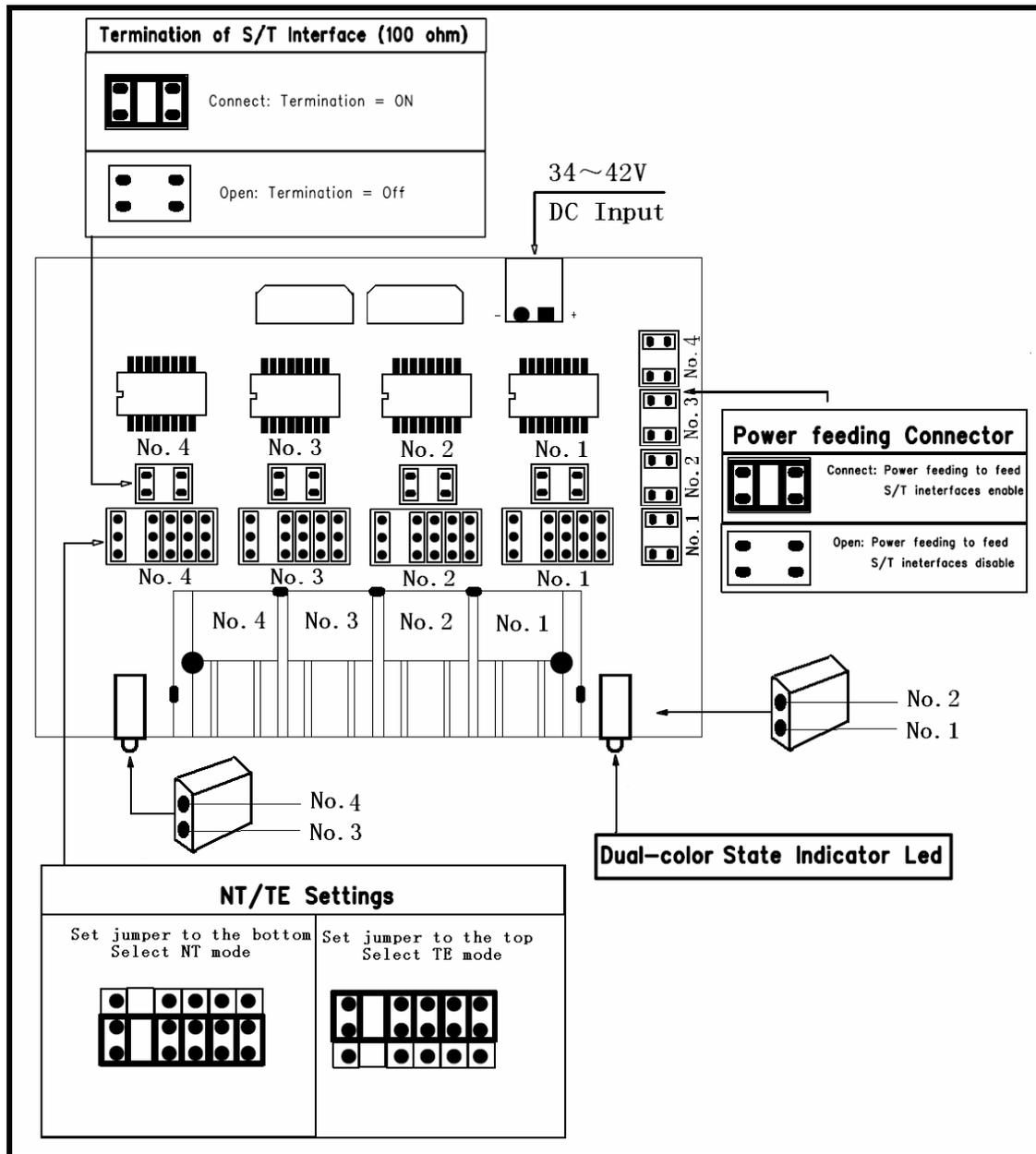
Chapter 3 Hardware Setting



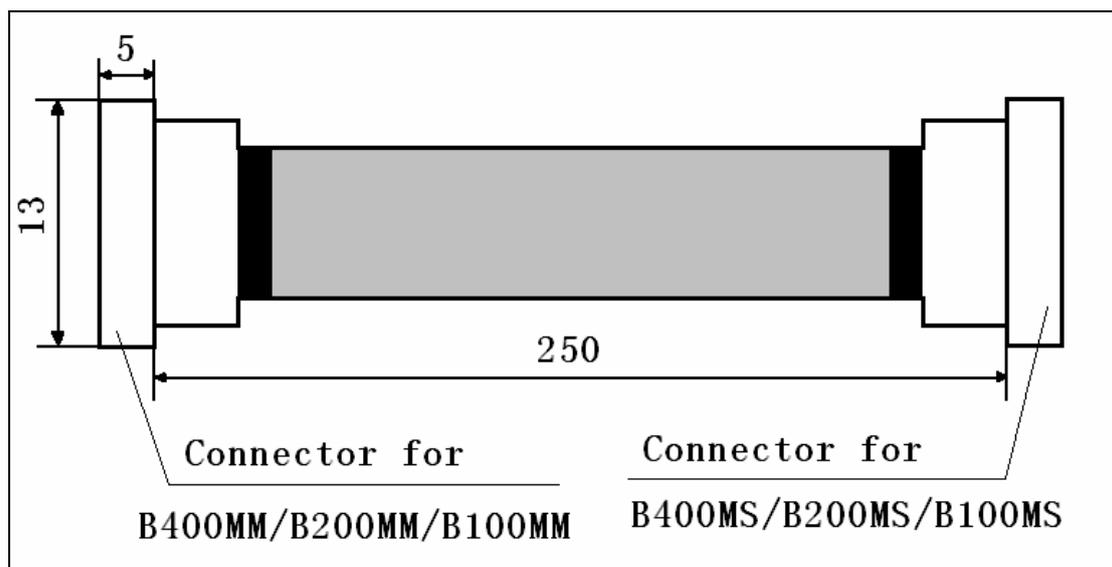


Card ID setting





B400MS Jumper Setting



BX00MM/BX00MS Connection Cable