Asterisk on RPI

Starting with Debian 3.2.27+

RPI-Debian # uname -a

Linux RPI-Debian 3.2.27+ #1 PREEMPT Sat Sep 22 06:10:01 EDT 2012 armv6l GNU/Linux

And after making a dd copy on 10/21/2012, (3767 seconds) lets add on asterisk

```
apt-get -y install make gcc g++ libxml2 libxml2-dev ssh libncurses5
libncursesw5 libncurses5-dev libncursesw5-dev linux-libc-dev sqlite libnewt-
dev libusb-dev zlib1g-dev libmysqlclient15-dev libsqlite0 libsqlite0-dev
bison openssl libssl-dev libeditline0 libeditline-dev libedit-dev mc sox
libedit2 libedit-dev curl libcurl4-gnutls-dev apache2 libapache2-mod-php5
php-pear openssh-server build-essential openssh-client zlib1g zlib1g-dev
libtiff4 libtiff4-dev libnet-telnet-perl mime-construct libipc-signal-perl
libmime-types-perl libproc-waitstat-perl mpg123 libiksemel-dev
```

Update Asterisk Install Pre-requisites

To start with, update the system and install the required component's by copying the command below:

apt-get update

apt-get -y install make gcc g++ libxml2 libxml2-dev ssh libncurses5 libncursesw5 libncurses5-dev libncursesw5-dev linux-libc-dev sqlite libnewt-dev libusb-dev zlib1g-dev libmysqlclient15-dev libsqlite0 libsqlite0-dev bison openssl libssl-dev libeditline0 libeditline-dev libedit-dev mc sox libedit2 libedit-dev curl libcurl4-gnutls-dev apache2 libapache2-mod-php5 php-pear openssh-server build-essential openssh-client zlib1g zlib1g-dev libtiff4 libtiff4-dev libnet-telnet-perl mime-construct libipc-signal-perl libmime-types-perl libproc-waitstat-perl mpg123 libiksemel-dev php5 php5-cli mysql-server php5-mysql php-db libapache2-mod-php5 php5-gd php5-curl mysql-client vim

apt-get -y install asterisk-core-sounds-en   asterisk-core-sounds-en-g722 asterisk-core-sounds-en-gsm   asterisk-core-sounds-en-wav

Dowload and Extract Asterisk

To Download and Extract asterisk, enter the commands as below:

cd /usr/src

wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-current.tar.gz

tar   xvfz   asterisk-1.8-current.tar.gz

cd asterisk-1.8*

Set Up Correct Compilation For ARM Platform
Using your favorite editor, make the change below.

vi makeopts.in

Search for the word "proc=" in the file and change this to read "proc=arm".

I was not required to add the PROC=arm line to makeopts.in as reported in other forums, the PROC setting seems to have been removed in Asterisk recently.

Compile and Install Asterisk
Using the following commands: (In make menuselect, add channel mgcp)

./configure
make menuselect
make
make install
make samples
make config

Finally, restart the PI by typing the following command:

reboot

Congratulations - Asterisk installed is now installed

To test the install, log in to the PI and type the following command:

asterisk -crvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

Change interface ETH0 to 192.168.5.1/24, change metric on eth0 and wlan0

vi /etc/network/interfaces

```
# The primary network interface
# auto eth0          If using ifplugd, don't use auto here
iface eth0 inet static
#metric 10
metric 200
address 192.168.5.1
netmask 255.255.255.0
network 192.168.5.0
broadcast 192.168.5.255
dns-nameservers 192.168.1.1

#address 192.168.0.89
#gateway 192.168.0.1
#netmask 255.255.255.0
#network 192.168.0.0
#broadcast 192.168.0.255
#dns-nameservers 192.168.1.1

# The wireless interface
auto wlan0
iface wlan0 inet dhcp
metric 10
#metric 200
dns-nameservers 192.168.1.1
```

/etc/init.d/networking restart

Install and enable DHCP

```
apt-get update
apt-get install isc-dhcp-server
```

```
vi /etc/default/isc-dhcp-server
```

Edit the following line

INTERFACES=”ETH0”

vi /etc/dhcp/dhcpd.conf

option domain-name "rpi.jj3601.com";
option domain-name-servers 192.168.1.1, 4.4.4.1;

default-lease-time 86000;
max-lease-time 86000;

subnet 192.168.5.0 netmask 255.255.255.0 {
range 192.168.5.100 192.168.5.199;
option routers 192.168.5.1;
option subnet-mask 255.255.255.0;

option broadcast-address 192.168.5.255;
#option domain-name-servers 192.168.1.1;

option ntp-servers 192.168.0.20;
#option netbios-name-servers 10.0.0.1;
#option netbios-node-type 8;
}

service isc-dhcp-server restart

Set up Linksys PAP-2T

PAP-2T runs on 5V DC so I made up a USB adapter cable to provide power. Plug a phone into the analog Phone 1 jack and (touch-tone only) enter **** to enter the configuration mode.

Reset the PAP-2T

Press 73738# followed by 1 to confirm.

Set up the PAP-2T for DHCP

Enable DHCP by pressing 101# (Enter 1 to Enable and 0 to Disable DHCP)
Press 1 # to confirm

