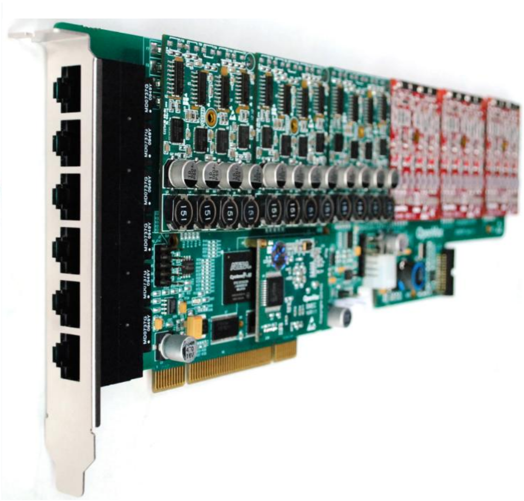
深圳开源通信有限公司

*OpenVox-Best Cost Effective Asterisk Cards*

# OpenVox A2410P/AE2410P on Trixbox2.8.0.3
# User Manual

A2410P



AE2410P

*Date:30/09/2010*

*Version: 1.1*

深圳开源通信有限公司

*OpenVox-Best Cost Effective Asterisk Cards*

OpenVox Communication Co. Ltd.
Address: F/3, Block No.127, Jindi Industrial Zone,
Shazui Road, Futian district, ShenZhen, Guangdong 518048, China
Tel:+86-755-82535095,82535461,Fax:+86-755-82535174
E-Mail：sales@openvox.cn support@openvox.cn

IM for Technical Support: support@openvox.cn James.zhu@openvox.cn
Business Hours: 9:00AM-18:00PM from Monday-Friday
URL：www.openvox.cn

*Thank You for Choosing OpenVox Products!*

# Table of Contents

# 1. Overview

## 1.1 What is A2410P/AE2410P

A2410P is a modular analog telephony interface product. AE2410P is A2410P with EC module. It is designed to build IP PBX.

A2410P/AE2410P must be used with FXO-400 or FXS-400 together to build a workable system. AE2410P has EC module, which must be used with FXO-400 or FXS-400 together.

Key Benefits:
Low CPU Payload : Firmware accelerate I/O access achieve high stability and highly decreased cpu payload
Scalable: Just add additional cards to extend system
Bus Master: Operation speed up to 132Mbytes/sec
Echo cancellation: Support high quality octasic echo cancellation DSP, each channel independent of 128ms or 1024 taps echo cancellation

RoHS compliant
Certificates: CE, FCC

Misc:
Temperature Operation: 0 to 50° C
Temperature Storage: -40 to 125° C
Humidity: 10 TO 90% NON-CONDENSING
Voltage: 3.3V
Board Power Dissipation： 6.89W

Disclaimers
Asterisk® is a registered trademark of Digium, Inc.

## 1.2 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.

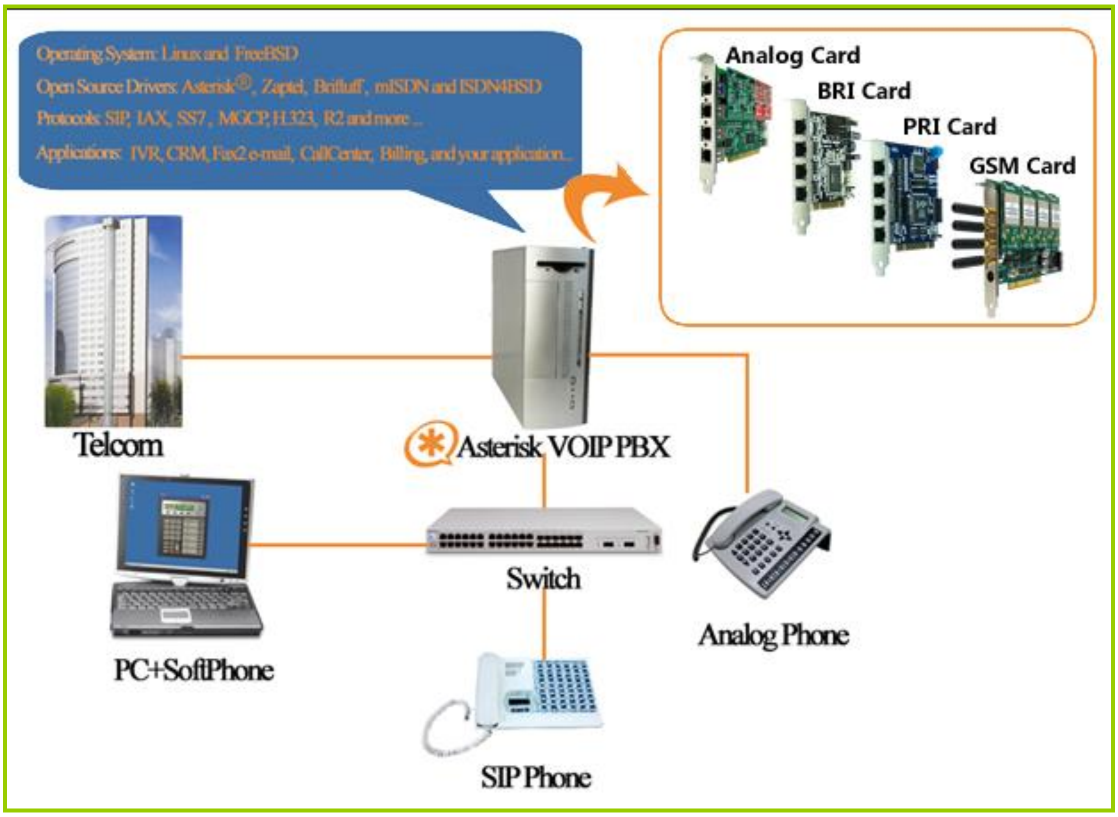

Operating System: Linux and FreeBSD
Open Source Drivers: Asterisk®, Zaptel, Brifluff, mISDN and ISDN4BSD
Protocols: SIP, IAX, SS7, MGCP, H.323, R2 and more...
Applications: IVR, CRM, Fax2 e-mail, CallCenter, Billing, and your application...
Analog Card
BRI Card
PRI Card
GSM Card
Telcom
Asterisk VOIP PBX
Switch
Analog Phone
PC+SoftPhone
SIP Phone

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).

# 2. Hardware Setting

To set the A2410P/AE2410P, user MUST go through these steps:

1. Checking power supply: Board must be provided power, please plug the power supply cable into power supply connector. (refer figure 2)

