CEM Solutions

Step by Step Guide

BRI Card Installation

Asterisk

Step by Step Guide

BRI Card Installation

(with Asterisk)

Version 1.0

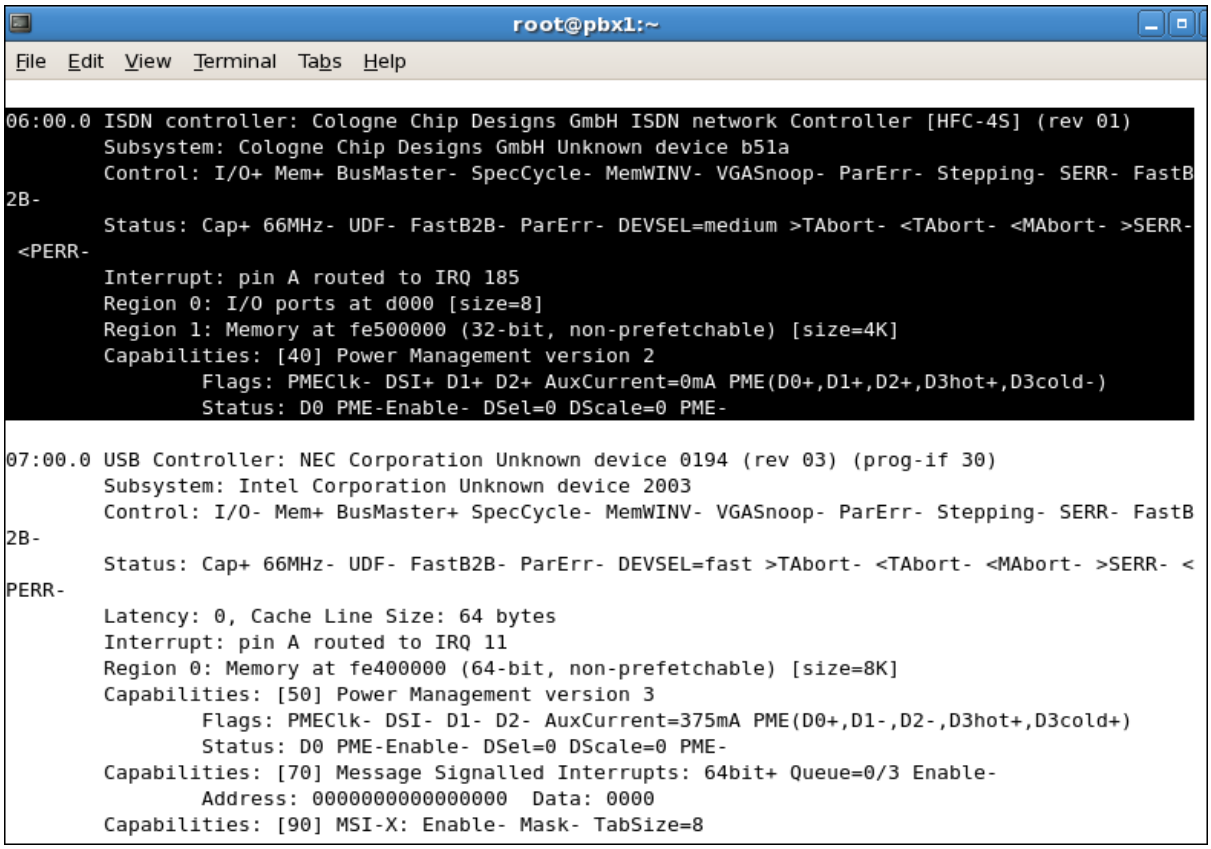
Table of Contents

Hardware Setup

1. Insert the BRI (PCI/PCIe) card in the corresponding slot
2. Check if the installed BRI card is detected using the below command

```
[root@pbx1 ~]# lspci -vvvvv
```

3. Check the output of the given command and ensure if there is a **`Cologne chip Unknown device`** with subsystem id **`b51a`**



Software Installation

Installation of Pre-requisite packages

1. Install all of Asterisk's dependencies that are required to compile asterisk.
    a. Run the followings commands to install the required packages needed for compiling DAHDI drivers from source.

For Centos/Redhat

```
[root@pbx1 ~]# yum –y install bison bison-devel ncurses
ncurses-devel zlib zlib-devel openssl openssl-devel gnutls-
devel gcc gcc-c++ libxml2
```

For Debian/Ubuntu

```
[root@pbx1 ~]# apt-get install debconf-utils proftpd-basic
build-essential libxml2-dev ncurses-dev bison flex libnewt-dev
```

