

AX4D+EC128

ATCOM® Digital Card AX4D/AXE4D

Product Guide

25/06/2011

Written by Robert.Ao

Content

CONTACT ATCOM	2
CHAPTER 1 THE INTRODUCTION OF AX4D	3
CHAPTER 2 HARDWARE INTRODUCTION	5
CHAPTER 3 SOFTWARE INSTALLATION.....	9
CHAPTER 4 SOFTWARE CONFIGURATION	11
CHAPTER 5 LOOPBACK TEST	14
CHAPTER 6 REFERENCE	19

Contact ATCOM

The Introduction of ATCOM

Founded in 1998, ATCOM technology has been always endeavoring in the R&D and manufacturing of the internet communication terminals. The product line of ATCOM includes IP Phone, USB Phone, IP PBX, VoIP gateway and Asterisk Card.

Contact sales:

Address	District C, east of 2nd floor, #3, Crown industry buildings, Chegongmiao Industry area, Futian district, Shenzhen, China
Tel	+(86)755-23487618
Fax	+(86)755-23485319
E-mail	sales@atcomemail.com

Contact Technical Support:

Tel	+(86)755-23481119
E-mail	Support@atcomemail.com

Website address: <http://www.atcom.cn/>

ATCOM Wiki Website: http://www.openippbx.org/index.php?title=Main_Page

Download Center: <http://www.atcom.cn/download.html>

Chapter 1 the Introduction of AX4D

Overview of the AX4D

AX4D Asterisk card is the telephony PCI card which supports four ISDN PRI E1/T1/J1 ports, optional with teleco-grade hardware echo cancellation. Using AX4D digital PRI card, open source Asterisk PBX and stand alone PC, users can create their IP PBX telephony solution include all the sophisticated features of traditional PBX, and extend features such as voicemail in IP PBX.

Features

AX4D: Four ISDN PRI E1/T1/J1 ports
Support original Dahdi, Zaptel(No patch)
Support Asterisk, Freeswitch, Yate
Support Elastix, Trixbox, AsteriskNOW, PBX in a Flash
100% compatible with all features of Asterisk PBX
32-bit bus master DMA data exchanges across PCI interface at 132 Mbytes/sec for minimum host processor intervention
Supports standard telephony and data protocols(Including PRI, CAS, CCS for voice and PPP, HDLC, Cisco HDLC and Frame Relay for data modes)
Supports chan_ss7 and Openr2
Optional hardware echo cancellation

Applications

ISDN PRI IP PBX
ISDN least cost router
Calling Card Platforms
IVR system
Call Center
Traditional Calls/VoIP Calls Conference
VoIP Gateway
Callback Service

Optional DSP Hardware Echo Cancellation Module

G.168–2002 echo cancellation in hardware
1024 taps/128 ms tail per channel on all channel densities
DTMF decoding and tone recognition
Voice quality enhancement: music protection, acoustic echo control, and adaptive noise reduction
Does not increase the physical size of the card, and no additional slot is required

Voice Modes

Pri CPE and PRI NET:

EuroISDN

4ESS(AT&T)

5ESS(Lucent)

