Setting up RPI as SIP Server And Client User Agent

By

Shally Verma

Contents

[Purpose 3](#_Toc442366888)

[Introduction 3](#_Toc442366889)

[Test Setup Overview 3](#_Toc442366890)

[Setting up sipwitch on Pi 4](#_Toc442366891)

[Configuring pre-installed sipwitch: 4](#_Toc442366892)

[Troubleshooting 4](#_Toc442366893)

[Setting up pjsua on Pi 5](#_Toc442366894)

[Building on Pi 5](#_Toc442366895)

[Few Checks 6](#_Toc442366896)

[Setting up microsip on 2nd window PC 7](#_Toc442366897)

[Client Registration 7](#_Toc442366898)

[Registering Pi to sipwitch server 7](#_Toc442366899)

[Registering linphone on window PC 1 8](#_Toc442366900)

[Registering Minisip to sipwitch server 11](#_Toc442366901)

[Checking client registration to Sipwitch 11](#_Toc442366902)

[Making calls 11](#_Toc442366903)

[Cross Compiling of PJSIP 13](#_Toc442366904)

[Appendix 13](#_Toc442366905)

[A Sip Servers And Clients for Pi 13](#_Toc442366906)

[B PJSIP/PJSUA Build Help Guide 14](#_Toc442366907)

# Purpose

This document is step by step guide to help user who intend to setup SIP server and SIP client on Raspberry Pi from scratch and how to test if call send and receive is working fine to / from Pi.

# Introduction

There are many SIP servers and clients (for list please see appendix) available on pi. We choose SIPwitch as SIP server user agent and pjsua (a pjsip based) SIP client for our demonstration and development purpose.

In following section, we will cover how final test setup looks like and walk through the steps right from setup to the point to be able to make calls between different soft phones

# Test Setup Overview

Our test setup include 2 windows based PC and a Raspberry Pi running linux debian wheezy 2015-05-05-raspbian-wheezy.img

Here’s is demo picture setup giving an overview of running system:



Above picture shows 2 windows machine and 1 Raspberry Pi where 1 machine running linphone softphone and another microsip and raspberry pi running both sip server “sipwitch” listening at port 5060 and a sip client “pjsua” listening at port 5061\*. So there are 3 clients registering themselves to sipwitch server at port 5060.

\*we used 5061 port because we are running server and client on same machine. By default any sip client and server use rtp port 5060 for communication. Example linphone and microsip using port 5060 when running on windows

# Setting up sipwitch on Pi

If you don’t have sipwitch installed, follow the link below else skip to instructions below

<https://www.packtpub.com/books/content/calling-your-fellow-agents>

## Configuring pre-installed sipwitch:

If you have sipwitch installed, do following

1. sudo nano /etc/default/sipwitch
   1. Uncomment  #PLUGINS="zeroconf scripting subscriber forward"  by removing # sign
2. sudo nano /etc/sipwitch.conf
3. goto <provision> and start adding lines under it.
   * 1. add following between <provision> and </provision> (make sure entries are outside <!-- and --> tags
     2. <user id=“string giving user a name”>. Note there should be no space around = sign
     3. <extension> integer value identifying extension of user</extension>
     4. <secret>password of the user </secret>
     5. Example, to add user with id “phone1” and extension 201 and password 201, entry in /etc/sipwitch.conf would be:

<provision>

<user id=“phone1”>

<extension>201</extension>

<secret>201</secret>

</user>

</provision>

* + 1. Save and close the file
    2. By default, it supports extension from 201 to 299. Softphones connected to the server can be assigned the numbers in this range

1. Restart sipwitch service by giving
   1. sudo service sipwitch restart
2. you can run following commands to check if sipwitch successfully started:
   1. sudo sipwitch registry
   2. sudo sipwitch dump

## Troubleshooting

Q. Getting message \*\*\* sipwitch: command: offline on running any sip command

A. Some system requires rebooting. Try reboot after setting up sipserver.

# Setting up pjsua on Pi

For making pjsua (a pjsip based softphone), we need to download the source code and compile for pi linux in use. Here we can download source code:

<http://www.pjsip.org/download.htm>

At the time of writing this document, latest release version is pjsip 2.4.5. Download it and copy on your Pi machine.

## Building on Pi

Before building, user needs to install certain packages on Pi (if not already available):

sudo apt-get install libasound2-dev

\*libasound2-dev is important if want to use alsa driver for audio devices.