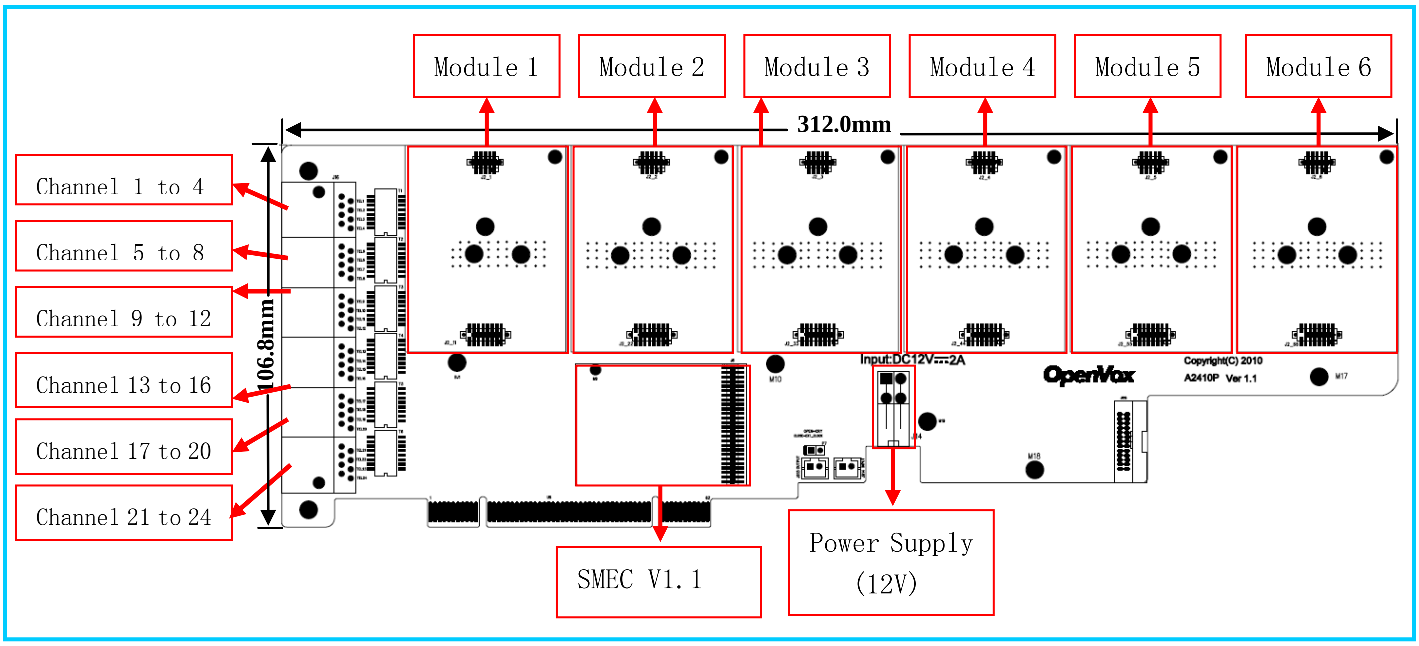


Figure 2: A2410P/AE2410P Hardware Configuration

2. Checking cabling/pin assignment: There are six modules on A2410P/AE2410P, each module (FXS400/FXO400) corresponds one RJ45 interface (refer figure 2). There are eight pins on each RJ45 interface. A2410P/AE2410P uses the two pins of it as a pair, connecting to two-wire telephone line, so each RJ45 interface can split into four telephone lines. Please see figure 3 for the setting of A2410P/AE2410P.

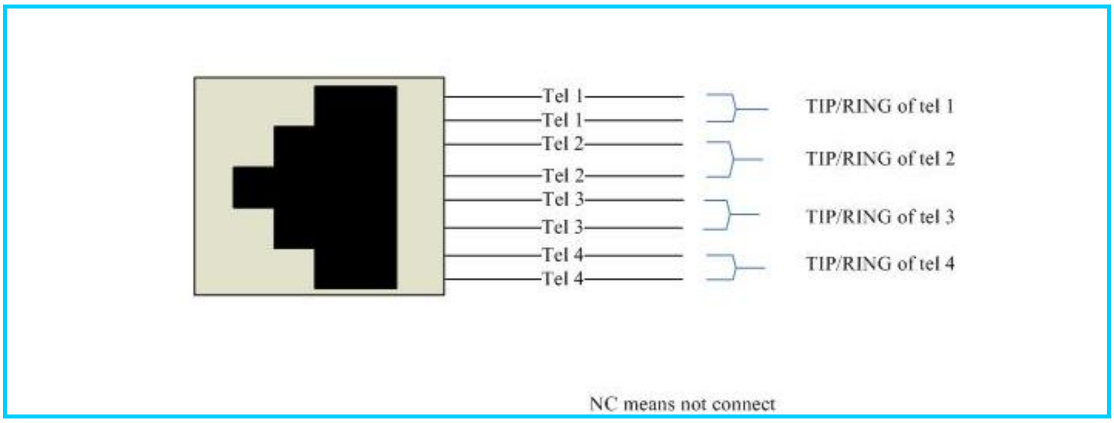


Figure 3: A2410P/AE2410P pin assignment

3. Splitting the RJ45 to RJ11: User has to use a splitter (refer figure 4) to split RJ45 interface to four RJ11 normal telephone line. And please connect PSTN line into FXO port and telephone into FXS port.

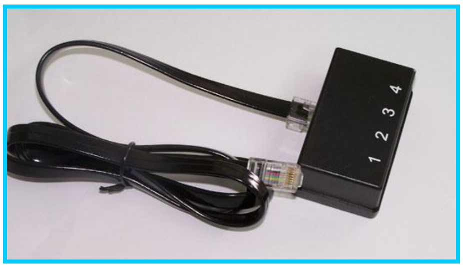


Figure 4: A2410P/AE2410P Splitter

# 3. Software Installation and Configuration

## 3.1 Download dahdi and driver

1) Checking the A2410P/AE2410P hardware by command: *#lspci -vvvvv*

From the following, user can see that there is a device called communication controller interface be found.

```
01:02.0 Communication controller: Unknown device 1b74:2410 (rev 01)
        Subsystem: Unknown device 1b74:0001
        Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV+ VGASnoop- ParErr- Stepping- SERR-
FastB2B-
        Status:  Cap-  66MHz-  UDF-  FastB2B-  ParErr-  DEVSEL=slow  >TAbort-  <TAbort-
<MAbort- >SERR- <PERR-
        Latency: 64, Cache Line Size: 16 bytes
        Interrupt: pin A routed to IRQ 6
              Region 0: Memory at dec00000 (32-bit, non-prefetchable) [size=1M]
```

2) Install compiling environment

```
# yum install gcc
# yum install kernel-devel-`uname –r`
```

3) Check the version of dahdi

If dahdi version on Trixbox is 2.2.0 or below,you must download dahdi 2.2.0.1 or above since dahdi 2.2.0 or below is not be used normally.

```
# modinfo /lib/modules/`uname -r`/dahdi/dahdi.ko
```

4) Download dahdi:

```
# cd /usr/src
# wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/releases/dahdi-linux-complete-2.2.0.2+2.2.0.tar.gz
# tar -zxvf dahdi-linux-complete-2.2.0.2+2.2.0
```

5)Download A2410P/AE2410P driver:

```
# cd /usr/src/ dahdi-linux-complete-2.2.0.2+2.2.0/linux/drivers/dahdi/
# wget http://downloads.openvox.cn/pub/drivers/dahdi-patches/a2410p/opvxa24xx_dahdi-linux.tar.gz
# tar -zxvf opvxa24xx_dahdi-linux.tar.gz
```

6)If user uses AE2410P,user should download firmware:

```
# cd /usr/src
# wget http://downloads.openvox.cn/pub/firmwares/opvx-dahdi-fw-oct6114-032-1.07.01.tar.gz
# tar -zxvf opvx-dahdi-fw-oct6114-032-1.07.01.tar.gz
# cp dahdi-fw-oct6114-032.bin /lib/firmware/
# mkdir -p /usr/lib/hotplug/firmware/
# mv dahdi-fw-oct6114-032.bin /usr/lib/hotplug/firmware/
```

## 3.2 Edit the configure file

To edit the configure file, user must follow these steps:

1. Add opvxa24xx in /usr/src/dahdi-xx/linux/build_tools/live_dahdi

   *# vi /usr/src/dahdi-xx/linux/build_tools/live_dahdi*

   add opvxa24xx shown in red

```
   MODULES_LOAD="$MODULES_LOAD xpp/xpp_usb"
          ;;
     wctdm24xxp | wct4xxp | wcte12xp | wctc4xp | opvxa24xx)
          MODULES_LOAD="$MODULES_LOAD $mod/$mod"
```

2. Add opvxa24xx in /usr/src/dahdi-xx/linux/drivers/dahdi/Kbuild

   *# vi /usr/src/dahdi-xx/linux/drivers/dahdi/Kbuild*

```
obj-$(DAHDI_BUILD_ALL)$(CONFIG_DAHDI_WCT4XXP)        += wct4xxp/
obj-$(DAHDI_BUILD_ALL)$(CONFIG_DAHDI_WCT4XXP)        += opvxa24xx/
obj-$(DAHDI_BUILD_ALL)$(CONFIG_DAHDI_WCTC4XXP)       += wctc4xxp/
```