DMS100

Hardware Requirement

1.6-Ghz Pentium IV

512 MB RAM

3.3V or 5V PCI 2.2 slot

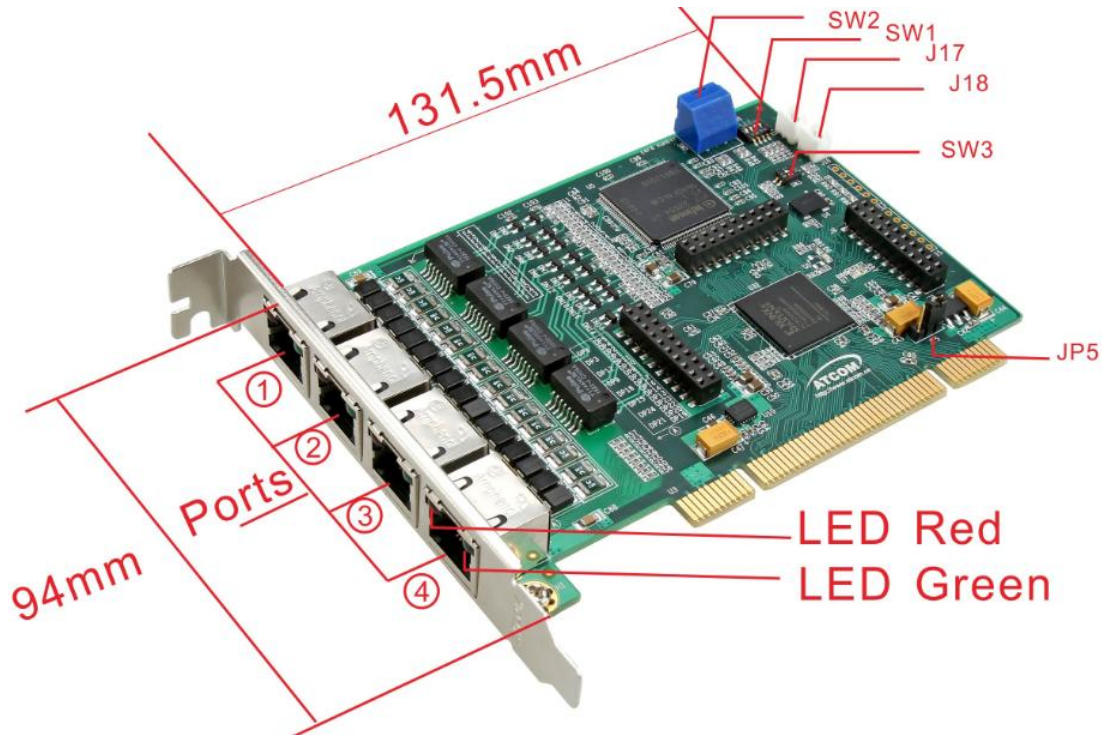
PCI Card Dimension

131.5mm (Length)*94mm (height)

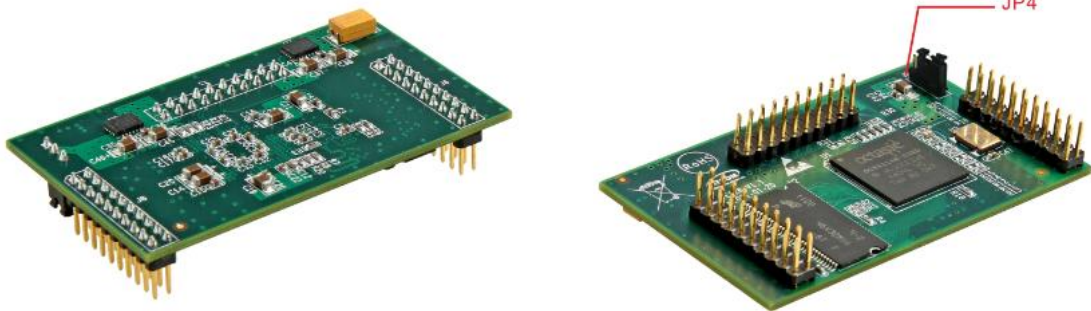
Operating System

Linux (all versions, releases and distributions from 1.0 up)

Chapter 2 Hardware Introduction



AX4D



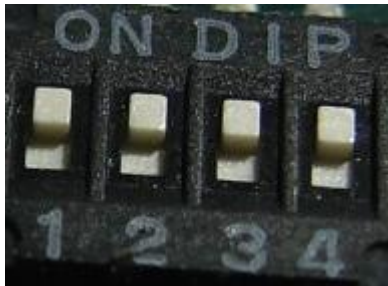
The Front and Rear View of EC128

LED Red: If the driver of the card is loaded correctly and the `/etc/dahdi/system.conf` file is configured correctly, the LED Red of the four ports will be red clearly.

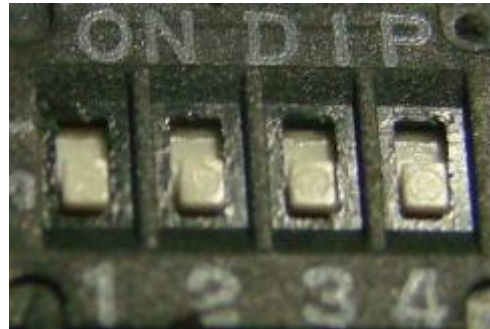
LED Green: If the PRI line is connected correctly, and it synchronizes to the other end successfully, then the LED Green will be green clearly.

If the PRI line is connected correctly, and it is synchronizing to the other end, then both of the LED Green and LED Red will be on clearly.