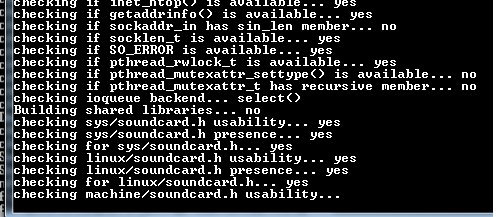
Since I didn’t want video for my setup and needed alsa as audio driver, thus I built sdk as

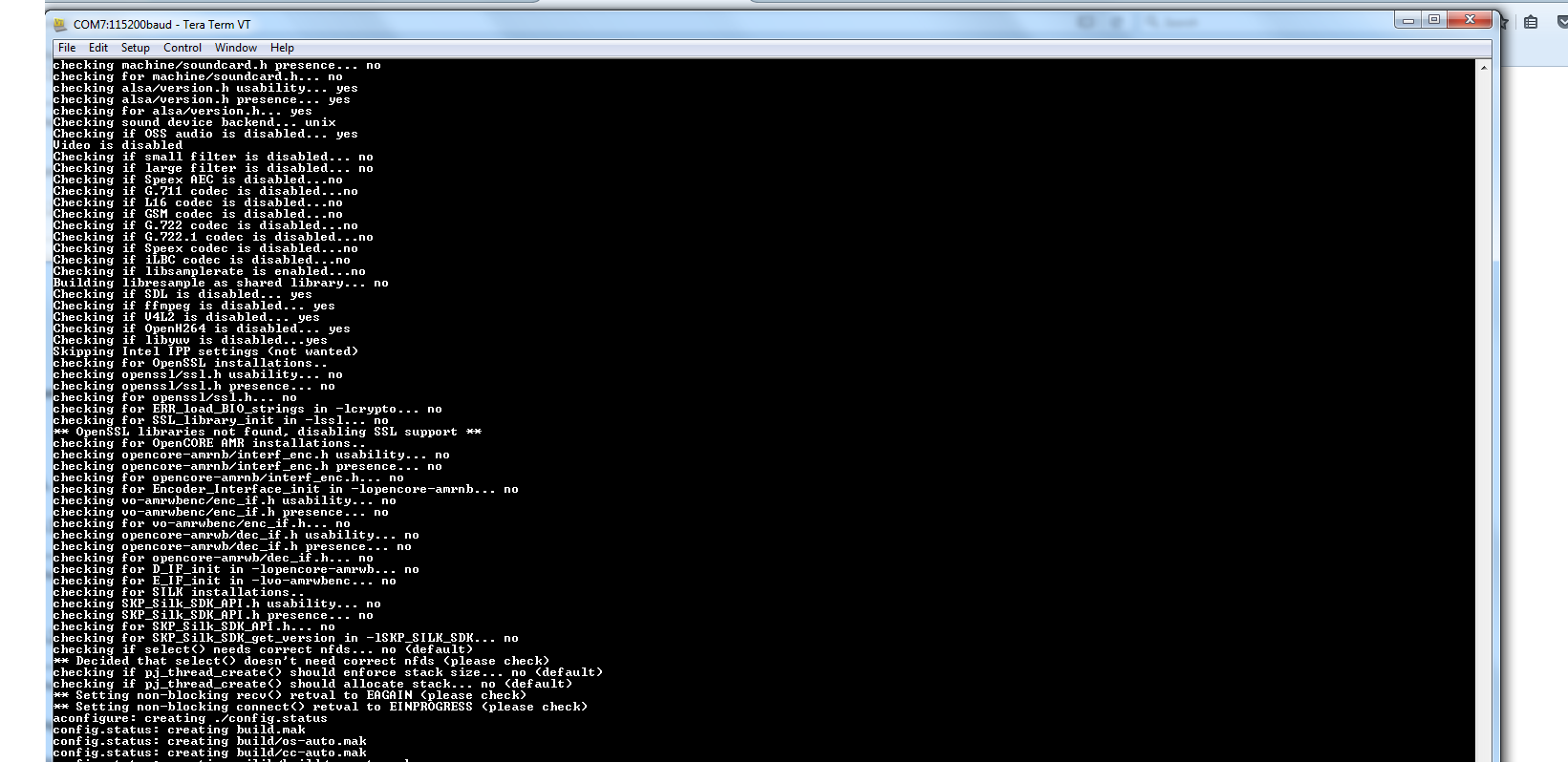
* ./configure –disable-video –disable-oss
* ./make dep
* ./make

\*--disable-oss is important because if we don’t mention it and alsa library and header is found then pjsip would compile for OSS library and that’s not what we want

Make sure that alsa driver is seen and visible to pjsip by checking output of ./configure command.

