OpenMCU-ru Administrator Guide

Revision 1.09

Author:

Eugene Blanchard
SAIT Polytechnic
Calgary, Alberta
Canada
eugene.blanchard@sait.ca

Table of Contents

1. Introduction

This manual is based on the original OpenMCU User Manual located here:

http://openmcu.ru/public/OpenMCU-ru/manual/user_guide_en.pdf

OpenMCU-ru is the fork of original OpenMCU with a lot of new features. OpenMCU is a simple Multi Conference Unit (Multipoint Control Unit) using the H.323 and SIP protocols. It requires a special version of the H323plus library modified by Varnavskiy Andrey Ivanovich. It is known to run on Linux, FreeBSD and Windows and should run on any platform supported by H323plus and Sofia-SIP. After version 3.48 and starting with version 4.1, only CentOS and Windows platforms are supported.

This guide is free to distribute, modify, edit, share as you see fit as long as you reference the authors.

OpenMCU-ru project:
* Andrey Varnavskiy 'muggot', 2009-2014
* Konstantin Yeliseyev 'kay27', 2012-2014
* Andrey Burbovskiy 'xak-mcu', 2013-2014

OpenMCU original project (now abandoned):
* updated by: Craig Southeren, 6 March 2006
* author: Roger Hardiman, 20 June 2001
* homepage: http://www.voxgratia.org/

What is the OpenMCU-ru Video Conference Server?

OpenMCU-ru is the multipoint control unit (MCU) engine or core that allows real-time video conferences to occur. Multiple clients attach to the video conference using video enabled IP phones, web browsers, streaming enabled webcams and RTSP streaming players. An MCU is needed whenever 3 or more parties want to be in an audio (or video) call. It acts as a mixer and if necessary, acts as a transcoder (translator) between the audio and video codecs.

Traditional Hardware MCUs were custom-built hardware devices full of digital signal processors (DSPs) and field programmable arrays (FPGAs) which are specialized chips that require programmers and hardware engineers to build. They were hard to manage and operate, and cost well into the 6 figures ($$). Hardware MCUs are the most inflexible platforms.

The latest advent of multicore processors, multicore graphic processor units and PC platforms provides the hardware processing power that allow software MCUs to be created. OpenMCU-ru is an open source software MCU, that is easily modified, very flexible and inexpensive.

The MCU needs to mix/translate all the audio and video so that every participant in the conference is seen and heard. This mixing of audio and video streams may involve *transcoding*, a technique that allows one audio/video codec format to be transformed to another in *real-time*. The transcoding of audio and video codecs is the issue that loads the processing capabilities of a video conference server.

The purpose of a video conference server is to save money. Employees do not have to travel for face to face meetings. Large companies will spend millions of dollars on travel cost for employees to attend meetings and conferences. Video conference room costs a fraction of the travel expenses and still provides face to face contact.

OpenMCU-ru differs from common video conferencing services like Skype clients in that it provides the back-end processing for video conferencing. OpenMCU-ru is not limited to one audio/video codec and can transcode between multiple clients running different codecs. In addition, custom and multiple conference room displays can be configured for different physical locations. OpenMCU-ru leads itself to the creation of physical conference rooms:

The goal is to allow conference clients to Bring Your Own Device (BYOD): IP phones, cell phones, softphones, PCs, tablets and streaming enabled webcams. In addition, the video conference stream can be monitored through a web browser or RTSP client like VLC media player.

2. Test Environment Specifications

This user manual is based on testing using the following hardware and software platforms, the testing was performed using the SIP and H.323 protocol. I used a virtualized environment just because it was easy to setup a virtual machine in my lab. Having a dedicated server would work just as well if not better.

- Proxmox 3.4-1 host :
  - Dual 8 core Intel 3.4 GHz processors (16 cores)
  - 72 GB RAM
  - RAID 10 300 GB
  - HP DL160 G6
- Standalone OpenMCU-ru virtual machine guest:
  - 1 GB memory
  - 1 core
  - 32 GB hard-drive
  - Hardware x86_64
  - OpenMCU-ru 4.1.6
  - CentOS 7.0
    - SELinux disabled
    - Firewall disabled
  - Linux 2.6.32
- FreePBX+OpenMCU-ru virtual machine guest:
  - 2 GB memory
  - 2 cores
  - 32 GB hard-drive
  - Hardware x86_64
  - OpenMCU-ru 4.1.6
  - FreePBX distribution 13
- 100 Mbps Ethernet network
- Layer 3 switch with interVLAN routing (4 VLANs)
- Tested across 5 routed networks
- 4 to 5 Video clients running at any time:
  - 4x Linphone 3.8.4 Desktop client softphone for Windows
    - 640x480 video out from phone
    - 1280x720 video in to phone
    - H.264 codec
  - Nortel 1535 IP phone
    - 172x144 video out from phone
    - 352x288 video in to phone

3. Features

OpenMCU contains the following features:

- configured by a web interface on port 1420 (use http://<host IP address>:1420 or localhost IP address)
- requires no codec hardware to operate
- supports all plugin audio codecs supported by OpenH323
- supports H.261, H.263, H.263+, VP8 and H.264 video up to full HD 1080P with caching
- supports G.711, G.722, G.723, G.726, G.728, G.729, SILK, Speex, Opus and other plugin audio codecs
- can accept multiple connections simultaneously
- many different conferences can be taking place at the same time using the ‘rooms’ feature
- displays statistics on calls in progress
- support the use of a gatekeeper (for example gnuGk)
- there is artificial limit of participants at 100 video endpoints
- default sound is transmitted and reproduced on all connected participants
- can initiate (invite) participants from the MCU to rooms

4. Operation

OpenMCU works by setting up a SIP or H.323 listener process and then waiting for incoming connections. Whenever an incoming connection is established, it determines which conference is required via the 'rooms' feature and adds the call to that conference. You call the MCU using the format 'room_name@server_name' or 'server_name##room_name' depending on the notation for your hard/software client.

For Example for myphone3 type:

Room101@mcu.myservers.com

New rooms are created automatically and there is a default room for people who do not specify a room or cannot specify a room (eg NetMeeting). The default room is called room101 (this can be changed in the Settings Section).

5. Command Line Options

OpenMCU-ru ***can*** be run by the command line but it is ***not*** necessary as the installer will automatically configure it to run properly. A web GUI interface is ***normally*** used to configure OpenMCU-ru. You will most likely ***never*** use these commands.

```
Openmcu [-c] –v|-d|-h|-x

  -h –help            output this help message and exit

  -v –version         display version information and exit

  -d –daemon          run as a daemon

  -u –uid uid         set user id to run as

  -g –gid gid         set group id to run as

  -p –pid-file        name or directory for pid file

  -t –terminate       orderly terminate process in pid file

  -k –kill            preemptively kill process in pid file

  -s –status          check to see if daemon is running

  -c –console         output messages to stdout rather than syslog

  -l –log-file file   output messages to file or directory instead of syslog

  -x –execute         execute as a normal program

  -i –ini-file        set the ini file to use, may be explicit file or

                       a ‘:’ separated set of directories to search.

  -H –handlemax n     set maximum number of file handles (set before uid/gid)

  -C –core-size       set the maximum core file size
```

6. Installation:

Section ***6*** describes installing OpenMCU-ru on a standalone server:

- Section 6.1.1 describes installing on a CentOS Full ISO distribution
- Section 6.1.2 describes installing on a CentOS Minimal ISO distribution
- Section 6.2 describes installing on a Windows 7 desktop

Section ***25*** describes installing OpenMCU-ru on an existing FreePBX server.

OpenMCU-ru can be installed in Linux or Windows. The operation of OpenMCU-ru is similar for both operating systems.

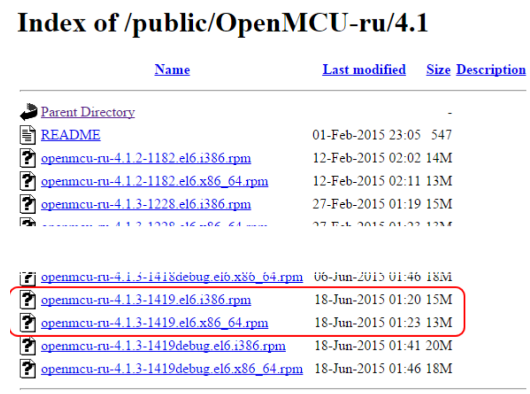
6.1.1 CentOS Full ISO Installation

Note: CentOS was chosen as it is the ***only*** Linux distribution supported past version 3.48 and it is the absolute easiest Linux installation method at this time! Version 3.48 is ***no*** longer supported and has ***many*** bugs and problems that have been solved with the 4.1 branch.

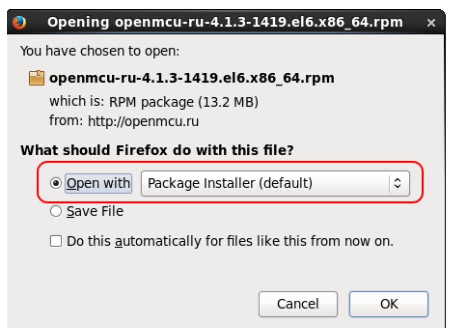
1. I installed CentOS 6.6 as a virtual machine under ProxMox 3.4.
    - Immediately did a Software Update of CentOS (twice to catch everything)
    - SELinux was disabled
    - Firewall was disabled with plans to enable it and open the needed ports later.
    - CentOS 7 is similar

2. CentOS installation: open a web browser and go to http://openmcu.ru/public/OpenMCU-ru/ and find the latest version of OpenMCU-ru. At the time of writing, 4.1.3 was the latest:

    Note: “x86_64” indicates the 64 bit version, “i386” indicates the 32 bit version. The version of OpenMCU-ru should match the version of CentOS (32 bit or 64 bit) that you are using.



3. Click on it and select “Open with Package Installer”. You will need root privileges.



4. Package Installer will automatically check for dependencies and install if needed.

5. OpenMCU-ru is installed in /opt/openmcu-ru directory:

```
[root@localhost openmcu-ru]# ls -l
total 80
-rw-r-r--  1 root root   320 Jun  5 15:23 AUTHORS
drwxr-xr-x  2 root root  4096 Jun 16 14:30 bin
drwxr-xr-x  2 mcu  mcu   4096 Jun 26 13:10 config
-rw-r-r--  1 root root 18305 Jun  5 15:23 COPYING
drwxr-xr-x  2 root root  4096 Jun 16 14:30 font
drwxr-xr-x 12 root root  4096 Jun 16 14:30 include
drwxr-xr-x  4 root root  4096 Jun 16 14:30 lib
drwxr-xr-x  2 mcu  mcu   4096 Jun 16 14:54 log
-rw-r-r--  1 root root  1069 Jun  5 15:23 NEWS
drwxr-xr-x  2 mcu  mcu   4096 Jun 26 13:12 pipe
-rw-r-r--  1 root root  2566 Jun  5 15:23 README
drwxr-xr-x  2 mcu  mcu   4096 Jun 26 14:45 records
drwxr-xr-x  2 root root  4096 Jun 16 14:33 resource
drwxr-xr-x  2 root root  4096 Jun 16 14:30 scripts
drwxr-xr-x  4 root root  4096 Jun 16 14:30 share
drwxr-xr-x  2 root root  4096 Jun  5 15:23 ssl
```

6. Once installed, check that OpenMCU-ru is running. As the root user, from a terminal window type:

   service openmcu-ru status
   Or
   /etc/init.d/openmcu-ru status

```
[root@localhost student]# service openmcu-ru status
openmcu-ru (pid  1467) is running...
```

7. Starting and stopping the server (rarely ever need to):

   service openmcu-ru start/stop
   Or
   /etc/init.d/openmcu-ru start/stop

8. Set a static IP address for the server. Go through the X windows GUI or through the command line, modify : "/etc/sysconfig/network-scripts/ifcfg-eth0" for CentOS 6.6. Use appropriate settings for your network.

```
DEVICE="eth0"
BOOTPROTO=static
ONBOOT=yes
HWADDR="1E:4B:6B:A5:4C:63"
TYPE="Ethernet"
IPADDR=192.168.204.251
NETMASK=255.255.255.0
GATEWAY=192.168.204.1
```

   - Restart the network services.

```
[root@localhost network] service network stop
[root@localhost network] service network start
```

9. Check if the SSHD daemon is running, this is for CentOS 6 and earlier
   - check if it has been installed by typing: "ls /etc/init.d" Look for sshd to be present.
   - If it is not installed, install it by typing: "yum –y install openssh-server openssh-clients"
   - To run sshd, type "service sshd start" or "/etc/init.d sshd start".
   - To make sshd start automatically each time the server boots, type "chkconfig sshd on".

10. Optional: uninstall X windows to save resources and minimize performance loading, SSH into the server:
    - Change the run level to full multiuser mode (command line only) from X11 windows by modifying /etc/inittab file line from "id:5:initdefault:" to id:3:initdefault:"
    - Uninstall X windows by typing: ***yum groupremove "X Window System"***

If you are running a dedicated server then you may want to uninstall X Windows. If you are experimenting with OpenMCU-ru, you may want to use X windows for troubleshooting and testing.

6.1.2 CentOS 7 Minimal ISO Installation

Note: CentOS was chosen as it is the ***only*** Linux distribution supported past version 3.48 and it is the absolute easiest installation method at this time! Version 3.48 is ***no*** longer supported and has ***many*** bugs and problems that have been solved with the 4.1 branch.

This installs creates a ***minimal*** installation with a dedicated server only for OpenMCU-ru. It should be the most efficient usage with all resources dedicated to OpenMCU-ru. No X windows environment.

1. Installed CentOS 7 15.11 64 bit minimalist. This provides a command line only interface to CentOS. No GUI or X windows. An alternative is that you can SSH to it.

The following commands assume that you are the root superuser or have sudo'd the commands:

2. With CentOS 7, "ifconfig" has been replaced with "ip addr". Eth0 is now called by a persistent name such as en018. Here's a reference if you want to continuing using "ifconfig" and to change back to eth0: https://www.unixmen.com/ifconfig-command-found-centos-7-minimal-installation-quick-tip-fix/

3. If "ip addr" command does not display an IPv4 address (see ens18 interface in example):

```
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether ee:51:2b:ee:03:6f brd ff:ff:ff:ff:ff:ff
[root@localhost ~]#
```

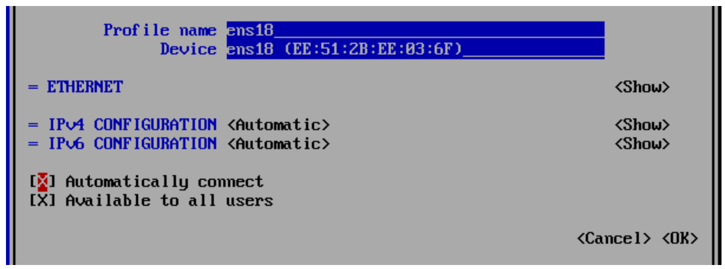
- Perform a "nmcli d" command to list the Ethernet interfaces. The example shows the Ethernet interface listed as "ens18" which verifies the "ip addr" command.

```
[root@localhost ~]# nmcli d
DEVICE  TYPE      STATE         CONNECTION
ens18   ethernet  disconnected  --
lo      loopback  unmanaged     --
```

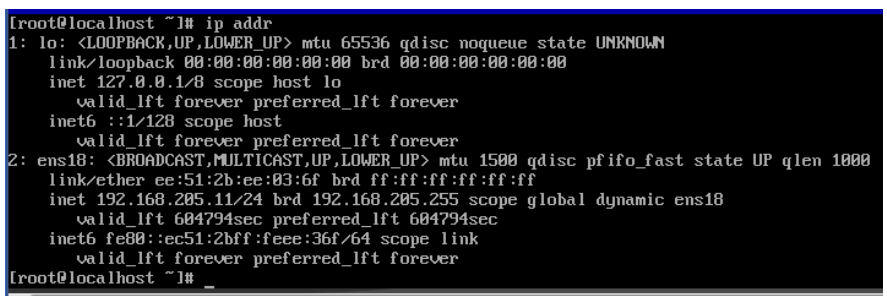
- Run the network manager user interface "nmtui"

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

- Edit the connection, select "Automatically connect" and OK then Quit



- Run "ip addr"again and the IPv4 address should show up:



4. After installation of CentOS 7, to update all packages (select "y" to all prompts as they are a lot smarter then we are!):

   # yum update

5. I like to use the nano text editor, to install:

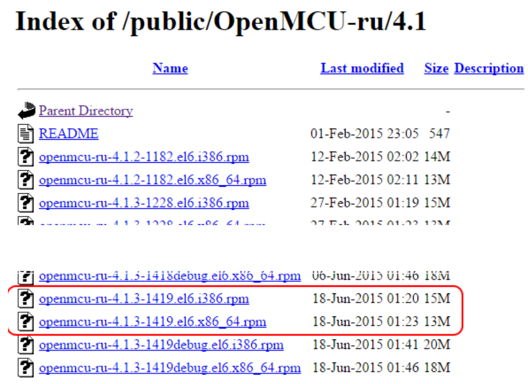
   # yum install nano

6. To download OpenMCU-ru from the repository, we'll use the wget program. To install wget:

   # yum install wget

7. Point your web browser to http://openmcu.ru/public/OpenMCU-ru/4.1/ (version 4.1 at the time of writing) and ***determine*** which is the latest stable release of OpenMCU-ru. Usually it is the 2nd from last on the webpage. You'll need to know the full name to download it. In this installation example the 2nd last file was "openmcu-ru-4.1.6-1448.el6.x86_64.rpm"

   Note: "x86_64" indicates the 64 bit version, "i386" indicates the 32 bit version. The version of OpenMCU-ru should match the version of CentOS (32 bit or 64 bit) that you are using.



Index of /public/OpenMCU-ru/4.1

Name	Last modified	Size	Description
Parent Directory		-	
README	01-Feb-2015 23:05	547	
openmcu-ru-4.1.2-1182.el6.i386.rpm	12-Feb-2015 02:02	14M	
openmcu-ru-4.1.2-1182.el6.x86_64.rpm	12-Feb-2015 02:11	13M	
openmcu-ru-4.1.3-1228.el6.i386.rpm	27-Feb-2015 01:19	15M	
openmcu-ru-4.1.3-1418debug.el6.x86_64.rpm	06-Jun-2015 01:46	18M	
openmcu-ru-4.1.3-1419.el6.i386.rpm	18-Jun-2015 01:20	15M	
openmcu-ru-4.1.3-1419.el6.x86_64.rpm	18-Jun-2015 01:23	13M	
openmcu-ru-4.1.3-1419debug.el6.i386.rpm	18-Jun-2015 01:41	20M	
openmcu-ru-4.1.3-1419debug.el6.x86_64.rpm	18-Jun-2015 01:46	18M	

8. Back on your server command line. Change directory (cd) to /usr/src

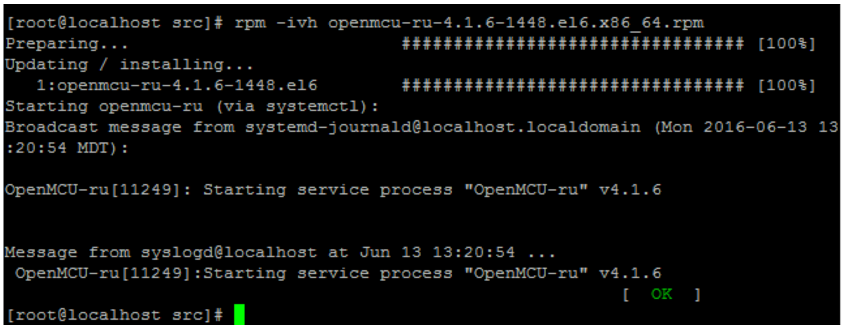
   # cd /usr/src

9. Download the latest version to your server:

   # wget http://openmcu.ru/public/OpenMCU-ru/4.1/openmcu-ru-4.1.6-1448.el6.x86_64.rpm

10. Install OpenMCU-ru by using the rpm command:

   # rpm -ivh openmcu-ru-4.1.6-1448.el6.x86_64.rpm ("--replacefiles" if errors)



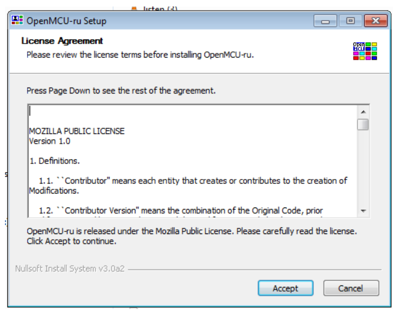
```
[root@localhost src]# rpm -ivh openmcu-ru-4.1.6-1448.el6.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:openmcu-ru-4.1.6-1448.el6        ################################# [100%]
Starting openmcu-ru (via systemctl):
Broadcast message from systemd-journald@localhost.localdomain (Mon 2016-06-13 13
:20:54 MDT):

OpenMCU-ru[11249]: Starting service process "OpenMCU-ru" v4.1.6


Message from syslogd@localhost at Jun 13 13:20:54 ...
 OpenMCU-ru[11249]:Starting service process "OpenMCU-ru" v4.1.6
                                                           [  OK  ]
[root@localhost src]#
```

6.2 Windows Installation

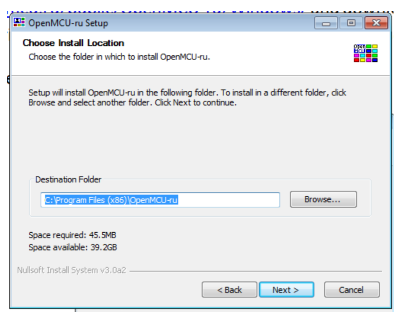
The Windows version is good to use for ***beta*** testing or if you just want to try it out. I would not recommend it for a production environment because of the load that the Windows GUI puts on the server. It takes a performance hit. If you want performance, go with the CentOS minimal install. The Windows installation requires 45.5 Mbytes of hard-drive space.

1. Go to http://openmcu.ru/public/OpenMCU-ru/Windows/ and download the latest version. At the time of this writing it was "openmcu-ru-4.1.6.1430-beta-win32_setup.exe"

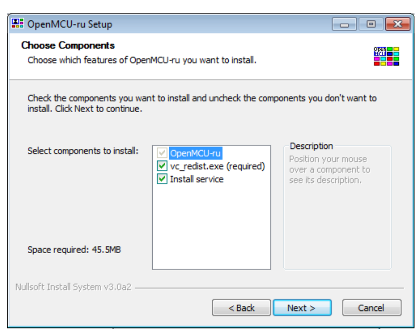
2. Run the executable and agree to the License Agreement



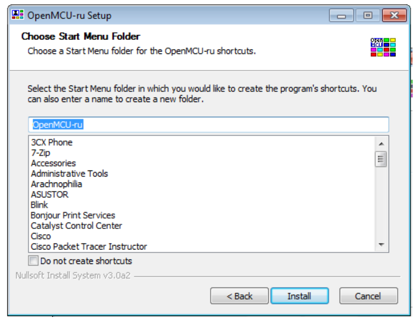
3. Next you are presented with the Install Location:



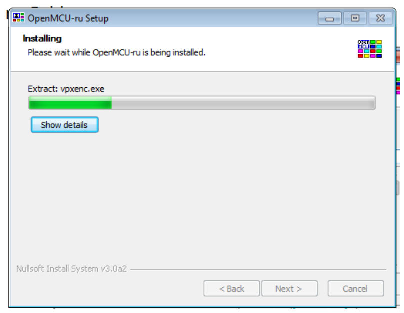
4. Select the components to install:



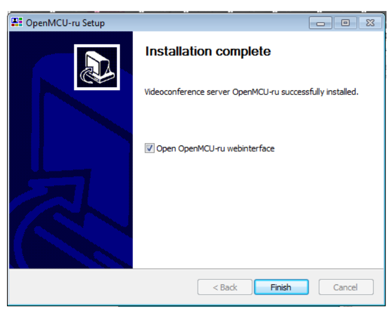
5. Choose a Start Menu Folder:



6. Select Install:

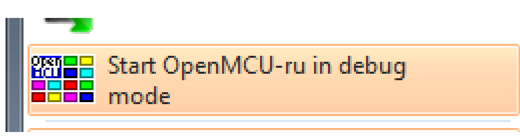


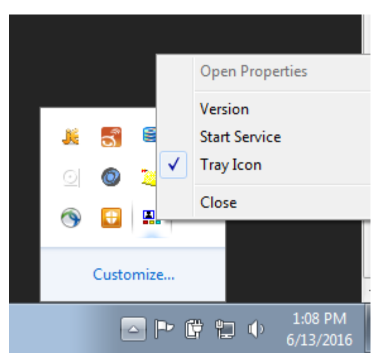
7. Finish and open the OpenMCU-ru Web Interface



8. OpenMCU-ru will be located at http://127.0.0.1:1420

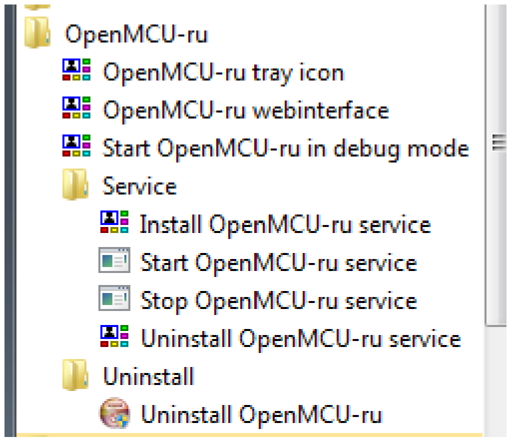
9. The Windows Start Menu, Taskbar and Tray will indicate:



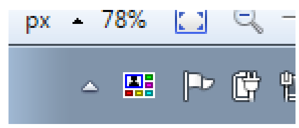


10. Windows specific information:

- Installed in C:\Program Files (x86)\OpenMCU-ru folder
- Start menu choices:



i. Selecting OpenMCU-ru tray icon, brings the OpenMCU-ru icon to the System Tray

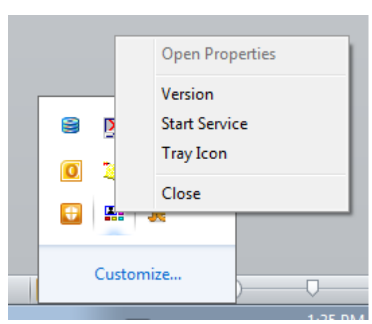


ii. Selecting Start OpenMCU-ru service from the Start menu doesn't appear to work even if selected as Run as Administrator.

iii. The Stop OpenMCU-ru service does work.