3. Add opvxa24xx in /usr/src/dahdi-xx/linux/drivers/dahdi/Kconfig

   *# vi /usr/src/dahdi-xx/linux/drivers/dahdi/Kconfig*

```
config DAHDI_WCT4XXP
 tristate "Digium Wildcard dual- and quad-T1/E1/J1 Support"
 depends on DAHDI && PCI
 default DAHDI
 ---help---
   This driver provides support for the following Digium
   Wildcard products:

    * TE205/206/207/210/211/212P (PCI/PCI-X)
   * TE220 (PCI-E)
    * TE405/406/407/410/411/412P (PCI/PCI-X)
   * TE420 (PCI-E)

   To compile this driver as a module, choose M here: the
   module will be called wct4xxp.

   If unsure, say Y.

 config DAHDI_OPVXA24XX
 tristate "OpenVox 24 ports analog card Support"
 depends on DAHDI && PCI
 default DAHDI
 ---help---
```

```
This driver provides support for the following OpenVox
Wildcard products:
* A2410P (PCI)

To compile this driver as a module, choose M here: the
module will be called opvxa24xx.

If unsure, say Y.
```

4. Add opvxa24xx in /usr/src/dahdi-xx/tools/xpp/perl_modules/Dahdi/Hardware/PCI.pm
   *# vi /usr/src/dahdi-xx/tools/xpp/perl_modules/Dahdi/Hardware/PCI.pm*

```
# from opvxa24xx
'1b74:2410'     => { DRIVER => 'opvxa24xx', DESCRIPTION => 'OpenVox A2410P' },

# from wctdm24xxp
'd161:2400'        => { DRIVER => 'wctdm24xxp', DESCRIPTION => 'Wildcard TDM2400P' },
'd161:0800'        => { DRIVER => 'wctdm24xxp', DESCRIPTION => 'Wildcard TDM800P' },
```

5. Add opvxa24xx in /usr/src/dahdi-xx/tools/xpp/perl_modules/Dahdi/Chans.pm
   *# vi /usr/src/dahdi-xx/tools/xpp/perl_modules/Dahdi/Chans.pm*

```
my $dahdi_cfg = $ENV{DAHDI_CFG} || '/usr/sbin/dahdi_cfg';
sub probe_type($) {
 my $self = shift;
 my $fqn = $self->fqn;
 my $num = $self->num;
 my $type;

 if($fqn =~ m:WCTDM/|WRTDM/|OPVXA1200/|OPVXA24XX/:) {
      my %maybe;

      undef %maybe;
      foreach my $sig (qw(fxo fxs)) {
           my $cmd = "echo ${sig}ks=$num | $dahdi_cfg -c /dev/fd/0";

           $maybe{$sig} = system("$cmd >/dev/null 2>&1") == 0;
      }
```

6. Add opvxa24xx in /usr/src/dahdi-xx/tools/modules.sample
   *# vi /usr/src/dahdi-xx/tools/modules.sample*

```
# Digium TE205P/TE207P/TE210P/TE212P: PCI dual-port T1/E1/J1
# Digium TE405P/TE407P/TE410P/TE412P: PCI quad-port T1/E1/J1
# Digium TE220: PCI-Express dual-port T1/E1/J1
# Digium TE420: PCI-Express quad-port T1/E1/J1
wct4xxp
# OpenVox A2410P: up to 24 analog ports
opvxa24xx
```

7. Add opvxa24xx in /usr/src/dahdi-xx/tools/blacklist.sample

   *# vi /usr/src/dahdi-xx/tools/blacklist.sample*

```
blacklist wctdm
blacklist opvxa24xx
blacklist wctc4xxp
blacklist wcb4xxp
```

## 3.3 Installation

User can install the driver via the following steps (assuming user has the source code of dahdi device driver installed in /usr/src/dahdi-2.2.XX directory):

Compiling

execute the commands:

```
# cd /usr/src/dahdi-linux-complete-XX
# ./configure
# make
# make install
# make config
```

## 3.4 Configure

1. Loading modules for opvxa24xx:

```
# modprobe dahdi
# modprobe opvxa24xx opermode=YOUR COUNTRY
# dahdi_genconf
# dahdi_cfg  -vvvv
```

Then run *# dmesg*

If user uses AE2410P, from the following, user can find EC module has been detected.

```
OpenVox A2410P version: 1.0
Module 0: Installed -- AUTO FXO (FCC mode)
Module 1: Installed -- AUTO FXO (FCC mode)
Module 2: Installed -- AUTO FXO (FCC mode)
Module 3: Installed -- AUTO FXO (FCC mode)
Module 4: Installed -- AUTO FXO (FCC mode)
Module 5: Installed -- AUTO FXO (FCC mode)
Module 6: Installed -- AUTO FXO (FCC mode)
Module 7: Installed -- AUTO FXO (FCC mode)
Module 8: Installed -- AUTO FXO (FCC mode)
Module 9: Installed -- AUTO FXO (FCC mode)
Module 10: Installed -- AUTO FXO (FCC mode)
Module 11: Installed -- AUTO FXO (FCC mode)
Module 12: Installed -- AUTO FXS/DPO
Module 13: Installed -- AUTO FXS/DPO
Module 14: Installed -- AUTO FXS/DPO
Module 15: Installed -- AUTO FXS/DPO
Module 16: Installed -- AUTO FXS/DPO
Module 17: Installed -- AUTO FXS/DPO
Module 18: Installed -- AUTO FXS/DPO
Module 19: Installed -- AUTO FXS/DPO
Module 20: Installed -- AUTO FXS/DPO
Module 21: Installed -- AUTO FXS/DPO
Module 22: Installed -- AUTO FXS/DPO
Module 23: Installed -- AUTO FXS/DPO
OpenVox VPM: echo cancellation supports 32 channels
OpenVox VPM: echo cancellation for 32 channels
OpenVox VPM: hardware DTMF disabled.
OpenVox VPM: Present and operational servicing 1 span(s)
Found an OpenVox A2410P: Version 1.0 (24 modules)
```

2. Checking the configure files

   The command *dahdi_genconf* will automatically generate the system.conf under /etc/dahdi and dahdi-channels.conf under /etc/asterisk. Please check the setting of system.conf, it looks like the following:

```
# Autogenerated by /usr/sbin/dahdi_genconf on Wed Sep 29 12:02:59 2010
# 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: OPVXA24XX/24 "OpenVox A2410 Board 25" (MASTER)
fxsks=1
fxsks=2
fxsks=3
fxsks=4
…
…
fxsks=12
fxoks=13
fxoks=14
fxoks=15
fxoks=16
…
…
fxoks=24



# Global data

loadzone        = us      ;Please change to your country
```

```
defaultzone     = us     ;Please change to your country
```

Edit /etc/asterisk/indications.conf

```
country = us     ;Please change to your country
```

File dahdi-channels.conf should look like the following:
FXO ports use FXS signaling, the configure file looks like this:

```
; Autogenerated by /usr/sbin/dahdi_genconf on Wed Sep 29 12:03:00 2010
; 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: DAHDI_DUMMY/1 "DAHDI_DUMMY/1 (source: Linux26) 1" (MASTER)

; Span 2: OPVXA24XX/24 "OpenVox A2410 Board 25"
;;; line="1 OPVXA24XX/24/0"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 1
callerid=
group=
context=default

;;; line="2 OPVXA24XX/24/1"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 2
callerid=
group=
context=default
…
…

;;; line="12 OPVXA24XX/24/11"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 11
callerid=
group=
context=default
```

