



# Step by Step Guide

E1/T1 PRI Card Installation

Asterisk

# **Step by Step Guide**

## **E1/T1 PRI Card Installation**

Version 2.0

# Table of Contents

|  |          |
|--|----------|
| <b>Hardware Setup</b> .....                  | <b>1</b> |
| <b>Software Installation</b> .....           | <b>2</b> |
| Test Environment.....                        | 2        |
| Installation of Pre-requisite packages ..... | 2        |
| Installation of Libpri package .....         | 2        |
| Installation of DAHDI package .....          | 4        |
| Installation of Asterisk Package .....       | 6        |
| <b>Software Configuration</b> .....          | <b>7</b> |

# Hardware Setup

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

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

3. Check the output of the given command and ensure if there is a line  
**Bridge: PLX Technology, Inc. Unknown device d44d (rev 01)**

```

root@pbx1:~
File Edit View Terminal Tabs Help
Bus: primary=03, secondary=04, subordinate=04, sec-latency=64
I/O behind bridge: 0000e000-0000e0ff
Memory behind bridge: fbf00000-fbffffff
Prefetchable memory behind bridge: fff00000-000ffffff
Secondary status: 66MHz+ FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- <SERR- <PERR-
BridgeCtl: Parity+ SERR+ NoISA- VGA- MAbort- >Reset- FastB2B-
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-
Capabilities: [50] Message Signalled Interrupts: 64bit+ Queue=0/0 Enable-
Address: 0000000000000000 Data: 0000
Capabilities: [60] Express PCI/PCI-X Bridge IRQ 0
Device: Supported: MaxPayload 128 bytes, PhantFunc 0, ExtTag-
Device: Latency L0s <64ns, L1 <1us
Device: AtnBtn- AtnInd- PwrInd-
Device: Errors: Correctable- Non-Fatal- Fatal- Unsupported-
Device: RlxdOrd- ExtTag- PhantFunc- AuxPwr- NoSnoop-
Device: MaxPayload 128 bytes, MaxReadReq 512 bytes
Link: Supported Speed 2.5Gb/s, Width x1, ASPM L0s L1, Port 0
Link: Latency L0s <1us, L1 <16us
Link: ASPM Disabled CommClk- ExtSynch-
Link: Speed 2.5Gb/s, Width x1

04:00.0 Bridge: PLX Technology, Inc. Unknown device d44d (rev 01)
Subsystem: PLX Technology, Inc. Unknown device 9030
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 185
Region 0: Memory at fbfffc00 (32-bit, non-prefetchable) [size=128]
Region 1: I/O ports at ec00 [size=128]
Region 2: Memory at fbfff000 (32-bit, non-prefetchable) [size=2K]
Region 3: Memory at fbffe800 (32-bit, non-prefetchable) [size=2K]
Capabilities: [40] Power Management version 1
Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold-)
Status: D0 PME-Enable- DSel=0 DScale=3 PME-
Capabilities: [48] #06 [0000]
Capabilities: [4c] Vital Product Data

[root@pbx1 ~]#

```

PLX Technology will be found, if you cannot see the PLX Technology, please poweroff your server and try another PCI slot, if it still does not help, you have to check the compatibility issue between the card and your PCI bus.

# Software Installation

---

## Test Environment

Libpri-1.4.14

dahdi-linux-complete-2.6.1

asterisk-1.8.20.1

centos 6.2 (kernel version: 2.6.32)

## 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 drivers from source.

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

## Installation of Libpri package

1. Go to `/usr/src` directory
2. Download libpri by running the following command

```
[root@localhost src]# wget
http://downloads.asterisk.org/pub/telephony/libpri/libpri-
1.4.14.tar.gz
```