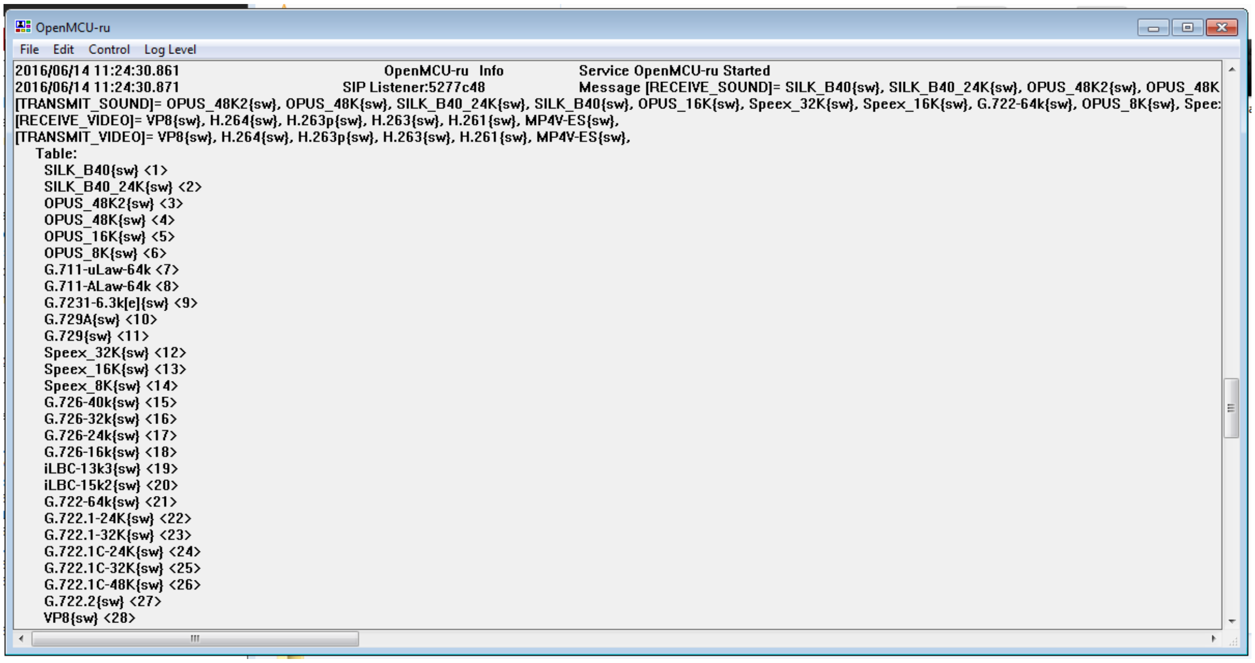
iv. Selecting OpenMCU-ru webinterface brings up the Web GUI when OpenMCU-ru is running

- Hidden Icons Tray or Tray Icon , right clicking brings up :



i. The Start Service choice does start OpenMCU-ru properly.

- Debug window and control:

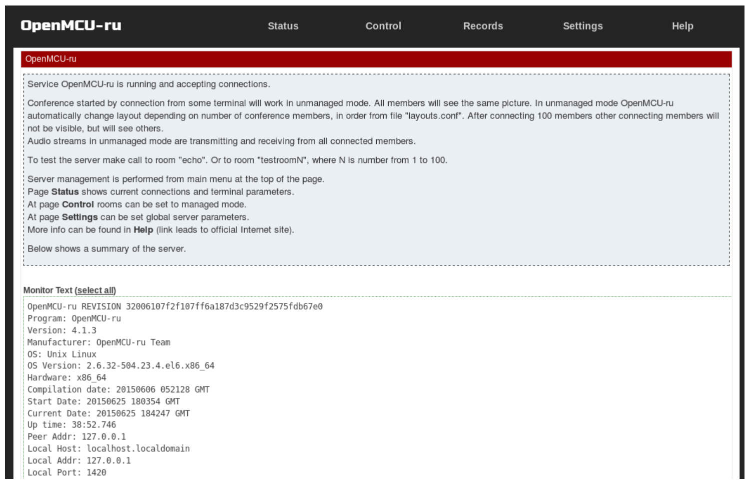


- Command line instructions:

OpenMCU-ru [Tray | NoTray | Version | Install | Remove | Start | Stop | Pause | Resume | Deinstall | NoWin]

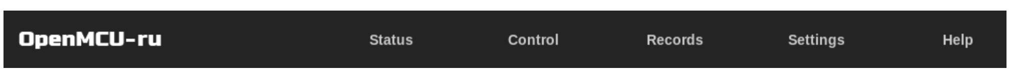
7. Exploring OpenMCU-ru

1. Open up a web browser and point it to http://<server IP address>:1420 to your server. There is no login prompt with the default configuration. You will start at the Welcome Page of OpenMCU-ru:



2. The Welcome Page is divided into 4 sections:

   - The Menu Bar



     - Clicking on the ***OpenMCU-ru logo*** returns to the Welcome Page
     - The Status option displays an information window on the currently running conference rooms
     - The Control option opens up the Main Configuration window for conference rooms
     - The Records option opens the Recording Management window that displays and manages recorded conference rooms.
     - The Settings option is a pull-down menu consisting of submenus for configuring OpenMCU-ru
     - The Help option directs you to the OpenMCU-ru wiki and forums.

- Information Section

OpenMCU-ru

Service OpenMCU-ru is running and accepting connections.

Conference started by connection from some terminal will work in unmanaged mode. All members will see the same picture. In unmanaged mode OpenMCU-ru automatically change layout depending on number of conference members, in order from file "layouts.conf". After connecting 100 members other connecting members will not be visible, but will see others.
Audio streams in unmanaged mode are transmitting and receiving from all connected members.

To test the server make call to room "echo". Or to room "testroomN", where N is number from 1 to 100.

Server management is performed from main menu at the top of the page.
Page **Status** shows current connections and terminal parameters.
At page **Control** rooms can be set to managed mode.
At page **Settings** can be set global server parameters.
More info can be found in **Help** (link leads to official Internet site).

Below shows a summary of the server.

The Information Section informs you that the server is running and general information.

- Server Status

Monitor Text (select all)

```
OpenMCU-ru REVISION 32006107f2f107ff6a187d3c9529f2575fdb67e0
Program: OpenMCU-ru
Version: 4.1.3
Manufacturer: OpenMCU-ru Team
OS: Unix Linux
OS Version: 2.6.32-504.23.4.el6.x86_64
Hardware: x86_64
Compilation date: 20150606 052128 GMT
Start Date: 20150625 180354 GMT
Current Date: 20150625 184247 GMT
Up time: 38:52.746
Peer Addr: 127.0.0.1
Local Host: localhost.localdomain
Local Addr: 127.0.0.1
Local Port: 1420
Room Count: 0
Max Room Count: 0
```

The Server Status section provides information on the configuration of the server

- Custom Logo Image Section

Custom logo iamge

This image is shown when no input video in mixer. Accepted only BMP, JPEG(maximum 500kB), PNG, GIF.

Change: Browse... No file selected. Submit Query

The Custom Logo Image section allows you to upload an image that will be used in a conference room when a participant initially connects and before the participant's video starts.

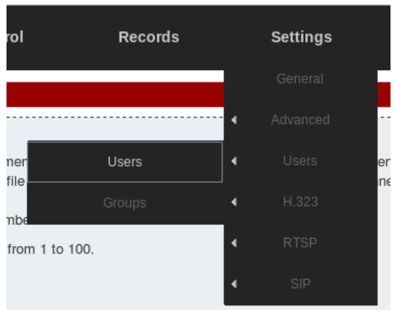
I created a simple 640x480 image to use (you can add your company logo):

Custom Logo Image Example

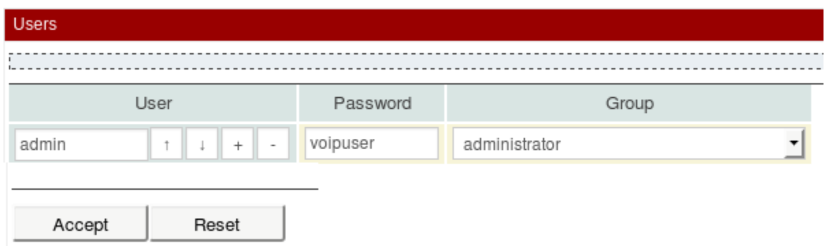
8. Initial Settings

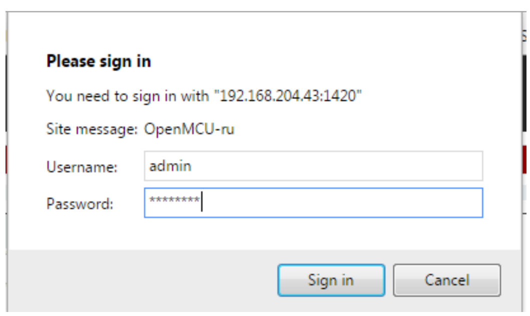
OpenMCU-ru works very well "right out of the box" and there are very few settings that need to be changed.

1. First step is to secure your server by configuring a password for the admin user. Go to Settings – Users – Users:

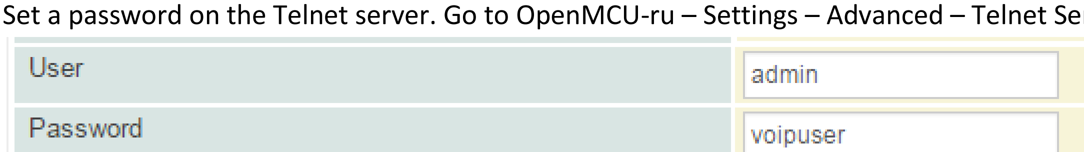


2. Add a password to the "admin" user and select Accept. Now when you connect to OpenMCU-ru, you will be asked for a username and password.

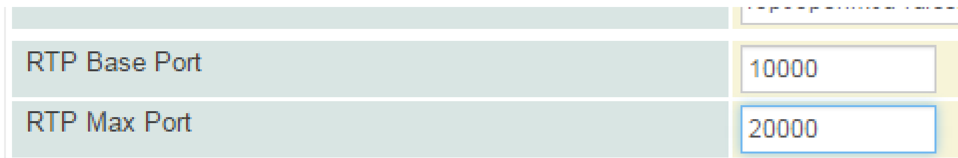




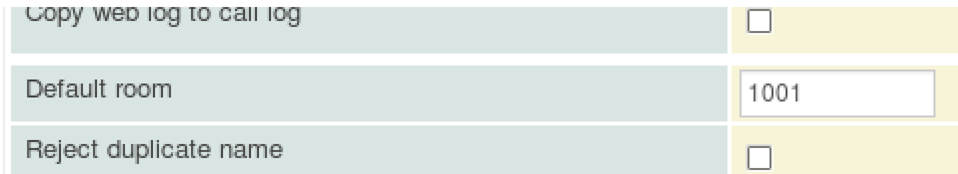
3. Set a password on the Telnet server. Go to OpenMCU-ru – Settings – Advanced – Telnet Server:



4. To make OpenMCU-ru compatible with FreePBX, go to Settings – General, set the RTP port range to match the FreePBX range: 10,000 – 20,000 :

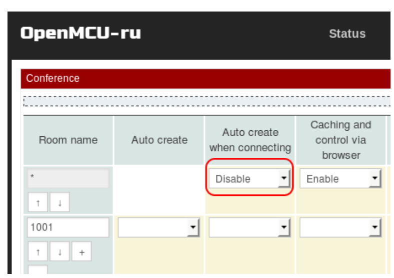


5. While in Settings – General: the default conference room name is “room101”. To make it easier to integrate with FreePBX (asterisk based PBX), I changed the default room to “1001” :



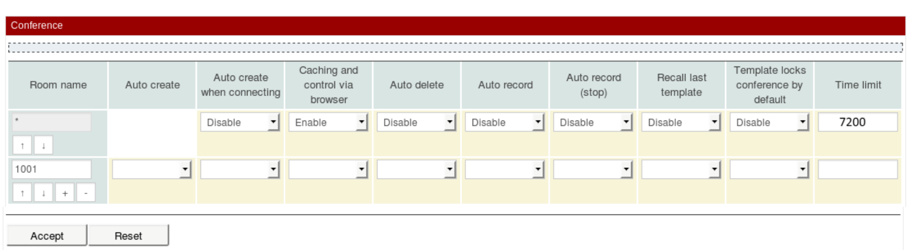
Note: OpenMCU-ru and FreePBX will both accept names for rooms such as “MonthlyMeeting” instead of numbers such as “1001”.

6. For security purposes, disable auto-create conference room on the wildcard room. Go to Settings – Advanced – Conferences and disable “Auto create when connecting”:

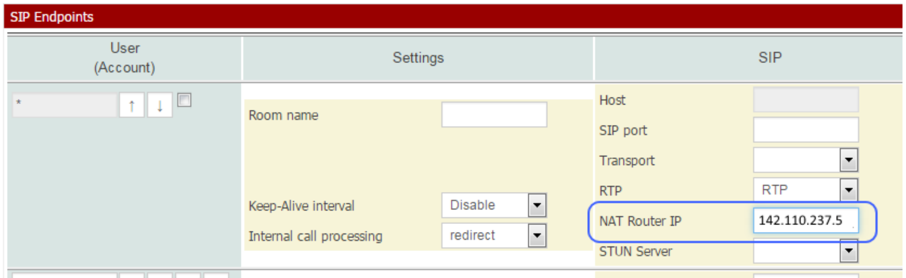


This will force all participants to connect to an existing conference room and prevent a denial of service attack by flooding OpenMCU-ru with requests to auto create conference rooms.

7. Set the time limit for conferences to 7200 seconds (2 hours) to prevent conferences from going on forever.



8. Set the public/WAN IP address: go to OpenMCU-ru – Settings – SIP – Endpoints, set NAT Router IP.



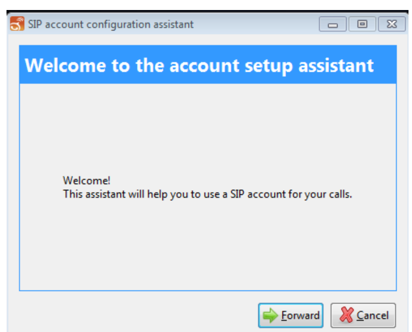
9. Configuring a SIP video softphone client

The free Linphone SIP softphone (http://www.linphone.org) is a good tool for testing OpenMCU-ru. It uses the H.264 codec which provides the best video quality and it allows you to connect directly to the conference room without needing to register with a PBX.

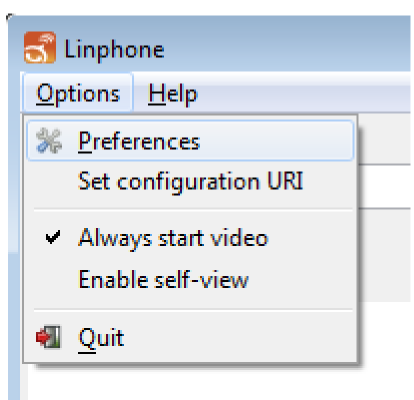
Most every other softphone requires that you register first with a PBX then dial through the PBX to get to OpenMCU-ru. Which means that you have to have routes configured between the PBX and OpenMCU-ru to test. This makes testing very difficult – use a Linphone and make your life easy. Later, we'll configure integration with the FreePBX (Asterisk based PBX) and go down that road.

1. Install Linphone on a PC or laptop with a web cam. Install the Cisco H.264 codec as it is the preferred codec for video.

2. When Linphone first starts up, the Account Setup Assistant will pop up. Just hit cancel for now.

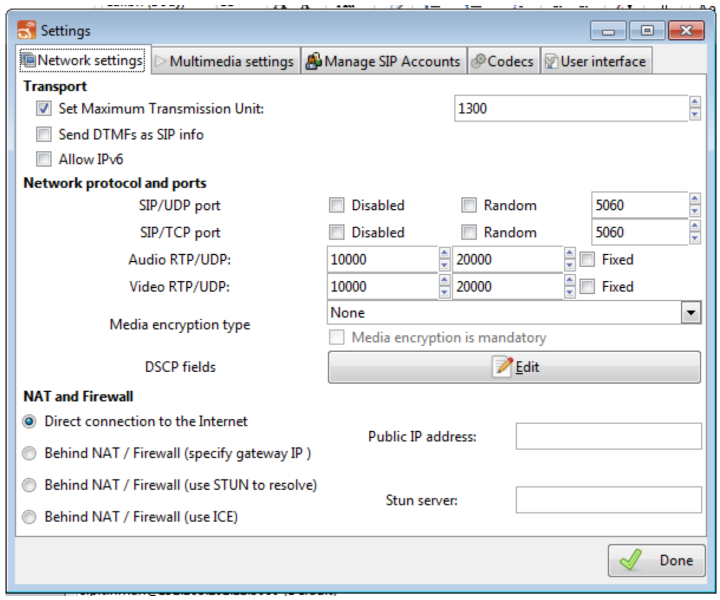


3. Go to Options:



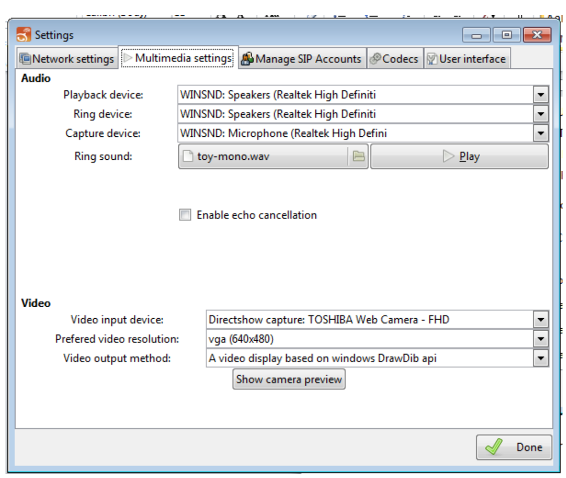
4. Make sure that “Always start video” is checked.

5. Go to Options – Preferences – Network settings:



6. Set the following parameters. These will be needed later for FreePBX (Asterisk) interoperability:
   - SIP UDP port: 5060 (SIP default standard port)
   - SIP TCP port: 5061 (SIPs - SIP secure port)
   - Audio RTP/UDP: 10000 – 20000
   - Video RTP/UDP 10000 – 20000

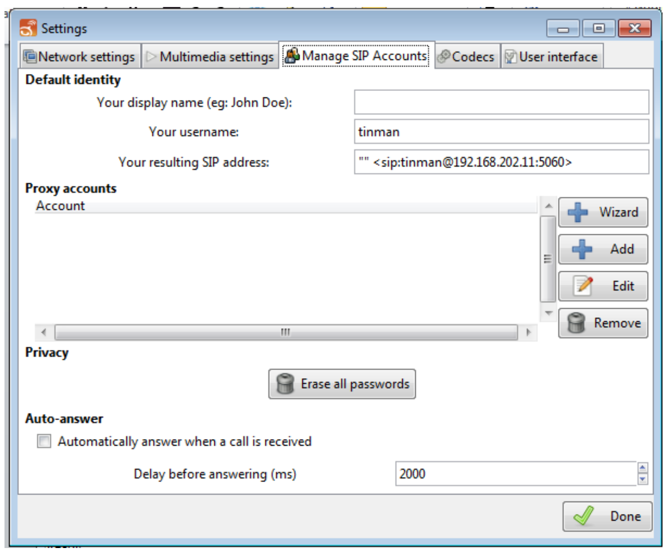
7. Next go to the Multimedia tab:



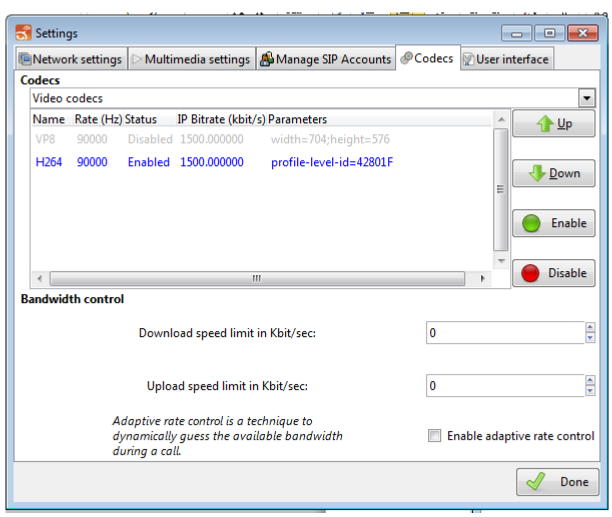
8. Check that the Audio devices are selected and work properly.

9. Select your Web Camera and a "Prefered video resolution" of "svga 800x600".

10. Go to Manage SIP Accounts tab and change "Your username". The default username is "toto", for testing purposes, I named my users after characters from the movie "The Wizard of Oz":



11. Go to the Codecs tab and disable the VP8 codec. The quality is very poor compared to the H.264 and the resolution is restricted to a small window.



12. There is the option to "Enable adaptive rate control" but I haven't evaluated it to make a decision whether it is a good thing or not.

10. Configuring a H.323 video softphone client

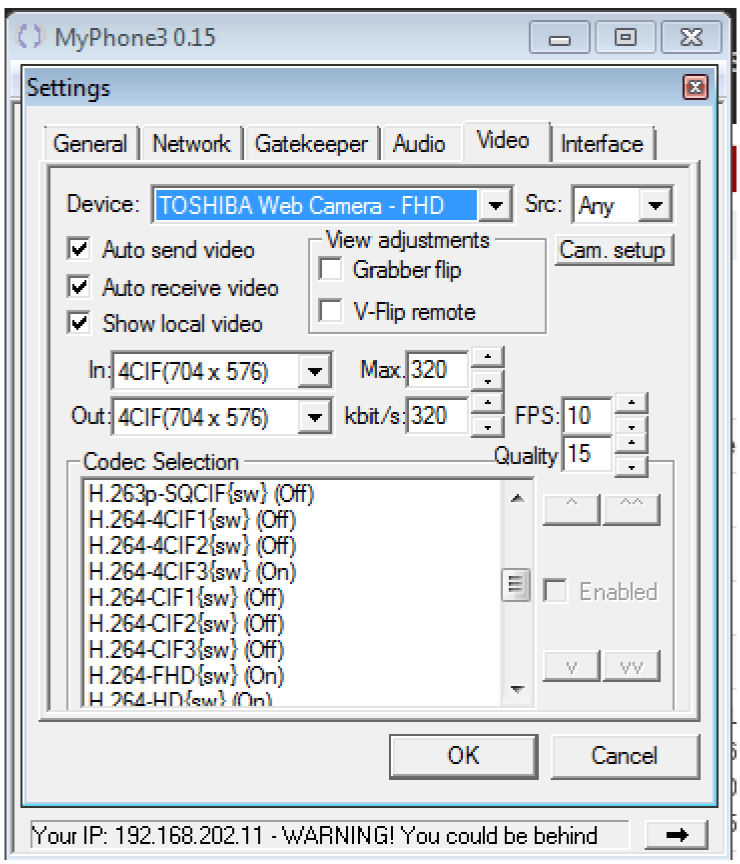
The free MyPhone3 h.323 softphone (http://openmcu.ru/wiki/en/terminals/myphone) is a good tool for testing OpenMCU-ru. It can use the H.264 codec which provides the best video quality and it allows you to connect directly to the conference room without needing to register with a PBX.

The MyPhone3 has a lot of video codecs and display formats available for testing and checking video quality with the Status window of OpenMCU-ru (more on this later).

1. After installing the MyPhone3, go to Settings – Video and configure for the best video quality:

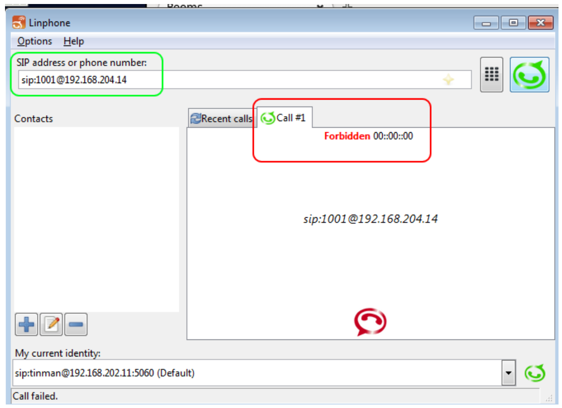
    - Set the In and Out Video for 4CIF (704x576). This will need to be fine-tuned to get the best video resolution for full screen.

    - Disable all video codecs except the H.264-4CIF3 codec for now.



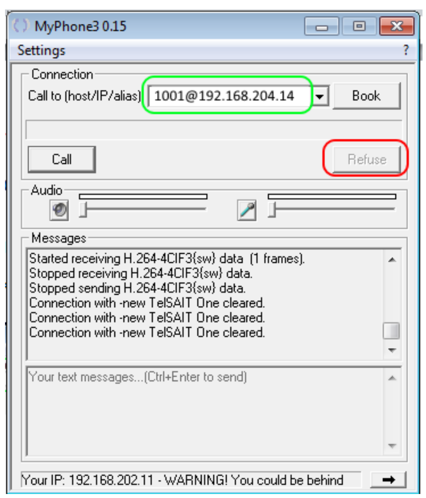
11. Verifying *disabled* auto conference room creation

1. From the SIP Linphone, dial 1001@<ipaddress of MCU>, see **green** highlight:



2. You should receive a Forbidden error message (highlighted in **red**). This tells us that auto creation of conference rooms is ***disabled*** which is what we want for security purposes!