FXS ports use FXO signalling, the configure file looks like this:

```
;;; line=″13 OPVXA24XX/24/12″
signalling=fxo_ks
callerid=″Channel 13″ <4013>
mailbox=4013
group=5
context=from-internal
channel => 13
callerid=
mailbox=
group=
context=default

;;; line=″14 OPVXA24XX/24/13″
signalling=fxo_ks
callerid=″Channel 14″ <4014>
mailbox=4014
group=5
context=from-internal
channel => 14
callerid=
mailbox=
group=
context=default
…
…

;;; line=″24 OPVXA24XX/24/23″
signalling=fxo_ks
callerid=″Channel 24″ <4024>
mailbox=4024
group=5
context=from-internal
channel => 24
callerid=
mailbox=
group=
context=default
```

3. Starting asterisk

Checking the dahdi channel loading from asterisk console:

*# asterisk –vvvvvvvgc*

Entering asterisk console, run command: *dahdi show channels*. If dahdi channels can be shown, which means the dahdi channels have been loaded into asterisk.

```
*CLI> dahdi show channels
   Chan Extension  Context         Language   MOH Interpret       Blocked    State
 pseudo            default         en         default                        In Service
      1            from-pstn       en         default                        In Service
      2            from-pstn       en         default                        In Service
      3            from-pstn       en         default                        In Service
   4          from-pstn      en        default                       In Service
   5          from-pstn      en        default                       In Service
   6          from-pstn      en        default                       In Service
   7          from-pstn      en        default                       In Service
   8          from-pstn      en        default                       In Service
   9          from-pstn      en        default                       In Service
  10          from-pstn      en        default                       In Service
  11          from-pstn      en        default                       In Service
  12          from-pstn      en        default                       In Service
  13          from-internal  en        default                       In Service
  14          from-internal  en        default                       In Service
  15          from-internal  en        default                       In Service
  16          from-internal  en        default                       In Service
  17          from-internal  en        default                       In Service
  18          from-internal  en        default                       In Service
  19          from-internal  en        default                       In Service
  20          from-internal  en        default                       In Service
  21          from-internal  en        default                       In Service
  22          from-internal  en        default                       In Service
  23          from-internal  en        default                       In Service
  24          from-internal  en        default                       In Service
```

The status of channel looks like the following. If user uses AE2410P, EC Status of active channel should be ON; otherwise it would be OFF.

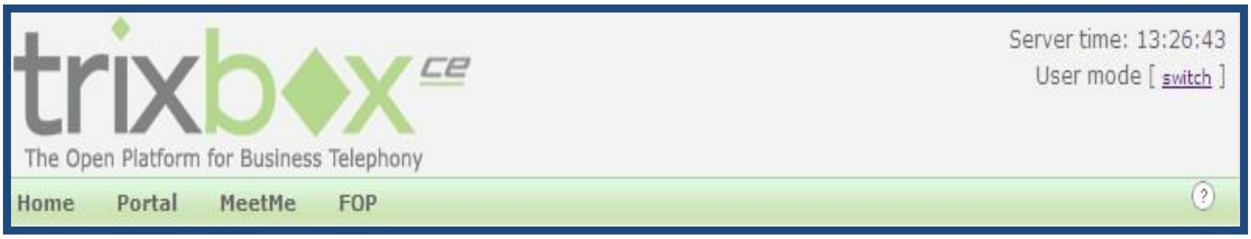
```
trixbox1*CLI> dahdi show channel 13
Channel: 13I>
File Descriptor: 25
Span: 11*CLI>
Extension: I>
Dialing: noI>
Context: from-internal
Caller ID: 111
Calling TON: 0
Caller ID name: device
Mailbox: 111@device
Destroy: 0LI>
InAlarm: 0LI>
Signalling Type: FXO Kewlstart
Radio: 0*CLI>
Owner: DAHDI/13-1
Real: DAHDI/13-1
Callwait: <None>
Threeway: <None>
```

```
Confno: -1LI>
Propagated Conference: -1
Real in conference: 0
DSP: yes*CLI>
Busy Detection: no
TDD: no1*CLI>
Relax DTMF: no
Dialing/CallwaitCAS: 0/0
Default law: ulaw
Fax Handled: no
Pulse phone: no
DND: no1*CLI>
Echo Cancellation:
trixbox1128 taps
trixbox1(unless TDM bridged) currently ON
Actual Confinfo: Num/0, Mode/0x0000
Actual Confmute: No
Hookstate (FXS only): Offhook
```

# 4. Configure on Trixbox webpage

## 4.1 Log into Trixbox Webpage

Please open your browser and enter the PBX IP address, then click “switch” button on the up right corner of the page in the following screen.

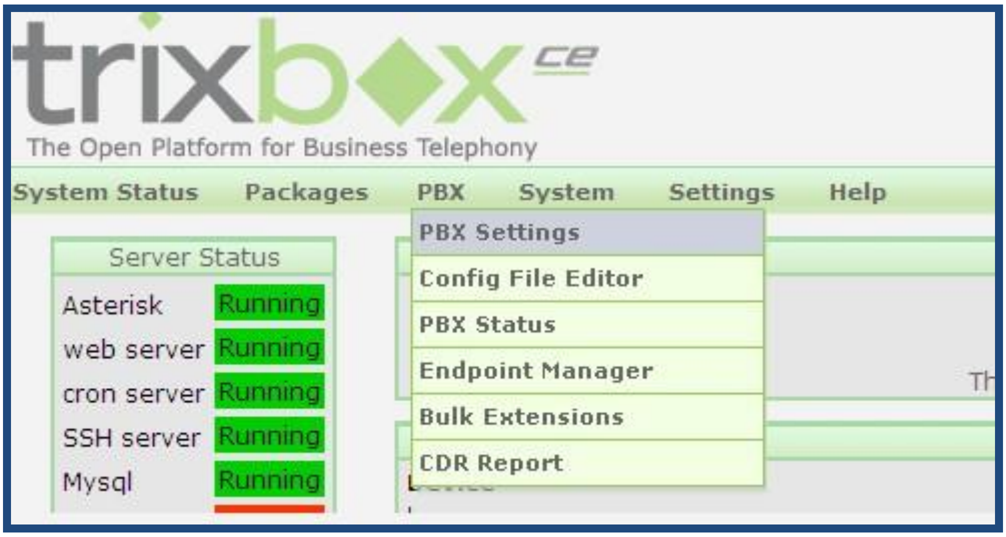


Input Username: maint; Password: password, then click on Submit button and you can enter the trixbox GUI.

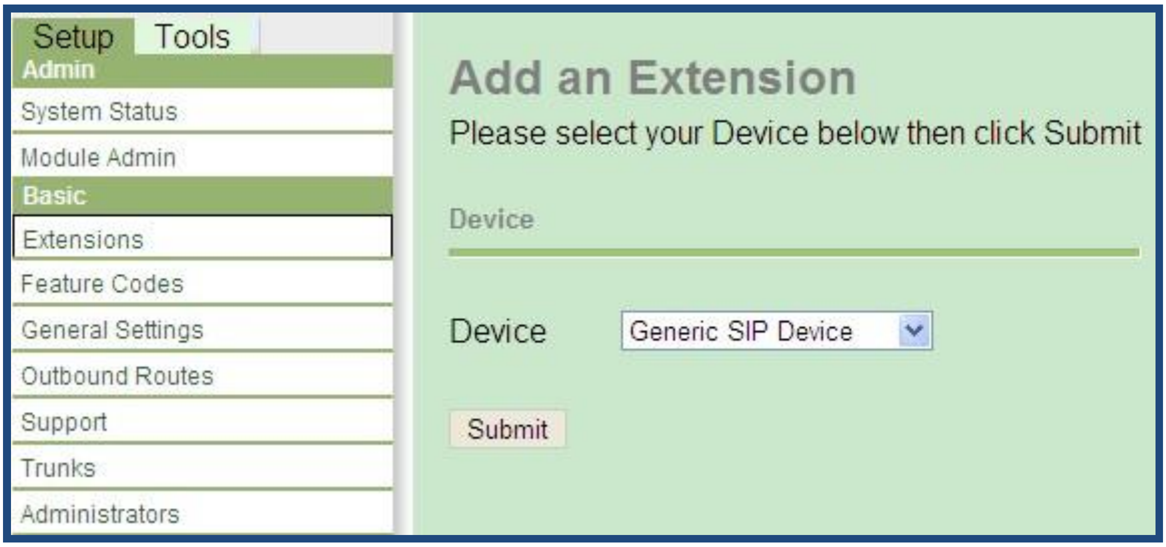
## 4.2 Create Extensions

Add two extensions, then user can plug two phones to FXS, after that the two phones can call each other.