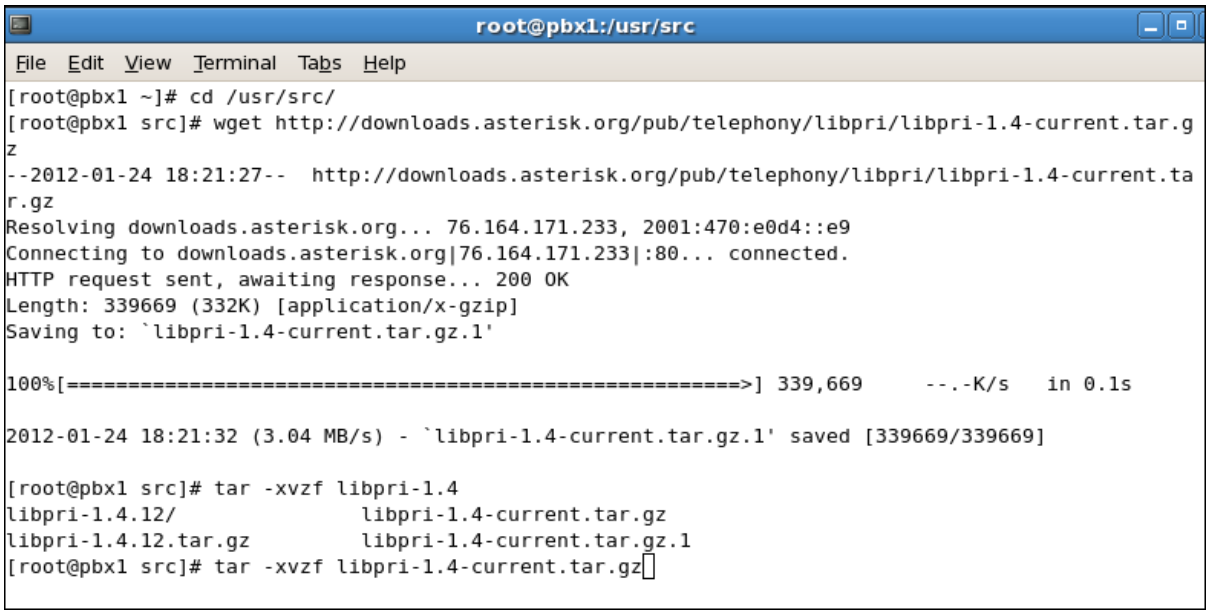
Installation of Libpri package

2. Go to `/usr/src` directory
3. Download the latest version of libpri from the source by running the following command

```
# wget
http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-
current.tar.gz
```

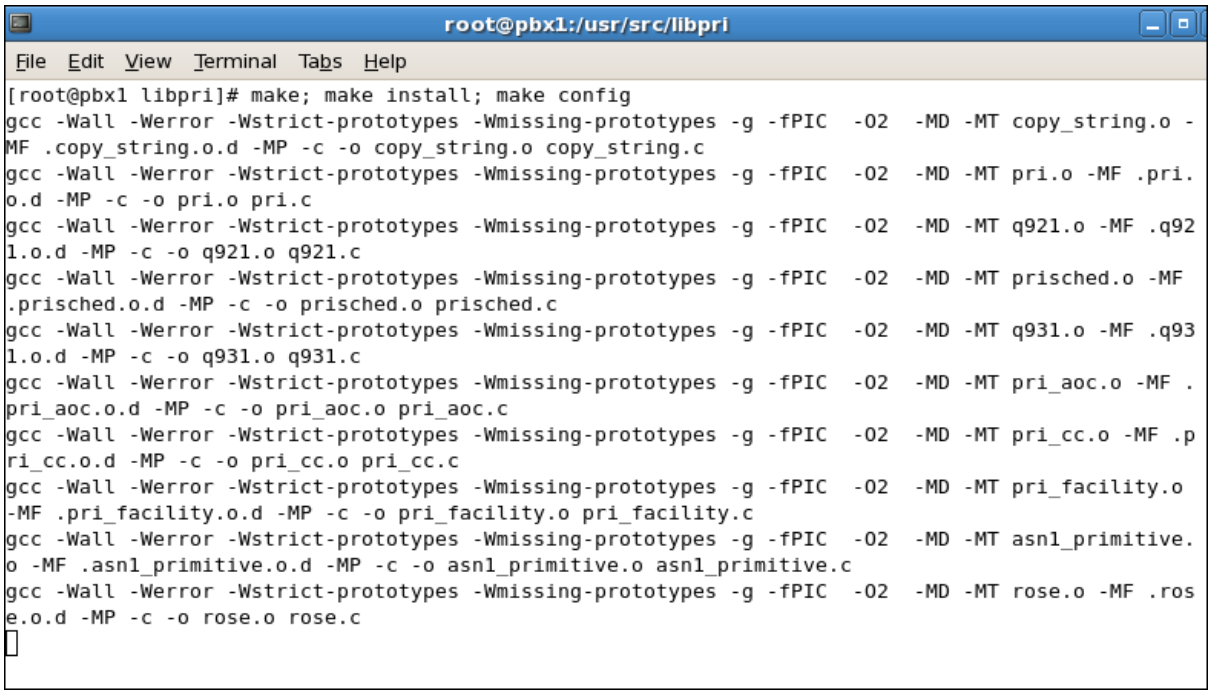
4. Expand the downloaded file

```
[root@srv1 src]# tar -xvzf libpri-1.4-current.tar.gz
```



5. Go to libpri folder and install the package using following commands as shown in the below screenshot

```
# cd libpri-1.4
# make clean; make ;  make install
```