3. From the H.323 MyPhone3, dial 1001@@<ipaddress of MCU>, see **green** highlight:

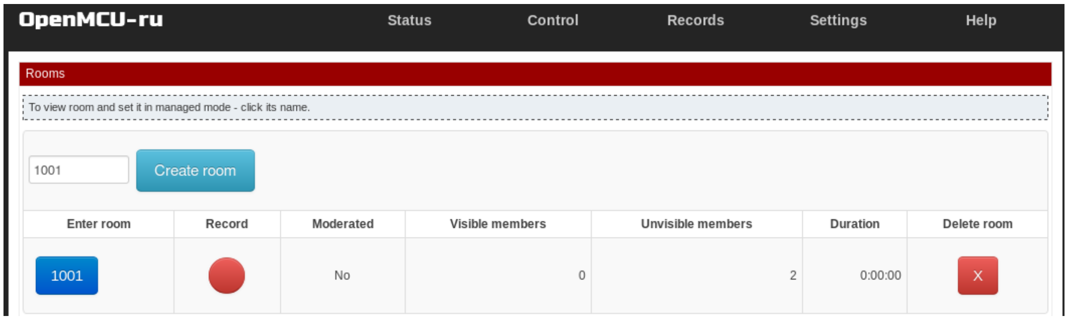


4. You should receive a Refuse error message (highlighted in **red**). This tells us that auto creation of conference rooms is disabled!

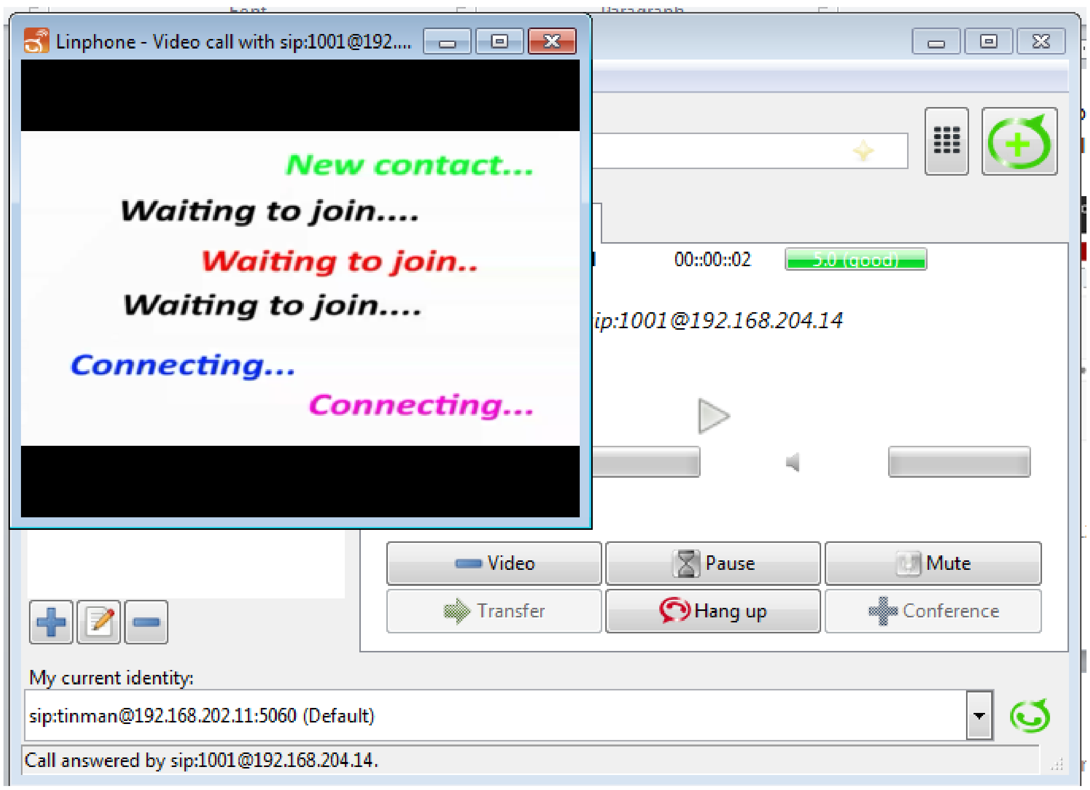
12. The first conference room

1. On OpenMCU-ru, create the conference room by going to the Control Option and selecting the Create Room button next to "1001". This will create conference room 1001.

Note: Room naming is not limited to just numbers. A valid room name, for example, is "TuesMeeting" or "Paris-London". In the following examples, room "1001" is used throughout. "Paris-London" can be substituted for "1001" and it would work fine.



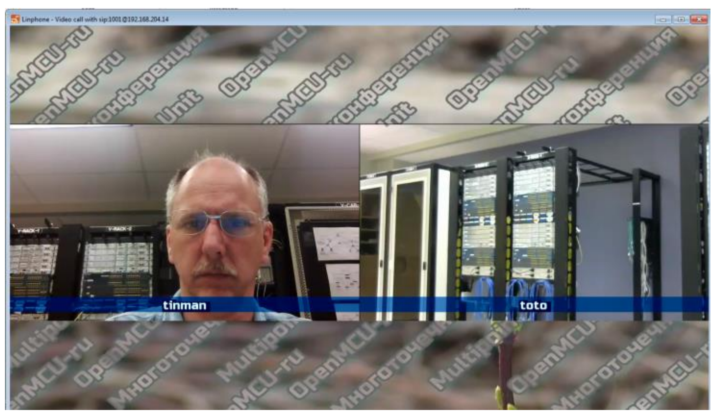
2. From the Linphone, now connect to sip:1001@<IP address of MCU>. You will see the Custom Logo Image momentarily flash on the screen while the Linphone (participant) joins the conference:



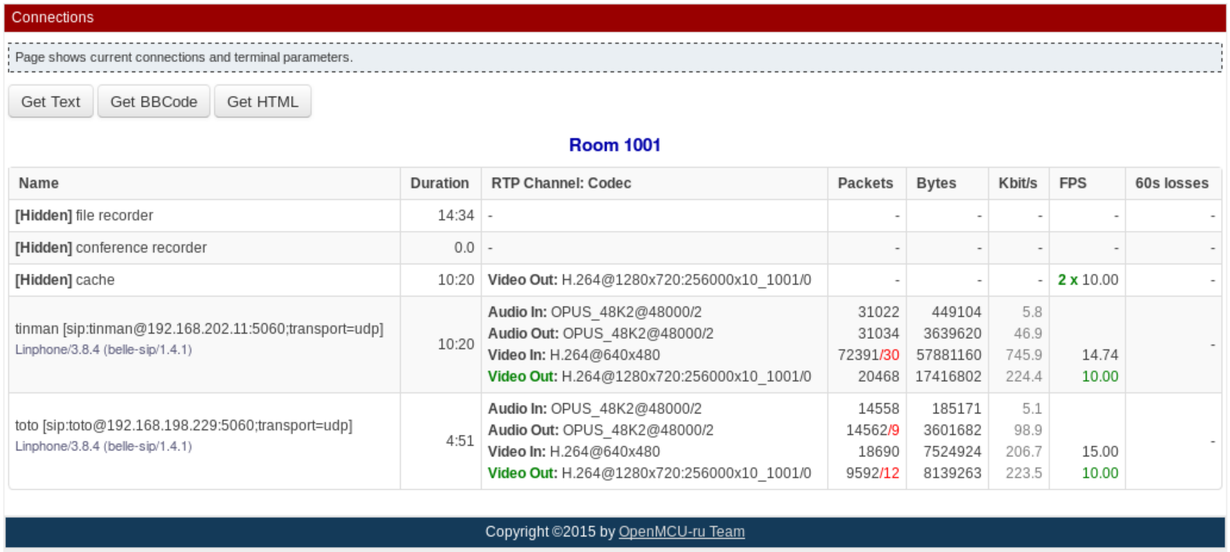
3. Then your web cam will be displayed:

4. It is difficult to see at this time but behind the conference participant's name are two Linphone options: Full Screen and Hangup. Full Screen sets the video image to full screen and Hangup ends the conference call for this participant.

5. When another participant calls in to 1001@<IP address of MCU>, you will see the Custom Image Logo while they are initially joining the conference. OpenMCU-ru is operating in ***Auto*** mode at this time and will automatically reconfigure the screen for the number of participants.



6. Monitor the Status of the conference room by going to the OpenMCU-ru Status Option:


Connections

Page shows current connections and terminal parameters.

Get Text | Get BBCode | Get HTML

Room 1001

Name	Duration	RTP Channel: Codec	Packets	Bytes	Kbit/s	FPS	60s losses
[Hidden] file recorder	14:34	-	-	-	-	-	-
[Hidden] conference recorder	0.0	-	-	-	-	-	-
[Hidden] cache	10:20	Video Out: H.264@1280x720:256000x10_1001/0	-	-	-	2 x 10.00	-
tinman [sip:tinman@192.168.202.11:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	10:20	Audio In: OPUS_48K2@48000/2 Audio Out: OPUS_48K2@48000/2 Video In: H.264@640x480 Video Out: H.264@1280x720:256000x10_1001/0	31022 31034 72391/30 20468	449104 3639620 57881160 17416802	5.8 46.9 745.9 224.4	14.74 10.00	-
toto [sip:toto@192.168.198.229:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	4:51	Audio In: OPUS_48K2@48000/2 Audio Out: OPUS_48K2@48000/2 Video In: H.264@640x480 Video Out: H.264@1280x720:256000x10_1001/0	14558 14562/9 18690 9592/12	185171 3601682 7524924 8139263	5.1 98.9 206.7 223.5	15.00 10.00	-

Copyright ©2015 by OpenMCU-ru Team

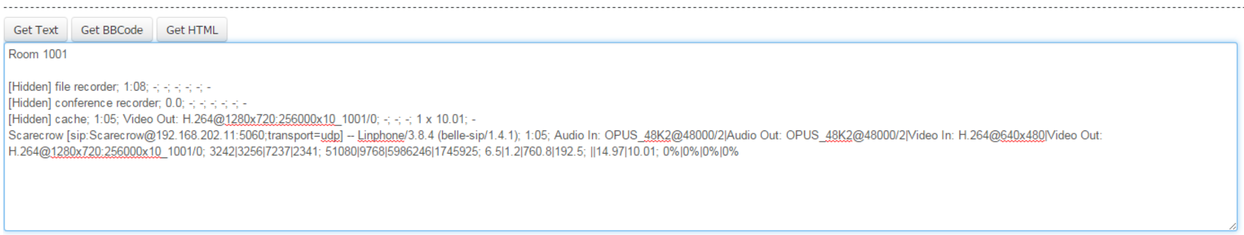
This shows:

- audio (OPUS) and video (H.264) codecs and resolution (640x480) that the participants are using.
- the Video Out codec (H.264) and resolution (1280x720).
- the Duration connected
- bandwidth used (Packets, Bytes and Kbit/s)
- frames per second (FPS) rate.

The Status screen is a ***very useful*** troubleshooting tool to see what codecs are being used, the bandwidth that each stream is taking and if audio/video is being received (In) and transmitted (out) for each participant.

Get Text

This provides a text based record of the conference for documentation purposes:

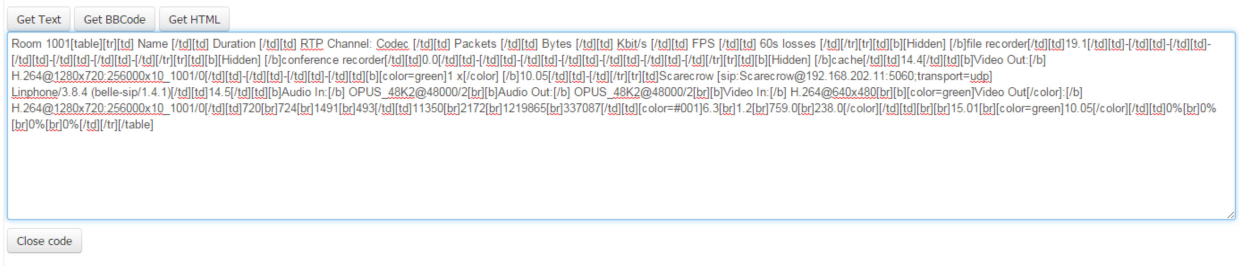

Get Text | Get BBCode | Get HTML

Room 1001

[Hidden] file recorder; 1:08; -; -; -; -; -; -
[Hidden] conference recorder; 0.0; -; -; -; -; -; -
[Hidden] cache; 1:05; Video Out: H.264@1280x720:256000x10_1001/0; -; -; -; 1 x 10.01; -
Scarecrow [sip:Scarecrow@192.168.202.11:5060;transport=udp] -- Linphone/3.8.4 (belle-sip/1.4.1); 1:05; Audio In: OPUS_48K2@48000/2|Audio Out: OPUS_48K2@48000/2|Video In: H.264@640x480|Video Out: H.264@1280x720:256000x10_1001/0; 3242|3256|7237|2341; 51080|9768|5986246|1745925; 6.5|1.2|760.8|192.5; ||14.97|10.01; 0%|0%|0%|0%

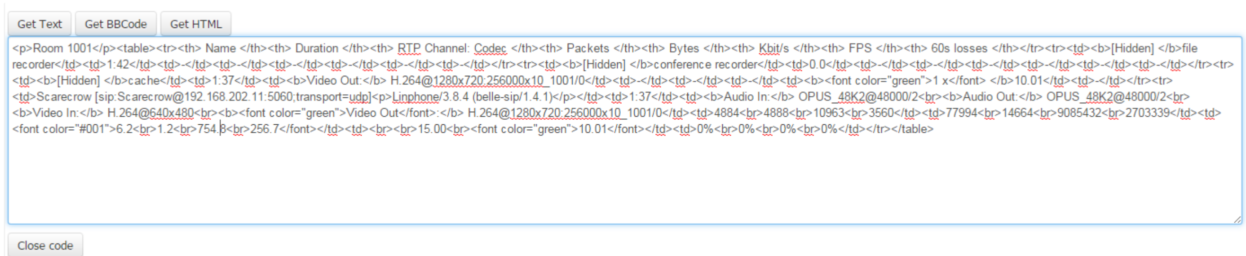
Get BBCode

This provides Bulletin Board code record of the conference for documentation purposes:



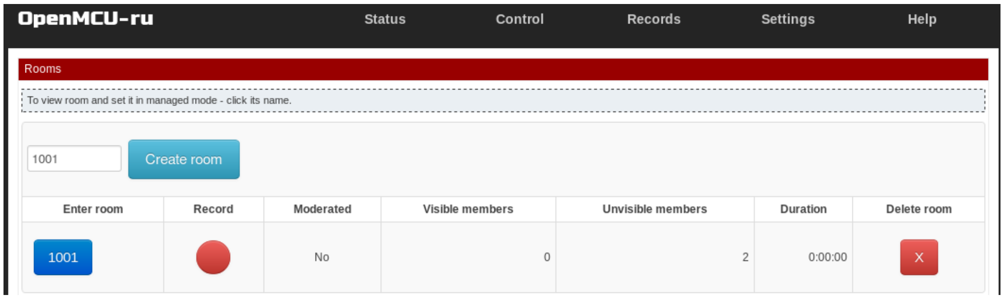
Get HTML

This provides HTML code record of the conference for documentation purposes:

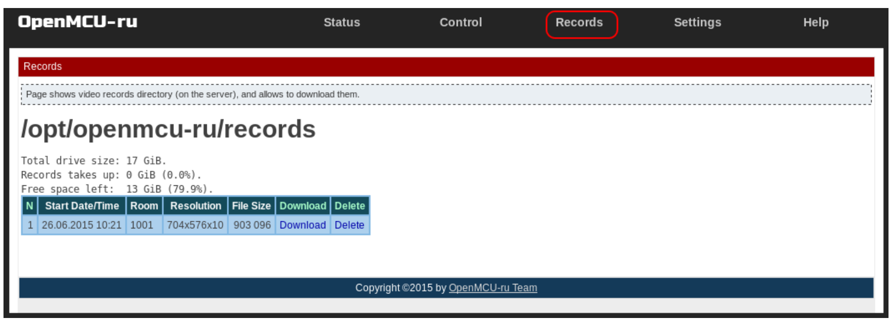


13. Recording a Conference

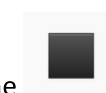
1. To record a conference, go to the Control option and press the Record button:

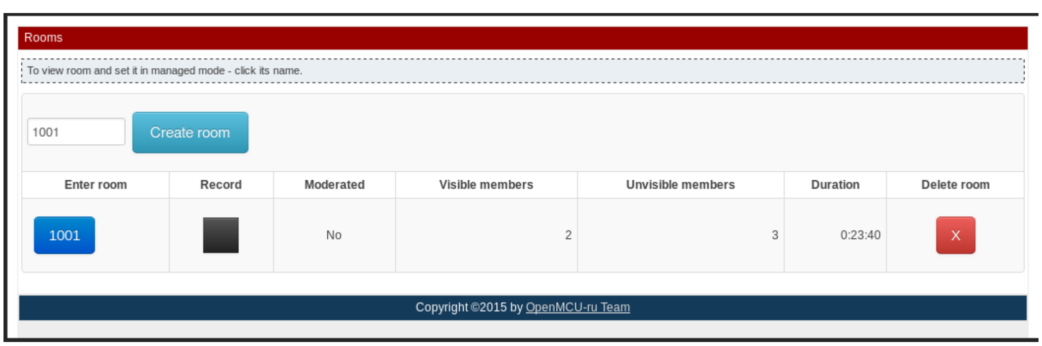


2. To monitor the recording, go to the Records Option:



You can download or delete the recording from this window. When you delete a recording, you will be prompted to verify the deletion.

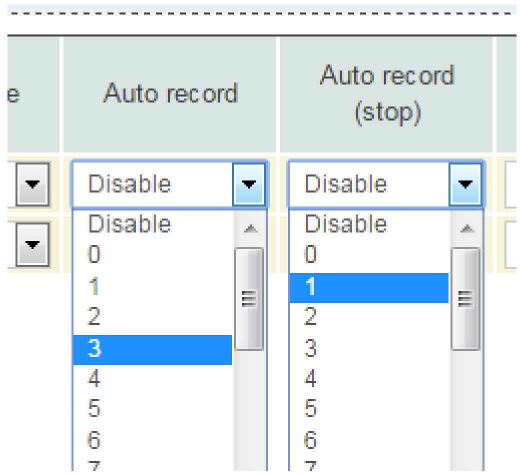
3. To stop recording, go back to the Control Option and press the  button.



4. The conference is recorded in .asf format and can be played back using VLC media player.

5. NOTE: I was only able to record a conference room's #0 display and not #1. So if you have a multi-display conference room be aware.

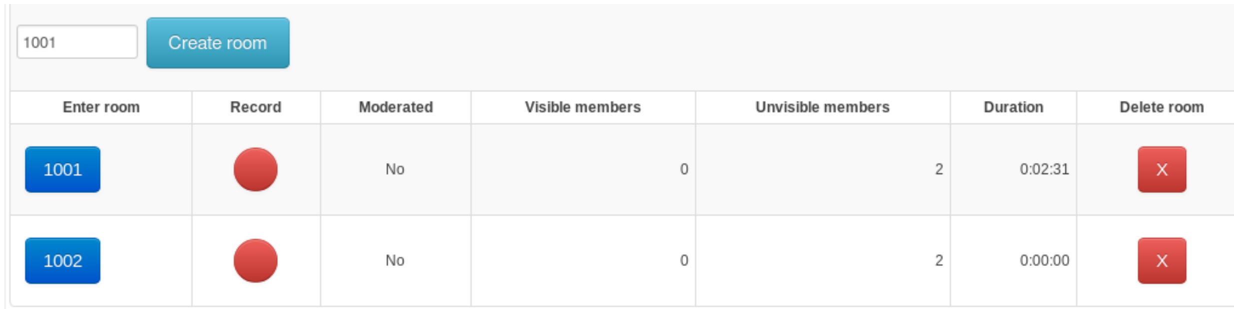
6. A conference can be auto recorded when started and the settings are configured from the Settings – Advanced – Conference menu:



- Auto record – conference room is automatically recorded when started. It can be fine-tuned to start recording when a specific number of participants enter the room.
- Auto record (stop) – recording is automatically stopped. It can be fine-tuned to stop recording when a specific number of participants leave/remain in the room.

14. Creating and deleting a Conference Room

1. To create room 1002, type 1002 in the textbox and select “Create Room”. A second conference room named 1002 will be created.



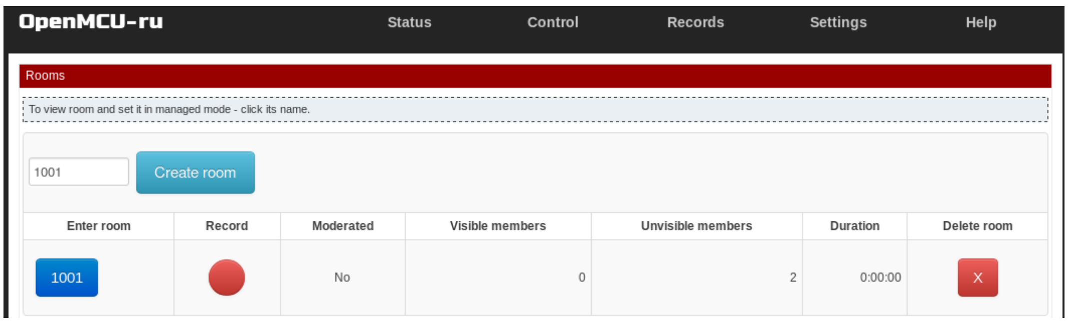
2. Participants will join the new room 1002 by dialing “1002@<IP address of MCU>. Room 1002 will run separately from room 1001.
3. To delete a room, press the delete room button 

, any participants who are connected will be disconnected.

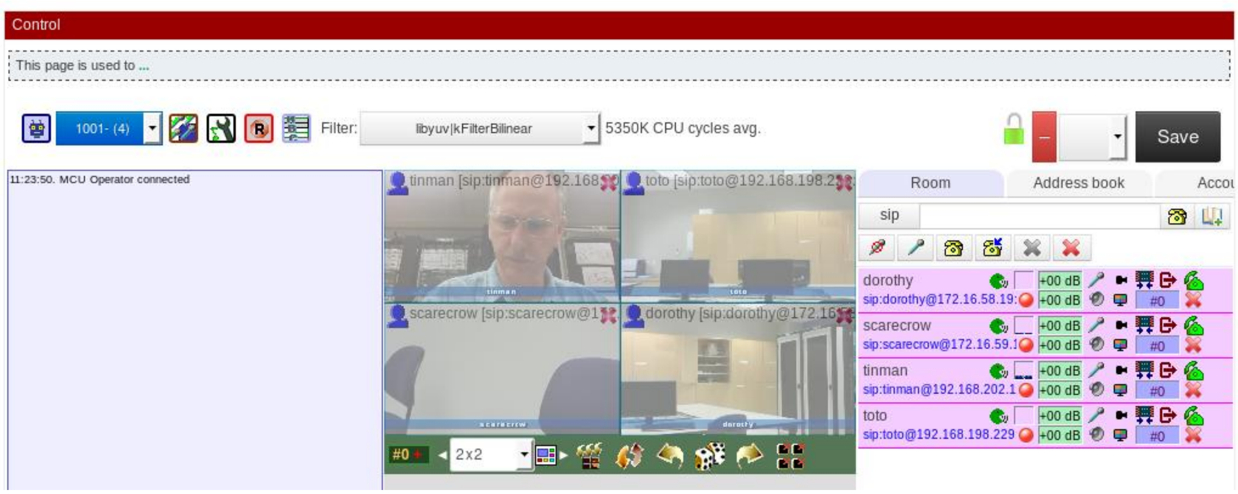
15. Taking Control of a Conference Room

Taking control of a conference room is the term used when the room is changed from automatic configuration mode to ***manual*** configuration mode. All parameters of the conference room are controlled by the Administrator. Ex. Placing participants into the window positions.

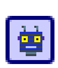
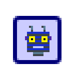
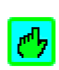
1. Go to the Control Option and select the room button 



2. This will open up room 1001 control:



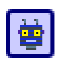
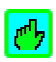
3. Hovering over any of the icons or buttons will bring up a help box for that item.

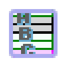
4. The  button is the ***Take Control*** button and is used to switch between auto mode  and manual configuration mode . In order to change the configuration of the conference room, you must take control of it. When you take control, you become the Operator.

5. If you attempt to change the configuration while in the auto mode, an error message will flash at the top of the screen:

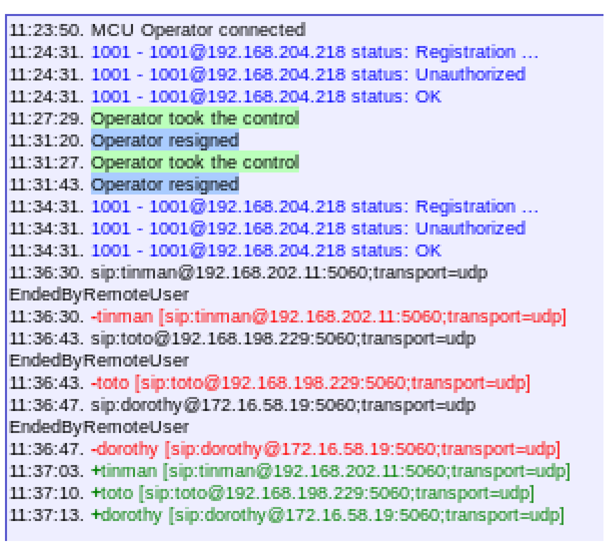
6. **NOTE:** When in Take Control mode, any participants that join the conference room will be added to the list on the right side. They will not be automatically placed in the conference room display. You will have to drag and drop each participant to a position in the display.