1) Click PBX→PBX Settings button in the following illustration:



2) Click on Extensions bar in the following illustration:



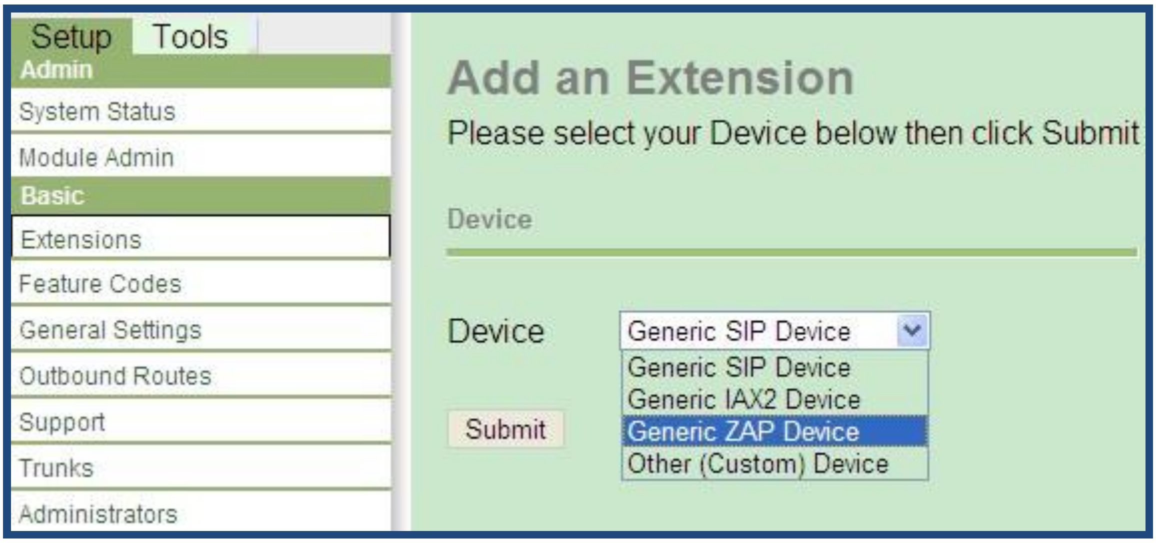
Add an Extension

Please select your Device below then click Submit

Device

Device: Generic SIP Device

From the drop down selection box, select Generic Zap Device since we are going to create zap extension, then click on submit button. Please refer to the following illustration:



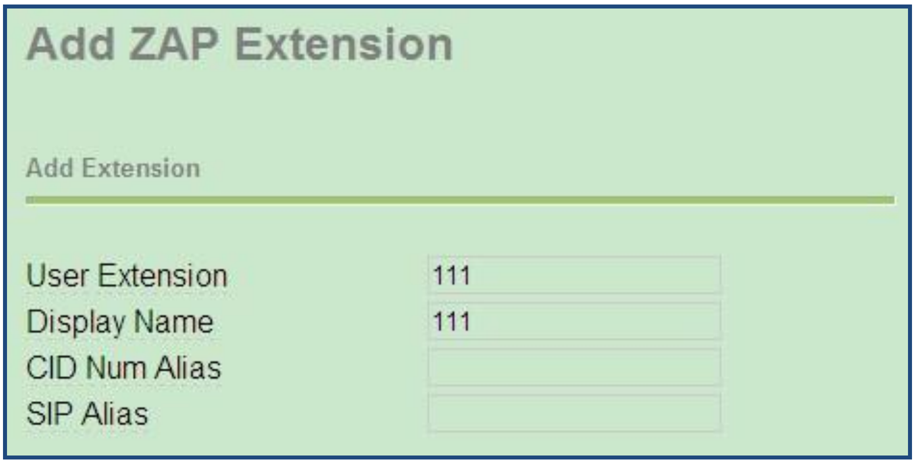
Add an Extension

Please select your Device below then click Submit

Device

Device: Generic SIP Device

3) Set up User Extension: 111 (that’ s the extension number I gave for reception); Display Name: 111 in the following illustration:

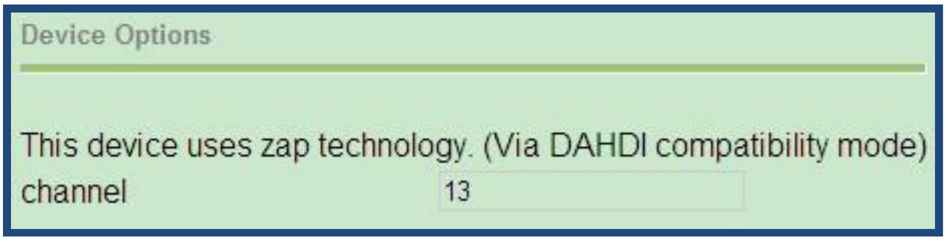


Add ZAP Extension

Add Extension

| | |
|---|---|
| User Extension | 111 |
| Display Name | 111 |
| CID Num Alias | |
| SIP Alias | |

4) Input a channel number for your extension, please refer to the following illustration.

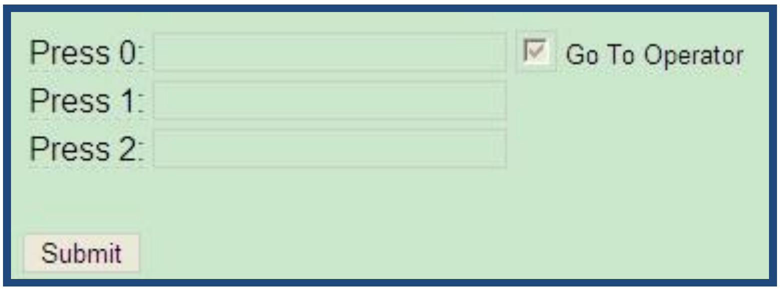


Device Options

This device uses zap technology. (Via DAHDI compatibility mode)

channel: 13

5) Then click on submit button.

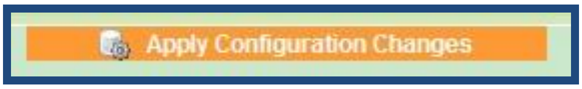


Press 0: ☑ Go To Operator

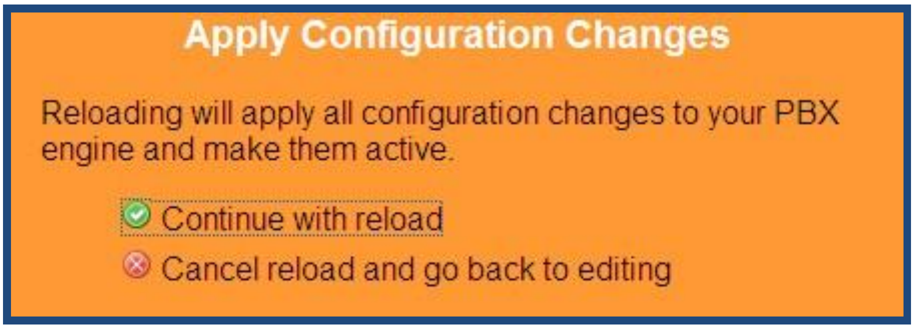
Press 1:

Press 2:

6) Click the “Apply Configuration Changes Here” bar in the top of the screen.



And click the “Continue with reload” in the popup dialog box.