Example console dump of ./configure output with alsa is here:





## Few Checks

1. Check if pjsip can see audio devices available in system
   1. Go to ~/pjproject-2.4.5/pjsip-apps/bin/samples/armv6l-unknown-linux-gnueabihf
   2. run ./auddemo . Command would display number of audio devices identified by pjsip in a system. Each device is assigned a number which is further used in another application ex. pjsua

Example console dump of “auddemo” on my system



# Setting up microsip on 2nd window PC

Follow the link under section “Setting up sipwitch on Pi” to get download site and setup

# Client Registration

## Registering Pi to sipwitch server

1. Now goto ~/pjproject-2.4.5/pjsip-apps/bin/ and run pjsua app with relevant input options. This application supports many features including registration
2. Sipwitch server already have a user created as “pi” with extension “200”. We can setup pjsua to register itself with same user id. However before that we need to setup password information for “pi” user on server side.
3. Run command “sudo sippasswd pi” on machine running sipwitch

After password for pi user is set, we can run following command to register pjsua as a pi user to server (in this command password is set to raspberry)

./pjsua-armv6l-unknown-linux-gnueabihf --id=sip:pi@10.60.132.106 --registrar=sip:10.60.132.106 --username=pi --password=raspberry --realm=\* --local-port=5061 --capture-dev=3 --playback-dev=3

Following is console snapshot of above command (you can see snd\_microsemi\_dac identified as playback and capture device)

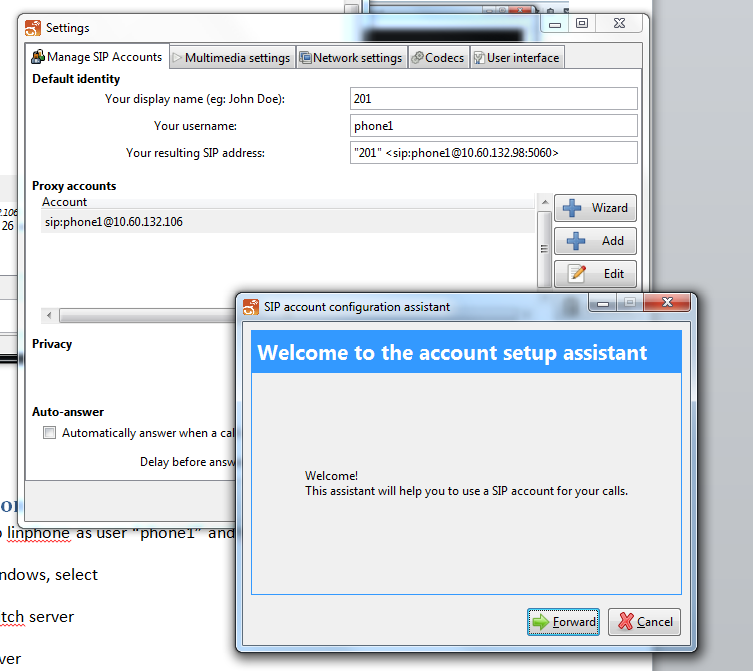


## Registering linphone on window PC 1

In current example, we setup linphone as user “phone1” with extension 201.

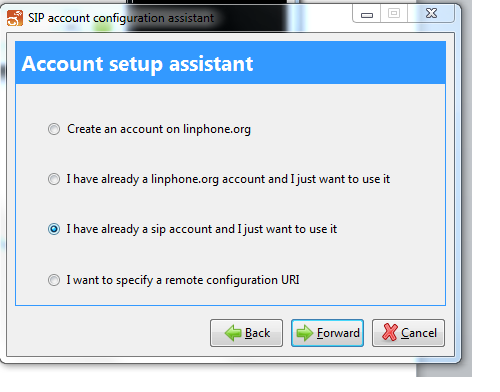
As you install linphone on windows, add account information:

1)Go to *Options->Preferences->Manage SIP Accounts*->*wizard*



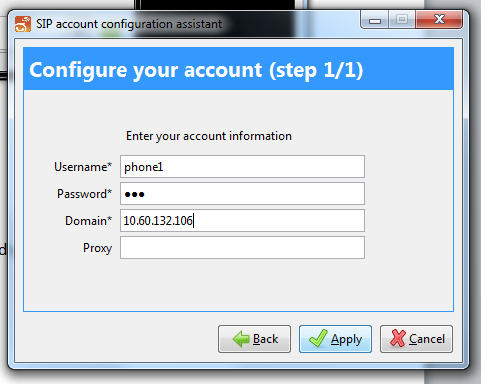
Click here

2) Since we already configured sipwitch server with user “phone1” information, we can select option *“I already have a sip account and I just want to use it”*

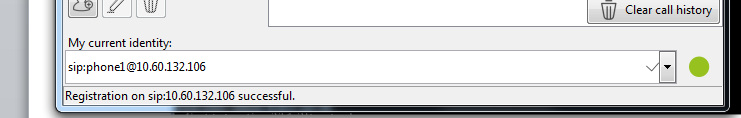


Select this

3) Select *forward* tab and to setup linphone as user “phone1”with *extension 201* to connect to sip server with *ipaddr 10.60.132.106* fill following and click *Apply*.



4) Successful registration to sip will show this status bar



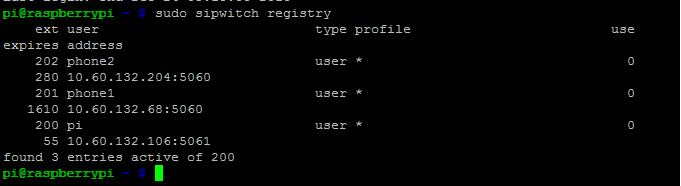
## Registering Minisip to sipwitch server

Information detailed in link given under section “Setting up sipwitch” you can register it as user “phone2” extn “202”. Password as you set in /etc/sipwitch.conf for user 202. In our example setup, password for 202 is also 202.

## Checking client registration to Sipwitch

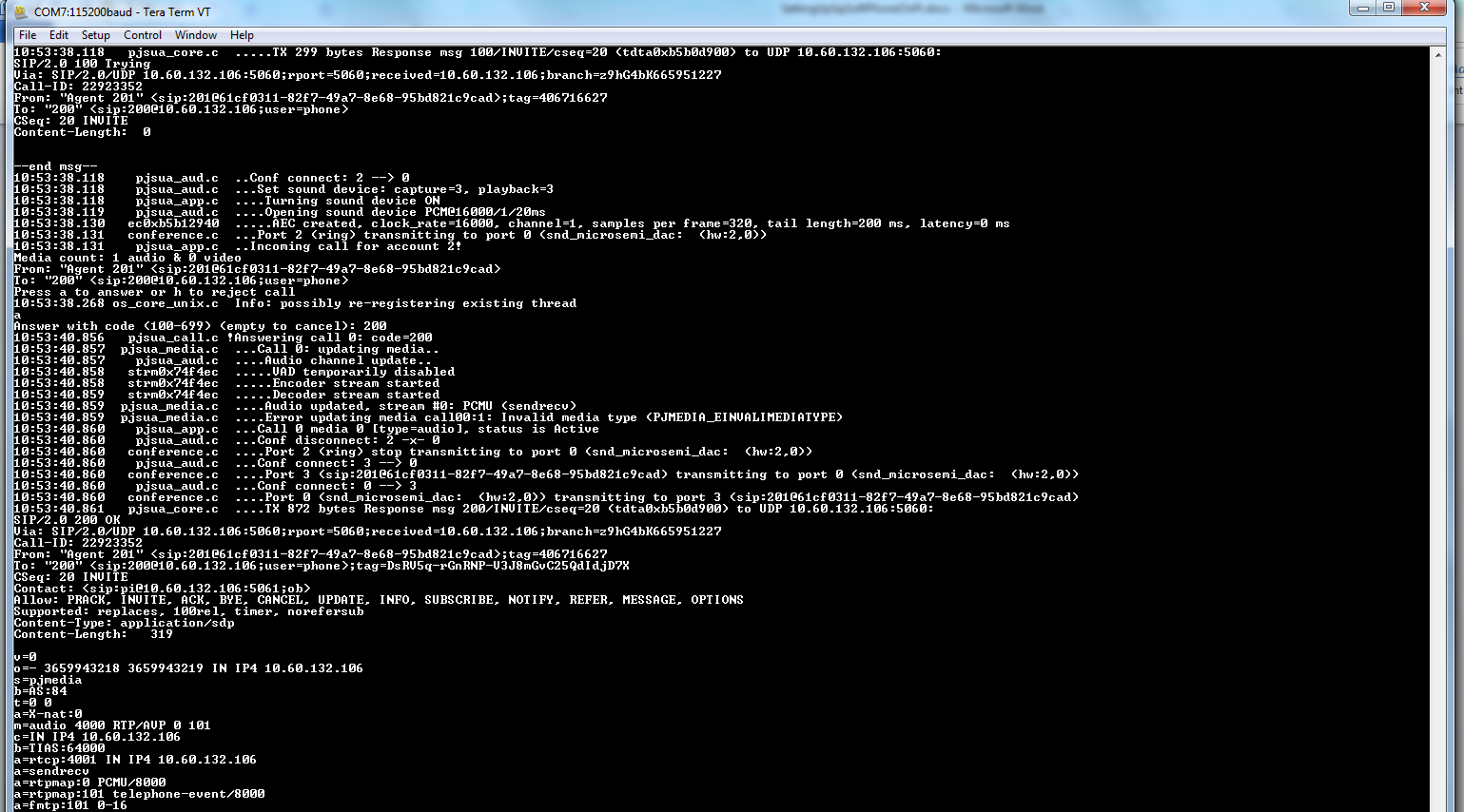
1. Go to Pi console and run command *sudo sipwitch registry*