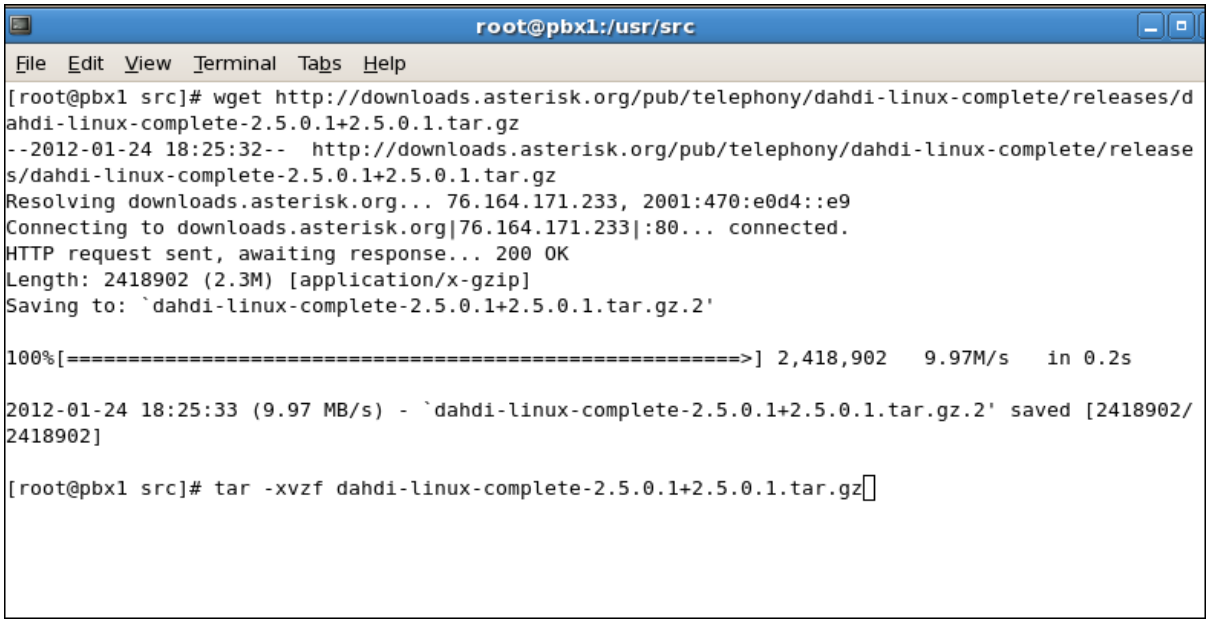
Next, we'll install DAHDI. DAHDI is the set of linux kernel modules and also a set of tools for interfacing with TDM cards. More importantly, DAHDI provides timing to several asterisk components, such as the MeetMe application as well as Music on Hold. If you don't have a proper timing source installed, you'll notice lots of stuttering pauses in any kind of audio playback (Music on Hold, IVR prompts, voicemail greetings) from asterisk. If you don't have any TDM hardware installed in your server, DAHDI also provides a "dummy" driver that will provide a timing source to asterisk.

Installation of DAHDI package

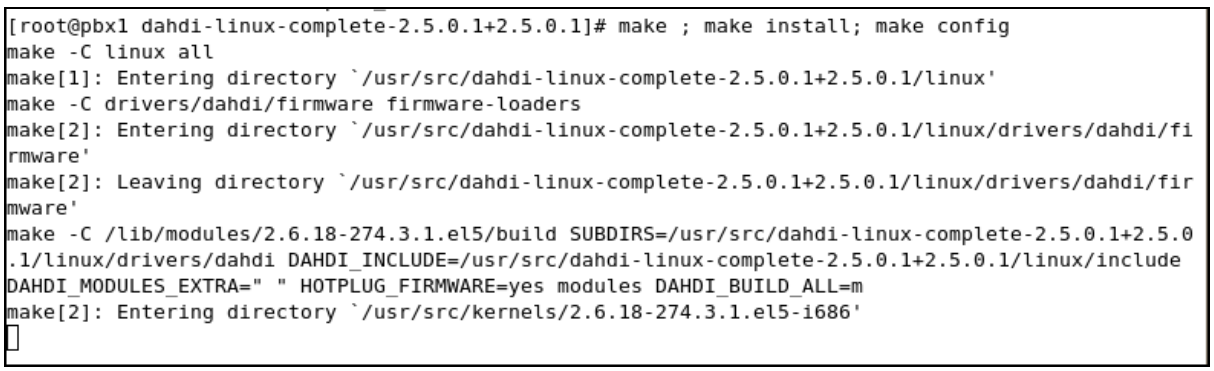
6. Download the DAHDI driver with tools, which are available at http://www.cem-solutions.net/ bri-card.html under 'Drivers and Manuals'.

```
#wget http://www.cem-solutions.net/firmware/bri-card/cem-dahdi-drivers/dahdi-linux-complete-2.5.0.1+2.5.0.1.tar.gz
```