3. Expand the downloaded file

```
[root@localhost src]# tar -xvzf libpri-1.4.14.tar.gz
```

```

root@localhost:~/usr/src
File Edit View Terminal Tabs Help
[root@localhost ~]# cd /usr/src/
[root@localhost src]# tar xvzf libpri-1.4.14.tar.gz
libpri-1.4.14/
libpri-1.4.14/doc/
libpri-1.4.14/doc/cc_qsig_monitor.fsm
libpri-1.4.14/doc/cc_ptmp_monitor_flattened.fsm
libpri-1.4.14/doc/cc_qsig_agent.fsm
libpri-1.4.14/doc/cc_ptp_monitor.fsm
libpri-1.4.14/doc/cc_qsig_monitor_flattened.fsm
libpri-1.4.14/doc/cc_ptmp_agent_flattened.fsm
libpri-1.4.14/doc/cc_ptmp_monitor.fsm
libpri-1.4.14/doc/cc_ptmp_agent.fsm
libpri-1.4.14/doc/cc_qsig_agent_flattened.fsm
libpri-1.4.14/doc/cc_ptp_agent.fsm
libpri-1.4.14/doc/cc_ptp_monitor_flattened.fsm
libpri-1.4.14/doc/cc_ptp_agent_flattened.fsm
libpri-1.4.14/rose.h
libpri-1.4.14/pri_aoc.c
libpri-1.4.14/testprilib.c
libpri-1.4.14/rose_qsig_diversion.c
libpri-1.4.14/pridump.c
libpri-1.4.14/rose_etsi_ect.c
libpri-1.4.14/prisched.c
libpri-1.4.14/rose_etsi_diversion.c
libpri-1.4.14/rosetest.c
libpri-1.4.14/rose_qsig_ct.c
libpri-1.4.14/libpri.h
libpri-1.4.14/rose_etsi_aoc.c
libpri-1.4.14/rose.c
libpri-1.4.14/pri_cc.c
libpri-1.4.14/asn1_primitive.c

```

4. Go to libpri-1.4.14 folder and install the package using following commands as shown in the below screenshot

```

[root@localhost src]# cd libpri-1.4.14
[root@localhost src]# make clean; make ; make install

```

```

root@localhost:~/usr/src/libpri-1.4.14
File Edit View Terminal Tabs Help
[root@localhost libpri-1.4.14]# make clean; make ; make install
rm -f *.o *.so *.lo
rm -f libpri.a libpri.so.1.4
rm -f pridump pritest rosetest testprilib
rm -f *.d
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT copy_string.o -MF .copy_string.o.d -MP -c -o copy_string.o copy_string.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT pri.o -MF .pri.o.d -MP -c -o pri.o pri.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT q921.o -MF .q921.o.d -MP -c -o q921.o q921.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT prisched.o -MF .prisched.o.d -MP -c -o prisched.o prisched.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT q931.o -MF .q931.o.d -MP -c -o q931.o q931.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT pri_aoc.o -MF .pri_aoc.o.d -MP -c -o pri_aoc.o pri_aoc.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT pri_cc.o -MF .pri_cc.o.d -MP -c -o pri_cc.o pri_cc.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT pri_facility.o -MF .pri_facility.o.d -MP -c -o pri_facility.o pri_facility.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT asn1_primitive.o -MF .asn1_primitive.o.d -MP -c -o asn1_primitive.o asn1_primitive.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT rose.o -MF .rose.o.d -MP -c -o rose.o rose.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT rose_address.o -MF .rose_address.o.d -MP -c -o rose_address.o rose_address.c
gcc -g -Wall -Werror -Wstrict-prototypes -Wmissing-prototypes -fPIC -O2 -MD -MT rose_etsi_aoc.o -MF .rose_etsi_aoc.o.d -MP -c -o rose_etsi_aoc.o rose_etsi_aoc.c

```

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

1. Download the DAHDI driver with tools, which are available at <http://www.allo.com/firmware/pri-card/drivers/dahdi-linux-complete-2.6.1+2.6.1.tar.gz>

```
[root@localhost src]# wget http://www.allo.com/firmware/pri-card/drivers/dahdi-linux-complete-2.6.1+2.6.1.tar.gz
```

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

```
[root@localhost src]# tar -xvzf dahdi-linux-complete-current.tar.gz
[root@pbx1 src]# cd dahdi-linux-complete-2.6.1+2.6.1
```