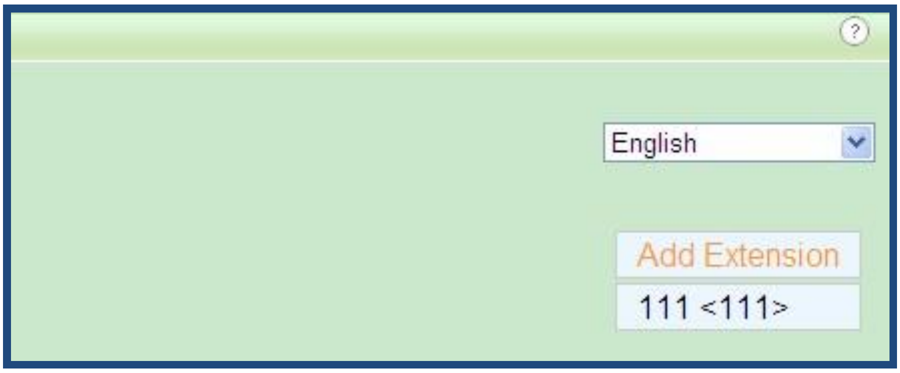


Apply Configuration Changes

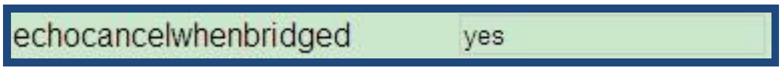
Reloading will apply all configuration changes to your PBX engine and make them active.

7) If you use AE2410P,please refer to the following steps,otherwise jump to 8).

a. Click on 111<111> button on the up right corner of the page, please refer to the following illustration:

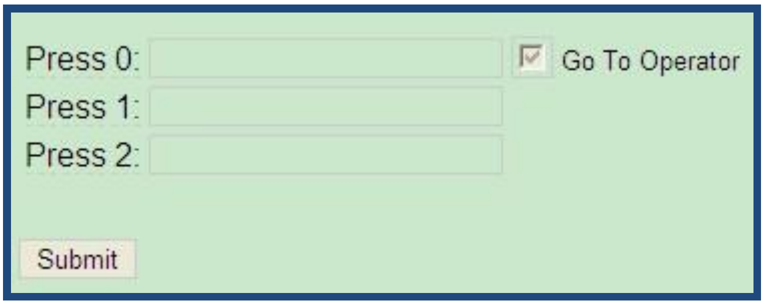


b. Set up echocancelwhenbridged :yes in the following instruction.



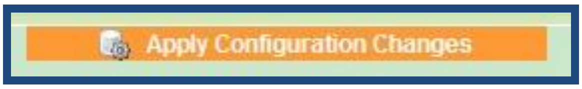
echocancelwhenbridged yes

c. Then click on submit button.

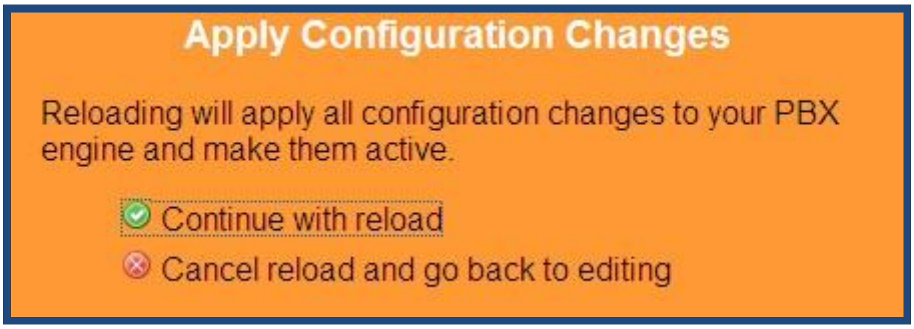


Press 0: ☑ Go To Operator
Press 1:
Press 2:

d. Click the “Apply Configuration Changes Here” bar in the top of the screen.



And click the “Continue with reload” in the popup dialog box.



Apply Configuration Changes

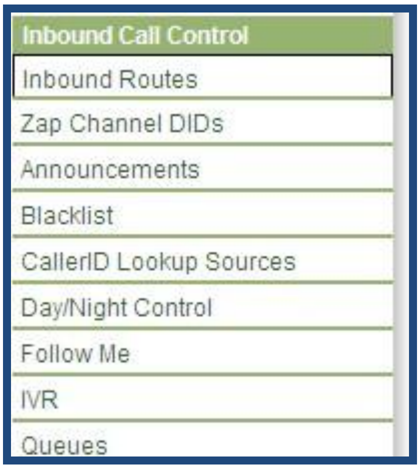
Reloading will apply all configuration changes to your PBX engine and make them active.

8) Please repeat the above steps to add another extension 222 on channel 6. After these, we can dial each other between 111 and 222.

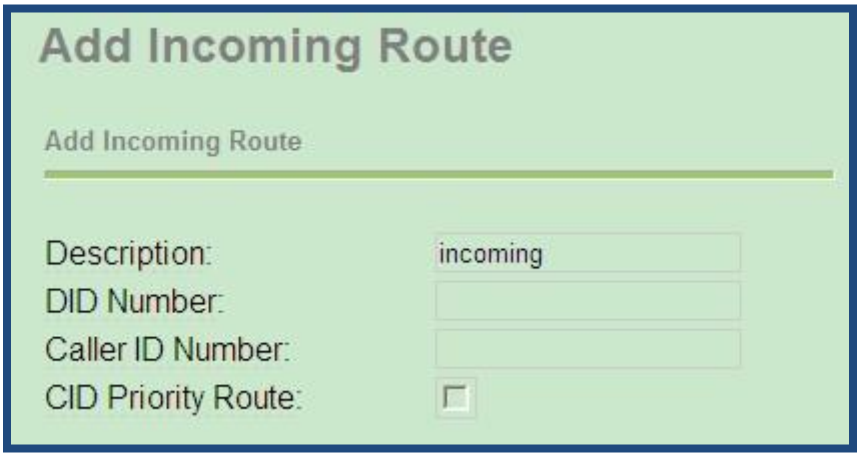
## 4.3 Add Incoming Route

Add an incoming route, it will help you get incoming calls.

1) Click on “Inbound Routes” bar.



2)  Input incoming1 in the textbox of Description as the following screen:



Add Incoming Route

Add Incoming Route

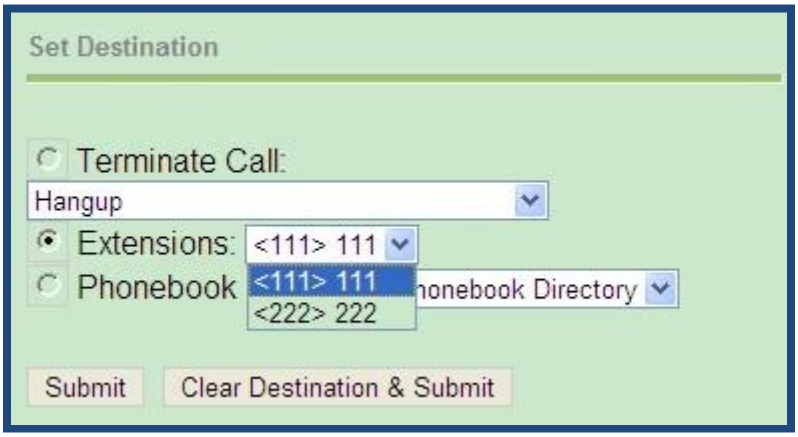
Description: incoming

DID Number:

Caller ID Number:

CID Priority Route: □

3) Click the Extensions textbox, choose an extension number from the drop down selection box. Then click submit. Please refer to the following illustration:



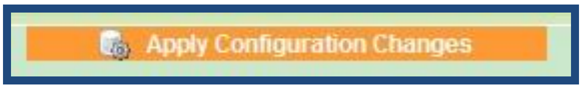
Set Destination

○ Terminate Call:
Hangup

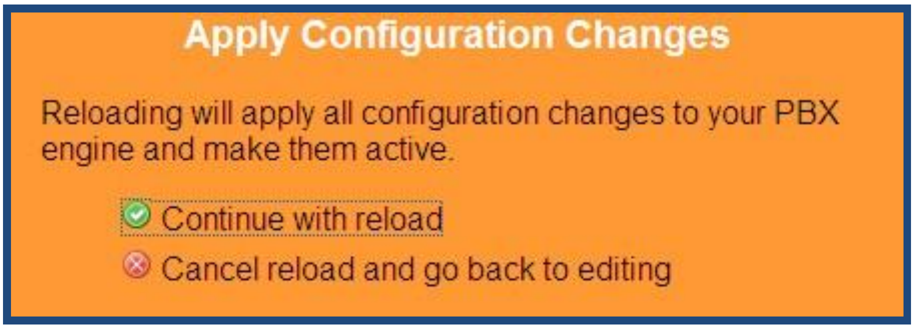
● Extensions: <111> 111
<111> 111
<222> 222

○ Phonebook  honebook Directory

4)  Click the “Apply Configuration Changes ” bar in the top of the screen.