Verify IP address on PAP-2T

Enter 110#

Verify PAP-2T gets a DHCP address

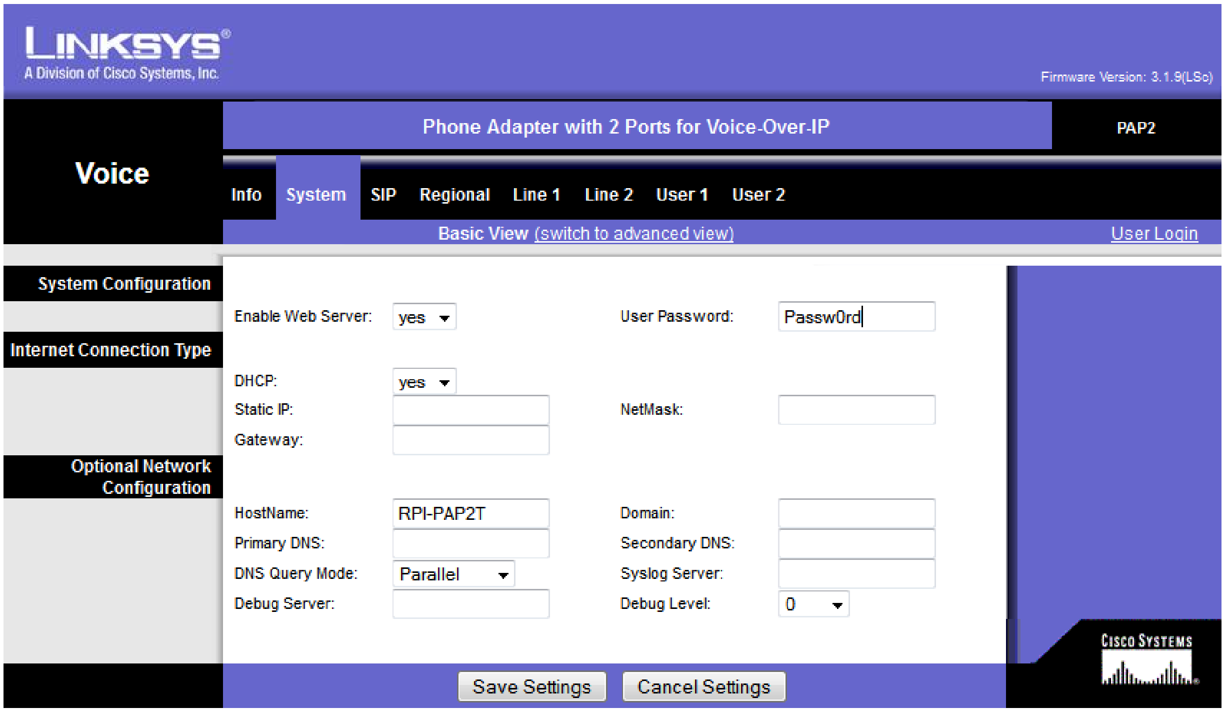
Plug the PAP-2T into the RPI   (a straight cable works fine)

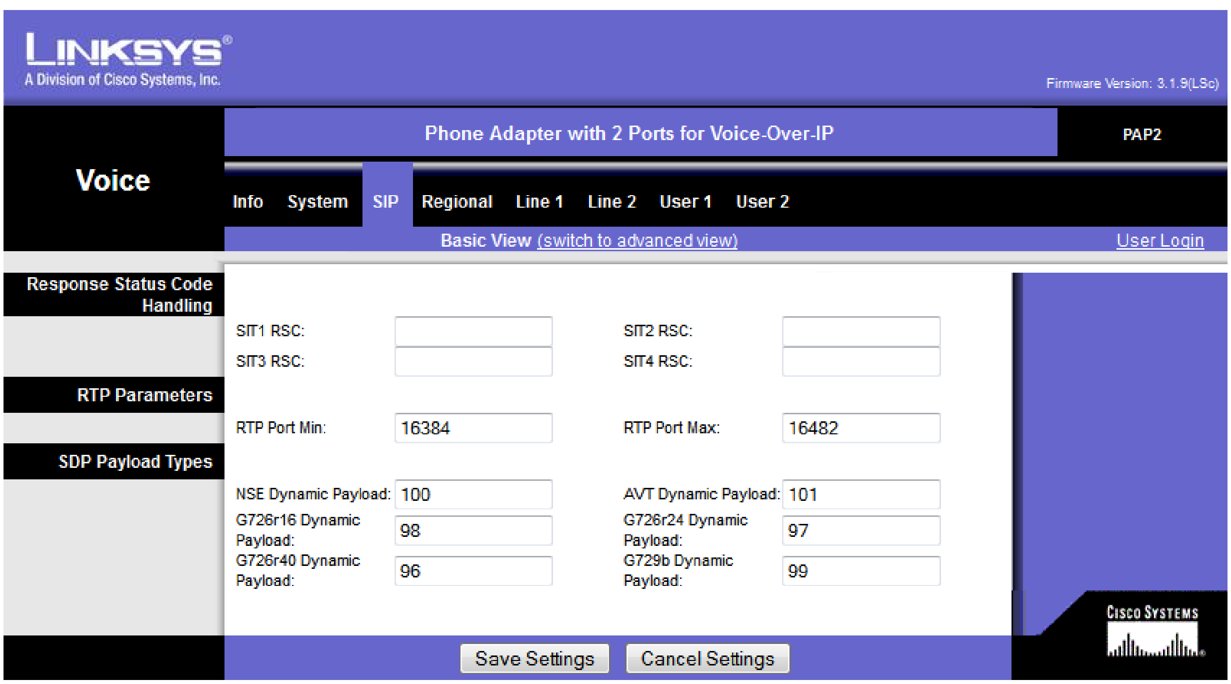
Verify it got an address by executing the arp command on the RPI.

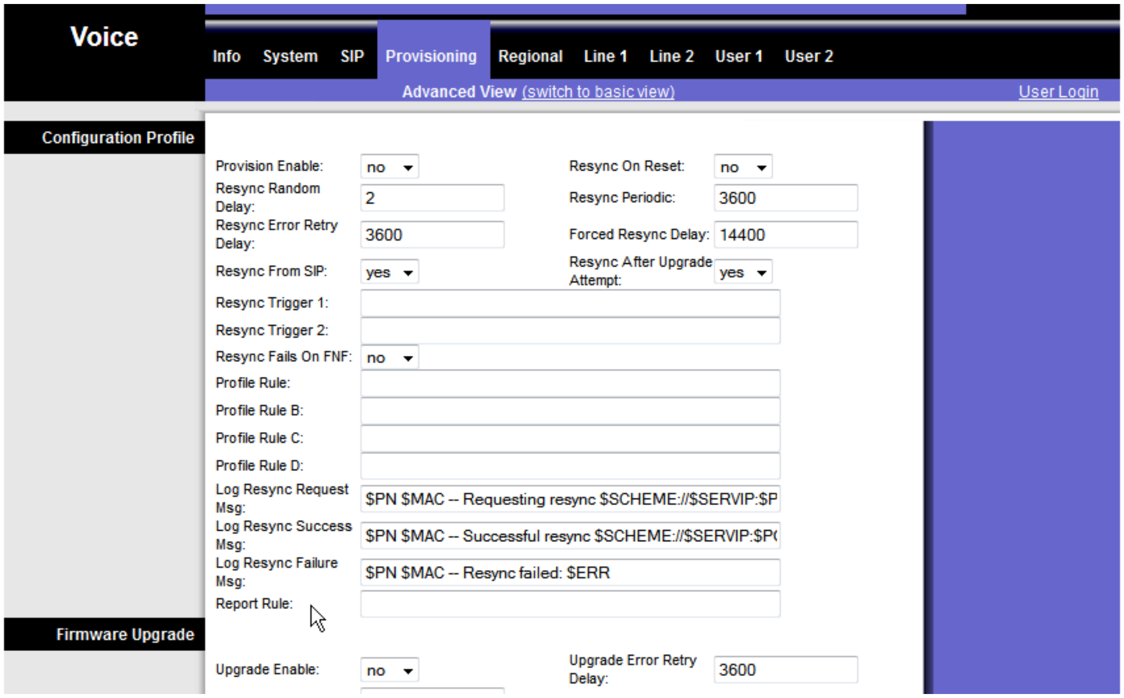
Settings- Cisco/Linksys PAP2t

The Cisco/Linksys PAP2T is a very popular 2 line Internet Phone Adapter or ATA device which can be connected up to your router. An analog phone can be connected to each of the two phone ports and if enabled with your VoISP the Cisco/Linksys PAP2T will support both lines. Each phone port operates independently with separate phone service and their own phone numbers, line 1 and line 2. The Cisco/Linksys ATA can be accessed from connecting a PC and typing the IP address of the ATA into a browser. To determine the current IP address connect a phone into phone port 1 while the ATA is powered up and connected to your router's LAN port and dial 4 stars **** then 110#. The IVR will say the IP address which you will need to type in the address of your browser.