3. Compile its contents, and install the dahdi driver as show in the below screenshot

```
[root@localhost dahdi-linux-complete-2.6.1+2.6.1]#make; make install; make config
```

```

root@localhost:/usr/src/dahdi-linux-complete-2.6.1+2.6.1
File Edit View Terminal Tabs Help
[root@localhost dahdi-linux-complete-2.6.1+2.6.1]# make; make install; make config
make -C linux all
make[1]: Entering directory `/usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux'
make -C drivers/dahdi/firmware firmware-loaders
make[2]: Entering directory `/usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/firmware'
make[2]: Leaving directory `/usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/firmware'
make -C /lib/modules/2.6.18-308.el5PAE/build SUBDIRS=/usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi DAHDI_INCLUDE=/usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/include DAHDI_MODULES_EXTRA=" " HOTPLUG_FIRMWARE=yes modules DAHDI_BUILD_ALL=m
make[2]: Entering directory `/usr/src/kernels/2.6.18-308.el5-PAE-1686'
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_adpcm_chan.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_channel.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_chip_open.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_chip_stats.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_conf_bridge.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_debug.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_events.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_interrupts.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_memory.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_miscellaneous.o
  CC [M] /usr/src/dahdi-linux-complete-2.6.1+2.6.1/linux/drivers/dahdi/tor3/./oct612x/octdeviceapi/oct6100api/oct6100_api/oct6100_mixer.o

```

If there is any problem with the driver patch used for installation, please contact [support@allo.com](mailto:support@allo.com)

4. If PRI card is with Hardware LEC module, Please download the firmware from here <http://www.allo.com/firmware/pri-card/OCT6-LEC-128.tar.gz> under /usr/lib/hotplug/firmware and /lib/firmware directories

```

[root@localhost ~]# cd /lib/firmware/
[root@localhost firmware]#
[root@localhost firmware]# wget -c
http://www.allo.com/firmware/pri-card/OCT6-LEC-128.tar.gz

```

- b. Extract the file in these two folders as shown below,

```

[root@localhost firmware]# tar xvzf OCT6-LEC-128.tar.gz
OCT6126E-128D.ima
dahdi-fw-oct6114-128.bin
[root@localhost firmware]# cd /usr/lib/hotplug/firmware/
[root@localhost firmware]# tar xvzf OCT6-LEC-128.tar.gz
OCT6126E-128D.ima
dahdi-fw-oct6114-128.bin

```



Reboot the machine

## Installation of Asterisk Package

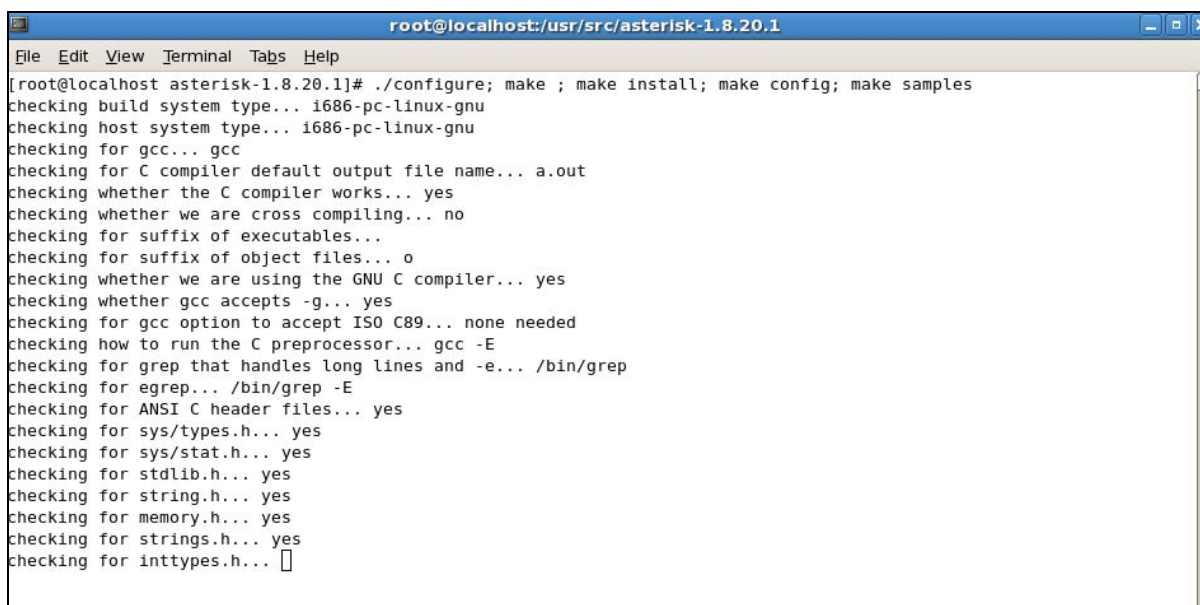
1. Download the Asterisk 1.8.20.1 version from <http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-current.tar.gz>

2. Expand the downloaded asterisk file as shown below

```
[root@pbx1 src]# tar xvzf asterisk-1.8-current.tar.gz
```

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

```
[root@pbx1 asterisk-1.8.20.1]# ./configure; make; make install;
make config; make samples
```



```

root@localhost:/usr/src/asterisk-1.8.20.1
File Edit View Terminal Tabs Help
[root@localhost asterisk-1.8.20.1]# ./configure; make ; make install; make config; make samples
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... 