And click the “Continue with reload” in the popup dialog box.



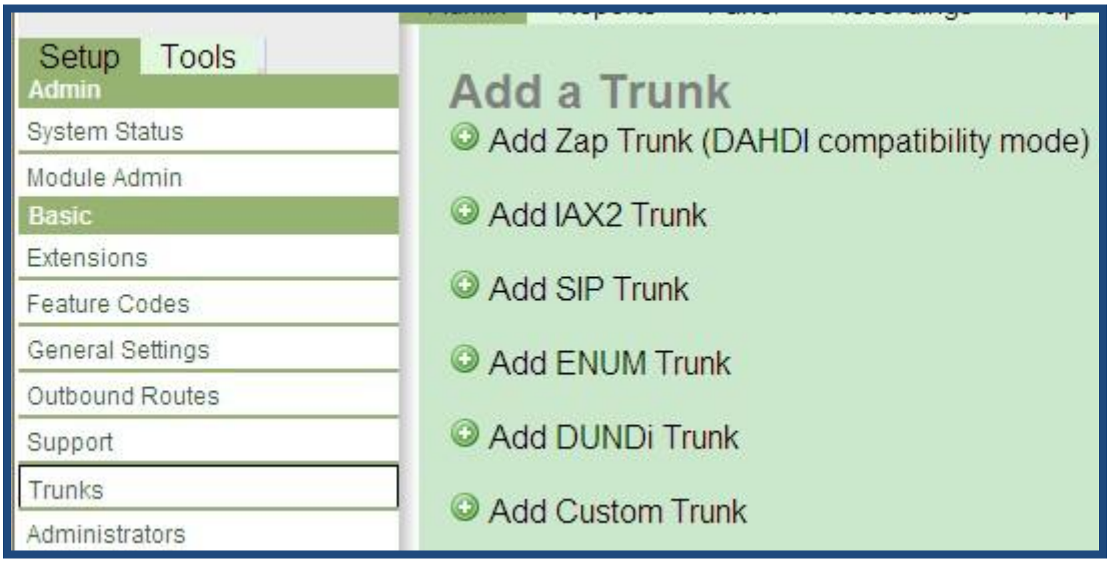
Apply Configuration Changes

Reloading will apply all configuration changes to your PBX engine and make them active.

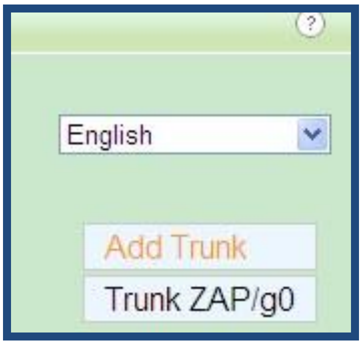
## 4.4 Create Trunk

To dial out, you have to create trunk.

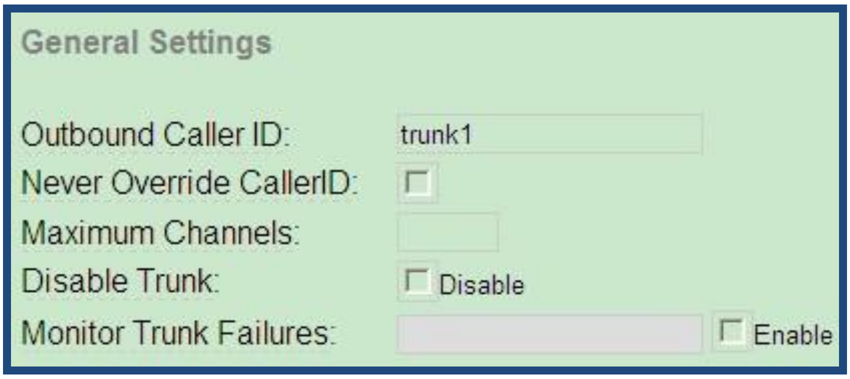
1) Click on Trunks bar as the following illustration:



2) Click on Trunk ZAP/g0 button on the up right corner of the page, please refer to the following illustration:



3) Input trunk1 in the textbox of Outbound Caller ID, please refer to the following illustration:



General Settings

Outbound Caller ID: trunk1

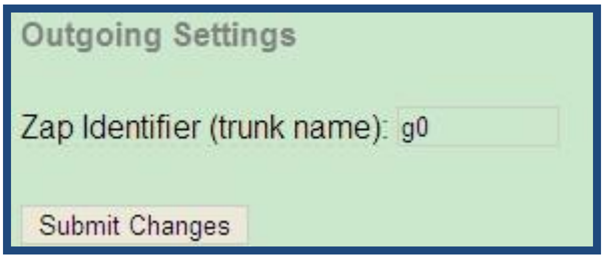
Never Override CallerID: □

Maximum Channels:

Disable Trunk: □ Disable

Monitor Trunk Failures: □ Enable

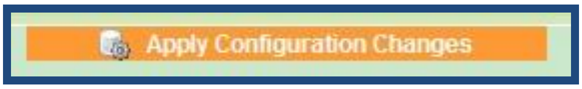
4) Click on Submit Changes button in the following screen:



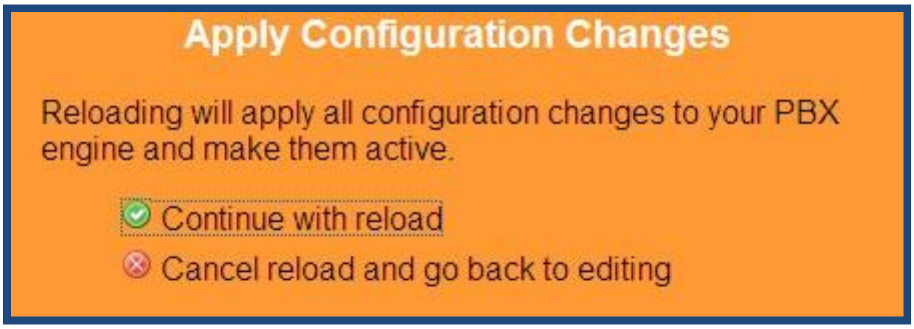
Outgoing Settings

Zap Identifier (trunk name): g0

5) Click the “Apply Configuration Changes Here” bar in the top of the screen.



And click the “Continue with reload” in the popup dialog box.