7. Expand the downloaded file and enter into that directory as shown in the below screenshot.



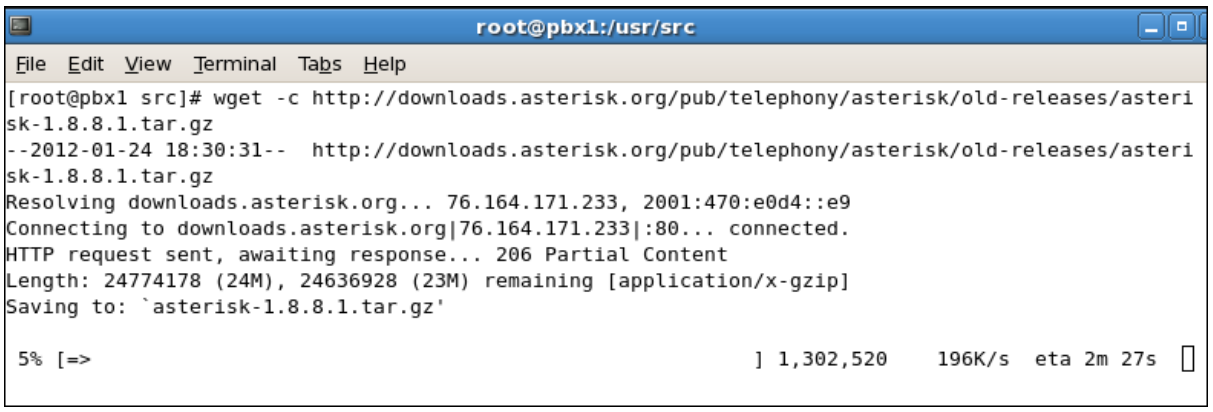
Install dahdi driver as show in the below screenshot



If there is any problem with the driver patch used for installation, please contact support@allo.com

Installation of Asterisk Package

8. Download the Asterisk 1.8 latest release version from http://downloads.asterisk.org/pub/telephony/asterisk/old-releases/