```

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

# Software Configuration

---

5. This session will provide steps for configuring signaling mode, once you are done with the signaling mode continue with module & channel configuration.

## E1/T1/MFCR2 mode settings

### E1 Mode

```
[root@localhost ~]# echo "tor3e" >> /etc/dahdi/modules
```

### T1 Mode

```
[root@localhost ~]# echo " options tor3e eltloVERRIDE=1 " >> /etc/modprobe.d/dahdi.conf
```

### MFCR2 Mode

```
[root@localhost ~]# echo "options tor3e mfcR2=1" >> /etc/modprobe.d/dahdi.conf
```

6. Load the DAHDI drivers

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

7. Use "dahdi\_genconf" to finish auto configuration:

```
[root@localhost ~]# dahdi_genconf -vvvvvvv
Default parameters from /etc/dahdi/genconf_parameters
Generating /etc/dahdi/system.conf
Generating /etc/asterisk/dahdi-channels.conf
```

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

```
[root@ localhost ~]# dahdi_cfg -vvvv
```

```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# dahdi_cfg -vvvvvvvvvvvvvv
DAHDI Tools Version - 2.6.1

DAHDI Version: 2.6.1
Echo Cancellor(s): HWEC
Configuration
=====

SPAN 1: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
SPAN 2: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
SPAN 3: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
SPAN 4: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)

Channel map:

Channel 01: Clear channel (Default) (Echo Canceler: none) (Slaves: 01)
Channel 02: Clear channel (Default) (Echo Canceler: none) (Slaves: 02)
Channel 03: Clear channel (Default) (Echo Canceler: none) (Slaves: 03)
Channel 04: Clear channel (Default) (Echo Canceler: none) (Slaves: 04)
Channel 05: Clear channel (Default) (Echo Canceler: none) (Slaves: 05)
Channel 06: Clear channel (Default) (Echo Canceler: none) (Slaves: 06)
Channel 07: Clear channel (Default) (Echo Canceler: none) (Slaves: 07)
Channel 08: Clear channel (Default) (Echo Canceler: none) (Slaves: 08)
Channel 09: Clear channel (Default) (Echo Canceler: none) (Slaves: 09)
Channel 10: Clear channel (Default) (Echo Canceler: none) (Slaves: 10)
Channel 11: Clear channel (Default) (Echo Canceler: none) (Slaves: 11)
Channel 12: Clear channel (Default) (Echo Canceler: none) (Slaves: 12)
Channel 13: Clear channel (Default) (Echo Canceler: none) (Slaves: 13)
Channel 14: Clear channel (Default) (Echo Canceler: none) (Slaves: 14)
Channel 15: Clear channel (Default) (Echo Canceler: none) (Slaves: 15)
Channel 16: D-channel (Default) (Echo Canceler: none) (Slaves: 16)

```

9. The following is an example system.conf file for E1 as shown in figure

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

```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# cat /etc/asterisk/dahdi-channels.conf
; Autogenerated by /usr/sbin/dahdi_genconf on Tue Mar  5 18:34:12 2013
; If you edit this file and execute /usr/sbin/dahdi_genconf again,
; your manual changes will be LOST.
; 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/chan_dahdi.conf that will include the global settings
;
; Span 1: Tor3/0/1 "ALLO (PCI) Quad E1 Card 0 Span 1" (MASTER) HDB3/CCS/CRC4 BLUE RED
group=0,11
context=from-pstn
switchtype = euroisdn
signalling = pri_cpe
channel => 1-15,17-31
context = default
group = 63

; Span 2: Tor3/0/2 "ALLO (PCI) Quad E1 Card 0 Span 2" HDB3/CCS/CRC4 BLUE RED
group=0,12
context=from-pstn
switchtype = euroisdn
signalling = pri_cpe
channel => 32-46,48-62
context = default
group = 63

; Span 3: Tor3/0/3 "ALLO (PCI) Quad E1 Card 0 Span 3" HDB3/CCS/CRC4 BLUE RED
group=0,13
context=from-pstn