7. The button functions on the upper left side are as follows:

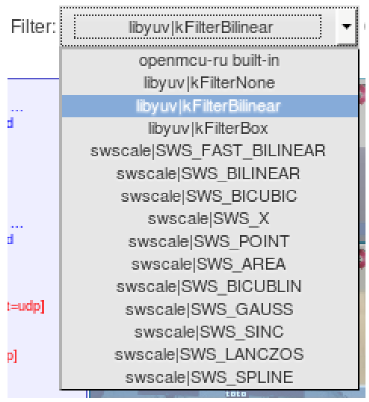
-  /  Take Control/Auto mode
-  Select room

  The (3) indicates # of participants
-  Mute/Unmute invisible members
-  VAD parameters (Voice Activity Detection)
-  Start/stop recording
-  Sort mode

8. Logging screen shows information concerning the conference rom:



9. Available video filtering algorithms available:



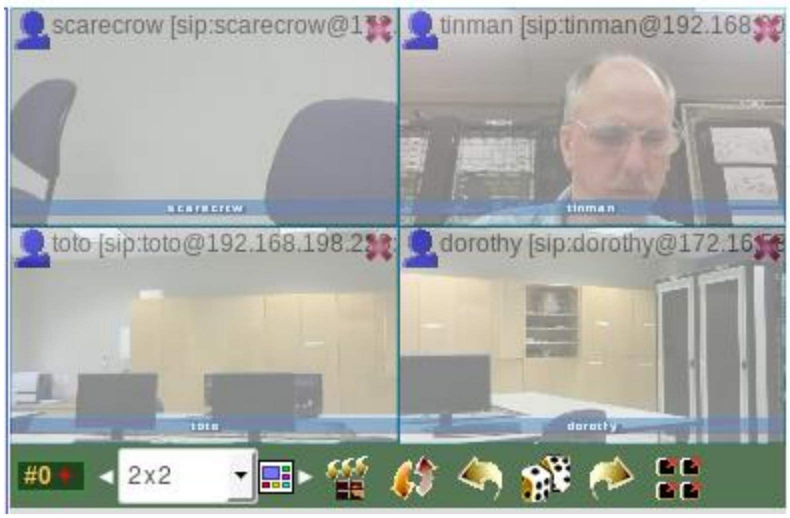
- **Openmcu-ru built-in**
- **Libyuv**
  An open source project that includes YUV scaling and conversion functionality.
  Scale YUV to prepare content for compression, with point, bilinear or box filter.
  Convert to YUV from webcam formats.
  Convert from YUV to formats for rendering/effects.
  Rotate by 90/180/270 degrees to adjust for mobile devices in portrait mode.
  Options filtering algorithms available:
    - kFilterNone
    - KfilterBox
- **Swscale**
  Used to display the video at a different pixel size/aspect ratio than it was encoded at when you don't have hardware video scaling support. Swscale also performs colorspace conversion between various RGB and YUV color formats, and conversion between packed (all channels in a single buffer) and planar (each channel has its own buffer) formats. All of these routines are highly optimized.

  Swscale used for for transcoding video, if the source video is not already in the format needed by the encoder. For instance, if your source video is RGB, you'll need to convert it to the appropriate YUV planar format, since most codecs work on YUV. This entails both colorspace conversion (an affine transformation of the R,G,B vectors) and actual scaling (resampling), since most YUV formats use half-resolution U and V planes (color planes) compared to the Y plane (luma, i.e. intensity data).
  Optional filtering algorithms available:
    - SWS_FAST_BILINEAR
    - SWS_BILINEAR
    - SWS_BICUBIC

- ○ SWS_X
- ○ SWS_POINT
- ○ SWS_AREA
- ○ SWS_BICUBIN
- ○ SWS_GAUSS
- ○ SWS_SNC
- ○ SWS_LANCZOS
- ○ SWS_SPLINE

10. Average CPU load is displayed: 9634K CPU cycles avg.

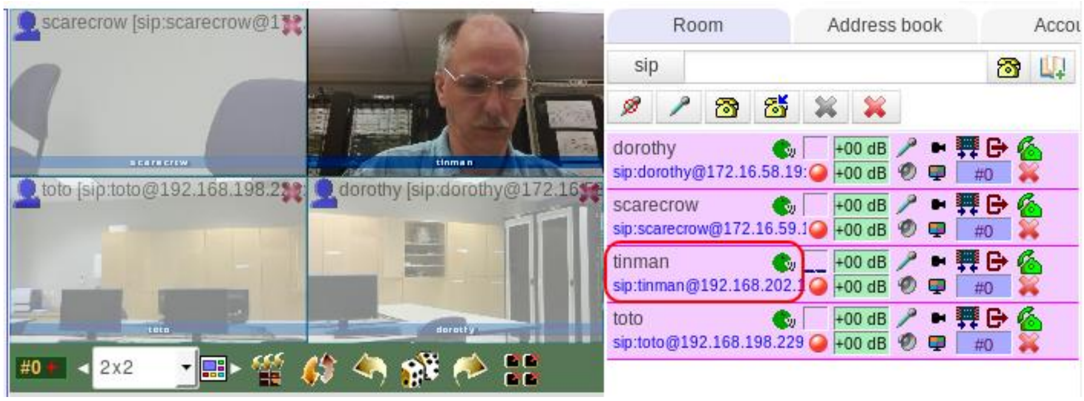
11. The Display will show the current state of the conference room and participants.



12. To remove a participant from a display position, click on the red **X**  in the corner of their video.

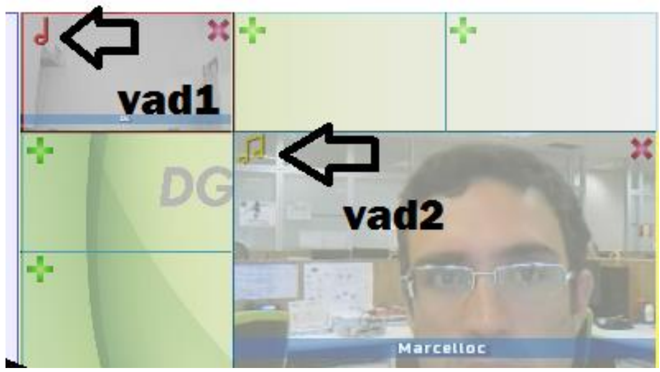
13. You can drag and drop participants onto displays or move them from one position to another.

14. Hovering over any of the participants highlighted in pink on the right side will show their position on the display. I hovered over the tinman (highlighted in **red**) and his video becomes highlighted:



15. At the top of each participant's ***position*** is the  information icon. It lists the username and URI. Clicking on the  icon cycles through the following options:

-  - participant name: the MCU places the participant at this specific location
-  - VAD: this position can be taken by a Voice Activated stream
-  - VAD2: this position takes priority for whoever is talking the most. It makes sense to use on a layout where a large screen (position 0) is surrounded with smaller screens.



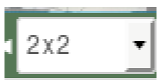
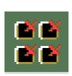
- The VAD2 position 0 will be taken by the most active endpoints. This activity is monitored by the amount of time that the sound is active.

In Take Control mode, the participants can be assigned as

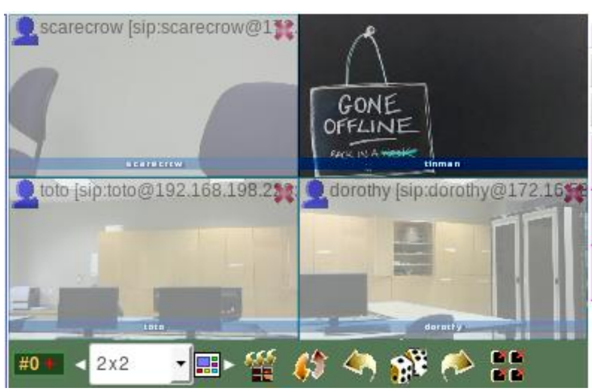
-  - no VAD: not allowed to talk
-  - VAD: can talk
-  - VAD2: King mode, has priority over everyone else

16. In Take Control mode, participants are not automatically added to the conference. The conference operator has to assign participants to a display position by dragging and dropping participants.

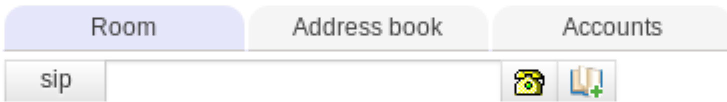
17. These options available at the bottom of the room display are described:

-  Identifies display number and allows you to add or delete a display
-  Pull-down menu that allows you to change the display matrix
-  Displays a pop-up window showing matrix choices (see section 21)
-  Add all participants listed in pink to the display
-  Rotate participants diagonally across display matrix
-  Rotate participants position right to left, bottom to top direction
-  Randomly position participants in display
-  Rotate participants position left to right , top to bottom direction
-  Kick participants out of display

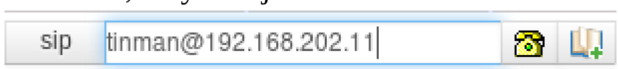
18. Indication that a participant has disconnected shows Gone Offline:



19. Participants can join the conference room by this section:

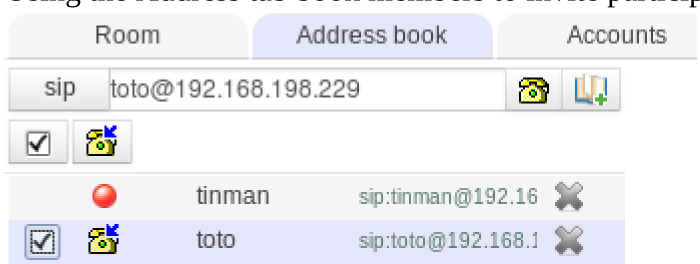


- Use the Room tab to manually invite participants by typing their URI and selecting the dial button . The participant's video phone will ring and when answered, they will join the room :



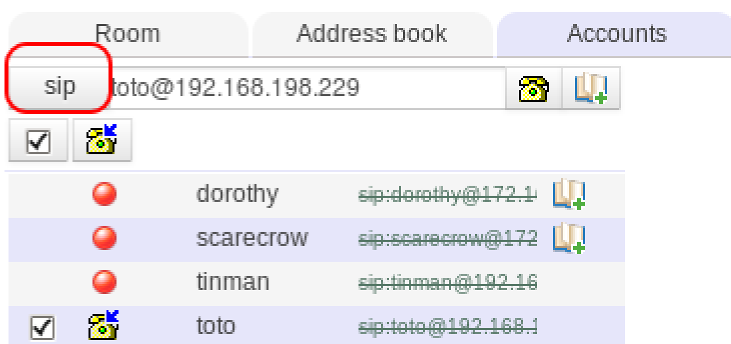
Tip! Before dialing, add the participant to the Address book by clicking on

- Using the Address tab book members to invite participants:



You can invite all address members or individual by using the checkboxes.

- The Account tab allows you to add existing participants to the Address book and invite them to the room:

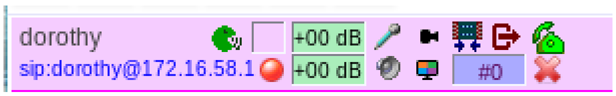


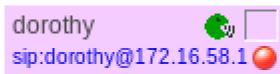
- Three protocols are allowed SIP, RTSP and H323. You can select by pushing on the highlighted button shown above.

20. The buttons functions on the right side of the screen are as follows:

- Mute all participants mics
- Unmute all participants mics
- Invite all active members
- Run continuous dialing all members
- Remove all inactive members
- Drop all active connections

21. Participant parameters:

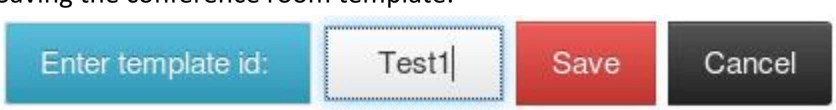


- 

identifies the user Dorothy and their URI
- indicates that the participant is active
- controls the Voice Activity Detection:
    - indicates enabled (default)
    - disabled
    - Chosen Van: will take VAD position permanently even if no voice comes from them.
- is an audio volume level meter
- 

allows you to adjust the audio input (mic) level and the audio output (speaker) level for each participant if needed.
- mutes the video of this participant to other participants. The other participants in the room see a screen capture (picture) instead of a live video feed. This would be used when there is a video problem with a participant
- mutes the room's video to this particular participant and replaces it with a black screen. Does not affect other participants.
- toggles the orientation of the video from horizontal to vertical
- removes users from video mixer
- #0 assigns which display the participant will view
- invites participant to room by dialing their URI
- indicates that the participant is connected to the room
- disconnects the participant from room
- removes participant from room list

22. Saving the conference room template:



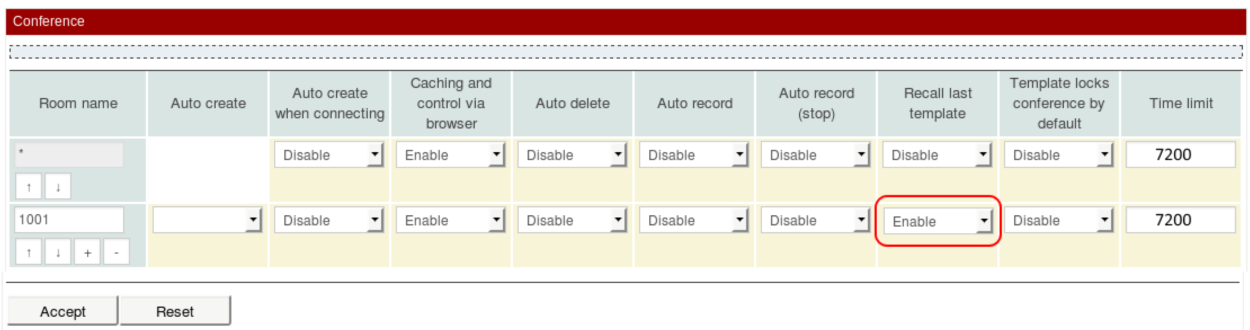
After saving, the template will appear as a pull-down option for other rooms:

Selecting the room template will bring up the address book and participants.

23. The lock icon  is used to lock the templates. Once locked no changes are allowed to the template. Templates can be locked by default when a room is first created by going to Settings – Advanced – Conferences and setting “Template locks conference by default” to Enabled.

    Template locks conference by default – conference follows previous template. No new participants or changes to conference room allowed.

24. To recall a previously saved template for a permanently defined conference room, you must enable “Recall last template”:



Conference

Room name	Auto create	Auto create when connecting	Caching and control via browser	Auto delete	Auto record	Auto record (stop)	Recall last template	Template locks conference by default	Time limit
* ↑ ↓		Disable	Enable	Disable	Disable	Disable	Disable	Disable	7200
1001 ↑ ↓ + -		Disable	Enable	Disable	Disable	Disable	Enable	Disable	7200

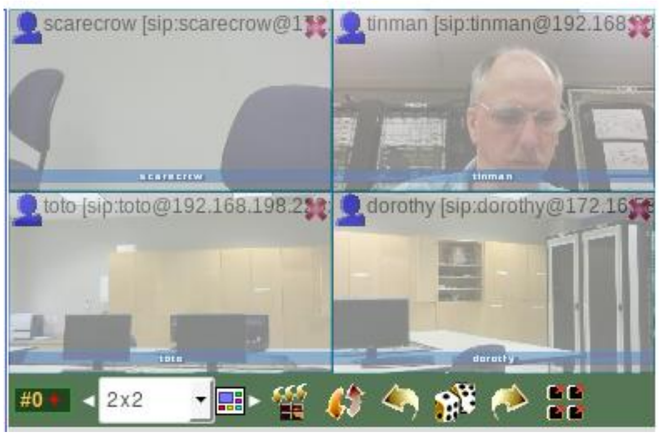
Accept Reset

16. Creating a multiple display conference room

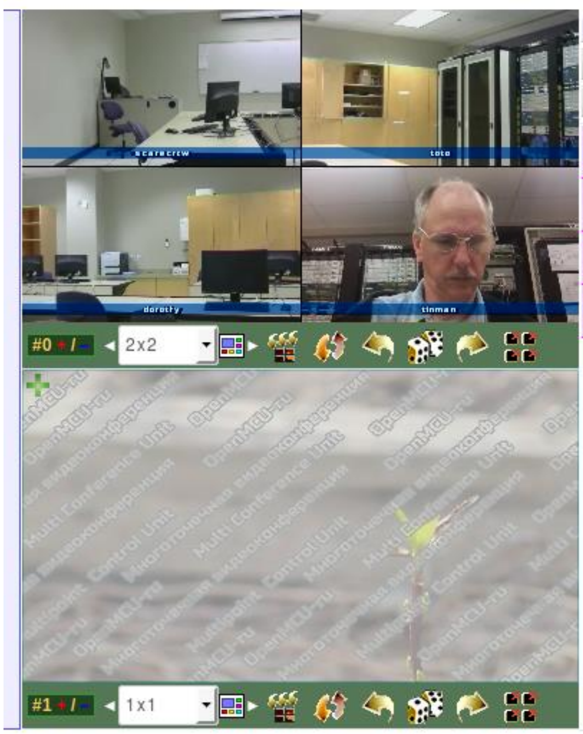
When creating a conference room where the participants are in two separate locations, often each location only wants to see the other location. They don't want to see themselves on screen. OpenMCU-ru allows the creation of additional displays that participants can be assigned to.

1. Go to the Control Option and select the conference room. Select the 

 button to create a second display

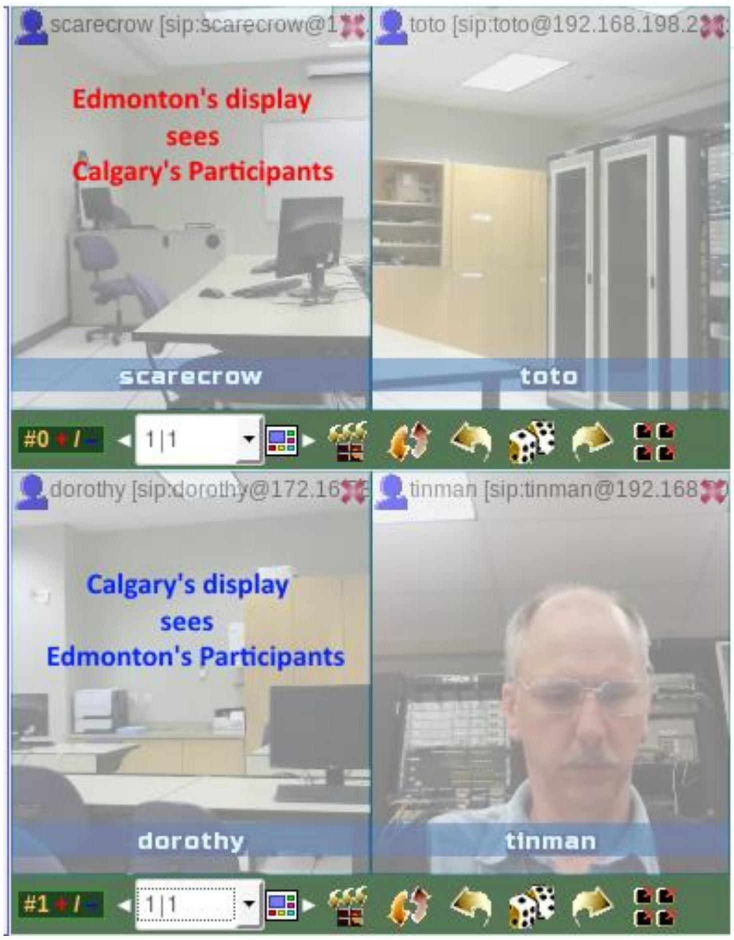


2. The window will change to show the new display #1 with no participants:

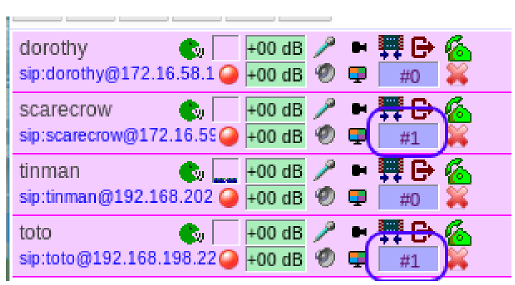


In this example, Scarecrow and Toto are located in Calgary and Dorothy and Tinman are located in Edmonton. The Edmonton participants only want to see Scarecrow and Toto who are located in Calgary. While the Calgary participants only want to see Dorothy and Tinman who are located in Edmonton.

3. Change the display matrix on both displays to 1|1 setting. This will show each city's participants side by side. Drag and drop the Calgary participants into the top display and the Edmonton participants into the bottom display:



4. Change the display that each participant sees so that they see the other city's participants. Go to their account and click on the display # to change:



Each city will now see the other city's participants.

5. The Status Window will now show two video out streams each going to 2 participants:

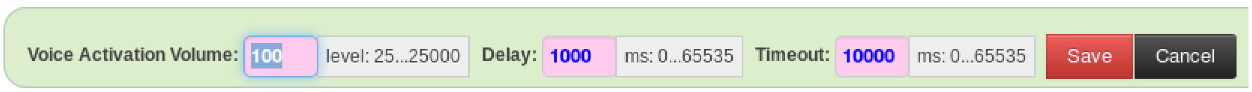
[Hidden] cache	54:53	**Video Out:** H.264@1280x720:256000x10_1001/0	-	-	-	**2 x** 10.00	-
[Hidden] cache	29:39	**Video Out:** H.264@1280x720:256000x10_1001/1	-	-	-	**2 x** 10.00	-

6. When recording conferences, only Display #0 is recorded

17. Voice Activity Detection (VAD)

Voice activity detection is used to reduce the amount of background noise. The audio input (mic level) from a participant is muted until a present level (threshold) is reached before the mic is turned on. This reduces the background noise that the participants hear.

The voice activity detection parameters are set separately for each conference room and are accessed after taking control of the room. When the VAD button  is pressed the following options appear:



The parameters that can adjusted are:

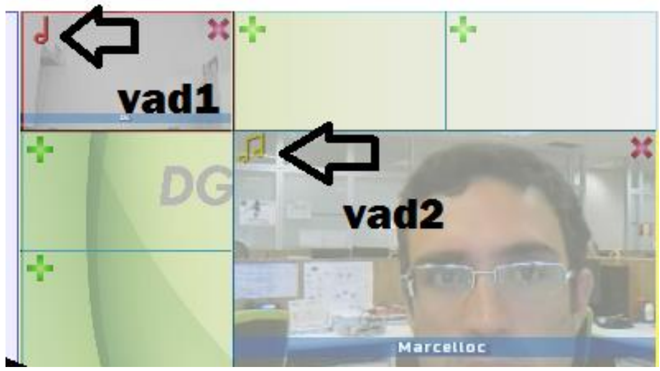
- Voice Activation Volume:
  - Range: 25-25000, default: 100
  - The minimum level of sound at which the MCU believes that there is voice activity. It is necessary to cut off the influence of background noise.
- Delay:
  - Range: 0 – 65535 ms, default: 1000 (1 sec)
  - This is the time delay in milliseconds before voice activation will be active. This is necessary to prevent activation at short interference (knocking, coughing, etc.).

- Timeout:
  - Range: 0-65535 ms, default 10000 (10 sec)
  - Once a participant is activated by voice, this is the time (in milliseconds) that no voice was heard before the participant is removed from the screen.

The default settings appear to work quite well for the noisy test environment that I worked in.

1. At the top of each participant's ***position*** is the 

information icon. It lists the username and URI. Clicking on the  icon cycles through the following options:
   -  - participant name: the MCU places the participant at this specific location
   -  - VAD: this position can be taken by a Voice Activated stream
   -  - VAD2: this position takes priority for whoever is talking the most. It makes sense to use on a layout where a large screen (position 0) is surrounded with smaller screens.



- The VAD2 position 0 will be taken by the most active endpoints. This activity is monitored by the amount of time that the sound is active.

In Take Control mode, the participants can be assigned as

-  - no VAD: not allowed to talk
-  - VAD: can talk
-  - VAD2: King mode, has priority over everyone else

18. Telnet to OpenMCU-ru Server

The Settings – Advanced – Telnet Server allows you to configure a telnet daemon so that you can telnet into the OpenMCU-ru software and manually create rooms through a command line. You will ***not*** normally be creating a room in this manner.