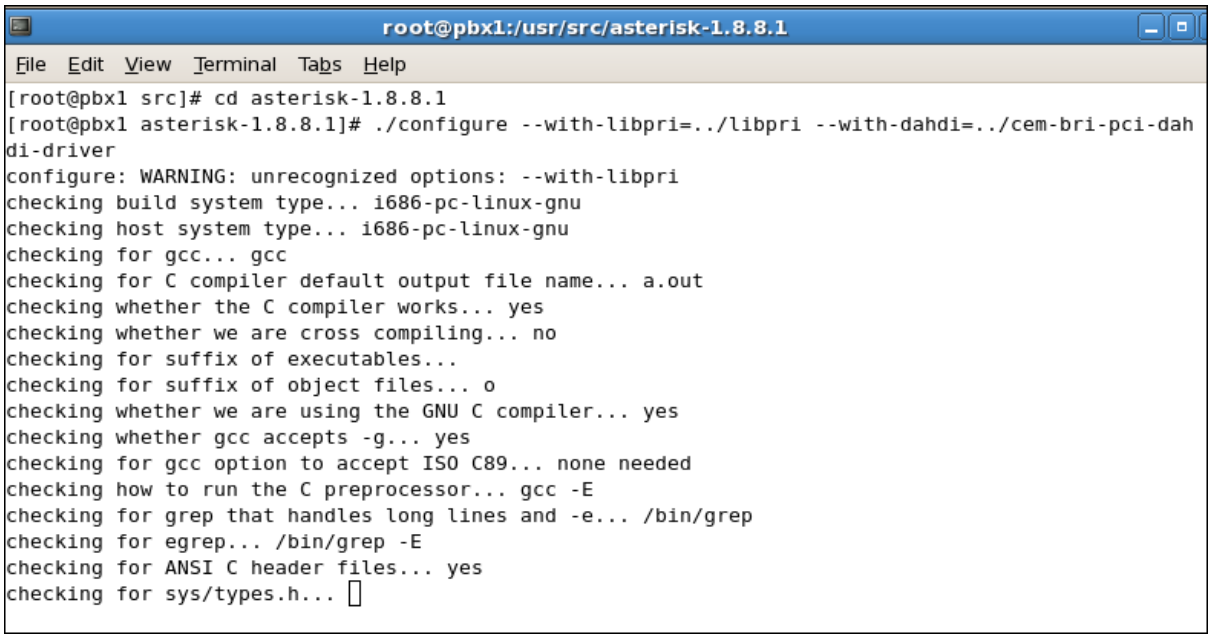


9. Expand the downloaded asterisk file as shown below

```
[root@pbx1 src]# tar xvzf asterisk-1.8.8.1.tar.gz
```

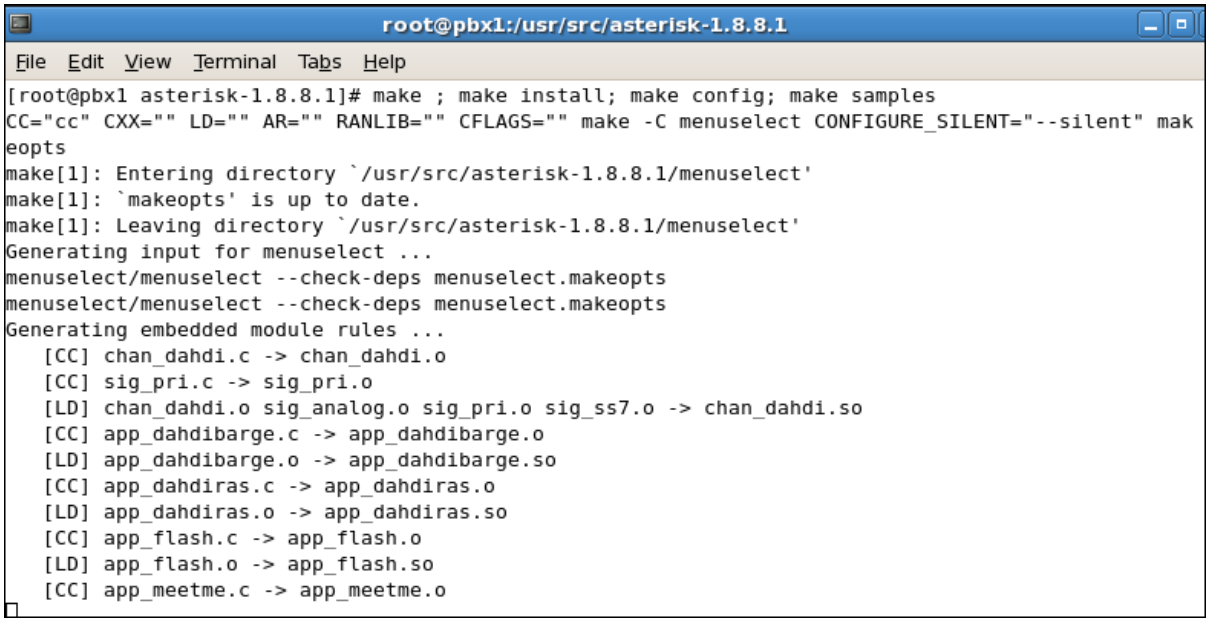
Go to asterisk folder and compile the packages as shown in the screenshot

```
[root@srv1 asterisk-1.8.8.1]# ./configure --with-
libpri=../libpri –with-  dahdi=../cem-bri-pci-dahdi-driver
```



10. Install the package by running the following command

```
[root@pbx1 asterisk-1.8.8.1]# make ; make install; make
config; make samples
```

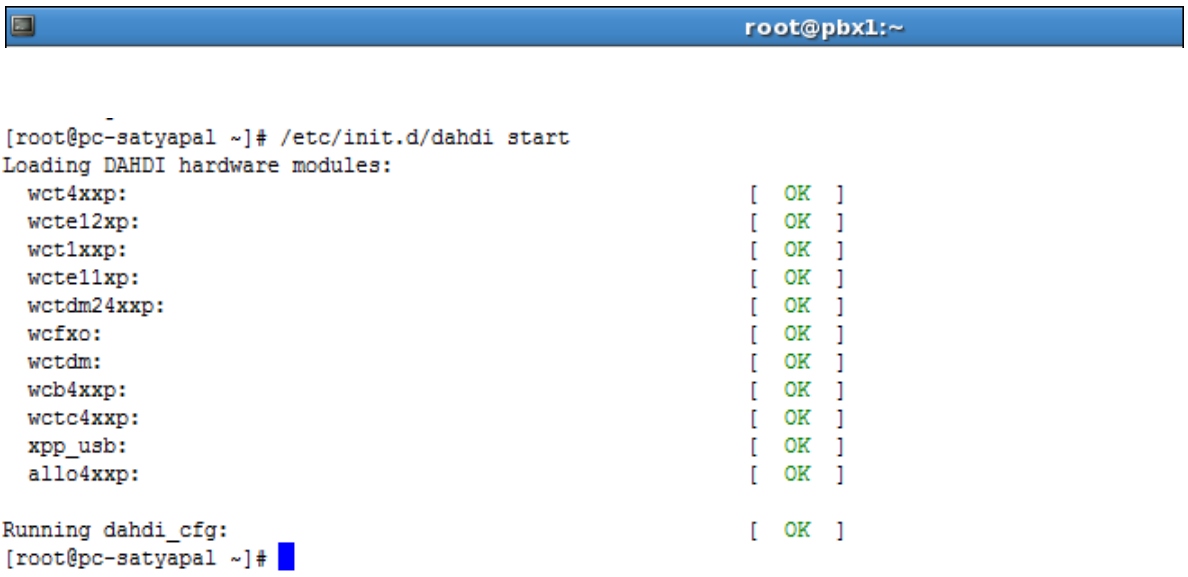


Now you have successfully compiled and installed Libpri, DAHDI and Asterisk.

Software Configuration

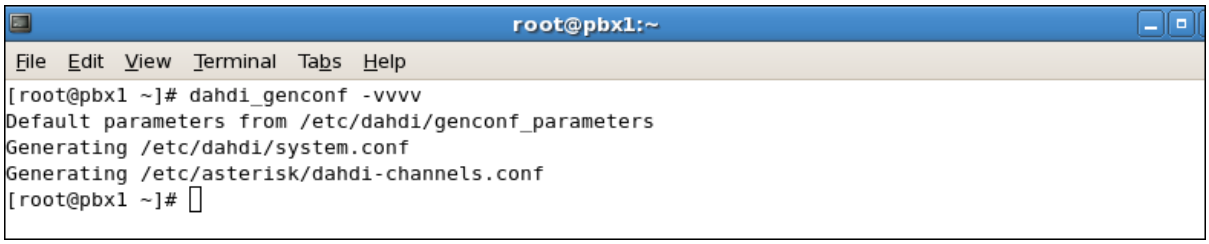
1. Please add the line "allo4xxp" at the end of the file in > /etc/dahdi/modules and start loading the driver by running

```
[root@srv1 asterisk-1.8.8.1]# /etc/init.d/dahdi start
```