Example console snapshot of the run :



# Making calls

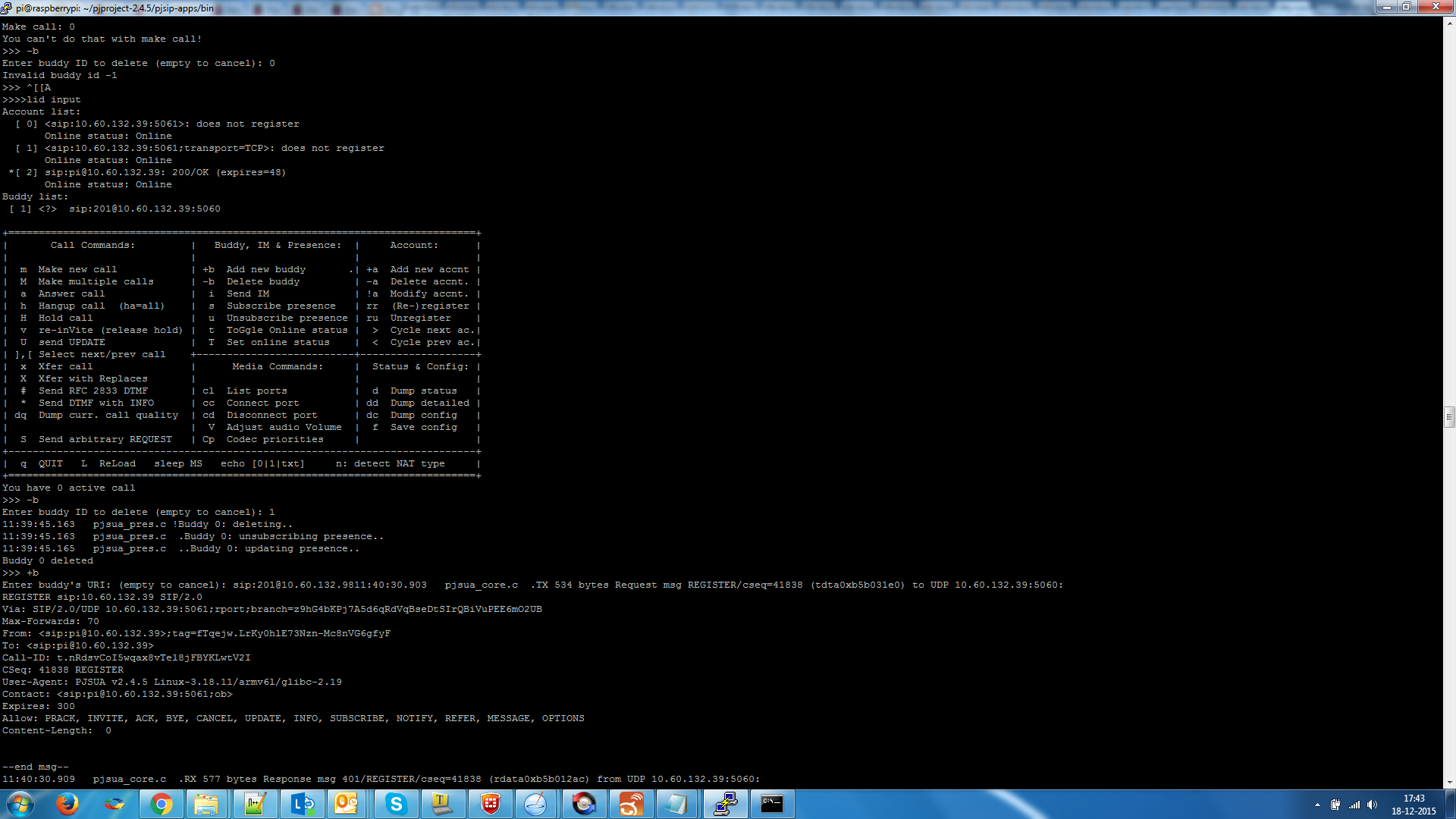
1. linphone->pjsua
   1. Dial 200
   2. Pjsua should respond with message “Press a to answer and h for hangup”
   3. Enter “a”
   4. It will ask for answer code
   5. To continue to talk, enter 200 and press enter
   6. Now your talk session is established