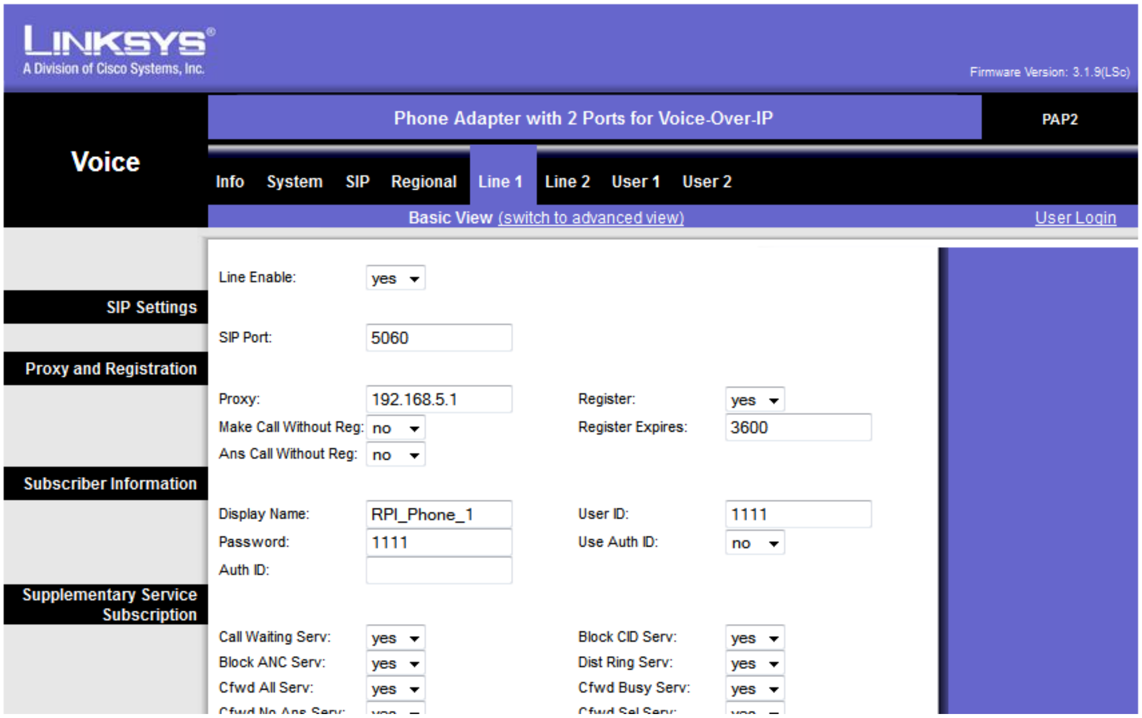
Click on the "Admin Login" button near the top right side of the screen, then click on the "Line 1" tab. The following instructions are typical instructions for line 1. In most cases you will need only modify a few parameters from the normal factory default settings.

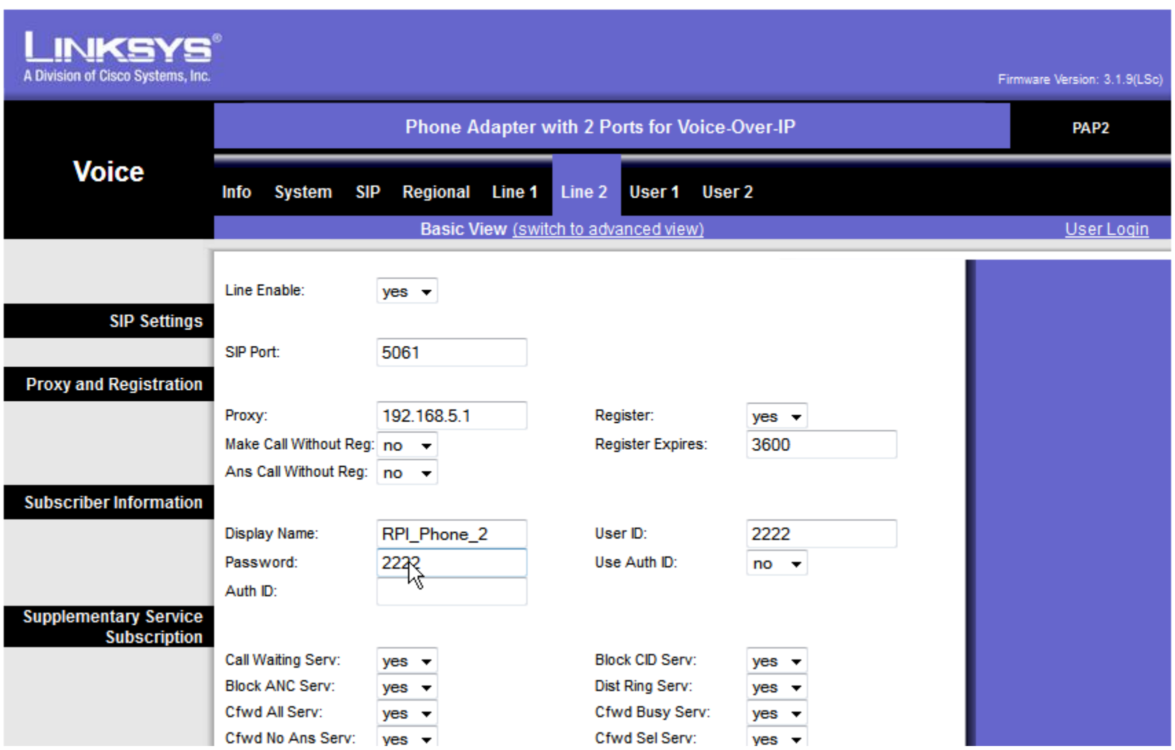

LINKSYS
A Division of Cisco Systems, Inc.
Firmware Version: 3.1.9(LSc)
Phone Adapter with 2 Ports for Voice-Over-IP
PAP2
Voice
Info System SIP Regional Line 1 Line 2 User 1 User 2
Basic View (switch to advanced view)
User Login
System Configuration
Enable Web Server: yes
User Password: Passw0rd
Internet Connection Type
DHCP: yes
Static IP:
NetMask:
Gateway:
Optional Network Configuration
HostName: RPI-PAP2T
Domain:
Primary DNS:
Secondary DNS:
DNS Query Mode: Parallel
Syslog Server:
Debug Server:
Debug Level: 0
Save Settings
Cancel Settings
Cisco Systems