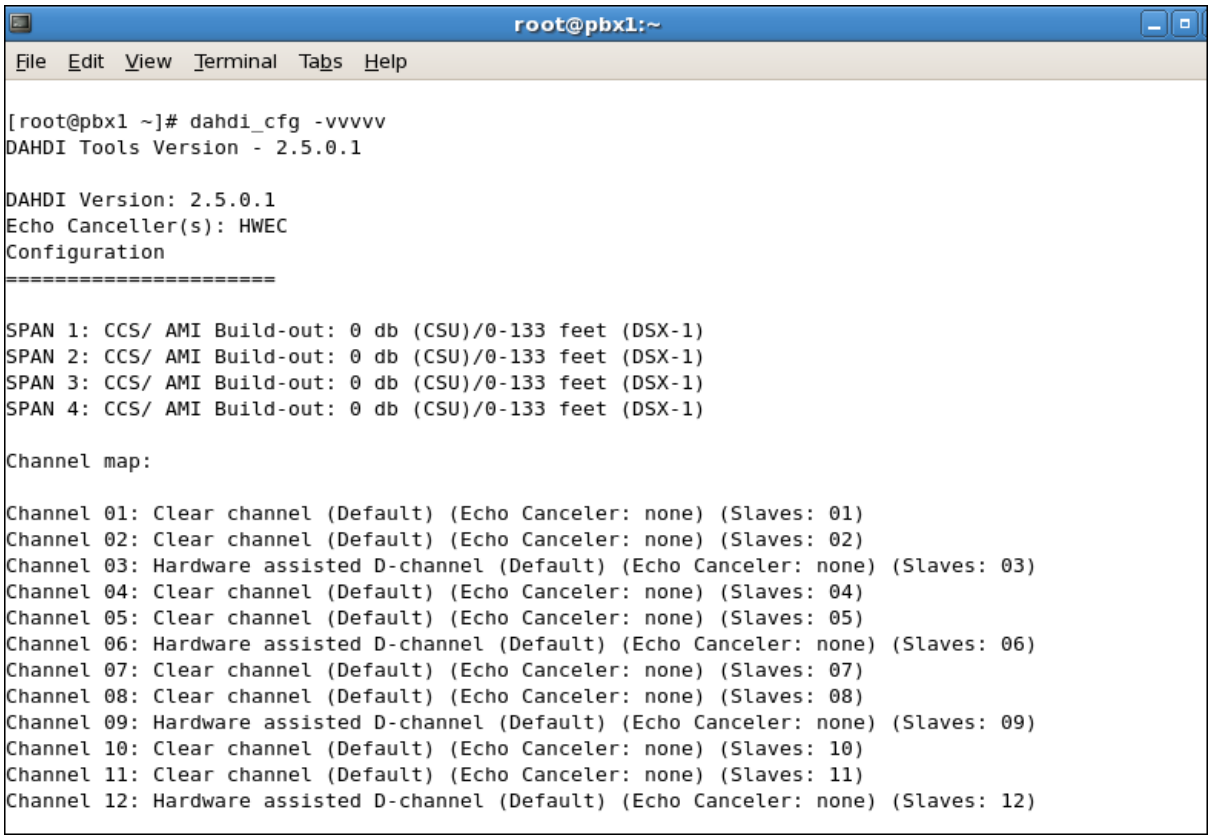
2. Generate config files using the following command

```
[root@pbx1 ~]# dahdi_genconf -vvvvvv
```



3. Check the output configured channels using the following commands. It will list the configure channels.

```
[root@pbx1 ~]# dahdi_cfg –vvvv
```



4. The following is a example `system.conf` file for BRI as shown in figure

```
[root@pbx1 ~]# vi /etc/dahdi/system.conf
```

```
# Autogenerated by /usr/sbin/dahdi_genconf on Fri Aug 10 11:28:22 2012
# If you edit this file and execute /usr/sbin/dahdi_genconf again,
# your manual changes will be LOST.
# Dahdi Configuration File
#
# This file is parsed by the Dahdi Configurator, dahdi_cfg
#
# Span 1: B4/0/1 "ALLO4XXP (PCI) Card 0 Span 1" (MASTER) AMI/CCS RED
span=1,1,0,ccs,ami
# termtype: te
bchan=1-2
hardhdlc=3
echocanceller=mg2,1-2

# Span 2: B4/0/2 "ALLO4XXP (PCI) Card 0 Span 2" AMI/CCS RED
span=2,2,0,ccs,ami
# termtype: te
bchan=4-5
hardhdlc=6
echocanceller=mg2,4-5

# Span 3: B4/0/3 "ALLO4XXP (PCI) Card 0 Span 3" AMI/CCS RED
span=3,3,0,ccs,ami
# termtype: te
bchan=7-8
hardhdlc=9
echocanceller=mg2,7-8

# Span 4: B4/0/4 "ALLO4XXP (PCI) Card 0 Span 4" AMI/CCS RED
span=4,4,0,ccs,ami
# termtype: te
bchan=10-11
hardhdlc=12
echocanceller=mg2,10-11

# Global data

loadzone        = us
defaultzone     = us
```