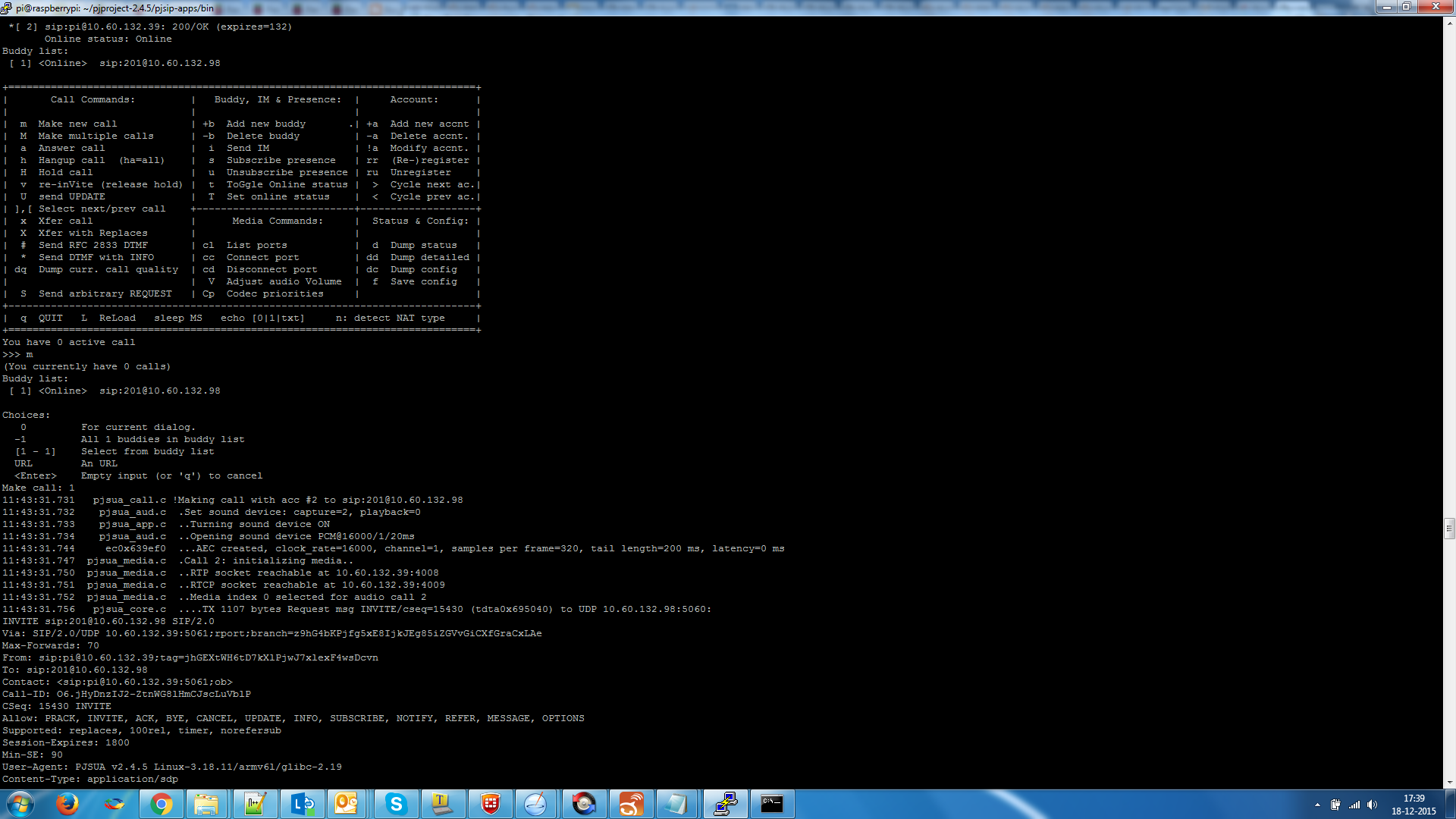


1. pjsua->linphone

To make a call from pi to linphone, 1st add linphone as buddy to pjusa

* 1. Enter +b option
  2. Enter uri [sip:phone1@<ip addr of windows machine running linphone>](sip:phone1@%3cip%20addr%20of%20windows%20machine%20running%20linphone%3e)
  3. Linphone may show message “user pi want to connect”
  4. Enter m option
  5. Phone1 will be shown “online” under buddy list, enter 1 to connect to
  6. Call is established
  7. Here ‘s snapshot of run





# Cross Compiling of PJSIP

We tried to cross compile this app for Pi however this exercise need libasound libs and other dependencies compiled on Pi and available on cross compiler machine. This is exercise “Yet to be done”

# Appendix

## A Sip Servers And Clients for Pi

List of other SIP server available for Raspberry Pi , here are listed some along with reason why they not been choosen for our testing

* miniSIPserver
  + got binary which is compiled for 2013-07-26-wheezy-raspbian so its very likely that it will not run on latest kernel we have.
  + Unable to get its source code so that it can be evaluated by doing compilation for native linux and run once to see if that works
* Freeswitch
* Astreik (little bulky)
* OpenSPIs (available for pi??)

Here are list of SIP clients on Pi

* -linphone
* miniSip
* pjsua

I was looking for sip softphone supporting command line interface so that I don’t have for display. Initial priority was to use installer thus I switched to linphone which comes with installer and tried to use linphonec on Pi. Whenever user was calling it could ring but as soon as call is connecting it was freezing and there was