

## Cum se verifica o placa Digium intr-un server de tip PBX

In cadrul acestui tutorial vom exemplifica un scenariu prin care se poate verifica o placa VoIP folosind un disc de boot "Live CD" care contine o aplicatie Asterisk.

Un disc de boot "Live CD" este o varianta a unui sistem de operare ce poate fi incarcat pe un calculator fara instalarea efectiva a sistemului de operare. Calculatorul va incarca sistemul de operare de pe CD/DVD si va porni sistemul de operare fara a accesa HDD-ul.

Folosirea unui "Live CD" este o metoda neinvaziva de a executa teste, ideala in acele cazuri in care nu avem acces la un sistem cu un PBX pre-instalat sau cand nu vrem sa modificam configuratia unei centrale folosite in productie. In aceste cazuri se identifica un sistem liber care poate fi folosit temporar pentru testele placii VoIP si care dupa executia testului redevine liber si fara modificari in continutul hard disk-ului.

In cele ce urmeaza vom folosi o editie "Live CD" produsa de Xorcom: [Live CD Quick Start v1.2](#). Dupa download-ul imaginii de CD de la adresa de mai sus, aceasta trebuie inregistrata pe un CD inscriptionabil.

Pentru a verifica functionalitatea unei placii VoIP Digium analogica urmatoorii pasi trebuie efectuati:

1. Se deconecteaza calculatorul de la sursa de curent.
2. Se verifica daca pe placa de baza exista un slot, PCI/PCI Express in functie de tipul de placa ales, liber.
3. Se introduce placa in slotul aferent de pe placa de baza.
4. Daca placa contine si module FXS se cupleaza si alimentarea in conectorul prezent pe placa.
5. Se fixeaza placa de carcasa calculatorului cu un surub.
6. Se cupleaza calculatorul la sursa de curent si se porneste.
7. Se introduce Live CD-ul in unitatea optica iar calculatorul va boota de pe CD fara a folosi hard disk-ul. In cazul in care calculatorul nu booteaza dupa CD se va schimba optiunea de bootare din BIOS.
8. Xorcom Rapid™ Live detecteaza placa Digium si incarca automat driverele necesare.
9. Verificarea/modificarea configuratiei echipamentului VoIP, prezent pe masina, se poate face prin doua metode: folosind Xorcom Rapid™ menu (interfata de configurare/administrare) sau direct din consola. In cazul exemplului nostru vom folosi metoda din consola. Pentru a reveni in Xorcom Rapid™ menu din consola se ruleaza comanda **rapid-menu**.
10. Din Rapid-menu se selecteaza optiunea "**Maintenance**" => "**Change root password**" si se introduce o parola noua. Folosind comanda Ctrl+C se paraseste Xorcom Rapid.
11. Se logheaza pe masina cu userul **root** si parola setata la punctul anterior.
12. Din consola se ruleaza comanda **lszaptel**:

```
tenora:~# lszaptel
### Span 1: WCTDM/0 "Wildcard TDM800P Board 1" (MASTER)
 1 FXO      FXOKS      (In use)
 2 FXS      FXSKS      (In use)
 3 FXS      FXSKS
 4 FXS      FXSKS
 5 FXS      FXSKS
 6 FXS      FXSKS
 7 FXS      FXSKS
 8 FXS      FXSKS
```

13. Pentru a verifica daca tipul de canal a fost configurat corespunzator de catre aplicatia Xorcom Rapid™, se ruleaza comanda **ztcfg -vvv**.

```
tenora:~# ztcfg -vvv
Zaptel Version: 1.4.10.1
Echo Cancellor: MG2
Configuration
=====
Channel map:
Channel 01: FXO Kewlstart (Default) (Slaves: 01)
Channel 02: FXS Kewlstart (Default) (Slaves: 02)
```

Dupa cum se observa canalul 1 este setat cu semnalizare de tip FXO Kewlstart, (ceea ce corespunde modului FXS), iar canalul 2 este setat cu semnalizare de tipul FXS Kewlstart (corespunde modului FXO). Ca o observatie, intotdeauna tipul de semnalizare care trebuie ales este cel opus tipului de modul.




14. In cazul folosirii Live CD-ului contextele si logica de tratare a unui apel (dialplan) de catre aplicatia Asterisk sunt generate automat si se gasesc in **/etc/asterisk/zapata-channels.conf**, pentru modulul fxo si in **/etc/asterisk/zapata\_additional.conf**, pentru modulul fxs.

Descrierea contextelor mai pe larg se poate gasi in articolul "[Despre VoIP-Cum ne conectam](#)" precum si in cartea "[Asterisk: The future of telephony](#)".