This is NOT the linux command line but a command line interface into the OpenMCU-ru server application. The Telnet port number is configurable and the default port number is:

Telnet port: 1423

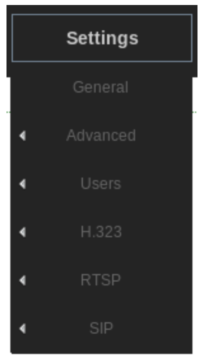
The available instructions are

```
# room 'name' ...
            create
            delete
            dial 'id'
            invite 'address'
            drop 'id'
            show members
            start_recorder
            stop_recorder
       "ctrl c" to exit
```

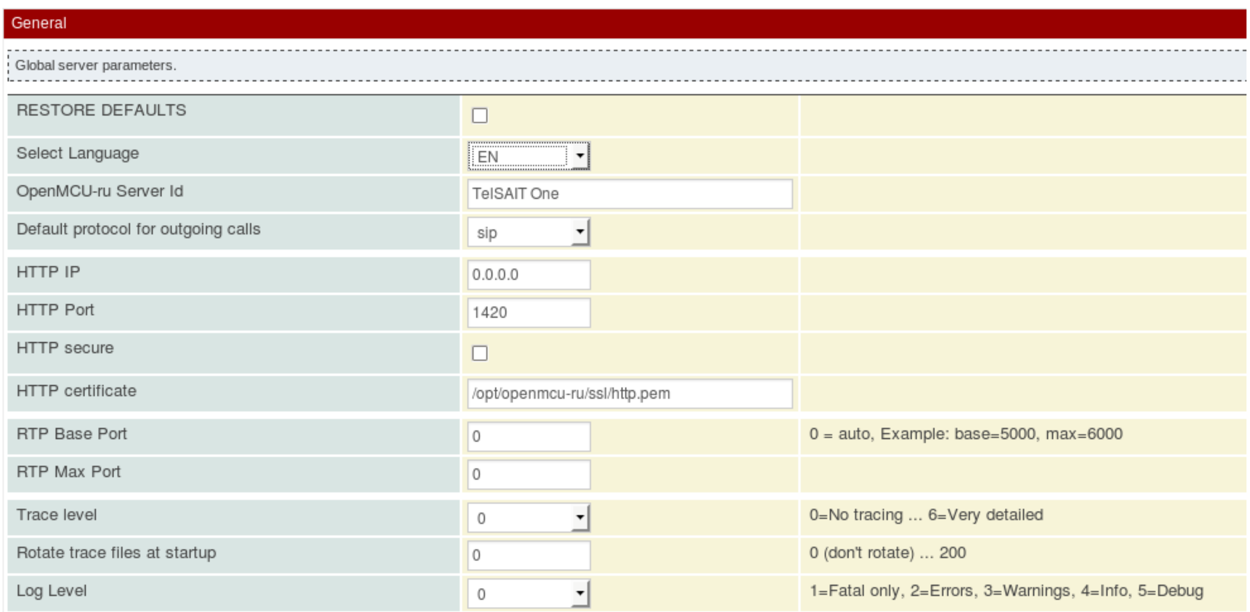
19. Settings

OpenMCU-ru provides complete control of the MCU with the option for the administrator to fine tune the MCU to their specific needs:

1. Hovering over the Settings Option, displays the submenus:



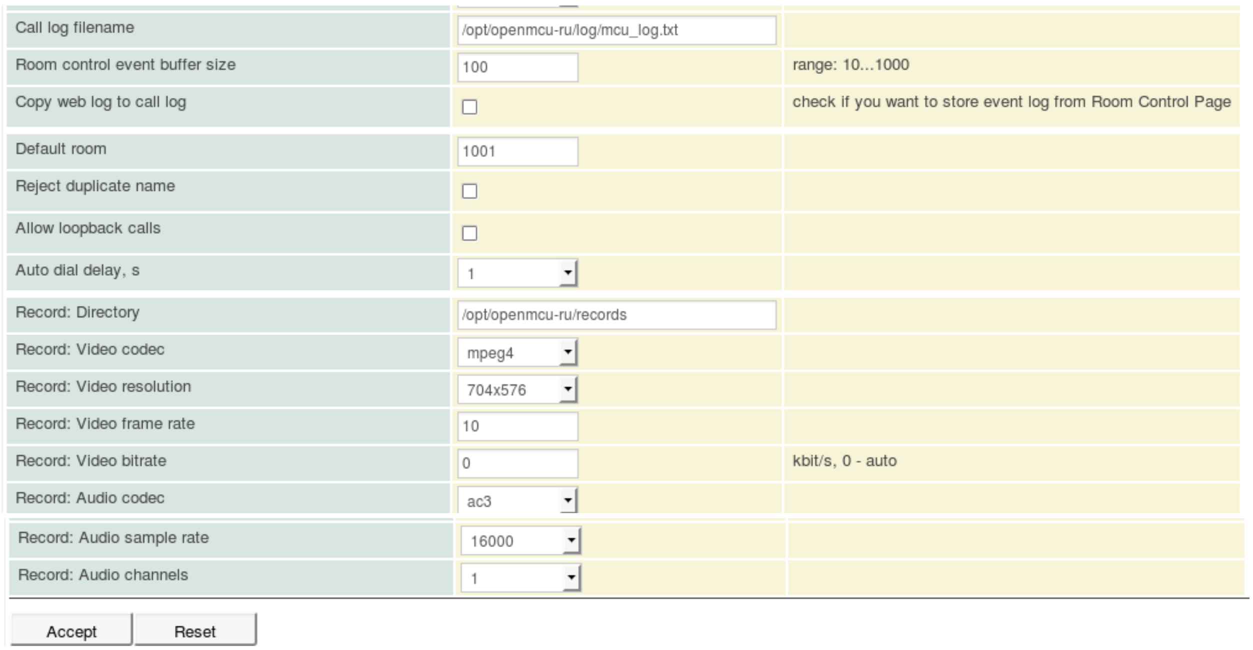
Settings – General



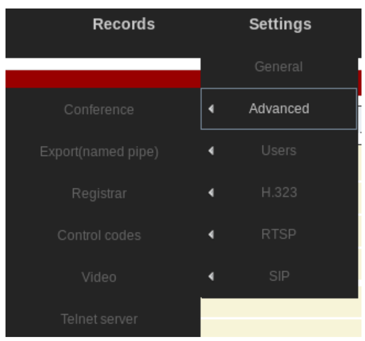
General

Global server parameters.

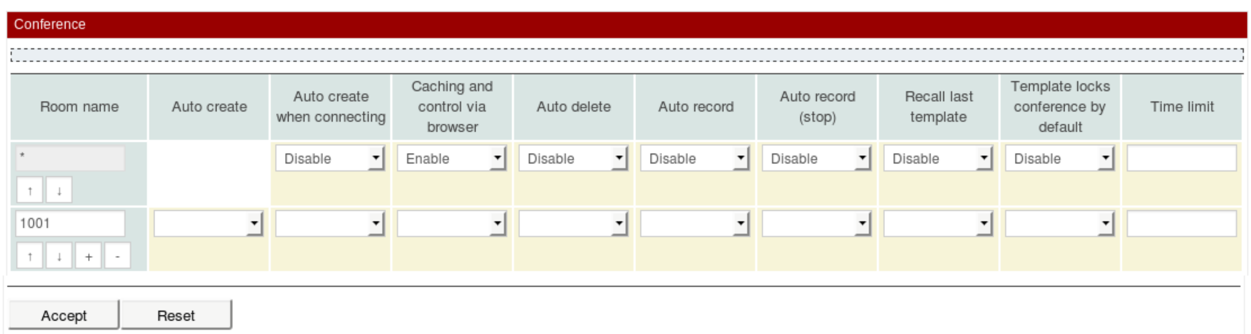
RESTORE DEFAULTS	☐	
Select Language	EN	
OpenMCU-ru Server Id	TelSAIT One	
Default protocol for outgoing calls	sip	
HTTP IP	0.0.0.0	
HTTP Port	1420	
HTTP secure	☐	
HTTP certificate	/opt/openmcu-ru/ssl/http.pem	
RTP Base Port	0	0 = auto, Example: base=5000, max=6000
RTP Max Port	0	
Trace level	0	0=No tracing ... 6=Very detailed
Rotate trace files at startup	0	0 (don't rotate) ... 200
Log Level	0	1=Fatal only, 2=Errors, 3=Warnings, 4=Info, 5=Debug



Settings – Advanced



Advanced – Conference

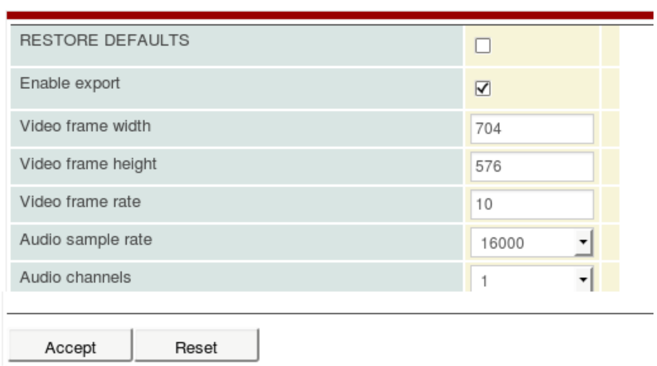


- “Auto create when connecting” – this allows users to dial into OpenMCU-ru server and auto create a conference. Should be disabled because it can permit a denial of service attack.

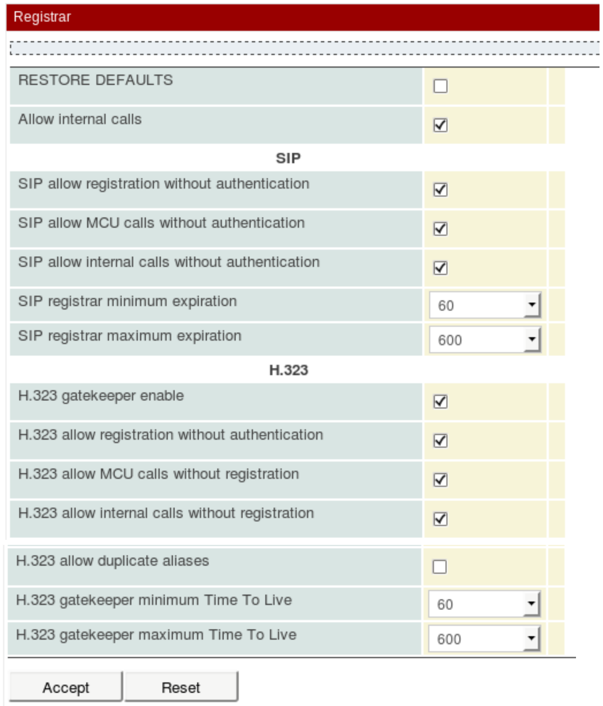
- "Caching and control via browser": Disable – creates classic mode when you see each other but don't see yourself. Every participant gets unique picture, it's impossible to use cached video stream, which creates quite a load on the server. This mode is not popular.
- Auto delete – conference room is automatically deleted when no participants
- Auto record – conference room is automatically recorded when started. It can be fine-tuned to start recording when a specific number of participants enter the room.
- Auto record (stop) – recording is automatically stopped. It can be fine-tuned to stop recording when a specific number of participants leave/remain in the room.
- Recall last template – loads a previously saved template of participants and displays for the room
- Template locks conference by default – conference follows previous template. No new participants or changes to conference room allowed.
- Time Limit: sets the time limit in ***seconds*** of the conference before it is stopped. Default is unlimited. "Auto delete" does not have to be set.

Advanced – Export (named pipe)

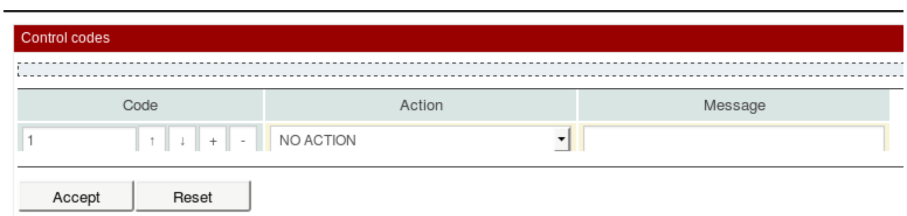
This is used to allow web streaming of conference rooms.



Advanced – Registrar



Advanced – Control Codes



Advanced – Video: Outgoing Video Quality

Video

Outgoing video quality.

RESTORE DEFAULTS	☐	
Enable video	☑	
Video frame rate	10	range: 1..999 (for outgoing video)
		H.263
H.263 Max bit rate		range 64..25000 kbit (for outgoing video, 0 disable)
H.263 Tx key frame period		range 0..600 (for outgoing video, the number of pictures in a group of pictures, or 0 for intra_only)
		H.263p
H.263p Max bit rate		range 64..25000 kbit (for outgoing video, 0 disable)
H.263p Tx key frame period		range 0..600, default 125 (for outgoing video, the number of pictures in a group of pictures, or 0 for intra_only)
		H.264
H.264 Max bit rate		range 64..25000 kbit (for outgoing video, 0 disable)
H.264 Encoding threads		range 0..64 (0 auto)
		VP8
VP8 Max bit rate		range 64..25000 kbit (for outgoing video, 0 disable)
VP8 Encoding threads		range 0..64 (0 default)
VP8 Encoding CPU used		range: 0..16 (Values greater than 0 will increase encoder speed at the expense of quality)

Accept | Reset

- Restore defaults: when checked restores default settings to Advanced Video settings
- Enable Video: when checked, allows video conferencing. Unchecked only audio.
- Video frame rate: sets the video frame rate for outgoing video - default 10 frames per second (fps). Setting this higher will increase the video frame but might load the CPU which could affect video quality and increase the network bit rate. This is dependant on your hardware platform.

Max Bit Rate:

Maximum bit rate that the codec will use. This allows traffic shaping and prevent a video codec from hogging the bandwidth on a shared link.

Tx key frame period:

Also known as Intra-frame period. In video compression, only changes that occur from one frame to the next are stored in the data stream, in order to greatly reduce the amount of information that must be stored. Most video sources have only small changes in the image from one frame to the next. Whenever a drastic change to the image occurs, a key frame must be created. The entire image for the frame must be output when the visual difference between the two frames is so great that representing the new image incrementally from the previous frame would require more data than recreating the whole image.

It is beneficial to include key frames at arbitrary intervals while encoding video. For example, a key frame may be output once for each 10 seconds of video, even though the video image does not change enough visually to warrant the automatic creation of the key frame.

- H.263 Max bit rate: set the maximum bit rate for the H.263 codec. “O” disables this codec.
- H.263 Tx key frame period: default 125

H.264 Encoding Threads

H.264 encoding speed can be increased by using multiple threads of the CPU core. Exploiting thread-level parallelism is an attractive approach to improving the performance of multimedia applications that are running on multithreading general-purpose processors.

Reference:
http://www.embedded.com/design/real-time-and-performance/4027585/Optimizing-Video-Encoding-using-Threads-and-Parallelism-Part-1--Threading-a-video-codec

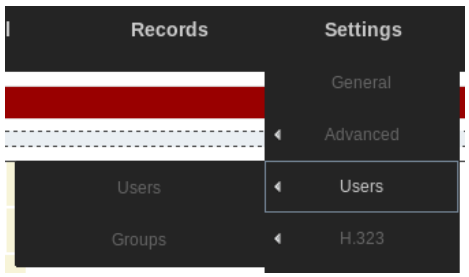
VP8 Encoding Threads:

VP8 supports the use of multiple threads in the encoder and decoder. This parameter determines the number of threads that will be allocated to the encode process. VP8 supports a mechanism whereby rows of macro-blocks can be simultaneously encoded on different threads. The decoder will usually automatically use an appropriate number of threads according to how many cores are available.

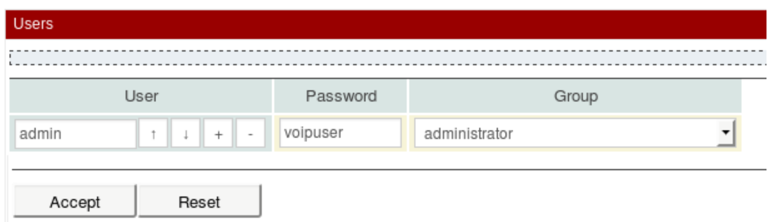
VP8 Encoding CPU Real-time mode allows the encoder to auto adjust the speed vs. quality trade-off in order to try and hit a particular cpu utilisation target. Legal values are 0-15. It is worth noting that the encode quality will depend on how hard a particular clip or section of a clip is and how fast the encoding machine is. In this mode the results will thus vary from machine to machine and even from run to run depending on what else you are doing.
Reference: http://www.webmproject.org/docs/encoder-parameters/

Advanced – Telnet

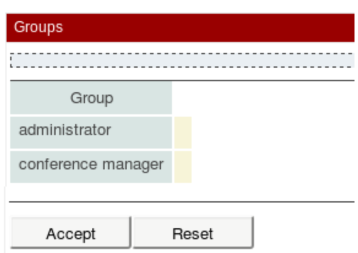
Settings – Users



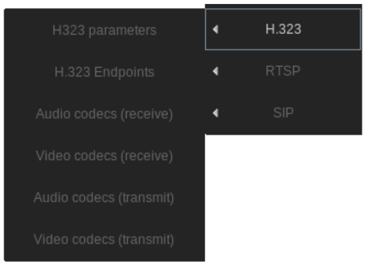
Users – Users



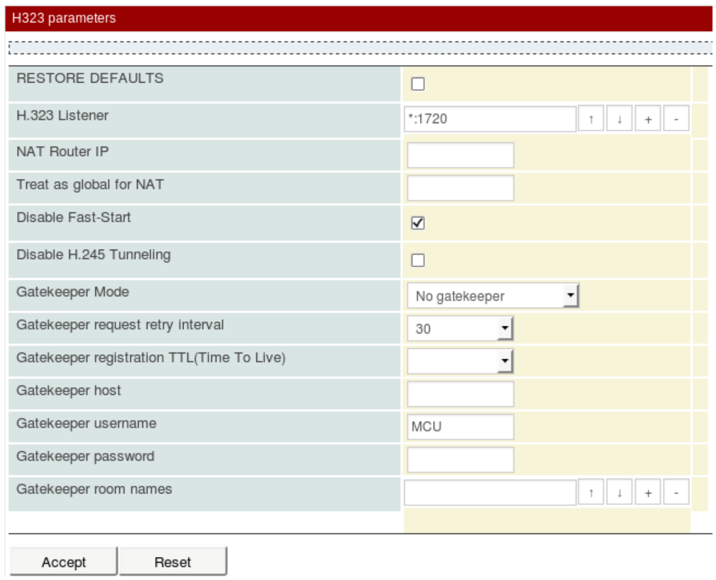
Users – Groups



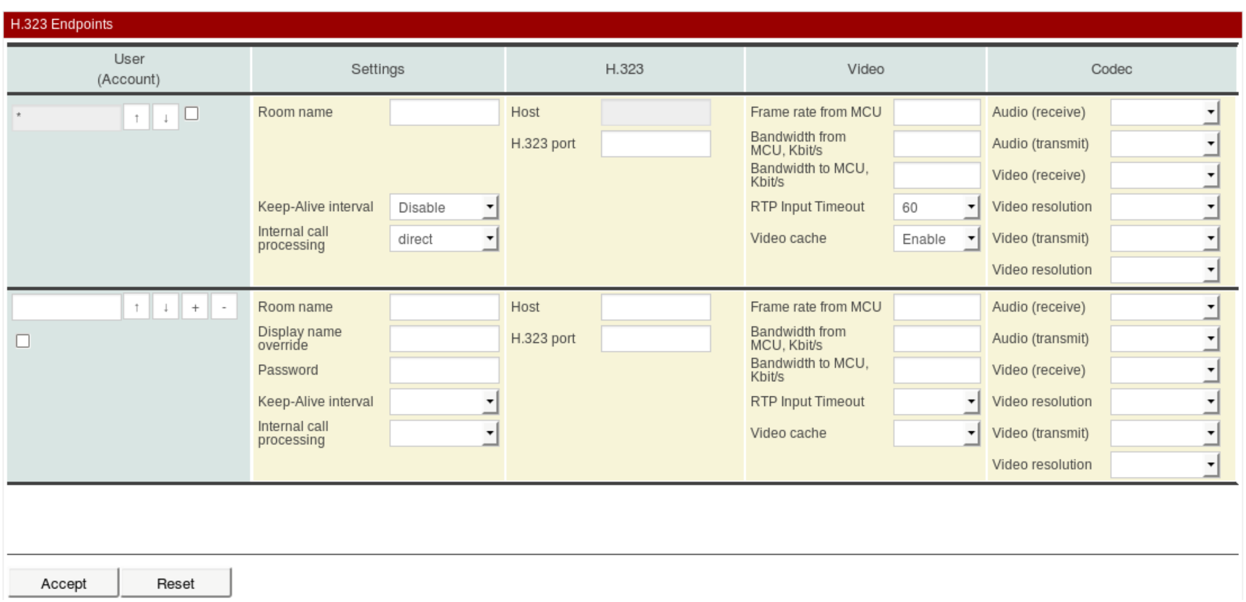
Settings – H.323



H.323 – H323 parameters



H.323 – H.323 Endpoints



H.323 – Audio codecs (receive)

Audio codecs (receive)

SILK_B40{sw}	☑	16000Hz	
SILK_B40_24K{sw}	☑	24000Hz	
OPUS_48K2{sw}	☑	48000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
OPUS_48K{sw}	☑	48000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
OPUS_16K{sw}	☑	16000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
OPUS_8K{sw}	☑	8000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
G.711-uLaw-64k{sw}	☑	8000Hz	
G.711-ALaw-64k{sw}	☑	8000Hz	
G.7231-6.3k[e]{sw}	☑	8000Hz	
G.729A{sw}	☑	8000Hz	
G.729{sw}	☑	8000Hz	
Speex_32K{sw}	☑	32000Hz	mode=6;vbr=off;
Speex_16K{sw}	☑	16000Hz	mode=6;vbr=off;
Speex_8K{sw}	☑	8000Hz	mode=3;vbr=off;
G.726-40k{sw}	☑	8000Hz	
G.726-32k{sw}	☑	8000Hz	
G.726-24k{sw}	☑	8000Hz	
G.726-16k{sw}	☑	8000Hz	
iLBC-13k3{sw}	☑	8000Hz	mode=30;
iLBC-15k2{sw}	☑	8000Hz	mode=20;
G.722-64k{sw}	☑	16000Hz	
G.722.1-24K{sw}	☑	16000Hz	bitrate=24000;
G.722.1-32K{sw}	☑	16000Hz	bitrate=32000;
G.722.1C-24K{sw}	☑	32000Hz	bitrate=24000;
G.722.1C-32K{sw}	☑	32000Hz	bitrate=32000;
G.722.1C-48K{sw}	☑	32000Hz	bitrate=48000;
G.722.2{sw}	☑	16000Hz	mode=7;octet-align=0;

Accept Reset

H.323 – Video codecs (receive)

Video codecs (receive)						
VP8{sw}	↑	↓	-	☑	352x288	max-fr=30;max-fs=0;
H.264{sw}	↑	↓	-	☑	704x576	
H.263p{sw}	↑	↓	-	☑	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
H.263{sw}	↑	↓	-	☑	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
H.261{sw}	↑	↓	-	☑	352x288	D=0;CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;

Accept Reset

H.323 – Audio codecs (transmit)

Audio codecs (transmit)						
OPUS_48K2{sw}	↑	↓	-	☑	48000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
OPUS_48K{sw}	↑	↓	-	☑	48000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
SILK_B40_24K{sw}	↑	↓	-	☑	24000Hz	
SILK_B40{sw}	↑	↓	-	☑	16000Hz	
OPUS_16K{sw}	↑	↓	-	☑	16000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
Speex_32K{sw}	↑	↓	-	☑	32000Hz	mode=6;vbr=off;
Speex_16K{sw}	↑	↓	-	☑	16000Hz	mode=6;vbr=off;
G.722-64k{sw}	↑	↓	-	☑	16000Hz	
OPUS_8K{sw}	↑	↓	-	☑	8000Hz	cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
Speex_8K{sw}	↑	↓	-	☑	8000Hz	mode=3;vbr=off;
G.711-uLaw-64k	↑	↓	-	☑	8000Hz	
G.711-ALaw-64k	↑	↓	-	☑	8000Hz	

G.7231-6.3k[e]{sw}	↑ ↓ -	☑	8000Hz	
G.729A{sw}	↑ ↓ -	☑	8000Hz	
G.729{sw}	↑ ↓ -	☑	8000Hz	
G.726-40k{sw}	↑ ↓ -	☑	8000Hz	
G.726-32k{sw}	↑ ↓ -	☑	8000Hz	
G.726-24k{sw}	↑ ↓ -	☑	8000Hz	
G.726-16k{sw}	↑ ↓ -	☑	8000Hz	
iLBC-13k3{sw}	↑ ↓ -	☑	8000Hz	mode=30;
iLBC-15k2{sw}	↑ ↓ -	☑	8000Hz	mode=20;
G.722.1-24K{sw}	↑ ↓ -	☑	16000Hz	bitrate=24000;
G.722.1-32K{sw}	↑ ↓ -	☑	16000Hz	bitrate=32000;
G.722.1C-24K{sw}	↑ ↓ -	☑	32000Hz	bitrate=24000;
G.722.1C-32K{sw}	↑ ↓ -	☑	32000Hz	bitrate=32000;
G.722.1C-48K{sw}	↑ ↓ -	☑	32000Hz	bitrate=48000;
G.722.2{sw}	↑ ↓ -	☑	16000Hz	mode=7;octet-align=0;

Accept Reset

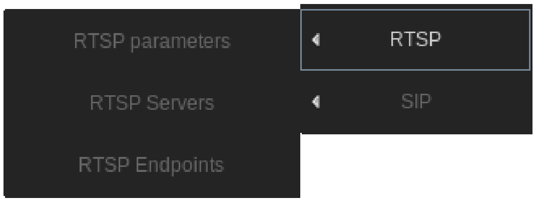
H.323 – Video codecs (transmit)

Video codecs (transmit)

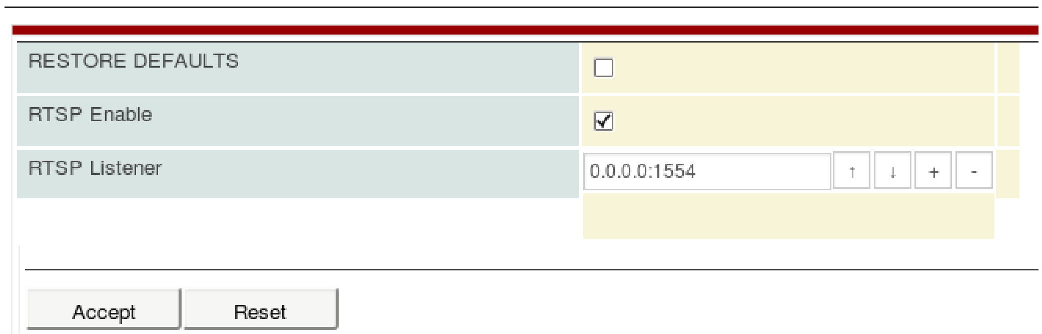
VP8{sw}	↑ ↓ -	☑	352x288	max-fr=30;max-fs=0;
H.264{sw}	↑ ↓ -	☑	704x576	
H.263p{sw}	↑ ↓ -	☑	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
H.263{sw}	↑ ↓ -	☑	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
H.261{sw}	↑ ↓ -	☑	352x288	D=0;CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;