LINKSYS
A Division of Cisco Systems, Inc.
Firmware Version: 3.1.9(LSc)
Phone Adapter with 2 Ports for Voice-Over-IP
PAP2
Voice
Info System SIP Regional Line 1 Line 2 User 1 User 2
Basic View (switch to advanced view)
User Login
Response Status Code Handling
SIT1 RSC:
SIT2 RSC:
SIT3 RSC:
SIT4 RSC:
RTP Parameters
RTP Port Min: 16384
RTP Port Max: 16482
SDP Payload Types
NSE Dynamic Payload: 100
AVT Dynamic Payload: 101
G726r16 Dynamic Payload: 98
G726r24 Dynamic Payload: 97
G726r40 Dynamic Payload: 96
G729b Dynamic Payload: 99
Save Settings
Cancel Settings
Cisco Systems

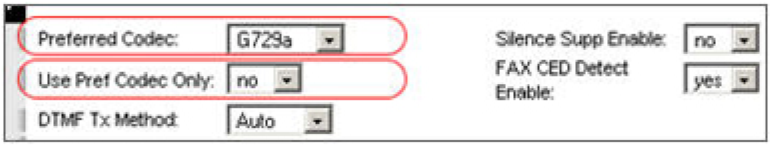


Regional Settings (Advanced View)


LINKSYS
A Division of Cisco Systems, Inc.
Firmware Version: 3.1.9(LSc)
Voice
Phone Adapter with 2 Ports for Voice-Over-IP
PAP2
Info System SIP Regional Line 1 Line 2 User 1 User 2
Basic View (switch to advanced view)
User Login
SIP Settings
Proxy and Registration
Subscriber Information
Supplementary Service Subscription
Line Enable: yes
SIP Port: 5060
Proxy: 192.168.5.1
Register: yes
Make Call Without Reg: no
Register Expires: 3600
Ans Call Without Reg: no
Display Name: RPI_Phone_1
User ID: 1111
Password: 1111
Use Auth ID: no
Auth ID:
Call Waiting Serv: yes
Block CID Serv: yes
Block ANC Serv: yes
Dist Ring Serv: yes
Cfwd All Serv: yes
Cfwd Busy Serv: yes


LINKSYS
A Division of Cisco Systems, Inc.
Firmware Version: 3.1.9(LSc)
Voice
Phone Adapter with 2 Ports for Voice-Over-IP
PAP2
Info System SIP Regional Line 1 Line 2 User 1 User 2
Basic View (switch to advanced view)
User Login
SIP Settings
Proxy and Registration
Subscriber Information
Supplementary Service Subscription
Line Enable: yes
SIP Port: 5061
Proxy: 192.168.5.1
Register: yes
Make Call Without Reg: no
Register Expires: 3600
Ans Call Without Reg: no
Display Name: RPI_Phone_2
User ID: 2222
Password: 2222
Use Auth ID: no
Auth ID:
Call Waiting Serv: yes
Block CID Serv: yes
Block ANC Serv: yes
Dist Ring Serv: yes
Cfwd All Serv: yes
Cfwd Busy Serv: yes
Cfwd No Ans Serv: yes
Cfwd Sel Serv: yes

The Preferred codec drop down box has several choices. Typically G711, which is an un-compressed codec would be the choice, but to conserve on bandwidth you might change Line 1 "Preferred Codec" to G729a. The "Use Pref Codec Only" setting to No would allow the call to negotiate the codec.



Note: The Cisco/Linksys PAP2 or PAP2T does not support usage of simultaneous calls using the G.729 codec. When one line port uses G.729, the other line port will use G.711. (Using compression like G729 does is processor intensive and both these ATA models do not have a capacity to handle simultaneous G729 calls.)

Asterisk Configuration

First, save off the original configuration files.

cd /etc/asterisk
cp iax.conf iax.conf.orig
cp sip.conf sip.conf.orig
cp extensions.conf extensions.conf.orig

Add the following line to /etc/asterisk/sip.conf

#include "/etc/asterisk/rpi_sip.conf"

Create /etc/asterisk/rpi_sip.conf with the following content.

[1111]
secret=1111
type=friend

```
deny=0.0.0.0/0.0.0.0
permit=192.168.5.0/255.255.255.0

host=dynamic
port=5060
qualify=yes
canreinvite=no
dtmfmode=inband
nat=no
context=internal
allow = all
mailbox = 1111

[2222]
secret=2222
type=friend

deny=0.0.0.0/0.0.0.0
permit=192.168.5.0/255.255.255.0

host=dynamic
port=5061
qualify=yes
canreinvite=no
dtmfmode=inband
nat=no
context=internal
allow = all
mailbox = 2222
```

From the Asterisk console, enter sip reload and sip show peers

You should see output similar to that listed below

```
RPI-Debian*CLI> sip reload
 Reloading SIP
  == Parsing '/etc/asterisk/sip.conf':   == Found
  == Parsing '/etc/asterisk/rpi_sip.conf':   == Found
  == Parsing '/etc/asterisk/users.conf':   == Found
  == Using SIP CoS mark 4
  == Parsing '/etc/asterisk/sip_notify.conf':   == Found
RPI-Debian*CLI> sip show peers
Name/username          Host                         Dyn Forcerport ACL Port     Status
1111/1111              192.168.5.100                 D            A  5060    OK (12 ms)
2222/2222              192.168.5.100                 D            A  5061    OK (12 ms)
2 sip peers [Monitored: 2 online, 0 offline Unmonitored: 0 online, 0 offline]
RPI-Debian*CLI>
```

On the PAP-2T, you should get dial tone on both ports.