SW1: It is used for selecting E1, T1, J1 mode, by default ATCOM will set it up according to your requirement before shipping out. Customer can set up E1, T1/J1 like the following:

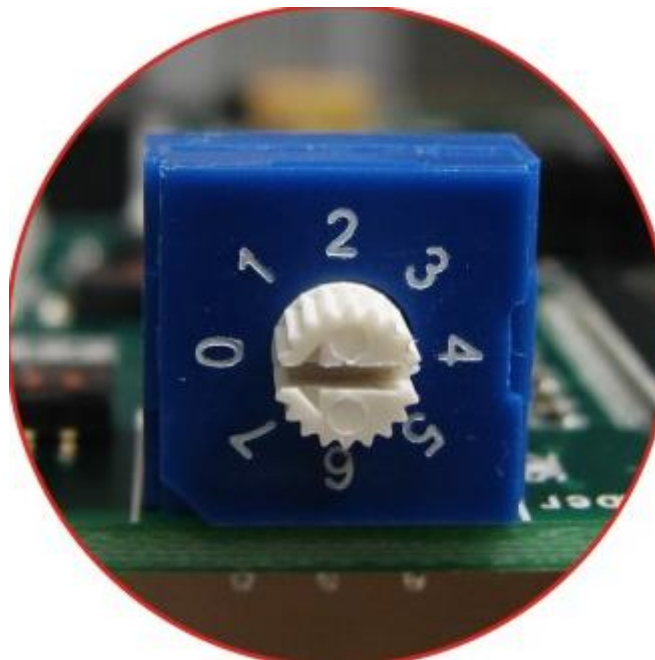


For E1 Mode

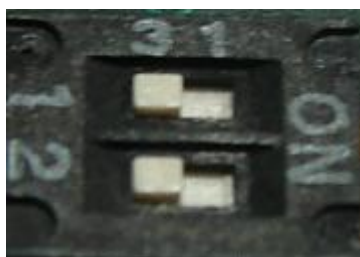


For T1 and J1 Mode

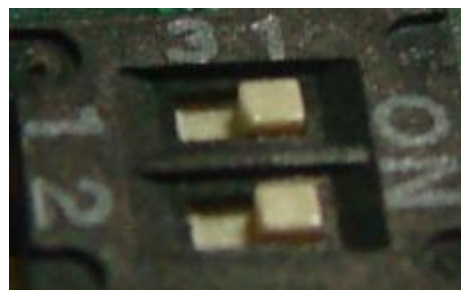
SW2: It is used when you have more than one PRI card in your server, if you have two cards in your server, then turn it to 0 in one card, and turn it to 1 in the other card; if you have four cards in your server, then turn it to 0,1,2,3 respectively.



SW3: It is used to select which clock does the card use for working, by default the card choose to use line clock, that is the way that most of E1 equipment use; the card can also use crystal on the card as working clock.