no voice setup. It was becoming difficult to debug so I switched to next recommended pjsua , a pjsip based softphone on Pi. Only disadvantage on using it on pi that there’s no pre-existing installer for it and we have to take source code and compile it on Pi machine we have.

Minisip didn’t try.

## B PJSIP/PJSUA Build Help Guide

For more information, these links is helpful ( you may directly refer to BUILD PJSIP section):

http://marpoz.blogspot.it/2013/05/build-door-berry-dependencies.html

<http://embbsys.blogspot.in/2015/04/steps-to-compile-pjsip-with-alsa.html> (please refer to only dependencies to be installed. We don’t need explicit setting of HAS\_PORT\_AUDIO 0. Page seems to be outdated. As long as configure script find alsa header file in /usr folder it will be enabled)

<https://trac.pjsip.org/repos/wiki/Getting-Started/Autoconf>

for full featured pjsip, please install whole list of dependent libraries

sudo apt-get install libv4l-dev libx264-dev libssl-dev libasound2-dev

\*libasound2-dev is important if want to use alsa driver for audio devices.

General building steps for pjsip are

* ./configure
* ./make dep
* ./make

if want to use raspberry built-in 3.5 mm jack as an output, then 1st load snd-dummy module and set this as capture device because raspberry don’t have mic-in ONLY speaker-out so and then need to set capture and playback device to a number corresponding to bcm2835 and dummy device (can be checked by running auddemo app)

Follow link here to get help on load snd-dummy

* <https://www.raspberrypi.org/forums/viewtopic.php?p=485842#p485842>
* <https://www.raspberrypi.org/forums/viewtopic.php?f=44&t=73645&p=532692#p532692>