Set up extensions.conf Insert this just before ;[context]

```
[internal]
exten => 1111,1,Answer()
exten => 1111,n,Set(CALLERID(num)=1111)
exten => 1111,n,Set(CALLERID(name)=Raspberry)
exten => 1111,n,Dial(SIP/1111,20,tr)
exten => 1111,n,VoiceMail(1111,u)
exten => 1111,n,Hangup

exten => 2222,1,Answer()
exten => 2222,n,Set(CALLERID(num)=1111)
exten => 2222,n,Set(CALLERID(name)=Raspberry)
exten => 2222,n,Dial(SIP/2222,20,tr)
exten => 2222,n,VoiceMail(2222,u)
exten => 2222,n,Hangup
exten => 8888,1,VoiceMailMain() ; to retrieve voicemail
```

Set up voicemail.conf   Insert this just after   [default]

; Added by JJ

```
;extension number  => password, User Name, email to send wav file to
1111 => 12345,Phone1,user@user.com
2222 => 12345,Phone2,user@user.com
```

Reference

Linksys PAP2T ATA Adapter Reset Procedure:

Sometimes it will be very helpful to reset your linksys ATA adapter to factory default settings. If you are using a used ATA adapter, then resetting your adapter to factory default settings is highly recommended. The procedure to reset your linksys adapter is as follows.

Step1: Connect a telephone connected to line1 of the ATA unit and power up your ATA unit using its power adapter.

Step2: Disconnect your PAP2T adapter from the internet connection or in other words just unplug the ethernet cable from the PAP2T hardware unit. Resetting with internet connecttion may mess up the unit making it completely useless.

Step3: Dial ****, and wait for the Interactive Voice Menu (IVM) to get activated. After hearing this message, type in the following number with the # symbol.

73738# This number spells RESET.

Step4: After this, you will be asked to confirm this by pressing 1.

Your linksys ATA unit will now go back to it factory default settings.

Dial 73738#
Password: 7756112# or 8995523#
enter 1 to confirm press # Hang up

Linksys PAP2T IP Configuration Procedure:

During the time where there is no ethernet cable connected to the PAP2T the 1 long 2 short flashing green light is normal because that means that there is no ethernet detected -- reset the PAP2T one more time while the ethernet is disconnected and phone configuration is accessible.

then try to set a static IP, network mask and default gateway on the PAP2T

-- disable DHCP by pressing **** 101 (Enter 1 to Enable and 0 to Disable DHCP)

-- after you have succesfully disabled DHCP press **** 111

-- Enter IP Address using numbers on the telphone keypad. Use the * (star) key when entering a decimal point

-- set a Network Mask by pressing **** 121

-- enter value using keypad

-- set Default Gateway by **** 131

-- enter value using keypad

-- set Primary DNS by **** 161

of course the IP settings you will enter on the PAP2T needs to be in the same range as that of your existing network

```
mount -t ntfs-3g -o rw,users /dev/sda1 /mnt/
```

Additional Steps

NOTE: It is necessary with ZoneEdit to create a new host at www.zoneedit.com before an update can be applied. Zoneedit will NOT create a new A record from an update. Go there and create a new record for raspberry.cnetvoip.com, assign IP address 1.1.1.1 and Save and Publish.

Add Dynamic DNS

apt-get update
apt-get install ddclient

A menu is auto-launched with choices for DDNS providers. I use Zoneedit.

Provide zoneedit username
Provide zoneedit password
Provide Fully Qualified Domain Name raspberry.cnetvoip.com

Edit /etc/rc.local and add the following line to auto start ddclient and run it every 20 minutes
/usr/sbin/ddclient -daemon 1200 –syslog

Here is the working /etc/ddclient.conf

```
# added by JJ
#ssl=yes
protocol=zoneedit1
#use=if, if=wlano   #comment out this line to not use the private IP address
use=web, web=myip.dnsomatic.com

login=jjones74, password='badpassword74'
raspberry.cnetvoip.com
```

To debug

```
ddclient -daemon=0 -debug -verbose -noquiet
```

To force an update

rm /var/cache/ddclient/ddclient.cache

ddclient –daemon=0

Here is a config using dnsomatic Note: you need to set up an account at dnsomatic.com with Zoneedit.com credentials and an entry for raspberry.cnetvoip.com

```
# Configuration file for ddclient generated by debconf
#
# /etc/ddclient.conf
ssl=yes
protocol=dyndns2
# added by JJ
#use=if, if=wlano   #comment out this line to not use the private IP address
use=web, web=myip.dnsomatic.com
server=updates.dnsomatic.com, login=jjones74, password='badpassword74'
all.dnsomatic.com
#raspberry.cnetvoip.com
```