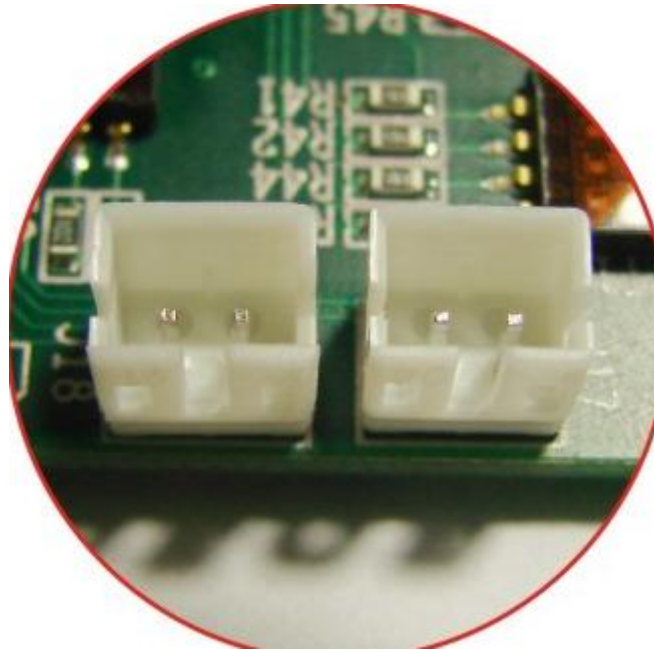


For Selecting Line Clock



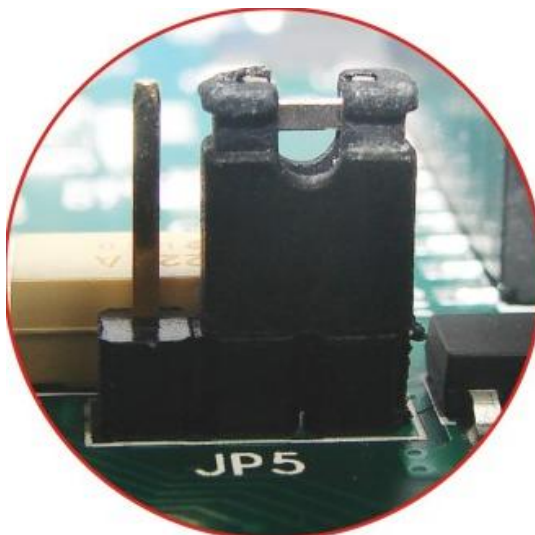
For Selecting Crystal Clock

J17, J18: They are used to connect to other cards, when you have several cards in your server. Using clock line to connect each other, all of the cards work in one server can synchronize accurately. If customer have PRI card and analog card in one server, the analog card can synchronize accurately with telecom in this way, and it is useful for fax.

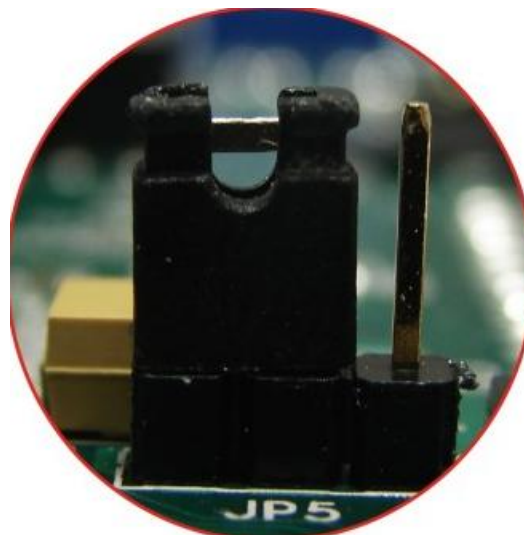


J18 and J17

JP5: It is used for selecting the PCI power feeding for the card, by default we choose the 3.3v for the card, customer do not need to reset the jumper; the card can also choose 5.0v for power feeding in the case that the 3.3v in the motherboard of PC is not enough for the card, but this phenomenon happens rarely.