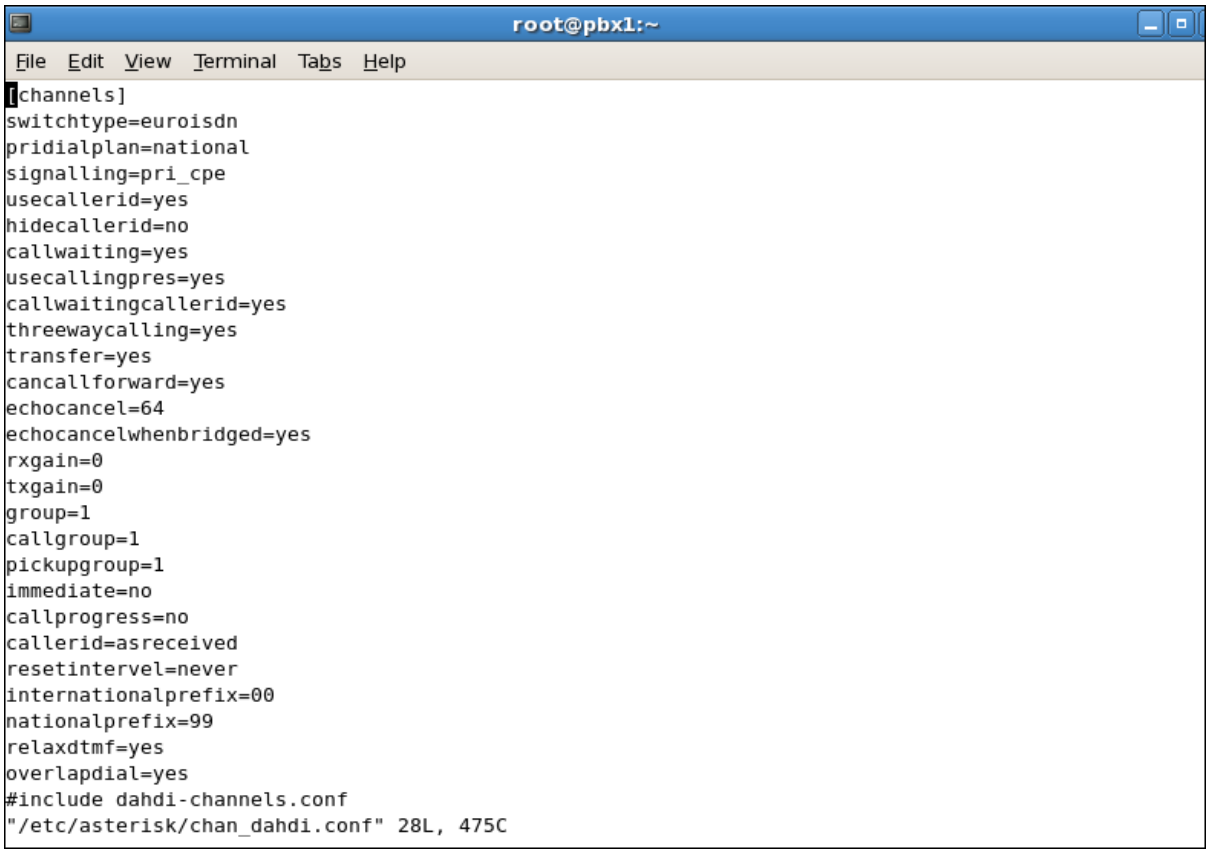
5. Configure the interface to Asterisk using dahdi
    a. We need to verify that asterisk installed correctly. We do this by manually starting asterisk from the command line. If everything starts up and there's not too many errors or warrnings, we're good to go
    b. You will need to modify the `chan_dahdi.conf` file which is located in the /etc/asterisk directory in order to configure the essential features of your card. This file is the configuration layer between DAHDI and Asterisk. Include the generated conf file `dahdi-channels.conf` in `chan_dahdi.conf` file

```
[root@pbx1 ~]# echo "#include dahdi-channels.conf" >>
/etc/asterisk/chan_dahdi.conf
```

Here is an example of `dahdi-channels.conf` file

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

; Span 1: B4/0/1 "ALLO4XXP (PCI) Card 0 Span 1" (MASTER) AMI/CCS RED
group=0,11
context=from-pstn
switchtype = euroisdn
signalling = bri_cpe_ptmp
channel => 1-2
context = default
group = 63

; Span 2: B4/0/2 "ALLO4XXP (PCI) Card 0 Span 2" AMI/CCS RED
group=0,12
context=from-pstn
switchtype = euroisdn
signalling = bri_cpe_ptmp
channel => 4-5
context = default
group = 63

; Span 3: B4/0/3 "ALLO4XXP (PCI) Card 0 Span 3" AMI/CCS RED
group=0,13
context=from-pstn
switchtype = euroisdn
signalling = bri_cpe_ptmp
channel => 7-8
context = default
group = 63

; Span 4: B4/0/4 "ALLO4XXP (PCI) Card 0 Span 4" AMI/CCS RED
group=0,14
context=from-pstn
switchtype = euroisdn
signalling = bri_cpe_ptmp
channel => 10-11
context = default
group = 63
```

Another example of a typical `chan_dahdi.conf` file



6. Start the asterik and connect the Asterisk CLI

```
[root@pbx1 ~]# /etc/init.d/asterisk start
```

7. Check the status of configured DAHDI channels in asterisk console