Accept Reset

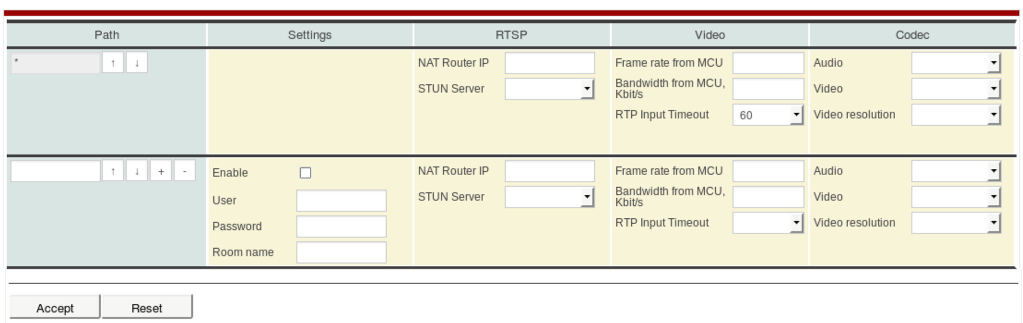
Settings – RTSP



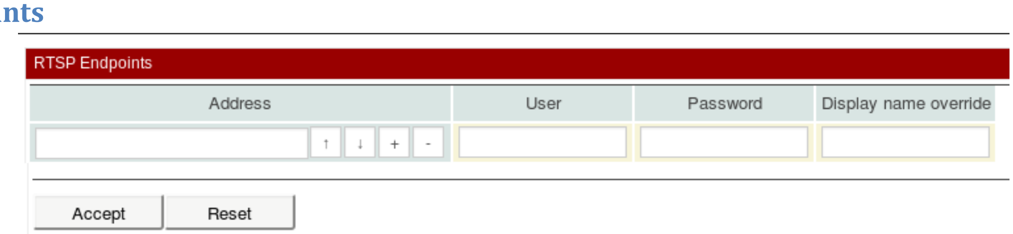
RTSP – RTSP parameters



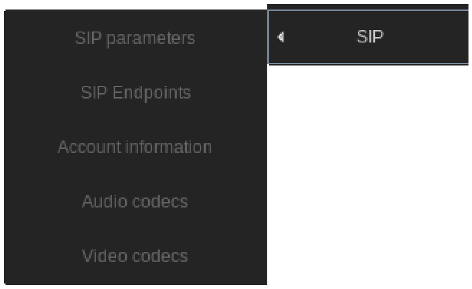
RTSP – RTSP Servers



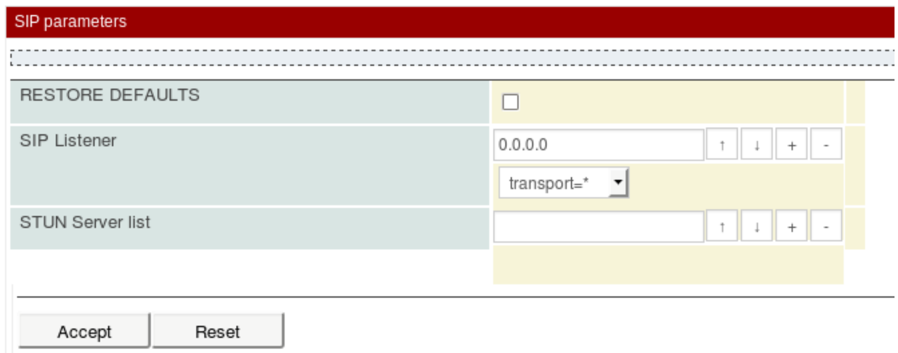
RTSP – RTSP Endpoints



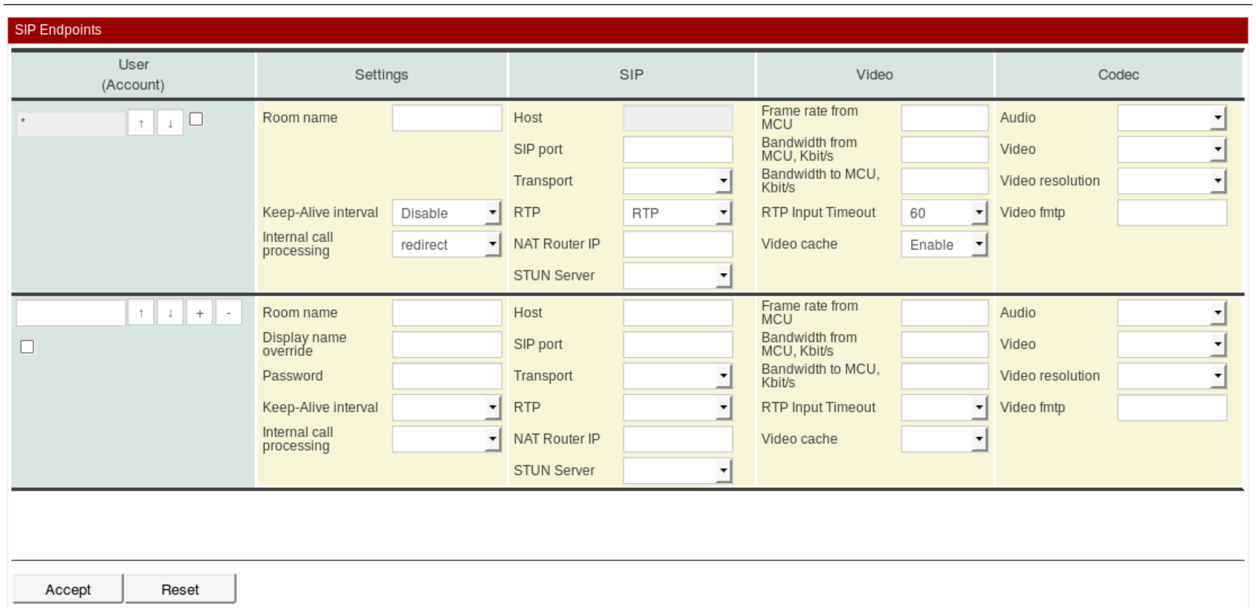
Settings – SIP



SIP – SIP parameters

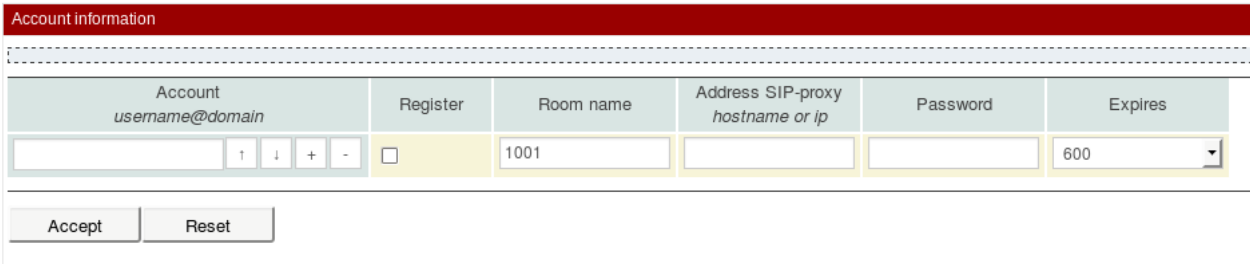


SIP – SIP Endpoints



- "Video resolution" – allows you to change the default video resolution to other settings
- "NAT Router IP" should be set to the public/WAN IP address of your network

SIP – Account information



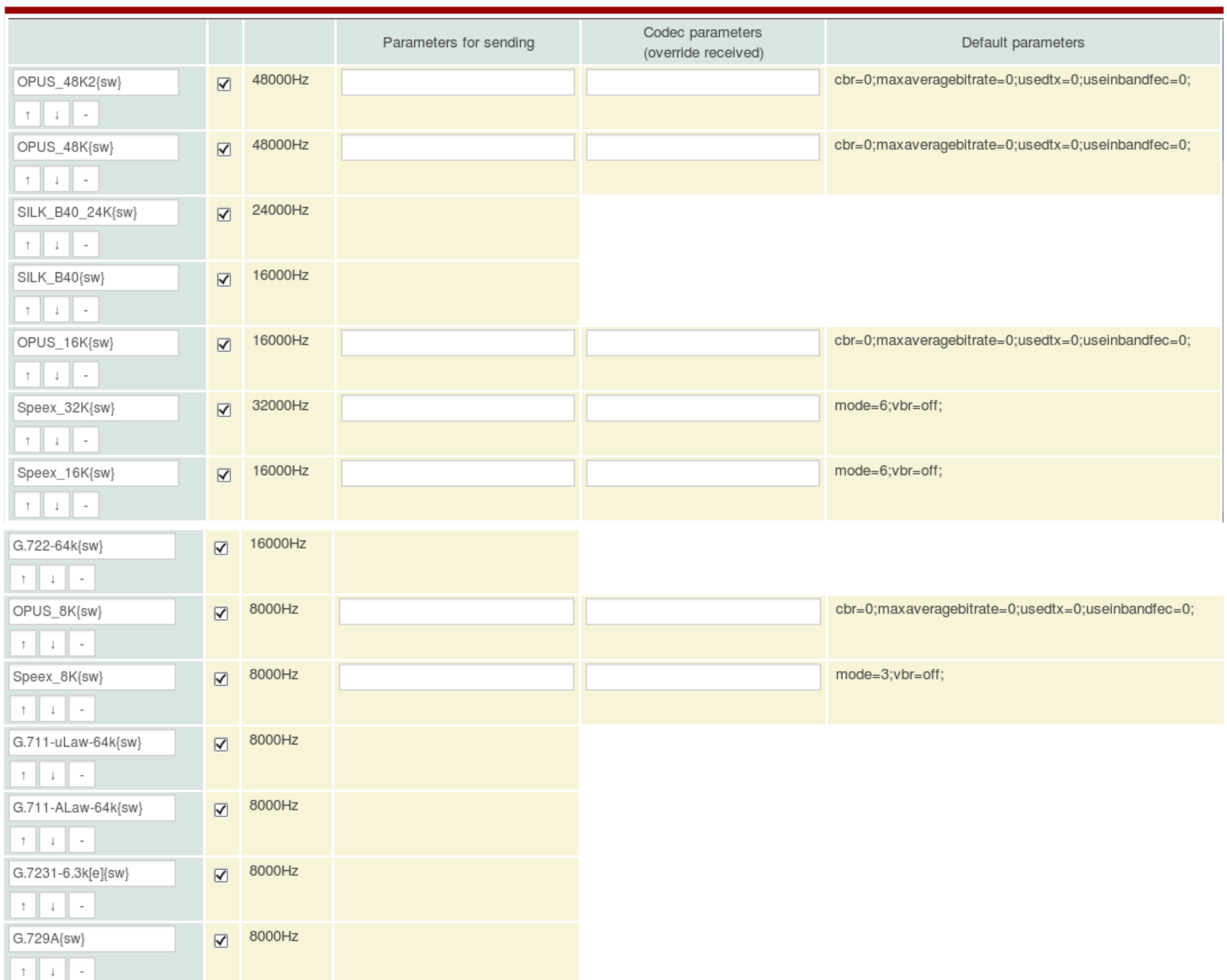
Account information

Account username@domain	Register	Room name	Address SIP-proxy hostname or ip	Password	Expires
↑ ↓ + -	☐	1001			600

Accept Reset

The SIP – Account information is used to connect to a Registrar's SIP extension. See Section 24, step 14 for an example of registering to a FreePBX extension.

SIP – Audio codecs



			Parameters for sending	Codec parameters (override received)	Default parameters
OPUS_48K2{sw} ↑ ↓ -	☑	48000Hz			cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
OPUS_48K{sw} ↑ ↓ -	☑	48000Hz			cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
SILK_B40_24K{sw} ↑ ↓ -	☑	24000Hz			
SILK_B40{sw} ↑ ↓ -	☑	16000Hz			
OPUS_16K{sw} ↑ ↓ -	☑	16000Hz			cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
Speex_32K{sw} ↑ ↓ -	☑	32000Hz			mode=6;vbr=off;
Speex_16K{sw} ↑ ↓ -	☑	16000Hz			mode=6;vbr=off;
G.722-64k{sw} ↑ ↓ -	☑	16000Hz			
OPUS_8K{sw} ↑ ↓ -	☑	8000Hz			cbr=0;maxaveragebitrate=0;usedtx=0;useinbandfec=0;
Speex_8K{sw} ↑ ↓ -	☑	8000Hz			mode=3;vbr=off;
G.711-uLaw-64k{sw} ↑ ↓ -	☑	8000Hz			
G.711-ALaw-64k{sw} ↑ ↓ -	☑	8000Hz			
G.7231-6.3k[e]{sw} ↑ ↓ -	☑	8000Hz			
G.729A{sw} ↑ ↓ -	☑	8000Hz			

G.729A{sw} ↑ ↓ -	☑	8000Hz	
G.729{sw} ↑ ↓ -	☑	8000Hz	
G.726-40k{sw} ↑ ↓ -	☑	8000Hz	
G.726-32k{sw} ↑ ↓ -	☑	8000Hz	
G.726-24k{sw} ↑ ↓ -	☑	8000Hz	
G.726-16k{sw} ↑ ↓ -	☑	8000Hz	
iLBC-13k3{sw} ↑ ↓ -	☑	8000Hz	mode=30;
iLBC-15k2{sw} ↑ ↓ -	☑	8000Hz	mode=20;
G.722.1-24K{sw} ↑ ↓ -	☑	16000Hz	bitrate=24000;
G.722.1-32K{sw} ↑ ↓ -	☑	16000Hz	bitrate=32000;
G.722.1C-24K{sw} ↑ ↓ -	☑	32000Hz	bitrate=24000;
G.722.1C-32K{sw} ↑ ↓ -	☑	32000Hz	bitrate=32000;
G.722.1C-48K{sw} ↑ ↓ -	☑	32000Hz	bitrate=48000;
G.722.2{sw} ↑ ↓ -	☑	16000Hz	mode=7;octet-align=0;

Accept Reset

SIP – Video codecs

			Default parameters
VP8{sw} ↑ ↓ -	☐	352x288	max-fr=30;max-fs=0;
H.264{sw} ↑ ↓ -	☑	704x576	
H.263p{sw} ↑ ↓ -	☐	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
H.263{sw} ↑ ↓ -	☐	352x288	CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;
MP4V-ES{sw} ↑ ↓ -	☑	352x288	config="";
H.261{sw} ↑ ↓ -	☐	352x288	D=0;CIF=1;CIF16=0;CIF4=0;QCIF=0;SQCIF=0;

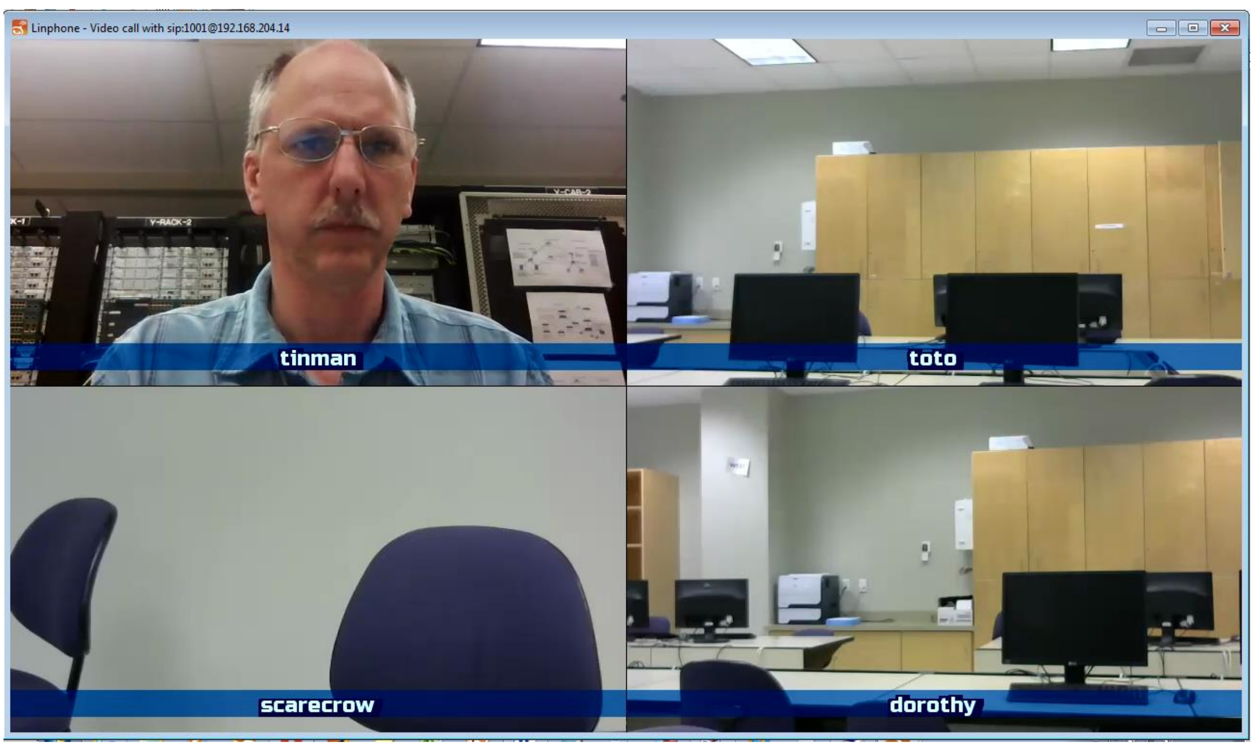
Accept Reset

20. Performance Results

This is section is to provide you with an idea of the computing resources used for a conference server with minimal resources (1 CPU/1 GB RAM). These results are based on a standalone server with X windows running.

Test specifications:

- 4 participants 640x480 H.264 video codec, OPUS audio codec
- Network 100 Mbps Ethernet
- Proxmox 3.4-1 host :
    - Dual 8 core 3.4 GHz processors
    - 72 GB RAM
    - RAID 10 300 GB
    - HP DL160 G6
- OenMCU-ru virtual machine guest:
    - CentOS 6.6
    - 1 GB memory
    - 1 core
    - 32 GB hard-drive



1. OpenMCU-ru status for room 1001:

Room 1001

Name	Duration	RTP Channel: Codec	Packets	Bytes	Kbit/s	FPS	60s losses
[Hidden] file recorder	16:29	-	-	-	-	-	-
[Hidden] conference recorder	0.0	-	-	-	-	-	-
[Hidden] cache	16:05	**Video Out:** H.264@1280x720:256000x10_1001/0	-	-	-	**4 x** 10.00	-
dorothy [sip:dorothy@172.16.58.19:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	8:08	**Audio In:** OPUS_48K2@48000/2 **Audio Out:** OPUS_48K2@48000/2 **Video In:** H.264@640x480 **Video Out:** H.264@1280x720:256000x10_1001/0	24393 24397 32432 15836	352142 6014205 6965769 13419081	5.7 98.7 99.5 172.2	14.96 10.00	0% 0% 0% 0%
scarecrow [sip:scarecrow@172.16.59.13:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	8:36	**Audio In:** OPUS_48K2@48000/2 **Audio Out:** OPUS_48K2@48000/2 **Video In:** H.264@640x480 **Video Out:** H.264@1280x720:256000x10_1001/0	25763 25838 32492 16984	366020 6345500 4138952 14418920	5.6 98.0 85.4 173.8	15.00 10.00	0% 0% 0% 0%
tinman [sip:tinman@192.168.202.11:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	16:05	**Audio In:** OPUS_48K2@48000/2 **Audio Out:** OPUS_48K2@48000/2 **Video In:** H.264@640x480 **Video Out:** H.264@1280x720:256000x10_1001/0	48244 48251 111533 31714	689430 11792518 89165822 26965421	5.6 98.7 742.9 173.8	14.80 10.00	0% 0% 0% 0%
toto [sip:toto@192.168.198.229:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	15:55	**Audio In:** OPUS_48K2@48000/2 **Audio Out:** OPUS_48K2@48000/2 **Video In:** H.264@640x480 **Video Out:** H.264@1280x720:256000x10_1001/0	47776/1 47781 64647 31288	591300 11791560 16287671 26596101	4.8 99.1 23.3 172.2	14.93 10.00	0% 0% 0% 0%

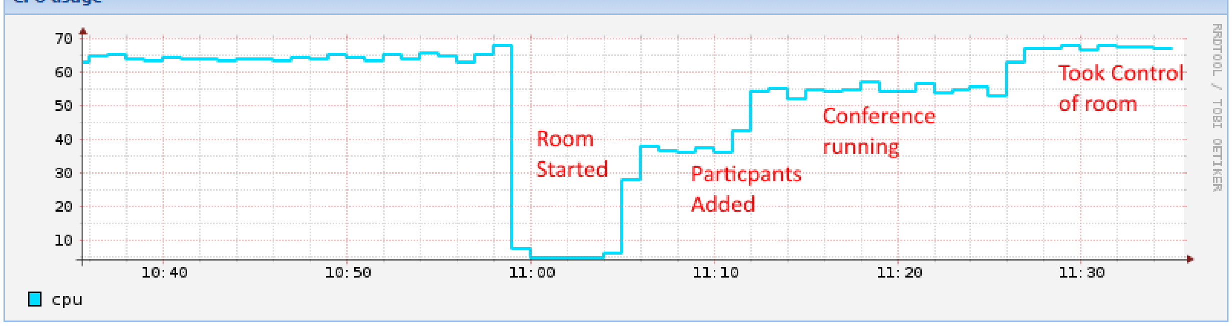
2. Proxmox VM performance monitoring:

Status

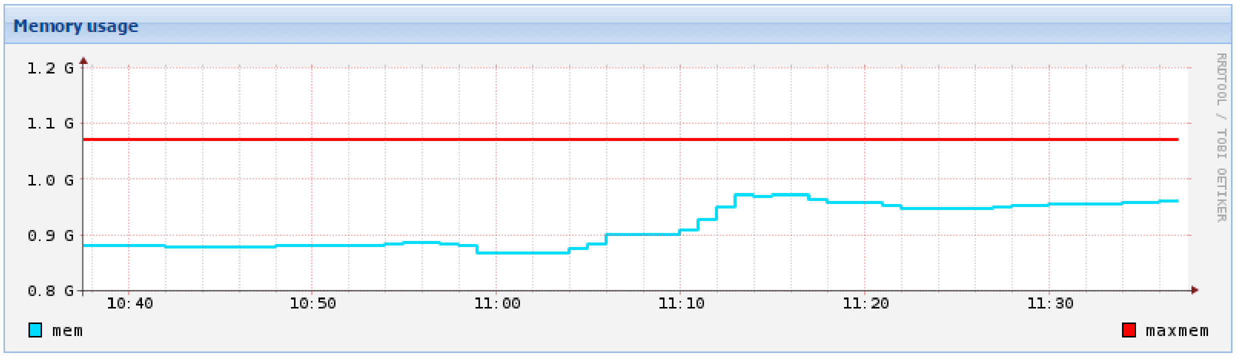
Name	TestMCU
Status	running
CPU usage	63.8% of 1CPU
Memory usage	Total: 1.00GB Used: 917MB
Uptime	23:51:40
Managed by HA	No

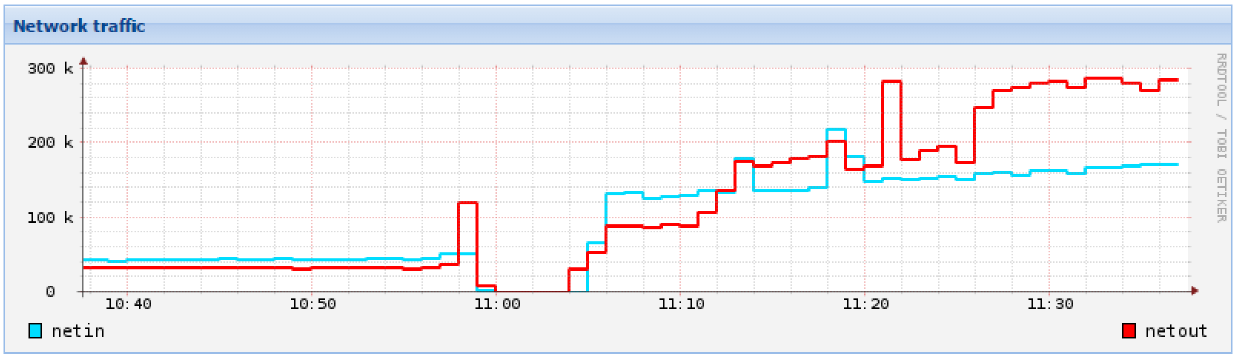
Notes

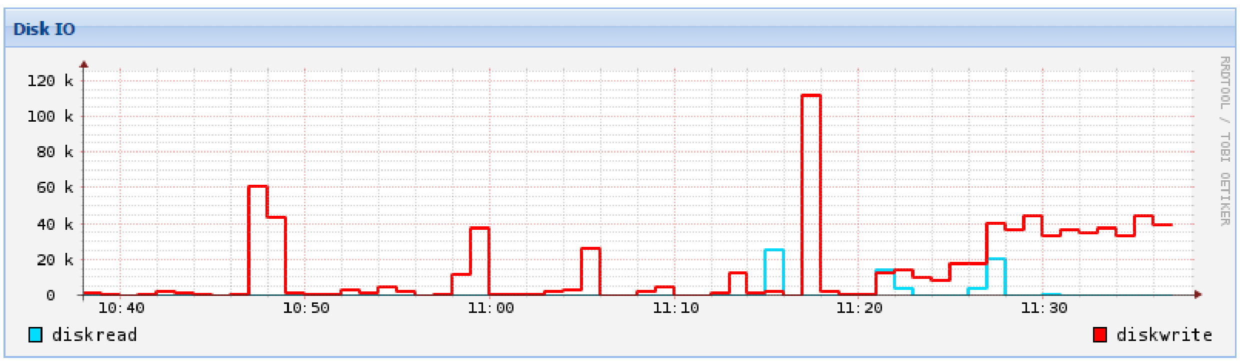
CPU usage



The initial jumps are indications of when participants are being added to the conference room.