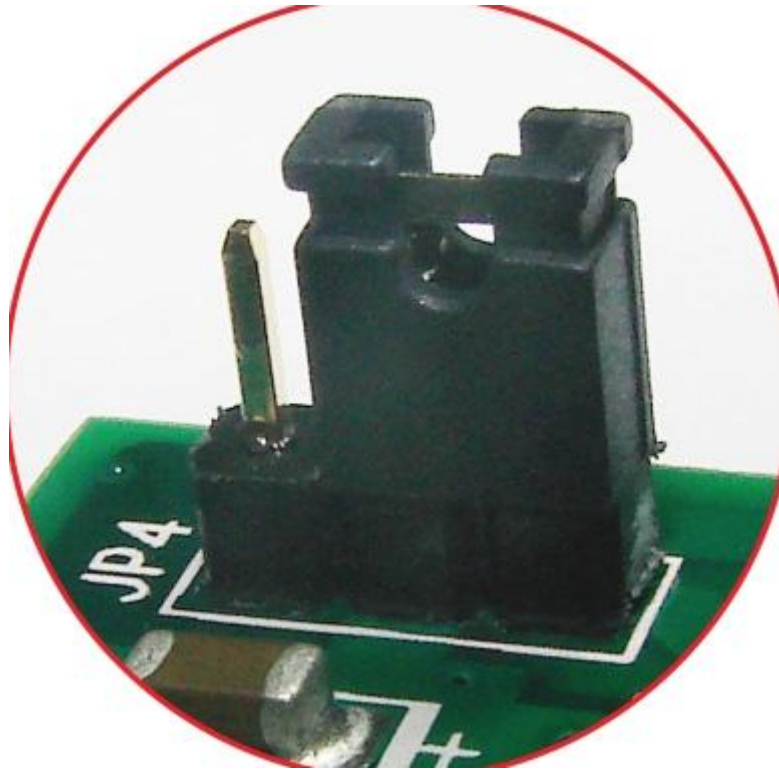


For Choosing 3.3v Power

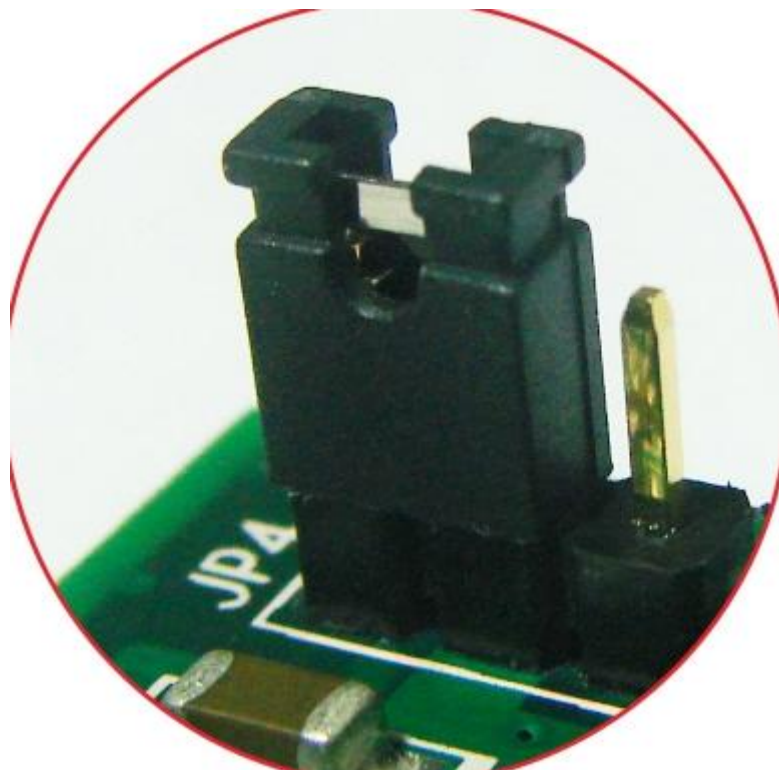


For Choosing 5.0v Power

JP4: It is used for selecting the power feeding for the hardware echo cancellation module, by default we choose the 3.3v for the card, customer do not need to reset the jumper; the card can also choose 5.0v for power feeding in the case that the 3.3v in the motherboard of PC is not enough for the card, but this phenomenon happens rarely.



For Selecting 3.3v Power



For Selecting 5.0v Power

Chapter 3 Software Installation

Test Environment:

libpri-1.4.10.2
dahdi-linux-2.3.0.1
dahdi-tools-2.3.0
asterisk-1.6.2.6
centos 5.4 (kernel version: 2.6.18-164.el5)

After inserting the card into your PCI slot and boot your server, please use the “lspci” command to check the PCI bus compatibility. From the correct output, you can see the following line:

```
-----  
03:01.0 Communication controller: Digium, Inc. Wildcard TE410P quad-span T1/E1/J1 card 3.3V  
(rev 02)  
-----
```

The TE410P will be found, if you can not see one line like the line above, please poweroff your server and try to use another PCI slot, if it still does not help, you have to check the compatibility issue between the card and your PCI bus.

1. To install asterisk and dahdi, we have to use “yum” command to install the following prerequisite packages:

```
bison bison-devel zlib zlib-devel openssl openssl-devel gnutls-devel gcc gcc-c++
```

2. Download libpri, dahdi-linux, dahdi-tools, and asterisk

```
[root@localhost src]#
```

```
wget http://downloads.asterisk.org/pub/telephony/libpri/releases/libpri-1.4.10.2.tar.gz
```

```
[root@localhost src]#
```

```
Wget
```

```
http://downloads.asterisk.org/pub/telephony/dahdi-linux/releases/dahdi-linux-2.3.0.1.tar.gz
```

```
[root@localhost src]#
```

```
wget
```

```
http://downloads.asterisk.org/pub/telephony/dahdi-tools/releases/dahdi-tools-2.3.0.tar.gz
```

```
[root@localhost src]#
```

```
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.6.2.6.tar.gz
```

3. Install libpri

```
1) [root@localhost src]# tar -xvzf libpri-1.4.10.2.tar.gz
```

```
2) [root@localhost libpri-1.4.10.2]# make
```

```
3) [root@localhost libpri-1.4.10.2]# make install
```