ADD C*Net Macros to Extensions.conf

```
;  Insert in [globals]

;
;Added by JJ
CNETANI=16877710                    ;
OFFICECODE=687
MYNAME=John Jones;                  for CallerID.


; The following is added for CNET

; This is the extensions.conf file.  It should be placed in the /etc/asterisk directory.
; THIS CONFIGURATION PROVIDES SIP AND IAX2 CONNECTIVITY TO LEGACY
; OR COLLECTIBLE TELEPHONE SYSTEMS.
;
; THIS EXAMPLE FILE ASSUMES A CERTAIN CONFIGURATION, BUT CAN EASILY BE
; MODIFIED TO PROVIDE FOR DIFFERENCES WITH INDIVIDUAL PHONE SWITCHES.
;
; THIS SAMPLE SYSTEM IS CONFIGURED AS FOLLOWS:
;      + THE CNET NETWORK HAS NO AREA CODE.
;      + THIS TANDEM HAS THE ASSIGNED OFFICE CODE OF 687.
;      + 2-DIGIT DIALING TOWARDS ELECTROMECHANICAL PHONE SYSTEM
;     + 7-DIGIT DIALING RECEIVED FROM ANALOG SWITCH FOR CNET
;       CALL DESTINATIONS IN NORTH AMERICA.
;     + 011 +  COUNTRY CODE + CITY CODE + NUMBER RECEIVED
;       FROM ANALOG SWITCH FOR INTERNATIONAL CNET DESTINATIONS
;
; ======================================================================
;

; Macros can be called from other contexts.  What they do is perform a certain
; function, and then return to the originating context.
;
[macro-dialswitch]
;
;
[macro-dialcnet]                    ;version bjc1.01 - asterisk 1.4 and asterisk 1.2

exten => s,1,Set(NUMBER=${ARG1})              ;Store number to be called
exten => s,2,GotoIf($[ ${ARG1:0:1} = "+"]?search)     ;Is number prefixed with '+'?
exten => s,3,Set(ARG1=+${ARG1})       ;Prefix number with '+',required to properly retrieve US ENUM
entries!!!
```

```
exten => s,4(search),Set(ENUM=${ENUMLOOKUP(${ARG1},ALL,,1,std.ckts.info)})    ;Search ENUM
database
exten => s,5,GotoIf($[${LEN(${ENUM})}=0]?no_uri)       ;Is ENUM record found?
exten => s,6,Gotoif(${DB_EXISTS(ELBD/${NUMBER})}?cdata) ;Does backup entry exist?
exten => s,7,Set(DB(ELBD/${NUMBER})=${ENUM})           ;No existing backup so store backup ENUM
data
exten => s,8(test),GotoIf($[${ENUM:0:3} = iax ]?iaxuri) ;Yes-IAX2 protocol
exten => s,9,GotoIf($[${ENUM:0:3} = sip ]?sipuri)      ;Yes-SIP protocol
exten => s,10,GotoIf($[${ENUM:0:3} = h32 ]?h323uri)     ;Yes-H323 protocol
exten => s,11(no_uri),Gotoif(${DB_EXISTS(ELBD/${NUMBER})}?seek)  ;ENUM="", is there a backup
entry?
exten => s,12,Macro(invalid-office-code,${NUMBER})             ;No valid ENUM and no backup entry
exten => s,13,Wait(5)                                ;Pause
exten => s,14,Hangup                                 ;Done - failed to make call - Goodbye
exten => s,15(seek),Set(ENUM=${DB_RESULT})             ;Set ENUM to backup base entry
exten => s,16,GotoIf(${REGEX("iax2|sip|h323"${ENUM})}?test)     ;If it's a single backup proceed to dial
out
exten => s,17,Set(NE=${CUT(ENUM,":",1)})              ;Get backup entry field 1
exten => s,18,Set(LU=${CUT(ENUM,":",2)})              ;Get backup entry field 2
exten => s,19,GotoIf($[${NE}>${LU}]?grec)             ;Was last used the last entry in list?
exten => s,20(frec),Set(LU=0)                         ;Reset entry pointer
exten => s,21(grec),Set(LU=$[${LU}+1])                ;Increment last used pointer
exten => s,22,Set(ENUM=${DB(ELBD/${NUMBER}/${LU})})     ;Read ENUM backup entry
exten => s,23,Set(DB(ELBD/${NUMBER})=${NE}:${LU})       ;Update last used record
exten => s,24,Goto(test)                             ;Proceed to dial out
exten => s,25(iaxuri),Set(DIALSTR=IAX2/${ENUM:5})       ;IAX2
exten => s,26,Goto(dodial)                           ;Make call
exten => s,27(sipuri),Set(DIALSTR=SIP/${ENUM:4})        ;SIP
exten => s,28,Goto(dodial)                           ;Make call
exten => s,29(h323uri),Set(DIALSTR=H323/${ENUM:5})      ;H323
exten => s,30,Macro(invalid,${NUMBER})                 ;Make Call
exten => s,31(dodial),NoOp(Outbound Caller ID is ${CALLERID(all)})
exten => s,32,Dial(${DIALSTR})                        ;Dial Out
exten => s,33,Hangup                                 ;Done -call attempted - Goodbye
exten => s,34(cdata),Gotoif($["${DB_RESULT}"="${ENUM}"]?test)    ;Does entry have a single backup?
exten => s,35,Set(SR=${DB_RESULT})                            ;Backup does not match current
exten => s,36,GotoIf(${REGEX("iax2|sip|h323"${SR})}?shuffle)    ;Is more than one backup entry stored?
exten => s,37,Set(NE=${CUT(SR,":",1)})                ;Get backup entry field 1
exten => s,38,Set(LU=${CUT(SR,":",2)})                ;Get backup entry field 2
exten => s,39,Gosub(rent)                            ;Check multiple backup entries
exten => s,40,Set(DB(ELBD/${NUMBER})=${NE}:${MR})       ;Store no. of entries and match as entry last
used
exten => s,41,GotoIf($[${MR}>0]?test)                 ;If backup exists proceed to dial out
exten => s,42,Set(NE=$[${NE}+1])                      ;Increment entries pointer
exten => s,43,Set(DB(ELBD/${NUMBER})=${NE}:${NE})       ;Update entry count and usage
exten => s,44,Set(DB(ELBD/${NUMBER}/${NE})=${ENUM})     ;Store nth backup entry
exten => s,45,Goto(test)                             ;Nth entry stored - proceed to dial out