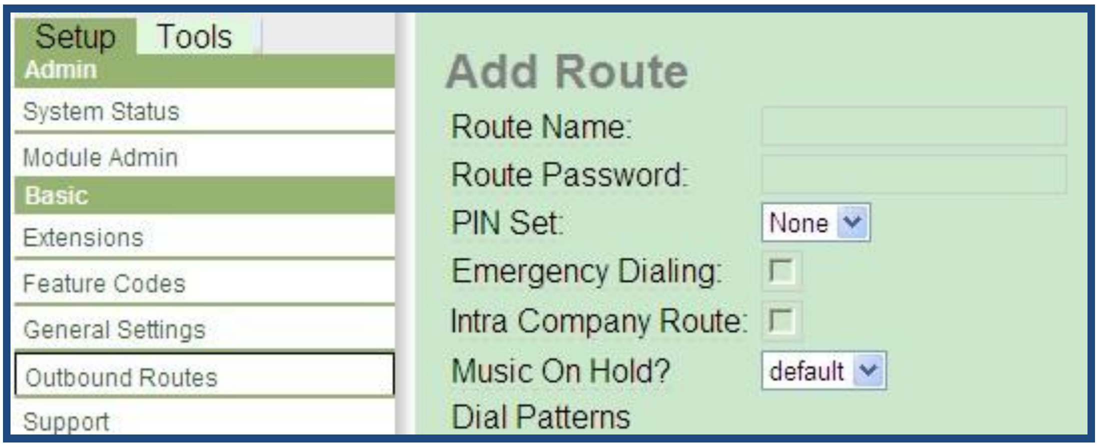
Apply Configuration Changes

Reloading will apply all configuration changes to your PBX engine and make them active.

## 4.5 Set up Outbound Route

Through outbound routes, user can dial out.

1) Please click on the Outbound Routes as the following illustration:



Add Route

Route Name:

Route Password:

PIN Set: None

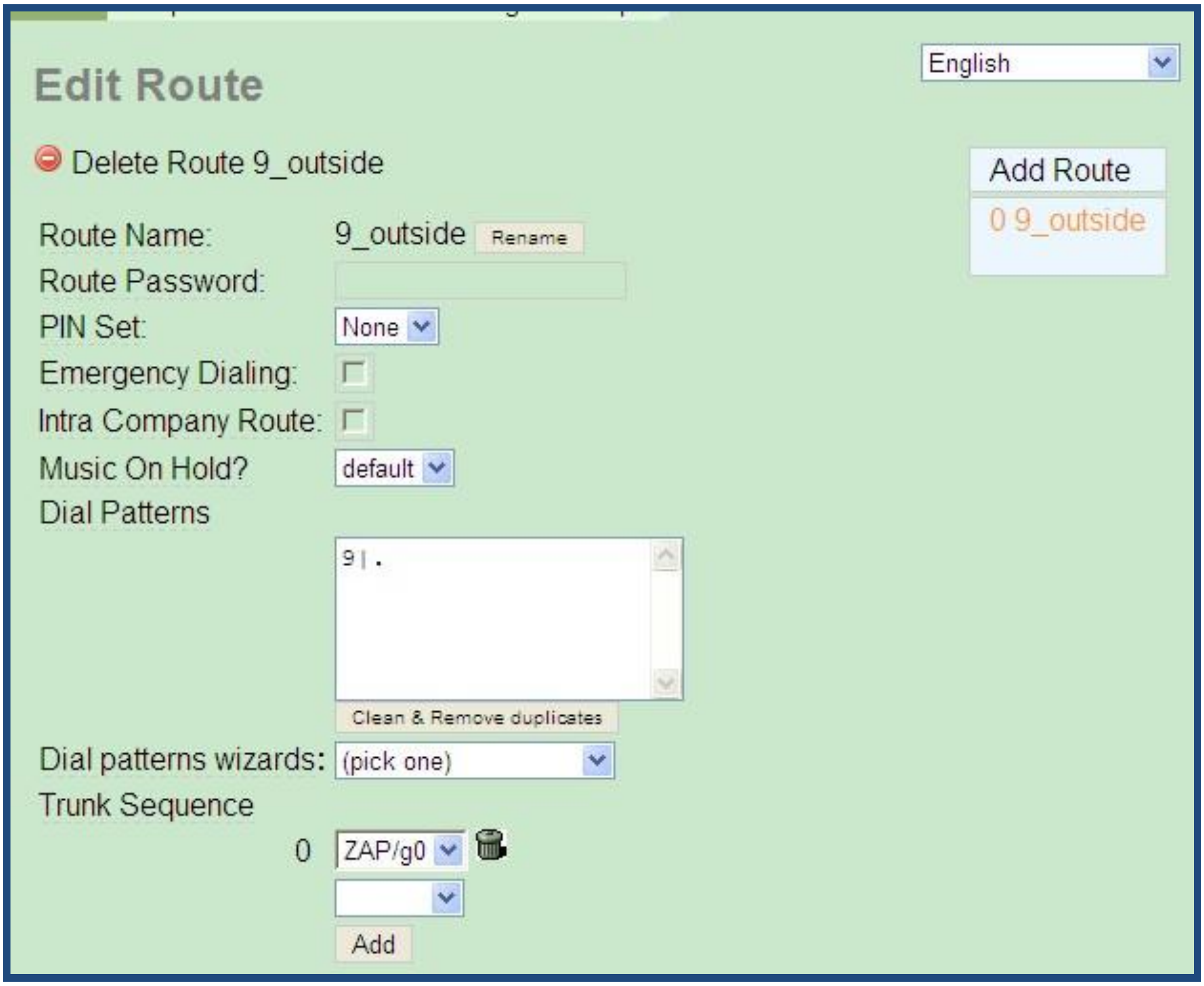
Emergency Dialing: □

Intra Company Route: □

Music On Hold? default

Dial Patterns

2) Please click “09_outside button”, keep the default parameters in the following screen. Please refer to the following illustration:



Edit Route

English

Delete Route 9_outside

Add Route
0 9_outside

Route Name: 9_outside Rename
Route Password:
PIN Set: None
Emergency Dialing: □
Intra Company Route: □
Music On Hold? default
Dial Patterns

```
9|.
```

Clean & Remove duplicates

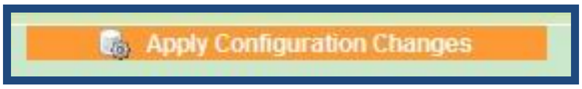
Dial patterns wizards: (pick one)

Trunk Sequence
0 ZAP/g0

Add

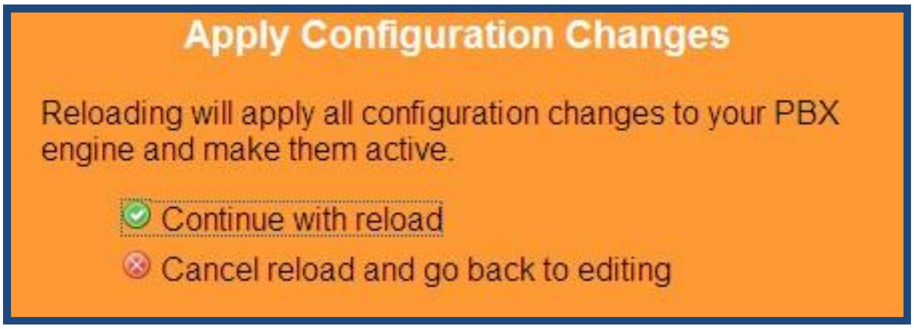
Here we use the default Dial Pattern 9|., so we have to add prefix 9, when we dial out. For example, if you want to call 123456, then you have to press 9123456 on your phone.

3) Click the “Apply Configuration Changes” bar in the top of the screen.



Apply Configuration Changes

And click the “Continue with reload” in the popup dialog box.



Apply Configuration Changes

Reloading will apply all configuration changes to your PBX engine and make them active.

Continue with reload
Cancel reload and go back to editing

Notes:

Test environments are:

Trixbox：2.8.0.3

Kernel version: 2.6.18-128.1.10.el5

Dahdi: dahdi-linux-complete-2.2.0.2+2.2.0

Asterisk: 1.6.0.10

Hardware: OpenVox A2410P/AE2410P

# 5. References

www.openvox.cn

www.digium.com

www.asterisk.org

www.voip-info.org

www.asteriskguru.com