4. Install dahdi-linux and dahdi-tools

- 1) [root@localhost src]# tar -xvzf dahdi-linux-2.3.0.1.tar.gz
- 2) [root@localhost src]# cd dahdi-linux-2.3.0.1/drivers/dahdi/
- 3) [root@localhost dahdi-linux-2.3.0.1]# make
- 4) [root@localhost dahdi-linux-2.3.0.1]# make install

Install dahdi-tools

- 1) [root@localhost src]# tar -xvzf dahdi-tools-2.3.0.tar.gz
- 2) [root@localhost dahdi-tools-2.3.0]# ./configure
- 3) [root@localhost dahdi-tools-2.3.0]# make
- 4) [root@localhost dahdi-tools-2.3.0]# make install
- 5) [root@localhost dahdi-tools-2.3.0]# make config

5. Install asterisk

- 1) [root@localhost src]# tar -xvzf asterisk-1.6.2.6.tar.gz
- 2) [root@localhost asterisk-1.6.2.6]# ./configure
- 3) [root@localhost asterisk-1.6.2.6]# make
- 4) [root@localhost asterisk-1.6.2.6]# make install
- 5) [root@localhost asterisk-1.6.2.6]# make samples

Chapter 4 Software Configuration

1. Please use the “dahdi_genconf” command to configure the /etc/dahdi/system.conf file and generate /etc/asterisk/dahdi-channels.conf file.

```
[root@localhost ~]# dahdi_genconf
```

It does not show any output if dahdi_genconf run successfully.

After running dahdi_genconf successfully, the system.conf file will get the following configuration:

/etc/dahdi/system.conf:

```
# Autogenerated by /usr/sbin/dahdi_genconf  do not hand edit
# This file is parsed by the Dahdi Configurator, dahdi_cfg
# Span 1: TE4/0/1 "T4XXP (PCI) Card 0 Span 1" (MASTER)
span=1,1,0,ccs,hdb3,crc4
# termtype: te
bchan=1-15,17-31
dchan=16
echocanceller=mg2,1-15,17-31

# Span 2: TE4/0/2 "T4XXP (PCI) Card 0 Span 2"
span=2,2,0,ccs,hdb3,crc4
# termtype: te
bchan=32-46,48-62
dchan=47
echocanceller=mg2,32-46,48-62

# Span 3: TE4/0/3 "T4XXP (PCI) Card 0 Span 3"
span=3,3,0,ccs,hdb3,crc4
# termtype: te
bchan=63-77,79-93
dchan=78
echocanceller=mg2,63-77,79-93

# Span 4: TE4/0/4 "T4XXP (PCI) Card 0 Span 4"
span=4,4,0,ccs,hdb3,crc4
# termtype: te
bchan=94-108,110-124
dchan=109
echocanceller=mg2,94-108,110-124
# Global data
loadzone          = us
defaultzone       = us
```

2. Please add the following line in the end of chan_dahdi.conf file
#include dahdi-channels.conf
3. Please run asterisk with the following command:
asterisk
asterisk -vvvgrc
reload
4. Please run dahdi show channels command
You should see 124 channels
5. How to check if the hardware echo cancellation works or not
 - 1) When you plug the EC128 into AX4D, please do not need to press hard, because the pin is a little longer than the slot.
 - 2) After booting your server with the AX4D and EC128, please run dmesg command to check if the hardware is detected by dahdi or not. You can get the following red information.

VPM450: echo cancellation for 128 channels

wct4xxp 0000:05:04.0: VPM450: hardware DTMF disabled.

wct4xxp 0000:05:04.0: VPM450: Present and operational servicing 4 span(s)

- 3) When you are using hardware echocancellation, please disable the software echocancellation in /etc/dahdi/system.conf, and enable the echocancel=yes line in /etc/asterisk/chan_dahdi.conf file. After setting, please restart dahdi and asterisk.
- 4) Run asterisk, and make a call by channel 1, you can get the following information marked with red line, if the EC128 is working.