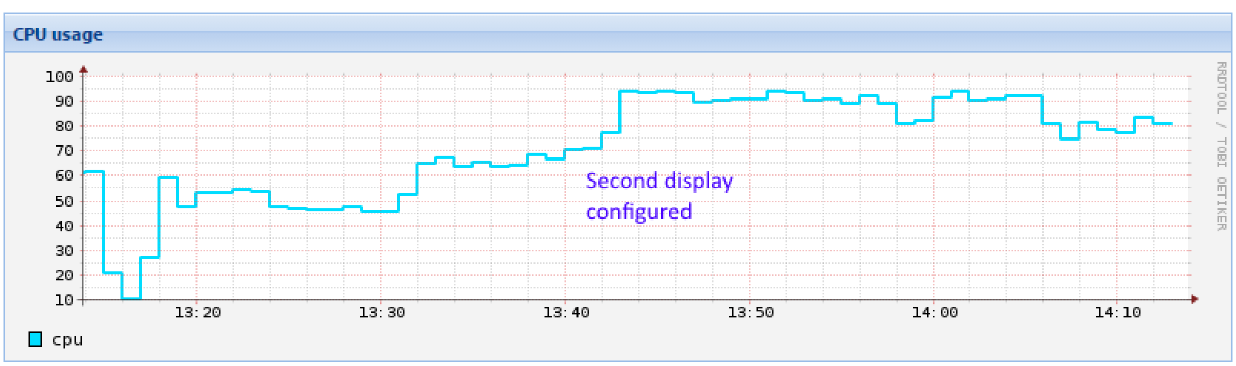






Two displays

This is the CPU usage after a 2nd display is configured. The CPU usage goes from roughly 70% to 90%.



Stability

I've run a conference room for hours at a time while testing and have had no issues. It just keeps working. No memory creep or unusual CPU spikes or anomalies.

Performance Test 2

I'm a teacher at the Southern Alberta Institute of Technolog (formerly SAIT Polytechnic) and had one of my classes connect to a video conference using OpenMCU-ru. OpenMCU-ru was run as a virtual machine under VMware Workstation on my laptop. The virtual machine was configured per:

- 2 CPUs
- 2 GB of RAM
- CentOS 7

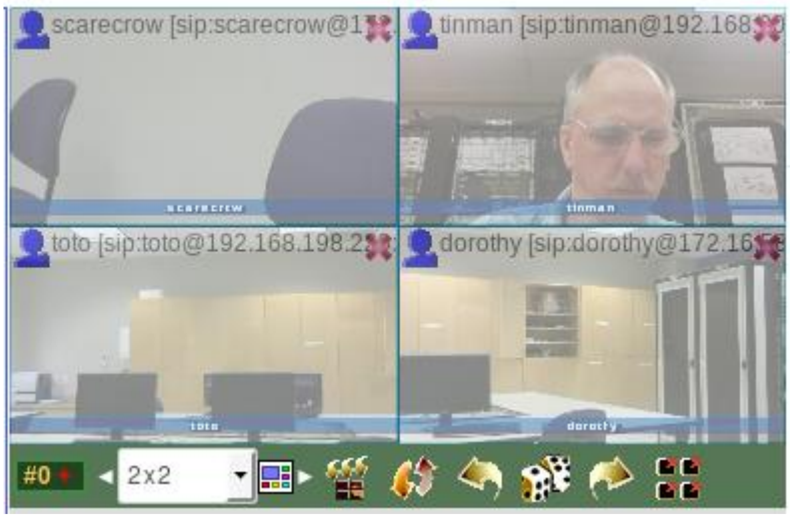
There were 14 Linphone clients running on the student's laptops running H.264 with 640x480 resolution. The CPU load was about 60% for each CPU and it worked flawlessly. The RAM was running at 1.9 GB and no swap file used. A separate PC running VLC RTSP client was monitoring the video stream also. The test lasted over 1 hour and it was quite impressive.

The conference room was initially setup in auto mode and then I took control of the room and was able to dynamically change the room display and configuration without breaking the conference.

Initially, I used a Web stream on the separate PC to monitor the conference room but the quality was poor. I switched to a VLC RTSP client to monitor the conference room and the quality was excellent – same as what the participants saw.

21. Display Matrix

When in a room's control and after taking control. The  button displays the matrix window



The matrices are defined in the text file /opt/openmcu-ru/config/layouts.conf. There are explanations on how to create a custom display matrix in the file.

Increase Display Frame Border Example:

To increase the display frame border width between participants for 2x1 display:

```
[My2x1]
frame_width=99;position_width=48
frame_height=4;position_height=2
(1,1);(50,1)
```

Contents of layouts.conf

```
# Layout definitions for OpenMCU 2.2.3.43 by muggot

# All parameters except "font" could be local (inside single
# layout) and global (for everything below). To redefine global
# parameter close the layout descriptor first (by typing "[]").

# [LAYOUD_ID] opens new layout descriptor. You don't have to
# close them, just type [NEXT_LAYOUT_ID] and continue, except
# cases you want redefine something global: use [] - it will
# mean that layout has ended.

# (x,y) defines video position in the frame. Video positions
# have to be defined inside layout descriptors. It is
# possible to define up to 100 video positions for each
# layout.

# Use # and // for writing comments.
```

```
# Dividers are: "Enter" (0xA) or semicolon (;).


# Initial parameters (global):

# Logical width and height of main video frame. openmcu will
# scale them, so you may use everything you want:
frame_width=704
frame_height=576

# mode_mask - where layout will used:
# 0 - never used;
# 1 - non-moderated mode only;
# 2 - moderated mode only;
# 3 - used in both modes.
mode_mask=3

# 1-pixel border around video position (1=yes; 0=no):
border=1

# Non-moderated mode behavior:
# new_members_first = 1|0 - add new members from begin (1) or
# from end (0) of the the list (default 1):
#new_members_first=1

# reallocate_on_disconnect = 1|0 - reallocate position to remove
# possible "holes" when somebody has disconnected, but layout
# not changed (default 0):
#reallocate_on_disconnect=1

# True type font for printing labels (subtitles):
#font=Russo_One.ttf

# Font size in pixels (or reltive to frame height):
#fontsize=1/16

# Cut username before bracket "[" or "(" =1 or don't =0:
#cut_before_bracket=1

# Label's options (bit mask):
#  + 1 = center horizontally;
#  + 2 = center vertically;
#  + 4 = paint from right;
#  + 8 = paint from bottom;
#  +16 = transparent pad;
#  +32 = disable label.
#  +64 = subtitle mode
#label_mask = 89

# Minimum width of video position to render lable (if it will
# smaller, the rendering will skipped). Just in pixels, or
# relative to frame width, eg. 1/5 (of frame that endpoint
# will receive):
#minimum_width_for_label = 1/5

# Color for label's pad:
#label_bgcolor=115599

# Label's borders:
#border_left=5/80 // of frame width
#border_right=5/80 // of frame width
#border_top=1/200 //of frame height
#border_bottom=1/100 //of frame height
#h_pad=1/16 //of frame width
```

```
#v_pad=1/24 //of frame height

#Shadow setup for 'subtitle mode':
#dropshadow_l=1/200 //of frame width
#dropshadow_r=1/80 //of frame width
#dropshadow_t=1/150 //of frame height
#dropshadow_b=1/65 //of frame height

# Logical width and height of member's video frame, openmcu
# will scale them, so you may use everything you want;
position_width=352
position_height=288

# Mock-up width & height - this size will used to display HTML
# macket of the layout in Room Control Page:
#mockup_width=500
#mockup_height=300

/////////////////////////////////////////////
///// 1. Start with several x*x layouts /////
/////////////////////////////////////////////

[1 x 1]
position_width=704; position_height=576; (0,0)

[2 x 2]
(0,0); (352,0)
(0,288); (352,288)

[3 x 3]
position_width=234; position_height=192
(0,0)  ;(234,0);  (468,0)
(0,192);(234,192);(468,192)
(0,384);(234,384);(468,384)

[4 x 4]
position_width=176; position_height=144
(0,  0); (176,  0); (352,  0); (528,0  )
(0,144); (176,144); (352,144); (528,144)
(0,288); (176,288); (352,288); (528,288)
(0,432); (176,432); (352,432); (528,432)

[5 x 5]
position_width=140; position_height=114
(0,  0); (141,  0); (282,  0); (423,  0); (564,  0)
(0,114); (141,114); (282,114); (423,114); (564,114)
(0,228); (141,228); (282,228); (423,228); (564,228)
(0,342); (141,342); (282,342); (423,342); (564,342)
(0,456); (141,456); (282,456); (423,456); (564,456)

[6 x 6]
position_width=116; position_height=96
(4,  0); (120,  0); (236,  0); (352,  0); (468,  0); (584,  0)
(4, 96); (120, 96); (236, 96); (352, 96); (468, 96); (584, 96)
(4,192); (120,192); (236,192); (352,192); (468,192); (584,192)
(4,288); (120,288); (236,288); (352,288); (468,288); (584,288)
(4,384); (120,384); (236,384); (352,384); (468,384); (584,384)
(4,480); (120,480); (236,480); (352,480); (468,480); (584,480)

[7 x 7]
frame_width=7; frame_height=7
position_width=1; position_height=1
(0,0); (1,0); (2,0); (3,0); (4,0); (5,0); (6,0)
(0,1); (1,1); (2,1); (3,1); (4,1); (5,1); (6,1)
```

```
(0,2); (1,2); (2,2); (3,2); (4,2); (5,2); (6,2)
(0,3); (1,3); (2,3); (3,3); (4,3); (5,3); (6,3)
(0,4); (1,4); (2,4); (3,4); (4,4); (5,4); (6,4)
(0,5); (1,5); (2,5); (3,5); (4,5); (5,5); (6,5)
(0,6); (1,6); (2,6); (3,6); (4,6); (5,6); (6,6)

[8 x 8]
frame_width=8; frame_height=8
position_width=1; position_height=1
(0,0); (1,0); (2,0); (3,0); (4,0); (5,0); (6,0); (7,0)
(0,1); (1,1); (2,1); (3,1); (4,1); (5,1); (6,1); (7,1)
(0,2); (1,2); (2,2); (3,2); (4,2); (5,2); (6,2); (7,2)
(0,3); (1,3); (2,3); (3,3); (4,3); (5,3); (6,3); (7,3)
(0,4); (1,4); (2,4); (3,4); (4,4); (5,4); (6,4); (7,4)
(0,5); (1,5); (2,5); (3,5); (4,5); (5,5); (6,5); (7,5)
(0,6); (1,6); (2,6); (3,6); (4,6); (5,6); (6,6); (7,6)
(0,7); (1,7); (2,7); (3,7); (4,7); (5,7); (6,7); (7,7)

[9 x 9]
frame_width=9; frame_height=9
position_width=1; position_height=1
(0,0); (1,0); (2,0); (3,0); (4,0); (5,0); (6,0); (7,0); (8,0)
(0,1); (1,1); (2,1); (3,1); (4,1); (5,1); (6,1); (7,1); (8,1)
(0,2); (1,2); (2,2); (3,2); (4,2); (5,2); (6,2); (7,2); (8,2)
(0,3); (1,3); (2,3); (3,3); (4,3); (5,3); (6,3); (7,3); (8,3)
(0,4); (1,4); (2,4); (3,4); (4,4); (5,4); (6,4); (7,4); (8,4)
(0,5); (1,5); (2,5); (3,5); (4,5); (5,5); (6,5); (7,5); (8,5)
(0,6); (1,6); (2,6); (3,6); (4,6); (5,6); (6,6); (7,6); (8,6)
(0,7); (1,7); (2,7); (3,7); (4,7); (5,7); (6,7); (7,7); (8,7)
(0,8); (1,8); (2,8); (3,8); (4,8); (5,8); (6,8); (7,8); (8,8)

[10x10]
frame_width=10; frame_height=10
position_width=1; position_height=1
(0,0); (1,0); (2,0); (3,0); (4,0); (5,0); (6,0); (7,0); (8,0); (9,0)
(0,1); (1,1); (2,1); (3,1); (4,1); (5,1); (6,1); (7,1); (8,1); (9,1)
(0,2); (1,2); (2,2); (3,2); (4,2); (5,2); (6,2); (7,2); (8,2); (9,2)
(0,3); (1,3); (2,3); (3,3); (4,3); (5,3); (6,3); (7,3); (8,3); (9,3)
(0,4); (1,4); (2,4); (3,4); (4,4); (5,4); (6,4); (7,4); (8,4); (9,4)
(0,5); (1,5); (2,5); (3,5); (4,5); (5,5); (6,5); (7,5); (8,5); (9,5)
(0,6); (1,6); (2,6); (3,6); (4,6); (5,6); (6,6); (7,6); (8,6); (9,6)
(0,7); (1,7); (2,7); (3,7); (4,7); (5,7); (6,7); (7,7); (8,7); (9,7)
(0,8); (1,8); (2,8); (3,8); (4,8); (5,8); (6,8); (7,8); (8,8); (9,8)
(0,9); (1,9); (2,9); (3,9); (4,9); (5,9); (6,9); (7,9); (8,9); (9,9)


///////////////////////////////////////////
///// 2. Several centered x*y layouts /////
///////////////////////////////////////////

[2 x 1]
position_width=352; position_height=288; (0,144); (352,144)

[3 x 2]
position_width=234; position_height=192
(1, 96); (235, 96); (469, 96)
(1,288); (235,288); (469,288)

[4 x 3]
frame_width=8; frame_height=8
position_width=2; position_height=2
(0,1); (2,1); (4,1); (6,1)
(0,3); (2,3); (4,3); (6,3)
(0,5); (2,5); (4,5); (6,5)
```

```
[5 x 4]
frame_width=10; frame_height=10
position_width=2; position_height=2
(0,1); (2,1); (4,1); (6,1); (8,1)
(0,3); (2,3); (4,3); (6,3); (8,3)
(0,5); (2,5); (4,5); (6,5); (8,5)
(0,7); (2,7); (4,7); (6,7); (8,7)

[6 x 5]
frame_width=12; frame_height=12
position_width=2; position_height=2
(0,1); (2,1); (4,1); (6,1); (8,1); (10,1)
(0,3); (2,3); (4,3); (6,3); (8,3); (10,3)
(0,5); (2,5); (4,5); (6,5); (8,5); (10,5)
(0,7); (2,7); (4,7); (6,7); (8,7); (10,7)
(0,9); (2,9); (4,9); (6,9); (8,9); (10,9)


//////////////////////////////////////////////////
///// 3. Layouts with 1 or 2 large positions /////
//////////////////////////////////////////////////

[1 + 5]
mode_mask=2
position_width=468; position_height=384; (235,192)
position_width=234; position_height=192
(1,  0);(235,0);(469,0)
(1,192)
(1,384)

[1 + 7]
mode_mask=2
position_width=528; position_height=432; (176,144)
position_width=176; position_height=144
(0,  0); (176,  0); (352,  0); (528,  0)
(0,144)
(0,288)
(0,432)

[1 + 9]
mode_mask=2
position_width=564; position_height=460; (140,116)
position_width=140; position_height=114
(0,  1); (141,  1); (282,  1); (423,  1); (564,  1)
(0,116)
(0,231)
(0,346)
(0,461)

[1 +11]
mode_mask=2
position_width=580; position_height=480; (120, 96)
position_width=116; position_height=96
(4,  0); (120,  0); (236,  0); (352,  0); (468,  0); (584,  0)
(4, 96)
(4,192)
(4,288)
(4,384)
(4,480)

[1 +12]
mode_mask=2
position_width=352; position_height=288; (0,0)
```

```
position_width=176; position_height=144
                    (352,  0); (528,0  )
                    (352,144); (528,144)
(0,288); (176,288); (352,288); (528,288)
(0,432); (176,432); (352,432); (528,432)

[1c+12]
mode_mask=2
position_width=352; position_height=288; (176,144)
position_width=176; position_height=144
(0,  0); (176,  0); (352,  0); (528,0  )
(0,144);                       (528,144)
(0,288);                       (528,288)
(0,432); (176,432); (352,432); (528,432)

[2 + 8]
mode_mask=2
position_width=352; position_height=288
(0,144); (352,144)
position_width=176; position_height=144
(0,  0); (176,  0); (352,  0); (528,0  )
(0,432); (176,432); (352,432); (528,432)

[2 + 3]
mode_mask=2
position_width=352; position_height=288
(0,48); (352,48)
position_width=234; position_height=192
(1,336); (235,336); (469,336)

[1 +20]
mode_mask=2
frame_width=6;frame_height=6
position_width=4;position_height=4;(1,1)
position_width=1;position_height=1
(0,0);(1,0);(2,0);(3,0);(4,0);(5,0)
(0,1);                        (5,1)
(0,2);                        (5,2)
(0,3);                        (5,3)
(0,4);                        (5,4)
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5)

[1c+24]
frame_width=7; frame_height=7
mode_mask=2;
position_width=5; position_height=5
(1,1)
position_width=1; position_height=1
(0,0);(1,0);(2,0);(3,0);(4,0);(5,0);(6,0)
(0,1);                              (6,1)
(0,2);                              (6,2)
(0,3);                              (6,3)
(0,4);                              (6,4)
(0,5);                              (6,5)
(0,6);(1,6);(2,6);(3,6);(4,6);(5,6);(6,6)

[1 +72]
mode_mask=2
frame_width=9; frame_height=9
position_width=3; position_height=3; (3,3)
position_width=1; position_height=1;
(0,0);(1,0);(2,0);(3,0);(4,0);(5,0);(6,0);(7,0);(8,0);
(0,1);(1,1);(2,1);(3,1);(4,1);(5,1);(6,1);(7,1);(8,1);
(0,2);(1,2);(2,2);(3,2);(4,2);(5,2);(6,2);(7,2);(8,2);
```

```
(0,3);(1,3);(2,3);                   (6,3);(7,3);(8,3);
(0,4);(1,4);(2,4);                   (6,4);(7,4);(8,4);
(0,5);(1,5);(2,5);                   (6,5);(7,5);(8,5);
(0,6);(1,6);(2,6);(3,6);(4,6);(5,6);(6,6);(7,6);(8,6);
(0,7);(1,7);(2,7);(3,7);(4,7);(5,7);(6,7);(7,7);(8,7);
(0,8);(1,8);(2,8);(3,8);(4,8);(5,8);(6,8);(7,8);(8,8);

[2 +|55|]
mode_mask=2
frame_width=22;frame_height=20;position_width=11;position_height=10;(0,0);(11,0)
position_width=2;position_height=2
( 0,10);( 2,10);( 4,10);( 6,10);(
8,10);(10,10);(12,10);(14,10);(16,10);(18,10);(20,10)
( 0,12);( 2,12);( 4,12);( 6,12);(
8,12);(10,12);(12,12);(14,12);(16,12);(18,12);(20,12)
( 0,14);( 2,14);( 4,14);( 6,14);(
8,14);(10,14);(12,14);(14,14);(16,14);(18,14);(20,14)
( 0,16);( 2,16);( 4,16);( 6,16);(
8,16);(10,16);(12,16);(14,16);(16,16);(18,16);(20,16)
( 0,18);( 2,18);( 4,18);( 6,18);(
8,18);(10,18);(12,18);(14,18);(16,18);(18,18);(20,18)

[2 +|75|]
mode_mask=2
frame_width=99;frame_height=9;position_width=33;position_height=3;(11,1);(55,1)
position_width=9;position_height=1
( 0,0);( 9,0);(18,0);(27,0);(36,0);(45,0);(54,0);(63,0);(72,0);(81,0);(90,0);
position_width=11
( 0,1);                            (44,1);                            (88,1)
( 0,2);                            (44,2);                            (88,2)
( 0,3);                            (44,3);                            (88,3);
position_width=9
( 0,4);( 9,4);(18,4);(27,4);(36,4);(45,4);(54,4);(63,4);(72,4);(81,4);(90,4)
( 0,5);( 9,5);(18,5);(27,5);(36,5);(45,5);(54,5);(63,5);(72,5);(81,5);(90,5)
( 0,6);( 9,6);(18,6);(27,6);(36,6);(45,6);(54,6);(63,6);(72,6);(81,6);(90,6)
( 0,7);( 9,7);(18,7);(27,7);(36,7);(45,7);(54,7);(63,7);(72,7);(81,7);(90,7)
( 0,8);( 9,8);(18,8);(27,8);(36,8);(45,8);(54,8);(63,8);(72,8);(81,8);(90,8)

[|2|+|67|]
mode_mask=2
frame_width=99;frame_height=9;position_width=33;position_height=4;(11,0);(55,0)
position_width=11;position_height=1
( 0,0);                            (44,0);                            (88,0)
( 0,1);                            (44,1);                            (88,1)
( 0,2);                            (44,2);                            (88,2)
( 0,3);                            (44,3);                            (88,3);
position_width=9
( 0,4);( 9,4);(18,4);(27,4);(36,4);(45,4);(54,4);(63,4);(72,4);(81,4);(90,4)
( 0,5);( 9,5);(18,5);(27,5);(36,5);(45,5);(54,5);(63,5);(72,5);(81,5);(90,5)
( 0,6);( 9,6);(18,6);(27,6);(36,6);(45,6);(54,6);(63,6);(72,6);(81,6);(90,6)
( 0,7);( 9,7);(18,7);(27,7);(36,7);(45,7);(54,7);(63,7);(72,7);(81,7);(90,7)
( 0,8);( 9,8);(18,8);(27,8);(36,8);(45,8);(54,8);(63,8);(72,8);(81,8);(90,8)

[|2|+|56|]
mode_mask=2
frame_width=99;frame_height=8;position_width=33;position_height=4;(11,0);(55,0)
position_width=11;position_height=1
( 0,0);                            (44,0);                            (88,0)
( 0,1);                            (44,1);                            (88,1)
( 0,2);                            (44,2);                            (88,2)
( 0,3);                            (44,3);                            (88,3);
position_width=9
( 0,4);( 9,4);(18,4);(27,4);(36,4);(45,4);(54,4);(63,4);(72,4);(81,4);(90,4)
( 0,5);( 9,5);(18,5);(27,5);(36,5);(45,5);(54,5);(63,5);(72,5);(81,5);(90,5)
```

```
( 0,6);( 9,6);(18,6);(27,6);(36,6);(45,6);(54,6);(63,6);(72,6);(81,6);(90,6)
( 0,7);( 9,7);(18,7);(27,7);(36,7);(45,7);(54,7);(63,7);(72,7);(81,7);(90,7)


//////////////////////////////////
///// 4. 'Picture in picture' /////
//////////////////////////////////

[2 [' ]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,16)

[2 [ ']]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (528,16)

[2 [. ]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,432)

[2 [ .]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (528,432)

[3 ['']]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,16); (528,16)

[3 [ :]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (528,16); (528,432)

[3 [..]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,432); (528,432)

[3 [: ]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,16); (20,432)

[3 ['.]]
label_mask=32
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,16); (528,432)

[3 [.']]
label_mask=32
```

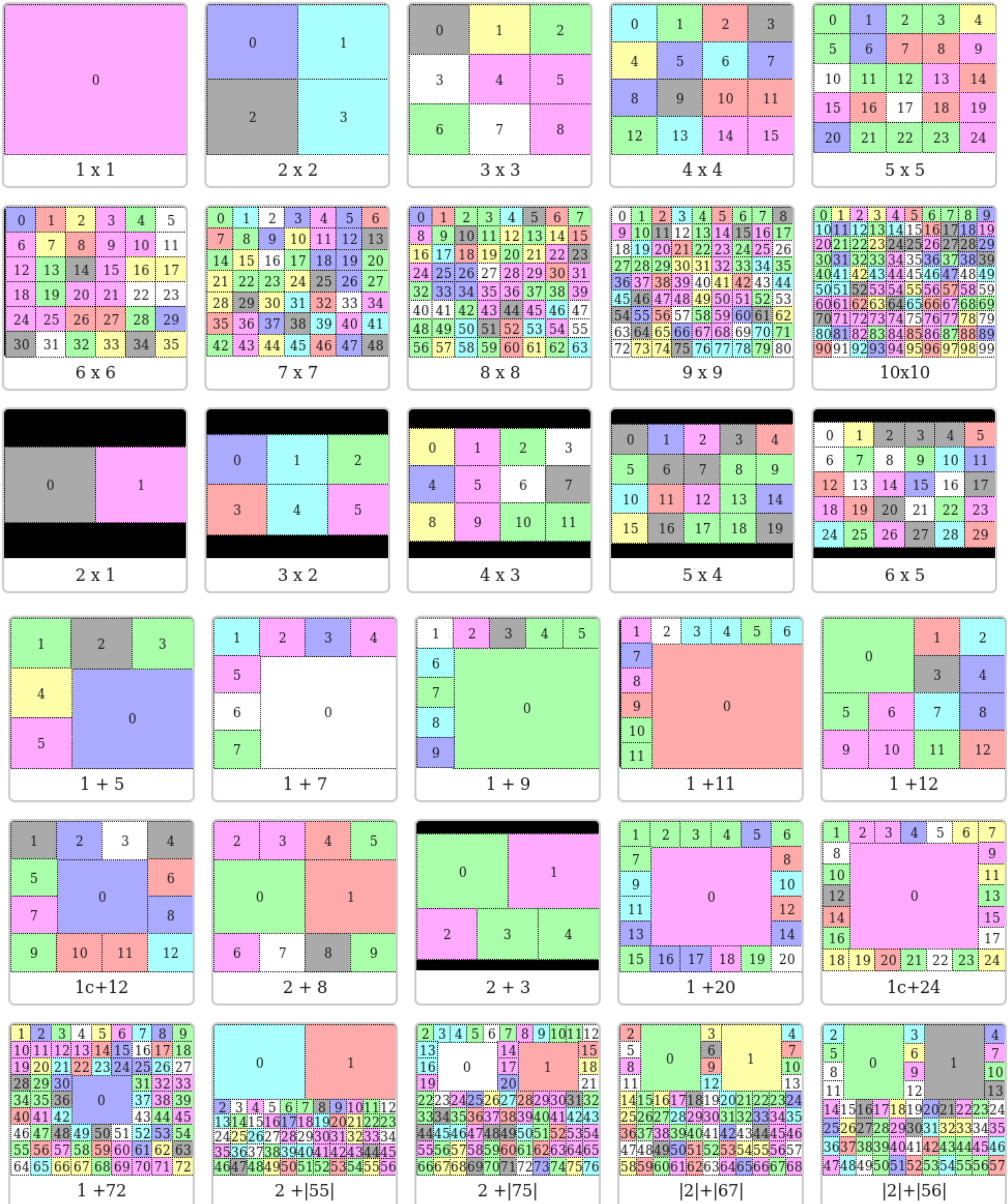
```
mode_mask=2
position_width=704; position_height=576; (0,0)
position_width=156; position_height=128; (20,432); (528,16);


///////////////////////////
///// 5. Other layouts /////
///////////////////////////

[|4|x3]
mode_mask=2
position_width=176; position_height=192
(  0,  0); (176,  0); (352,  0); (528,  0)
(  0,192); (176,192); (352,192); (528,192)
(  0,384); (176,384); (352,384); (528,384)

[1 +10]
mode_mask=2
position_width=352; position_height=192; (176,192)
position_width=176
(  0,  0); (176,  0); (352,  0); (528,  0)
(  0,192);                       (528,192)
(  0,384); (176,384); (352,384); (528,384)

[2 +10]
mode_mask=2
frame_width=10; frame_height=5
 position_width=5; position_height=3
  (0,1); (5,1)
 position_width=2; position_height=1
  (0,0); (2,0); (4,0); (6,0); (8,0)
  (0,4); (2,4); (4,4); (6,4); (8,4)

[1 | 1]
mode_mask=2
position_width=352; position_height=576
(0,0); (352,0)

[1w+6]
mode_mask=2
position_width=704; position_height=192
(0,384)
position_width=234
(1,  0); (235,  0); (469,  0)
(1,192); (235,192); (469,192)

[1w+8]
mode_mask=2
position_width=704; position_height=288
(0,288)
position_width=176; position_height=144
(0,  0); (176,  0); (352,  0); (528,0  )
(0,144); (176,144); (352,144); (528,144)

[1-2]
position_width=352; position_height=288
(176,0)
(0,288); (352,288)

[2-3]
frame_width=6; frame_height=6;
position_width=2; position_height=2;
(1,1); (3,1); (0,3); (2,3); (4,3)
```