### zapata-channels.conf

```
; Span 1: WCTDM/0 "Wildcard TDM800P Board 1" (MASTER)
;;; line="1 WCTDM/0/0"
```

## Meniu Principal

-  [Home](#)
-  [Modulo Consulting](#)
-  [Produse VoIP](#)
-  [Proiecte software](#)
-  [Articole](#)
-  [Contact](#)

## Tutoriale

-  [Instalarea placilor telefonice](#)
-  [Cum se verifica o placa Digium intr-un server de tip PBX](#)
-  [Configurarea Linksys SPA962 cu Linksys SPA932](#)

```
;;; line="2 WCTDM/0/1"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 2
```

#### zapata\_additional.conf

```
;;;; [401]
signalling=fxo_ks
record_out=Adhoc
record_in=Adhoc
mailbox=401
immediate=no
group=5
echotraining=no
echocancelwhenbridged=no
echocancel=yes
context=from-internal
callerid=ZAP channel 1 <401>
channel=>1
```

In cazul de fata contextele nu vor fi modificate si se vor pastra configuratiile existente.

15. Se ruleaza comanda asterisk -r si se verifica aparitia celor doua canale in cadrul modulului zaptel folosind comanda **zap show channels**.

```
tenora:~# asterisk -r
tenora*CLI> zap show channels
Chan Extension Context Language MOH Interpret
pseudo from-internal en default
1 from-internal en default
2 from-pstn en default
```

16. Pentru a vedea daca placa functioneaza corespunzator se ruleaza comanda **ztest**. Acest test este folosit pentru determinarea eventualele pierderi de intreruperi in comunicarea dintre placa VoIP si procesor. Placile au nevoie de **1000 intreruperi/sec** pentru a nu intampina probleme in ceea ce priveste calitatea audio. Conform Digium, daca in urma testului rezulta valori de cel putin **99.98%** nu vor fi probleme cu calitatea sunetului. Daca valorile sunt sub **99.975%** calitatea sunetului va fi afectata. In acest caz se vor face verificari suplimentare cum ar fi:
  - placa trebuie sa aiba alocata o intrerupere (IRQ) care sa nu fie impartita cu un alt modul hardware (**lspci -v, cat /proc/interrupts**).Cel mai simplu mod de rezolvare a acestei probleme este re-instalarea placii VoIP intr-un alt slot. Daca aceasta nu rezolva problema de mai sus, se incearca dezactivarea, din BIOS, a modulelor hardware de pe placa de baza care nu sunt absolute necesare, ca de exemplu: portul paralel, porturile seriale, canalul audio, porturile USB, etc.
  - se verifica daca modul X-window este activat (**ps aux | grep X**) si in caz afirmativ se renunta la acest mod.
  - se verifica daca exista hard disk-uri IDE si sunt setate in modul UDMA2 (**hdparm -vi /dev/hda**).Pentru setarea in modul UDMA2 se poate folosi comanda .

## Teste functionale

In continuare vom efectua o serie de teste astfel incat:

- apelurile dinspre telefonul analogic cuplat la portul 1 (modulul FXS) sa fie rutate catre o extensie SIP, pe care urmeaza sa o definim, in cazul in care se formeaza aceasta extensie si catre o linie analogica, daca se formeaza un numar care incepe cu 9.
- apelurile sosite pe linia analogica cuplata la portul 2 (modulul FXO) sa fie redirectionate simultan catre telefonul analogic si catre extensia SIP.

Pentru efectuarea testelor va trebui sa configuram/definim extensiile folosind aplicatia **freePBX**, prezenta pe Live CD.

1. Se creeaza o extensie SIP folosind aplicatia freePBX, prezenta pe Live CD.
2. Se acceseaza interfata web a CD-ului prin intermediul unui browser introducand adresa IP. Adresa IP este afisata in Xorcom Rapid™ menu. Din interfata web se alege freePBX din meniul afisat in partea stanga. Userul si parola default pentru accesarea aplicatiei sunt: **user: admin , passwd: admin**.
3. Din meniul aplicatiei freePBX se selecteaza optiunea Extensions. Dupa cum se poate vedea si din imaginea de mai jos, pentru modulul FXS a fost deja creata o extensie, in cazul acestui exemplu Zap chanel 1<401>.



In cazul in care aceasta extensie nu a fost creata automat, se selecteaza din drop down-ul de pe pagina Generic ZAP device. In pagina de configurare a extensiei nou create, se introduc parametri doriti pentru User Extension si Display Name iar pentru channel se introduce numarul canalului corespunzator modulului FXS (vezi pct.12).

4. Se salveaza modificarile facute si se verifica in /etc/asterisk/zapata.additional.conf daca extensia a fost creata.
5. In continuare se creeaza o extensie SIP. Se selecteaza din drop down-ul de pe pagina Generic SIP device. In pagina de configurare a extensiei nou create, se introduc parametri doriti pentru User Extension, Display Name si secret.
6. Se salveaza modificarile.
7. Se configureaza un telefon IP pentru aceasta extensie. In cazul exemplului nostru am folosit un terminal Linksys SPA922.