exten => s,46(shuffle),Set(DB(ELBD/${NUMBER}/1)=${SR})   ;Move stored entry to backup entry /1
```

```
exten => s,47,Set(DB(ELBD/${NUMBER}/2)=${ENUM})         ;Create second backup entry /2
exten => s,48,Set(DB(ELBD/${NUMBER})=2:2)              ;Store no. of backup entries & last called
exten => s,49,Goto(test)                              ;Alternative backup entry stored - proceed to dial out
exten => s,50(rent),Set(i=0)                          ;Load test counter
exten => s,51,Set(MR=0)                               ;Load match store
exten => s,52,While($[${i}<${NE}])                    ;Check multiple entries for a NUMBER looking for a
match
exten => s,53,GotoIf($["${DB(ELBD/${NUMBER}/$[${i}+1])}"="${ENUM}"]?mark)       ;Mark entry
matching ENUM result
exten => s,54,Set(i=$[${i}+1])                        ;Increment counter
exten => s,55,EndWhile                                ;Look at next entry if one exists
exten => s,56,Return                                  ;Done searching for ENUM result backup record
exten => s,57(mark),Set(MR=$[${i}+1])                 ;ENUM result already backed up
exten => s,58,Set(i=${NE})                            ;Load counter with no. of last entry
exten => s,59,ContinueWhile                           ;Stop searching

[macro-invalid]
exten => s,1,Answer
exten => s,2,Wait(1)
exten => s,3,Playtones(480*20/2000,0/4000)
exten => s,4,Wait(8)
exten => s,5,Playtones(!950/330,!1400/330,!1800/330,0)
exten => s,6,Wait(2)
exten => s,7,Playback(discon-or-out-of-service)
exten => s,8,Wait(1)
exten => s,9,SayDigits($OfficeCode)
exten => s,10,Wait(5)
exten => s,11,Hangup
;
[macro-Milliwatt]
exten => s,1,Answer
exten => s,2,Playtones(1004)
exten => s,3,Wait(${ARG1})
exten => s,4,Hangup
;

[cnet-out]

exten => _1NXXXXXX,1,Set(CALLERID(name)='John Jones') ; make sure you dont use any space character

exten => _X.,n,NoOp(${EXTEN})
exten => _X.,n,NoOP(${CALLERID(num)})

exten => _1NXXXXXX,n,Set(CALLERID(number)=16877710) ; make sure you dont use any space
character;
exten => _1NXXXXXX,n,Macro(dialcnet,${EXTEN})
exten => _1NXXXXXX,n,Congestion
 ;
```

```
exten => _011.,1,Set(CALLERID(name)='John Jones') ; make sure you dont use any space character
exten => _011.,n,Set(CALLERID(number)=16877710) ; make sure you dont use any space character;
exten => _011.,n,Macro(dialcnet,${EXTEN:3})
exten => _011.,n,Congestion
 ;

[cnet-callerid-in]

;***************   Shane's CLID lookup logic ******************************
;   CURL doesn't exist is OpenWRT version of Asterisk
exten =>
_X.,1,Set(cnetuser=${CURL(https://www2.shaneyoung.com/cquery/?ip=${IAXPEER(CURRENTCHANNEL)}
)})


exten => _256XXXXXXX,n,Set(CALLERID(number)=1${CALLERID(number:4:10)})   ;  For Peter Voetsch
exten => _256XXXXXXX,n,NoOp(${CALLERID(number)}) ; log callerID string
exten => _256XXXXXXX,n,Goto(cnet-in,${EXTEN},1)

exten => _1256XXXXXXX,n,Set(CALLERID(number)=1${CALLERID(number:5:11)})  ;  For Peter Voetsch
exten => _1256XXXXXXX,n,NoOp(${CALLERID(number)}) ; log callerID string
exten => _1256XXXXXXX,n,Goto(cnet-in,${EXTEN},1)



;****************   START of CallerId logic   ****************************


exten => _X.,n,GotoIf($["${CALLERID(number)}" != ""]?checkit:done)
exten => _X.,n(checkit),GotoIf($[${LEN(${CALLERID(number)})} = 8] & $["${CALLERID(number):0:1}" =
"1"]?done)
exten => _X.,n,GotoIf($[${LEN(${CALLERID(number)})} = 7]?sevendigits:not7digits)
exten => _X.,n(not7digits),GotoIf($[${LEN(${CALLERID(number)})}> 9]?done)
exten => _X.,n,GotoIf($[${LEN(${CALLERID(number)})}< 8]?done)
exten => _X.,n,Set(CALLERID(number)=011${CALLERID(number)})   ; append 011 to all other numbers
Most International num
exten => _X.,n,NoOp(${CALLERID(number)}) ; log callerID string
exten => _X.,n,Goto(cnet-in,${EXTEN},1)
exten => _X.,n(sevendigits),Set(CALLERID(number)=1${CALLERID(number)})    ; if it is 7 digits, append a 1
to seven dig
exten => _X.,n(done),NoOp(${CALLERID(number)}) ; log callerID string
exten => _X.,n,Goto(cnet-in,${EXTEN},1)

;***************   END of CallerId logic     ****************************
```

```
[cnet-in]

exten => _16877710,1,Goto(internal,1111,1)

;   Add at end of   [internal]

;      Special Extensions

exten => 579,1,Answer
exten => 579,n,Playback(vm-extension)
exten => 579,n,SayDigits(${CALLERID(num)})
exten => 579,n,Wait(2)
exten => 579,n,SayDigits(${CALLERID(num)})
exten => 579,n,Wait(1)
exten => 579,n,Playback(vm-goodbye)
exten => 579,n,Wait(1)
exten => 579,n,Hangup

exten => 1212,1,Answer
exten => 1212,n,SayUnixTime( , EST5EDT , I)
exten => 1212,n,SayUnixTime( | | M)
exten => 1212,n,SayUnixTime( | | P)
exten => 1212,n,SayUnixTime( , EST5EDT , IMp)
exten => 1212,n,Wait(1)
exten => 1212,n,Hangup

;
; add to enable calls to cnet

include => cnet-out

; Invalid Extension
; Invalid incoming calls should have been caught before
; this point, but, just incase they weren't...
```

```
;
exten => i,1,Macro(invalid)
;
; Hangup
exten => h,1,Hangup
;
; Timeout
exten => t,1,congestion
```