Notice: If you enable the software echocancellation, you can also get the following red lines, so please disable the software echocancellation when you use EC128.

```
localhost*CLI> dahdi show channel 1
```

```
Channel: 1
```

```
File Descriptor: 17
```

```
Span: 1
```

```
Extension: 300
```

```
Dialing: no
```

```
Context: from-pstn
```

```
Caller ID: 900
```

```
Calling TON: 33
```

```
Caller ID name: 900
```

```
Mailbox: none
```

```
Destroy: 0
```

```
InAlarm: 0
```

```
Signalling Type: ISDN PRI
```

Radio: 0
Owner: DAHDI/1-1
Real: DAHDI/1-1
Callwait: <None>
Threeway: <None>
Confno: -1
Propagated Conference: -1
Real in conference: 0
DSP: yes
Busy Detection: no
TDD: no
Relax DTMF: no
Dialing/CallwaitCAS: 0/0
Default law: alaw
Fax Handled: no
Pulse phone: no
DND: no
Echo Cancellation:
 128 taps
 currently ON
Wait for dialtone: 0ms
PRI Flags: Call
PRI Logical Span: Implicit

Chapter 5 Loopback Test

1 LEDs status

LED Red: If the driver of the card is loaded correctly and the /etc/dahdi/system.conf file is configured correctly, the LED Red of the four ports will be red clearly.

LED Green: If the PRI line is connected correctly, and it synchronizes to the other end successfully, then the LED Green will be green clearly.

If the PRI line is connected correctly, and it is synchronizing to the other end, then both of the LED Green and LED Red will be on clearly.

2 Use E1 cross-over cables to connect port 1 and port 4, port 2 and port 3, if the connection is correct, the LEDs will turn to green.

3 Make a E1 crossover cable, the E1 crossover cable connection pins are as the following:

1 < --- > 4

2 < --- > 5

4 < --- > 1

5 < --- > 2

For the detail E1 crossover cable connection pins, please refer to the following link:

<http://www.chebucto.ns.ca/Chebucto/Technical/Manuals/Max/max6000/gs/cables.htm#17372>

Please edit dahdi-channels.conf file like the following:

```
; Autogenerated by /usr/sbin/dahdi_genconf on Mon Mar 22 18:10:35 2010 -- do not hand edit  
; Dahdi Channels Configurations (chan_dahdi.conf)
```

```
; This is not intended to be a complete chan_dahdi.conf. Rather, it is intended  
; to be #include-d by /etc/asterisk/chan_dahdi.conf that will include the global settings
```

```
; Span 1: TE4/0/1 "T4XXP (PCI) Card 0 Span 1" (MASTER)
```

```
group=1  
context=from-internal  
switchtype = euroisdn  
signalling = pri_net  
channel => 1-15,17-31  
context = default
```

```
; Span 2: TE4/0/2 "T4XXP (PCI) Card 0 Span 2"
```

```
group=2  
context=from-internal  
switchtype = euroisdn  
signalling = pri_net  
channel => 32-46,48-62  
context = default  
group = 63
```

```
; Span 3: TE4/0/3 "T4XXP (PCI) Card 0 Span 3"
```

```
group=3  
context=from-pstn  
switchtype = euroisdn  
signalling = pri_cpe  
channel => 63-77,79-93  
context = default  
group = 63
```

```
; Span 4: TE4/0/4 "T4XXP (PCI) Card 0 Span 4"
```

```
group=4  
context=from-pstn  
switchtype = euroisdn  
signalling = pri_cpe  
channel => 94-108,110-124  
context = default
```


3 Check the ports status

1) Start asterisk

```
[root@localhost asterisk]# asterisk -vvgc
```

```
*CLI> reload
```

2) Run pri show spans command

You can get the following port information, all of the four ports are in up status.

```
*CLI> pri show spans
```

```
PRI span 1/0: Provisioned, Up, Active
```

```
PRI span 2/0: Provisioned, Up, Active
```

```
PRI span 3/0: Provisioned, Up, Active
```

```
PRI span 4/0: Provisioned, Up, Active
```

4 Edit a dial-plan in extensions.conf

```
[from-internal]
```

```
exten=>300,1,Dial(dahdi/g1/100)
```

```
exten=>300,2,hangup()
```

```
exten=>400,1,Dial(dahdi/g2/200)
```

```
exten=>400,2,hangup()
```

```
[from-pstn]
```

```
exten=>100,1,Playback(hello-world)
```

```
exten=>100,2,Dial(dahdi/g1/${EXTEN})
```

```
exten=>100,3,Hangup()
```

```
exten=>200,1,Playback(hello-world)
```

```
exten=>200,2,Dial(dahdi/g2/${EXTEN})
```

```
exten=>200,3,Hangup()
```

5 Use a registered sip phone to call from the 30 channels of port 1 to the 30 channels of port 4, and call from the 30 channels of port 2 to the 30 channels of port 3