1. **Apeluri dinspre telefonul analogic spre estensia SIP** In portul 1 al placii, port ce corespunde modulului FXS se cupleaza un telefon analogic. De pe acest telefon se efectueaza un apel catre extensia SIP creata in prelabil (in exemplul nostru aceasta extensie este 100). In momentul generarii semnalului de apel, telefonul SIP "suna". Acesta va continua sa "sune" pana in momentul in care se decide de a se raspunde apelului. Daca se ruleaza comanda **asterisk -r** se poate urmari desfasurarea apelului folosind

```
tenora:~# asterisk -r
tenora*CLI>
-- Starting simple switch on 'Zap/1-1'
-- Executing [100@from-internal:1] Macro("Zap/1-1", "exten-vm|novm|100") in new sta
-- Executing [s@macro-exten-vm:1] Macro("Zap/1-1", "user-callerid") in new stack
-- Executing [s@macro-user-callerid:1] NoOp("Zap/1-1", "user-callerid: ZAP channel
-- Executing [s@macro-user-callerid:2] Set("Zap/1-1", "AMPUSER=401") in new stack
-- Executing [s@macro-user-callerid:3] GotoIf("Zap/1-1", "0?report") in new stack
```

Desfasurarea unui apel SIP este descrisa mai pe larg in articolul ["Despre\\_VoIP\\_- Ce\\_inseamna\\_SIP"](#).

2. **Apeluri dinspre telefonul analogic catre linia analogica daca se formeaza 9 la inceputul numarului.** In cazul Live CD-ului de la Xorcom, asterisk-ul vine configurat initial pentru efectuarea de apeluri folosind un prefix (de exemplu 9) la inceputul numarului. Configuratia se gaseste in meniul aplicatiei freePBX la optiunea "Outbound Routes". De pe telefonul analogic se formeaza un numar care incepe cu 9 si urmat de numarul la care vrem sa sunam. Terminalul cuplat la linia analogica "suna" in momentul in care apelul este initiat. In log-ul preluat din consola asterisk se poate vedea pas cu pas desfasurarea evenimentului:

```
tenora*CLI>
-- Hungup 'Zap/2-1'
== Spawn extension (macro-dialout-trunk, s, 25) exited non-zero on 'Zap/1-1' in mac
== Spawn extension (macro-dialout-trunk, s, 25) exited non-zero on 'Zap/1-1'
-- Executing [h@macro-dialout-trunk:1] Macro("Zap/1-1", "hangupcall") in new stack
-- Executing [s@macro-hangupcall:1] ResetCDR("Zap/1-1", "w") in new stack
-- Executing [s@macro-hangupcall:2] NoCDR("Zap/1-1", "") in new stack
-- Executing [s@macro-hangupcall:3] GotoIf("Zap/1-1", "1?skiprg") in new stack
```

## Teste aditionale

### Apeluri sosite pe linia analogica.

1. Pentru desfasurarea acestui test se configureaza o "coada" de asteptare din aplicatia freePbx.
2. Se selecteaza din meniul optiunea "Queues" si se creeaza o "coada" de asteptare noua. In exemplul nostru aceasta se cheama "300:reception".
3. Se introduc parametrii pentru "queue number", "queue name" si "static agents". Pentru static agents (extensiile care vor fi mereu in "linia" de asteptare) se introduc extensiile corespunzatoare modulelor FXS si FXO, in cazul nostru acestea fiind "100" si "401".
4. In portul 2, port aferent modulului FXO, se cupleaza o linie analogica si se efectueaza un apel, de pe un telefon care nu este asociat sistemului de test, catre numarul aferent acesteia. Conform dial plan-ului stabilit, atat telefonul asociat extensiei SIP (extensia 100) definite cat si telefonul analogic (zap 401), trebuie sa sune. Telefonul analogic "suna" in momentul in care primeste un apel, prin intermediul modulului FXS care initiaza si trimite impulsurile electrice necesare apelului, pe linia analogica. Mai jos se poate vedea log-ul afisat in consola asterisk in momentul efectuarii apelului:

```
tenora*CLI>
-- Starting simple switch on 'Zap/2-1'
-- Executing [s@from-pstn:1] Set("Zap/2-1", "__FROM_DID=s") in new stack
-- Executing [s@from-pstn:2] Gosub("Zap/2-1", "app-blacklist-check|s|1") in new sta
-- Executing [s@app-blacklist-check:1] LookupBlacklist("Zap/2-1", "") in new stack
-- Executing [s@app-blacklist-check:2] GotoIf("Zap/2-1", "0?blacklisted") in new st
```

```
-- Executing [s@app-blacklist-check:3] Return("Zap/2-1", "") in new stack
-- Executing [s@from-pstn:3] GotoIf("Zap/2-1", "1 ?cidok") in new stack
```