```

10. This session will provide steps for updating LEC module,

a. Download the LEC module from <http://www.allo.com/firmware/pri-card/OCT6-LEC-128.tar.gz> into /usr/lib/hotplug/firmware and /lib/firmware directories

b. Extract the file in these two folders as shown below,

```
[root@localhost ~]# cd /lib/firmware/
[root@localhost firmware]# tar xvzf OCT6-LEC-128.tar.gz
OCT6126E-128D.ima
dahdi-fw-oct6114-128.bin
[root@localhost firmware]# cd /usr/lib/hotplug/firmware/
[root@localhost firmware]# tar xvzf OCT6-LEC-128.tar.gz
OCT6126E-128D.ima
dahdi-fw-oct6114-128.bin
```

Reboot the machine

11. Include the dahdi-channels.conf in chan\_dahdi.conf file to configure dahdi channels with asterisk.

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

12. Edit the dialplan (/etc/asterisk/extensions.conf)

```
[from-pstn]
exten => s,1,Answer() // answer the inbound call
exten => s,n,Playback(cc_welcome) //please message
exten => s,n,Hangup()

[from-internal]
exten => _X.,1,Dial(dahdi/g1/${EXTEN})
exten => _X.,n,Hangup
```

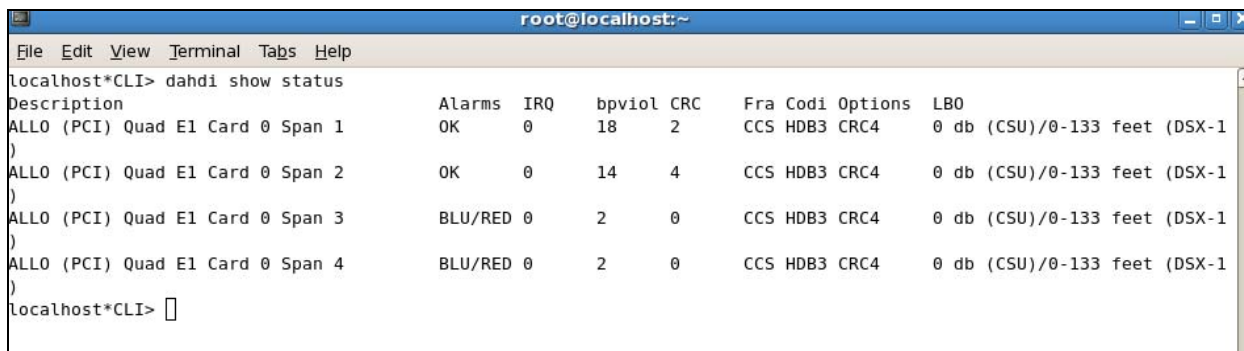
13. Start the asterisk and connect the Asterisk CLI

```
[root@localhost ~]# /etc/init.d/asterisk start
Starting asterisk:
```