- 1) Dial 300, it will call 30 times between port 1 and port 4, the following information is only a part of outputs of the right results. In this way, we can check if all the channels of port 1 and port 4 can communicate.

```
*CLI>          -- Executing [300@from-internal:1] Dial("SIP/500-00000001",
"dahdi/g1/100") in new stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g1/100
  -- Accepting call from '500' to '100' on channel 0/1, span 4
  -- Executing [100@from-pstn:1] Playback("DAHDI/94-1", "hello-world") in new
stack
  -- <DAHDI/94-1> Playing 'hello-world' (language 'en')
  -- DAHDI/1-1 is proceeding passing it to SIP/500-00000001
  -- DAHDI/1-1 answered SIP/500-00000001
  -- Executing [100@from-pstn:2] Dial("DAHDI/94-1", "dahdi/g1/100") in new stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g1/100
  -- Accepting call from '500' to '100' on channel 0/2, span 4
  -- Executing [100@from-pstn:1] Playback("DAHDI/95-1", "hello-world") in new
stack
  -- <DAHDI/95-1> Playing 'hello-world' (language 'en')
  -- DAHDI/2-1 is proceeding passing it to DAHDI/94-1
  -- DAHDI/2-1 answered DAHDI/94-1
  -- Native bridging DAHDI/94-1 and DAHDI/2-1

  -- Executing [100@from-pstn:2] Dial("DAHDI/123-1", "dahdi/g1/100") in new
stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g1/100
  -- Accepting call from '500' to '100' on channel 0/31, span 4
  -- Executing [100@from-pstn:1] Playback("DAHDI/124-1", "hello-world") in new
stack
  -- <DAHDI/124-1> Playing 'hello-world' (language 'en')
  -- DAHDI/31-1 is proceeding passing it to DAHDI/123-1
  -- DAHDI/31-1 answered DAHDI/123-1
  -- Native bridging DAHDI/123-1 and DAHDI/31-1
  -- Executing [100@from-pstn:2] Dial("DAHDI/124-1", "dahdi/g1/100") in new
stack
```

- 2) Dial 400, it will call 30 times between port 2 and port 3, the following information is only a part of outputs of the right results. In this way, we can check if all the channels of port 2 and port 3 can communicate.

```
*CLI>      -- Executing [400@from-internal:1] Dial("SIP/500-00000003",
"dahdi/g2/200") in new stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g2/200
  -- Accepting call from '500' to '200' on channel 0/1, span 3
  -- Executing [200@from-pstn:1] Playback("DAHDI/63-1", "hello-world") in
  new stack
  -- <DAHDI/63-1> Playing 'hello-world' (language 'en')
  -- DAHDI/32-1 is proceeding passing it to SIP/500-00000003
  -- DAHDI/32-1 answered SIP/500-00000003
  -- Executing [200@from-pstn:2] Dial("DAHDI/63-1", "dahdi/g2/200") in new
  stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g2/200
  -- Accepting call from '500' to '200' on channel 0/2, span 3
  -- Executing [200@from-pstn:1] Playback("DAHDI/64-1", "hello-world") in
  new stack
  -- <DAHDI/64-1> Playing 'hello-world' (language 'en')
  -- DAHDI/33-1 is proceeding passing it to DAHDI/63-1
  -- DAHDI/33-1 answered DAHDI/63-1
  -- Native bridging DAHDI/63-1 and DAHDI/33-1

  -- Executing [200@from-pstn:2] Dial("DAHDI/92-1", "dahdi/g2/200") in new stack
  -- Requested transfer capability: 0x00 - SPEECH
  -- Called g2/200
  -- Accepting call from '500' to '200' on channel 0/31, span 3
  -- Executing [200@from-pstn:1] Playback("DAHDI/93-1", "hello-world") in
  new stack
  -- <DAHDI/93-1> Playing 'hello-world' (language 'en')
  -- DAHDI/62-1 is proceeding passing it to DAHDI/92-1
  -- DAHDI/62-1 answered DAHDI/92-1
  -- Native bridging DAHDI/92-1 and DAHDI/62-1
  -- Executing [200@from-pstn:2] Dial("DAHDI/93-1", "dahdi/g2/200") in new
  stack
```

Chapter 6 Reference

<http://www.asteriskguru.com/>

<http://www.asterisk.org/downloads>

http://www.openippbx.org/index.php?title=Main_Page

<http://www.atcom.cn/>