```
root@pbx1:~
File  Edit  View  Terminal  Tabs  Help
pc-satyapal*CLI> dahdi show status
Description                          Alarms  IRQ    bpviol CRC    Fra Codi Options  LBO
ALLO4XXP (PCI) Card 0 Span 1         RED     0      0      0      CCS AMI           0 db (CSU)/0-133 feet (DSX-1)
ALLO4XXP (PCI) Card 0 Span 2         RED     0      0      0      CCS AMI           0 db (CSU)/0-133 feet (DSX-1)
ALLO4XXP (PCI) Card 0 Span 3         OK      0      0      0      CCS AMI           0 db (CSU)/0-133 feet (DSX-1)
ALLO4XXP (PCI) Card 0 Span 4         RED     0      0      0      CCS AMI           0 db (CSU)/0-133 feet (DSX-1)
pc-satyapal*CLI>
```

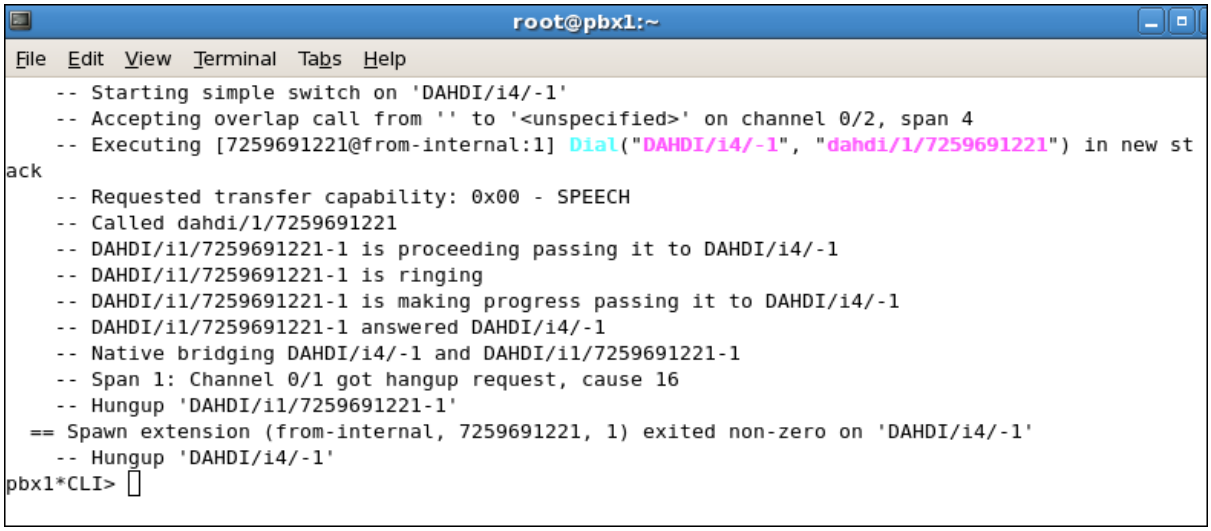
8. At this point we are ready to write a Dial Plan in `/etc/asterisk/extensions.conf.`

Here is an example of writing a Dial Plan syntax to make a outbound and inbound calls.

```
root@pbx1:~
File  Edit  View  Terminal  Tabs  Help
; one function. Remember that function names are UPPER CASE.

[from-pstn]
exten => _X.,1,Dial(SIP/100)
exten => _X.,n,Hangup()

[from-internal]
exten => _X.,1,Dial(dahdi/1/${EXTEN})
exten => _X.,n,Hangup()
exten => s,1,Answer
exten => s,2,Playtones(dial)
;use DigitTimeout previous to Asterisk 1.2
exten => s,3,Set(TIMEOUT(digit)=5)
exten => s,4,WaitExten(60)
;exten => s,5,Dial(dahdi/1/${EXTEN})
;exten => 1000,1,Dial(SIP/${EXTEN})

[from-internal1]
exten => _X.,1,Dial(dahdi/1/${EXTEN})
exten => _X.,n,Hangup()

;exten => 100,1,Dial(dahdi/7/100)

;exten => 100,2,Dial(SIP/100)


[from-internal2]
exten => _X.,1,Dial(dahdi/13/${EXTEN})
```

Now the system is ready to make calls.

Here is an example output of outbound call which is using DAHDI channel 1.


root@pbx1:~
File Edit View Terminal Tabs Help
-- Starting simple switch on 'DAHDI/i4/-1'
-- Accepting overlap call from '' to '<unspecified>' on channel 0/2, span 4
-- Executing [7259691221@from-internal:1] Dial("DAHDI/i4/-1", "dahdi/1/7259691221") in new st
ack
-- Requested transfer capability: 0x00 - SPEECH
-- Called dahdi/1/7259691221
-- DAHDI/i1/7259691221-1 is proceeding passing it to DAHDI/i4/-1
-- DAHDI/i1/7259691221-1 is ringing
-- DAHDI/i1/7259691221-1 is making progress passing it to DAHDI/i4/-1
-- DAHDI/i1/7259691221-1 answered DAHDI/i4/-1
-- Native bridging DAHDI/i4/-1 and DAHDI/i1/7259691221-1
-- Span 1: Channel 0/1 got hangup request, cause 16
-- Hungup 'DAHDI/i1/7259691221-1'
== Spawn extension (from-internal, 7259691221, 1) exited non-zero on 'DAHDI/i4/-1'
-- Hungup 'DAHDI/i4/-1'
pbx1*CLI>