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

#### 14. Check the status of dahdi spans in asterisk CLI.

Here is an example shows the spans status of CP400P/CP400E card

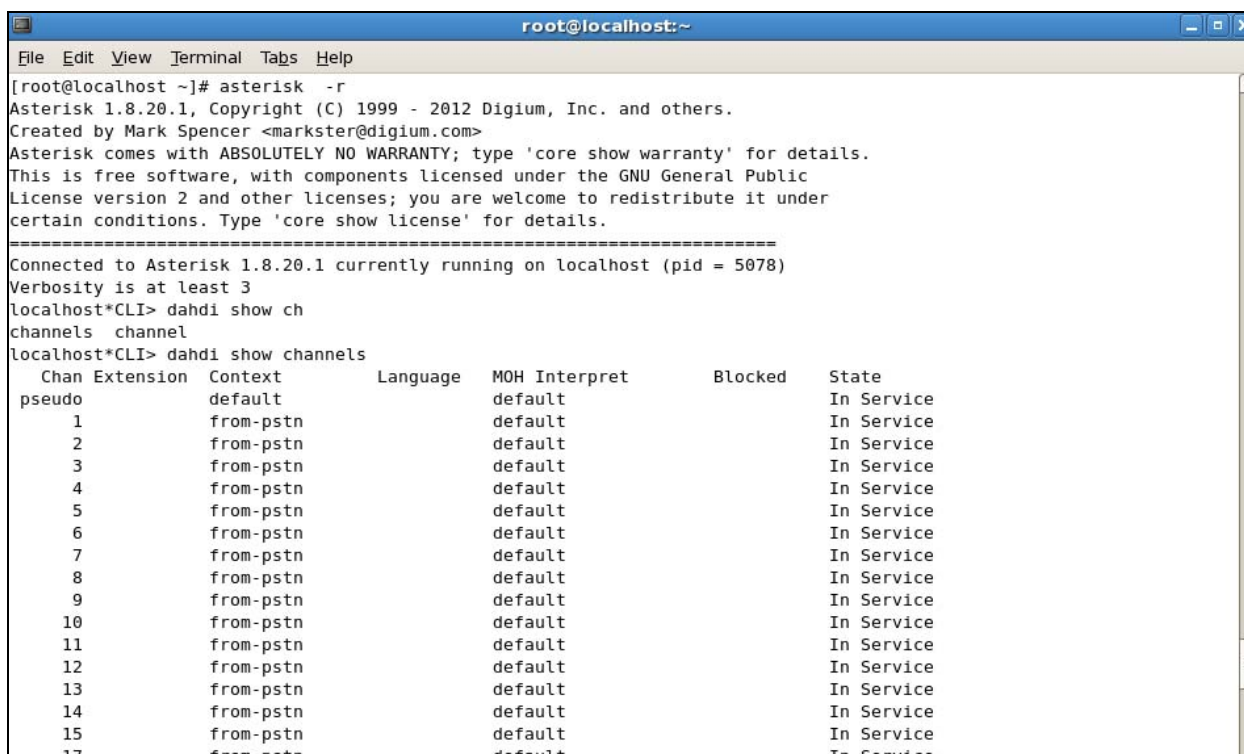


```

root@localhost:~
File Edit View Terminal Tabs Help
localhost*CLI> dahdi show status
Description                               Alarms  IRQ    bpviol  CRC    Fra Codi Options  LBO
ALLO (PCI) Quad E1 Card 0 Span 1          OK      0     18     2     CCS HDB3 CRC4    0 db (CSU)/0-133 feet (DSX-1
)
ALLO (PCI) Quad E1 Card 0 Span 2          OK      0     14     4     CCS HDB3 CRC4    0 db (CSU)/0-133 feet (DSX-1
)
ALLO (PCI) Quad E1 Card 0 Span 3          BLU/RED 0     2     0     CCS HDB3 CRC4    0 db (CSU)/0-133 feet (DSX-1
)
ALLO (PCI) Quad E1 Card 0 Span 4          BLU/RED 0     2     0     CCS HDB3 CRC4    0 db (CSU)/0-133 feet (DSX-1
)
localhost*CLI>

```

#### 15. Check the configured dahdi channels in asterisk using “dahdi show channels” as shown in the screenshot.



```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# asterisk -r
Asterisk 1.8.20.1, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.20.1 currently running on localhost (pid = 5078)
Verbosity is at least 3
localhost*CLI> dahdi show ch
channels channel
localhost*CLI> dahdi show channels
Chan Extension Context Language MOH Interpret Blocked State
pseudo default default default
1 from-pstn default In Service
2 from-pstn default In Service
3 from-pstn default In Service
4 from-pstn default In Service
5 from-pstn default In Service
6 from-pstn default In Service
7 from-pstn default In Service
8 from-pstn default In Service
9 from-pstn default In Service
10 from-pstn default In Service
11 from-pstn default In Service
12 from-pstn default In Service
13 from-pstn default In Service
14 from-pstn default In Service
15 from-pstn default In Service
17 from-pstn default In Service

```

#### 16. Then check the check the PRI status of all spans

Here is an example output of PRI spans



```
root@localhost:~  
File Edit View Terminal Tabs Help  
localhost*CLI> pri show spans  
PRI span 1/0: Up, Active  
PRI span 2/0: Up, Active  
PRI span 3/0: In Alarm, Down, Active  
PRI span 4/0: In Alarm, Down, Active  
localhost*CLI> █
```

Now the system is ready to make calls using these configured dahdi channels.