log in to the asterisk console and execute

dialplan reload

Adding BitWizard LCD display for basic Asterisk & system info

Edited the i2clcd94.c program so it would accept strings directly (no –t or –T required) and so that a whole line of text can be piped in. This is listed below.

RPI-Breadboard # cat /rpi/gpio/rpitools/bw_rpi_tools/bw_i2c_lcd/i2clcd94.c

```
/*
 * bw_i2c_lcd.c.
 *
 * Control the BitWizard I2C-LCD expansion boards.
 *
 * based on:
 * SPI testing utility (using spidev driver)
 *
 * Copyright (c) 2007  MontaVista Software, Inc.
 * Copyright (c) 2007  Anton Vorontsov <avorontsov@ru.mvista.com>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; version 2 of the License.
 *
 * Cross-compile with cross-gcc -I/path/to/cross-kernel/include
 *
 *
 *
 * Compile on raspberry pi with
 *
 * gcc -Wall -O2 i2clcd94.c -o  /usr/sbin/lcd
 *
 */


#include <stdint.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <getopt.h>
#include <fcntl.h>
#include <string.h>
#include <sys/ioctl.h>
#include <linux/types.h>
#include <linux/spi/spidev.h>
#include <linux/i2c-dev.h>

#define ARRAY_SIZE(a) (sizeof(a) / sizeof((a)[0]))


static const char *device = "/dev/i2c-1";
// static uint8_t mode;
// static uint8_t bits = 8;
static uint32_t speed = 500000;
static uint16_t delay = 15;
static int cls = 0;
static int reg = -1;
static int val = -1;
static int x = -1, y = -1, addr=0x94;
static int backlight = -1;
static int contrast = -1;
```

```
static int text = 0;


static void pabort(const char *s)
{
  perror(s);
  abort();
}

void transfer (int fd, int len, char *buf)
{
   write (fd, buf+1, len-1);
}


static void send_text (int fd, char *str)
{
  char *buf;
  int l;
  l = strlen (str);
  buf = malloc (l + 5);
  buf[0] = addr;
  buf[1] = 0;
  strcpy (buf+2, str);
  transfer (fd, l+2, buf);
  free (buf);
}


static void set_reg_value8 (int fd, int reg, int val)
{
  char buf[5];
  buf[0] = addr;
  buf[1] = reg;
  buf[2] = val;
  transfer (fd, 3, buf);
}



static void print_usage(const char *prog)
{
  printf("Usage: %s [-DsbdlHOLC3]\n", prog);
  puts("  -D --device   device to use (default /dev/spidev1.1)\n"
       "  -s --speed    max speed (Hz)\n"
       "  -d --delay    delay (usec)\n"
       "  -b --bpw      bits per word \n"
       "  -l --loop     loopback\n"
       "  -H --cpha     clock phase\n"
       "  -O --cpol     clock polarity\n"
       "  -L --lsb      least significant bit first\n"
       "  -C --cs-high  chip select active high\n"
       "  -3 --3wire    SI/SO signals shared\n");
  exit(1);
}

static const struct option lopts[] = {

  // SPI options.
  { "device",  1, 0, 'D' },
  { "speed",   1, 0, 's' },
  { "delay",   1, 0, 'd' },
```

```
  // text display options.
  { "reg",       1, 0, 'r' },
  { "val",       1, 0, 'v' },
  { "pos",       1, 0, 'p' },
  { "addr",      1, 0, 'a' },
  { "text",      1, 0, 't' },
  { "ptext",     1, 0, 'T' },
  { "backlight", 1, 0, 'b' },
  { "contrast",  1, 0, 'c' },
  { "file",      1, 0, 'f' },

  { "cls",       0, 0, 'C' },

  { NULL, 0, 0, 0 },
};



static int parse_opts(int argc, char *argv[])
{
  while (1) {
    int c;

    c = getopt_long(argc, argv, "D:s:d:r:v:p:a:tT:b:c:f:C", lopts, NULL);

    if (c == -1)
      break;

    switch (c) {
    case 'D':
      device = strdup (optarg);
      break;
    case 's':
      speed = atoi(optarg);
      break;
    case 'd':
      delay = atoi(optarg);
      break;
    case 'r':
      reg = atoi(optarg);
      break;
    case 'v':
      val = atoi(optarg);
      break;
    case 'p':
      sscanf (optarg, "%d,%d", &x, &y);
      break;

    case 'a':
      sscanf (optarg, "%x", &addr);
      break;
    case 'T':
      sscanf (optarg, "%d,%d", &x, &y);
      // fallthrough
    case 't':
      text = 1;
      break;
    case 'b':
      backlight = atoi (optarg);
      break;
    case 'c':
```

```
      contrast = atoi (optarg);
      break;
    case 'C':
      cls = 1;
      break;
    default:
      text = 1;
      break;
    }
  }
  return optind;
}



int main(int argc, char *argv[])
{
  int ret = 0;
  int fd;
  int nonoptions;
  char buf[0x100];
  int i;

// printf("argc is %d argv[0] is ..%s..argv[1] is ..%s..\n\\r",argc,argv[0],argv[1]);

  nonoptions = parse_opts(argc, argv);

  fd = open(device, O_RDWR);
  if (fd < 0)
    pabort("can't open device");

  // .xx.
  addr = addr >> 1;
  if (ioctl(fd, I2C_SLAVE, addr) < 0)
    pabort ("cant set slave addr");

  if (cls) set_reg_value8 (fd, 0x10, 0xaa);

  if (contrast != -1) set_reg_value8 (fd, 0x12, contrast);

  if (backlight != -1) set_reg_value8 (fd, 0x13, backlight);

  if ((x != -1) && (y != -1)) {
    set_reg_value8 (fd, 0x11, (y << 5) | x);
  }

  if (reg != -1)
    set_reg_value8 (fd, reg, val);

  //printf ("text = %d, nonoptions = %d.\n", text, nonoptions);
  if (text) {
    buf [0] = 0;
    for (i=nonoptions; i < argc;i++) {
      if (i != nonoptions) strcat (buf, " ");
      strcat (buf, argv[i]);
    }
    send_text (fd, buf);
  }
  if ( argc == 2 && argv[1][0] != '-')
    {
      buf[0] =0;
      strcat (buf, argv[1]);
```

```
    send_text (fd, buf);
   }
  if (argc == 1 && strcmp(argv[0],"lcd") ==0 ) {      // when text is piped in
    char stringin[50];

    scanf("%[^\n]s",stringin);    // look for a whole line
    send_text (fd, stringin);
    }
  close(fd);

  return ret;
}
```

RPI-Breadboard #

Next, create /etc/asterisk/lcd.bat to send the desired information

RPI-Breadboard # cat lcd.bat

```
# script to send ifconfig output and Asterisk active call info to lcd
#
# Every 1 minutes refresh ifconfig.txt file

 while [ 1 -eq 1 ]
 do
   lcd -C
   rm /etc/asterisk/ifconfig.txt
   ifconfig | grep addr: | sed  's/inet addr://' |  awk '{ print $1 }'  >> /etc/asterisk/ifconfig.txt
   for c in 1 2 3
     do
       lcd -p 0,0
       IP=$(cat /etc/asterisk/ifconfig.txt | head -n 1 | tail -n 1)
       echo $IP | lcd
       lcd -p 0,1
       ACTIVE=$(asterisk -vvvvvrx 'core show channels' | tail -n 3 | head -n 1)
       echo $ACTIVE | lcd
       sleep 5
       lcd -p 0,0
       lcd "                                "
       lcd -p 0,1
       lcd "                                "

       lcd -p 0,0
       IP=$(cat /etc/asterisk/ifconfig.txt | head -n 2 | tail -n 1)
       echo $IP | lcd
       lcd -p 0,1
       ACTIVE=$(asterisk -vvvvvrx 'sip show peers' | tail -n 2 | head -n 1 | awk '{ print $1 , $2 , $3  }' )
       echo $ACTIVE | lcd
       sleep 5
       lcd -p 0,0
```

```
    lcd "                    "
    lcd -p 0,1
    lcd "                    "

    lcd -p 0,0
    IP=$(cat /etc/asterisk/ifconfig.txt | head -n 3 | tail -n 1)
    echo $IP | lcd
    lcd -p 0,1
    ACTIVE=$(asterisk -vvvvvrx 'core show channels' | tail -n 3 | head -n 1)
    echo $ACTIVE | lcd
    sleep 5
    lcd -p 0,0
    lcd "                    "
    lcd -p 0,1
    lcd "                    "

    lcd -p 0,0
    IP=$(cat /etc/asterisk/ifconfig.txt | head -n 4 | tail -n 1)
    echo $IP | lcd
    lcd -p 0,1
    ACTIVE=$(asterisk -vvvvvrx 'iax2 show peers' | tail -n 2 | head -n 1 | awk '{ print $1 , $2 , $3 }' )
    echo $ACTIVE | lcd
    sleep 5
    lcd -p 0,0
    lcd "                    "
    lcd -p 0,1
    lcd "                    "
  done
done
```

RPI-Breadboard #

Next, make /etc/asterisk/lcd.bat start automatically at reboot

```
vi /etc/rc.local
```

Add the python line so the file now looks like this:

```
 # rc.local
 #
 # This script is executed at the end of each multiuser runlevel.
 # Make sure that the script will "exit 0" on success or any other
 # value on error.
 #
 # In order to enable or disable this script just change the execution
 # bits.
 #
 # By default this script does nothing.
 # Print the IP address
 _IP=$(hostname -I) || true
 if [ "$_IP" ]; then
   printf "My local IP address is %s\n" "$_IP"
   python /usr/sbin/email_power_up.py
 fi

# Start the lcd.bat script to display status on BitWizard 1602 LCD
/etc/asterisk/lcd.bat &
printf "/etc/asterisk/lcd.bat started in /etc/rc.local\n"

exit 0
```