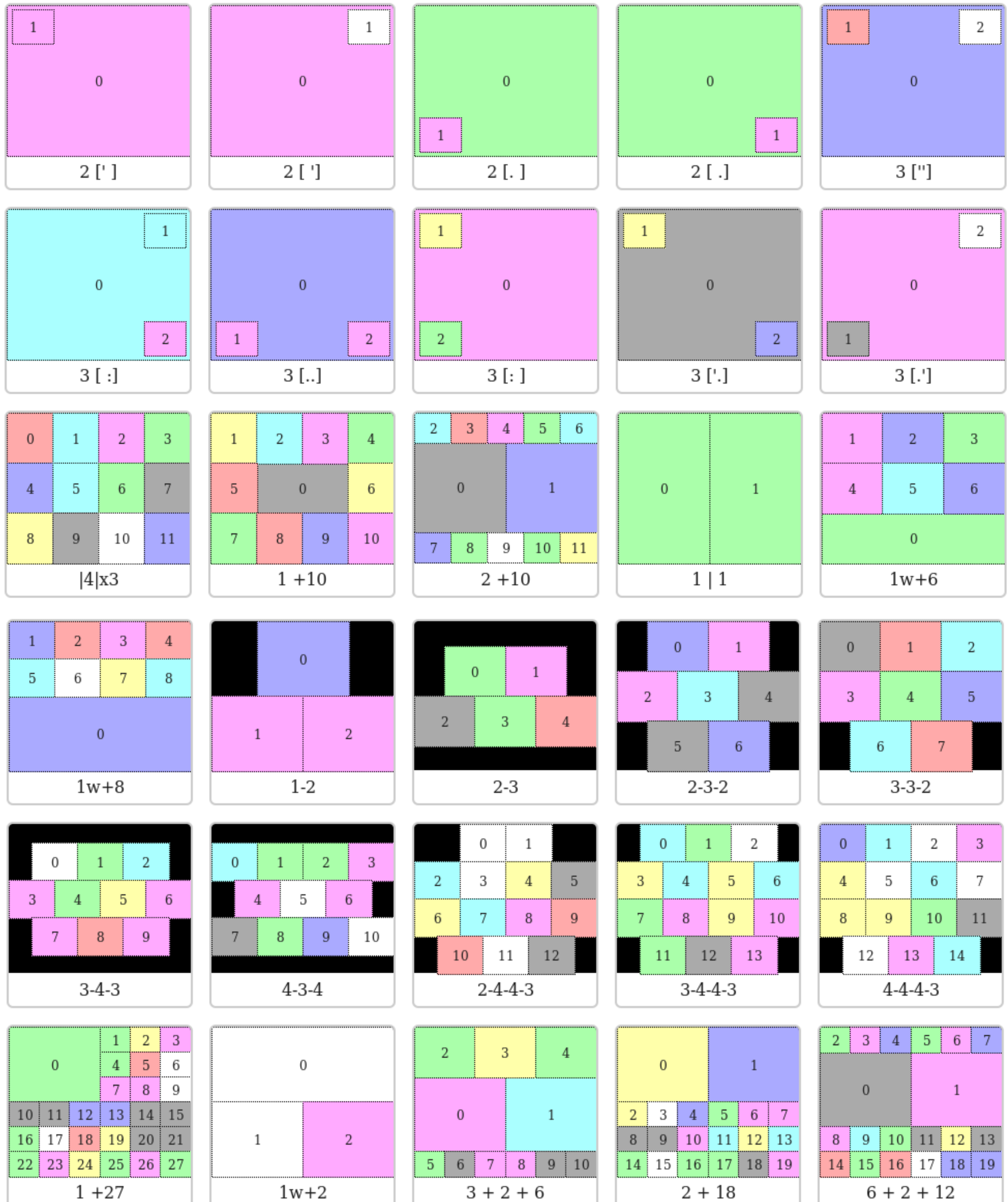
```
[2-3-2]
position_width=234; position_height=192
     (118,  0) ; (352,  0)
(  1,192) ; (235,192) ; (469,192)
     (118,384) ; (352,384)

[3-3-2]
position_width=234; position_height=192
(1,  0);(235,  0);(469,  0)
(1,192);(235,192);(469,192)
   (118,384);(352,384)

[3-4-3]
frame_width=8; frame_height=8; position_width=2; position_height=2
  (1,1);(3,1);(5,1)
(0,3);(2,3);(4,3);(6,3)
  (1,5);(3,5);(5,5)

[4-3-4]
frame_width=8; frame_height=8; position_width=2; position_height=2
(0,1);(2,1);(4,1);(6,1)
  (1,3);(3,3);(5,3)
(0,5);(2,5);(4,5);(6,5)

[2-4-4-3]
frame_width=8; frame_height=8; position_width=2; position_height=2
      (2,0);(4,0)
(0,2);(2,2);(4,2);(6,2)
(0,4);(2,4);(4,4);(6,4)
   (1,6);(3,6);(5,6)

[3-4-4-3]
frame_width=8; frame_height=8; position_width=2; position_height=2
   (1,0);(3,0);(5,0)
(0,2);(2,2);(4,2);(6,2)
(0,4);(2,4);(4,4);(6,4)
   (1,6);(3,6);(5,6)

[4-4-4-3]
frame_width=8; frame_height=8; position_width=2; position_height=2
(0,0);(2,0);(4,0);(6,0)
(0,2);(2,2);(4,2);(6,2)
(0,4);(2,4);(4,4);(6,4)
   (1,6);(3,6);(5,6)


/////////////////////////////////////////
///// 6. Sent by openmcu.ru members /////
/////////////////////////////////////////

[1 +27] // By vol4enok
mode_mask=2
frame_width=6;frame_height=6
position_width=3;position_height=3;(0,0)
position_width=1;position_height=1
                  (3,0);(4,0);(5,0)
                  (3,1);(4,1);(5,1)
                  (3,2);(4,2);(5,2)
(0,3);(1,3);(2,3);(3,3);(4,3);(5,3)
(0,4);(1,4);(2,4);(3,4);(4,4);(5,4)
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5)

[1w+2] // By vol4enok
mode_mask=2
```

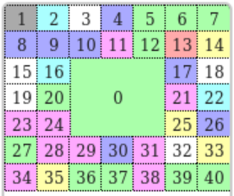
```
frame_width=2;frame_height=2
position_width=2;position_height=1;(0,0)
position_width=1;position_height=1;
(0,1);(1,1)

[3 + 2 + 6] // By vol4enok
mode_mask=2
frame_width=6;frame_height=6
position_width=3;position_height=3;(0,2);(3,2)
position_width=2;position_height=2;(0,0);(2,0);(4,0)
position_width=1;position_height=1
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5)

[2 + 18] // By vol4enok
mode_mask=2
frame_width=6;frame_height=6
position_width=3;position_height=3;(0,0);(3,0)
position_width=1;position_height=1
(0,3);(1,3);(2,3);(3,3);(4,3);(5,3)
(0,4);(1,4);(2,4);(3,4);(4,4);(5,4)
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5)

[6 + 2 + 12] // By vol4enok
mode_mask=2
frame_width=6;frame_height=6
position_width=3;position_height=3;(0,1);(3,1)
position_width=1;position_height=1
(0,0);(1,0);(2,0);(3,0);(4,0);(5,0)
position_width=1;position_height=1
(0,4);(1,4);(2,4);(3,4);(4,4);(5,4)
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5)

[1c+40] // by palexa
mode_mask=2
//ÐšÐ¾Ð»Ð¸Ñ‡ÐµÑ□Ñ‚Ð²Ð¾ Ð¿Ð¾Ð·Ð¸Ñ†Ð¸Ð¹ Ð¿Ð¾ ÑˆÐ¸Ñ€Ð¸Ð½Ðµ Ð¸ Ð²Ñ‹Ñ□Ð¾Ñ‚Ðµ
frame_width=7;frame_height=7
//Ð¦ÐµÐ½Ñ‚Ñ€Ð°Ð»ÑŒÐ½Ð°Ñ□ Ð¿Ð¾Ð·Ð¸Ñ†Ð¸Ñ□ 3Ñ…3 Ð¸ ÐµÑ‘ Ð¿Ð¾Ð»Ð¾Ð¶ÐµÐ½Ð¸Ðµ
position_width=3; position_height=3; (2,2)
//ÐžÑ□Ñ‚Ð°Ð»ÑŒÐ½Ñ‹Ðµ Ð¿Ð¾Ð·Ð¸Ñ†Ð¸Ð¸ 1Ñ…1 Ð¸ Ð¸Ñ… Ð¿Ð¾Ð»Ð¾Ð¶ÐµÐ½Ð¸Ðµ
position_width=1; position_height=1
(0,0);(1,0);(2,0);(3,0);(4,0);(5,0);(6,0)
(0,1);(1,1);(2,1);(3,1);(4,1);(5,1);(6,1)
(0,2);(1,2);                  (5,2);(6,2)
(0,3);(1,3);                  (5,3);(6,3)
(0,4);(1,4);                  (5,4);(6,4)
(0,5);(1,5);(2,5);(3,5);(4,5);(5,5);(6,5)
(0,6);(1,6);(2,6);(3,6);(4,6);(5,6);(6,6)

[2v + 12_]
mode_mask=2
frame_width=12; frame_height=6
position_width=6; position_height=3
(0,0); (6,0)
position_width=2; position_height=1
(0,3); (2,3); (4,3); (6,3); (8,3); (10,3)
(0,4); (2,4); (4,4); (6,4); (8,4); (10,4)

[2v + 8]
mode_mask=2
frame_width=8; frame_height=4
position_width=4; position_height=2
(0,0); (4,0)
position_width=2; position_height=1
(0,2); (2,2); (4,2); (6,2)
```

```
(0,3); (2,3); (4,3); (6,3)

[2v + 32]
mode_mask=2
frame_width=16; frame_height=8
position_width=8; position_height=4
(0,0); (8,0)
position_width=2; position_height=1
(0,4); (2,4); (4,4); (6,4); (8,4); (10,4); (12,4); (14,4)
(0,5); (2,5); (4,5); (6,5); (8,5); (10,5); (12,5); (14,5)
(0,6); (2,6); (4,6); (6,6); (8,6); (10,6); (12,6); (14,6)
(0,7); (2,7); (4,7); (6,7); (8,7); (10,7); (12,7); (14,7)

[2v + 21]
mode_mask=2
frame_width=14; frame_height=7
position_width=7; position_height=4
(0,0); (7,0)
position_width=2; position_height=1
(0,4); (2,4); (4,4); (6,4); (8,4); (10,4); (12,4)
(0,5); (2,5); (4,5); (6,5); (8,5); (10,5); (12,5)
(0,6); (2,6); (4,6); (6,6); (8,6); (10,6); (12,6)

[2v + 10]
mode_mask=2
frame_width=10; frame_height=5
position_width=5; position_height=3
(0,0); (5,0)
position_width=2; position_height=1
(0,3); (2,3); (4,3); (6,3); (8,3)
(0,4); (2,4); (4,4); (6,4); (8,4)

[2v + 12]
mode_mask=2
frame_width=12; frame_height=6
position_width=6; position_height=4
(0,0); (6,0)
position_width=2; position_height=1
(0,4); (2,4); (4,4); (6,4); (8,4); (10,4)
(0,5); (2,5); (4,5); (6,5); (8,5); (10,5)
```

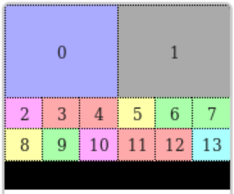
Available Display Matrices Layout:



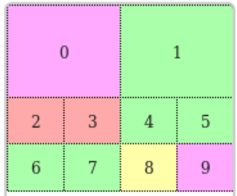

2 [']
2 [']
2 [.]
2 [.]
3 ['']
3 [:]
3 [..]
3 [:]
3 ['.]
3 [.']
|4|x3
1 +10
2 +10
1 | 1
1w+6
1w+8
1-2
2-3
2-3-2
3-3-2
3-4-3
4-3-4
2-4-4-3
3-4-4-3
4-4-4-3
1 +27
1w+2
3 + 2 + 6
2 + 18
6 + 2 + 12



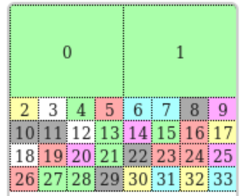
1c+40



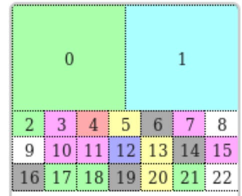
2v + 12_



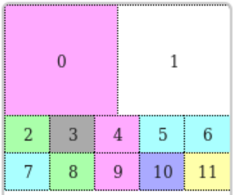
2v + 8



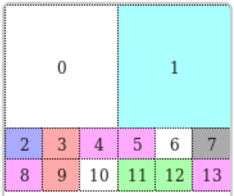
2v + 32



2v + 21



2v + 10



2v + 12

22. Quality of Service

Note: This is a simple overview of QoS that is to provide an awareness of the issues of running a video conference server with voice and data traffic. QoS is internally configured within OpenMCU-ru and there is no mechanism to easily change QoS settings.

There are two basic QoS mechanisms in place for IP packets: Type of Service (ToS) and Differentiated Services Code Point (DiffServ or DSCP) which use the IP packet's ToS/DSCP field (8 bits). Diffserv is an improved version of ToS and is backwards compatible.

Type of Service (ToS):

- ToS utilizes the first 3 bits and allows a priority range of 0 to 7
  - Larger ToS value means traffic has higher priority
  - Data traffic has a default priority of 0 for ToS
  - Call setup protocols like SIP have a default priority of 3 for ToS
  - Video traffic has a default priority of 4 for ToS
  - Voice RTP traffic has a default priority of 5 for ToS

DiffServ:

- DiffServ uses 6 bits of the ToS/DSCP field and provides the DiffServ "value"
  - The first 3 bits are now called the Precedence bits and provide the priority (just like ToS) which allows backwards compatibility.
  - The next 3 bits provide the Drop Probability value which is used to "fine-tune" priorities.
  - Priority bits (3) and the Drop Probability bits (3) combine to provide a DiffServ range of values from 0 to 63.
  - The larger the DiffServ value, the higher the priority
  - Data traffic has a default value of 0 for DiffServ (best effort)
  - Video traffic has a default value of 32-40 for ToS
  - Voice traffic has a default priority of 46 for ToS
  - The Drop Probability value is used to "fine-tune" priorities. For example, to give an e911 trunk priority over normal IP telephony traffic:

    - IP telephony traffic is assigned the DiffServ value of 40 (101000) – priority 5 (101), drop probability 0 (000).
    - E911 trunk is assigned a DiffServ value of 46 (101110) – still priority 5 (101) but drop probablity 6 (110)
- Voice RTP traffic has priority over video RTP traffic and call setup protocols like SIP and H.323.

Important

When configuring/designing your network traffic flow, be aware that heavy ***voice*** traffic can cause OpenMCU-ru ***video*** to break up. The solution is to move the OpenMCU-ru server to its own separate video subnet.

This could be a problem if OpenMCU-ru is installed on FreePBX. The voice traffic (priority 5) from FreePBX could interfere with the video traffic (priority 4) from OpenMCU-ru.

23. OpenMCU-ru Initialization File

The OpenMCU-ru GUI modifies the openmcu.ini text file located at /opt/openmcu-ru/config directory. You can edit it with a text editor or using WinSCP to access the server. Normally, you would not modify this file directly.

Reasons to modify this file:

For example if you set the https protocol ***on*** and lock yourself out from the web GUI. You could change this line to "`Enable HTTP secure=`**`FALSE`**" after which you would restart the server ("service openmcu-ru restart") and be able to login using the regular HTTP protocol.

Example openmcu.ini file:

```
[RECEIVE_SOUND]
SILK_B40{sw}=True
SILK_B40_24K{sw}=True
OPUS_48K2{sw}=True
OPUS_48K{sw}=True
OPUS_16K{sw}=True
OPUS_8K{sw}=True
G.711-uLaw-64k{sw}=True
G.711-ALaw-64k{sw}=True
G.7231-6.3k[e]{sw}=True
G.729A{sw}=True
G.729{sw}=True
Speex_32K{sw}=True
Speex_16K{sw}=True
Speex_8K{sw}=True
G.726-40k{sw}=True
G.726-32k{sw}=True
G.726-24k{sw}=True
G.726-16k{sw}=True
iLBC-13k3{sw}=True
iLBC-15k2{sw}=True
MS-IMA-ADPCM{sw}=True
G.728-16k[e]{sw}=True
MS-GSM{sw}=True
G.722-64k{sw}=True
GSM-06.10{sw}=True
G.722.1-24K{sw}=True
G.722.1-32K{sw}=True
G.722.1C-24K{sw}=True
G.722.1C-32K{sw}=True
G.722.1C-48K{sw}=True
G.722.2{sw}=True

[RECEIVE_VIDEO]
VP8{sw}=True
H.264{sw}=True
H.263p{sw}=True
H.263{sw}=True
H.261{sw}=True
MP4V-ES{sw}=True

[TRANSMIT_SOUND]
OPUS_48K2{sw}=True
OPUS_48K{sw}=True
SILK_B40_24K{sw}=True
SILK_B40{sw}=True
OPUS_16K{sw}=True
Speex_32K{sw}=True
Speex_16K{sw}=True
G.722-64k{sw}=True
OPUS_8K{sw}=True
Speex_8K{sw}=True
G.711-uLaw-64k=True
G.711-ALaw-64k=True
G.7231-6.3k[e]{sw}=True
```

```
G.729A{sw}=True
G.729{sw}=True
G.726-40k{sw}=True
G.726-32k{sw}=True
G.726-24k{sw}=True
G.726-16k{sw}=True
iLBC-13k3{sw}=True
iLBC-15k2{sw}=True
G.722.1-24K{sw}=True
G.722.1-32K{sw}=True
G.722.1C-24K{sw}=True
G.722.1C-32K{sw}=True
G.722.1C-48K{sw}=True
G.722.2{sw}=True

[TRANSMIT_VIDEO]
VP8{sw}=True
H.264{sw}=True
H.263p{sw}=True
H.263{sw}=True
H.261{sw}=True
MP4V-ES{sw}=True

[SIP Audio]
OPUS_48K2{sw}=True
OPUS_48K{sw}=True
SILK_B40_24K{sw}=True
SILK_B40{sw}=True
OPUS_16K{sw}=True
Speex_32K{sw}=True
Speex_16K{sw}=True
G.722-64k{sw}=True
OPUS_8K{sw}=True
Speex_8K{sw}=True
G.711-uLaw-64k{sw}=True
G.711-ALaw-64k{sw}=True
G.7231-6.3k[e]{sw}=True
G.729A{sw}=True
G.729{sw}=True
G.726-40k{sw}=True
G.726-32k{sw}=True
G.726-24k{sw}=True
G.726-16k{sw}=True
iLBC-13k3{sw}=True
iLBC-15k2{sw}=True
G.722.1-24K{sw}=True
G.722.1-32K{sw}=True
G.722.1C-24K{sw}=True
G.722.1C-32K{sw}=True
G.722.1C-48K{sw}=True
G.722.2{sw}=True

[Managing Groups]

[H323 Parameters]
H.323 Listener=*:1720

[H323 Endpoint *]

[H323 Endpoint empty]

[SIP Endpoint *]

[SIP Endpoint empty]

[Export Parameters]
RESTORE DEFAULTS=FALSE
Enable export=TRUE
Video frame width=704
Video frame height=576
Video frame rate=10
Audio sample rate=16000
Audio channels=1

[Managing Users]
admin=voipuser,administrator

[Telnet Server]
RESTORE DEFAULTS=FALSE
Enable=TRUE
Telnet Listener=*:1423
Username=admin
Password=voipuser
```

```
[Conference *]
Auto create when connecting=Disable
Force split screen video=Enable
Auto delete empty=Disable
Auto record start=Disable
Auto record stop=Disable
Recall last template=Disable
Template locks conference by default=Disable

[Conference 1001]

[SIP Video]
VP8{sw}=FALSE
H.264{sw}=TRUE
H.263p{sw}=FALSE
H.263{sw}=FALSE
MP4V-ES{sw}=TRUE
H.261{sw}=FALSE

[SIP Proxy Account 1001@192.168.204.245]
Enable=FALSE
Room=1001
Address=192.168.204.245
Password=1001
Expires=600

[SIP Proxy Account 1001@192.168.204.218]
Enable=TRUE
Room=1001
Address=192.168.204.218
Password=3951e4d7
Expires=600

[Address book scarecrow]
Scheme=sip
Host=192.168.202.11
Display name=scarecrow

[Parameters]
RESTORE DEFAULTS=FALSE
OpenMCU-ru Server Id=OpenMCU-ru v4.1.3
Default protocol for outgoing calls=sip
HTTP IP=0.0.0.0
HTTP Port=1420
Enable HTTP secure=FALSE
HTTP Certificate=/opt/openmcu-ru/ssl/http.pem
RTP Base Port=10000
RTP Max Port=20000
Trace level=0
Rotate trace files at startup=0
Log Level=0
Call log filename=/opt/openmcu-ru/log/mcu_log.txt
Room control event buffer size=100
Copy web log to call log=FALSE
Default room=1001
Reject duplicate name=FALSE
Allow loopback calls=FALSE
Auto dial delay=1
Video Recorder directory=/opt/openmcu-ru/records
Video Recorder video codec=mpeg4
Video Recorder resolution=704x576
Video Recorder frame rate=10
Video Recorder video bitrate=0
Video Recorder audio codec=ac3
Video Recorder sound rate=16000
Video Recorder sound channels=1
```

24. Connecting to FreePBX/Asterisk

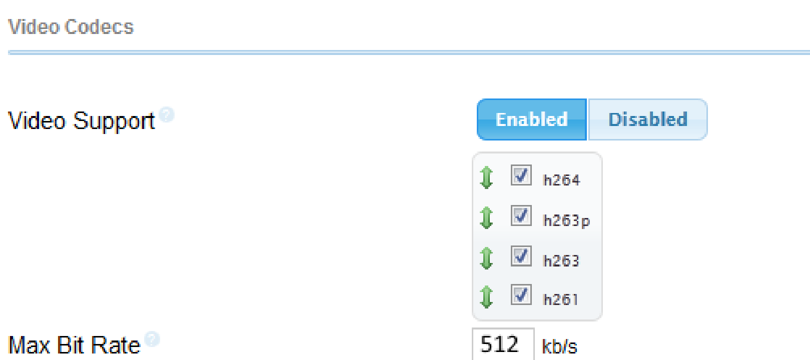
The following steps document connecting to an Asterisk PBX using the FreePBX distribution of Asterisk. It is assumed that you are familiar with FreePBX. The basic steps are:

- Enable video support in FreePBX
- Create extensions for Linphones
- Enable H.264 codec, disable VP8 codec
- Create an extension for OpenMCU-ru in FreePBX
- Configure OpenMCU-ru Account Information to use FreePBX extension

Enable video support in FreePBX

1. Go to FreePBX – Settings - Asterisk SIP Settings – Video Codecs and Enable Video Support then Submit and Apply .

2. Go back to Enable Video Support and select H.264 only to start.
    - Max bit rate: **512** kb/s



3. Set “Reinvite Behavior” to “Yes”

4. Scroll down to the bottom and at "Other SIP Settings" add "directrtpsetup = yes"

5. Submit and Apply

Configure Linphones as extensions on FreePBX

6. Configure SIP extensions for the Linphones on FreePBX.

7. Set “Canreinvite: YES”

8. Disable VP8 codec so only H.264 codec is present.

9. Configure Linphone account to register to your PBX

10. After registering to your PBX, perform *65 (speak your extension) to verify that the phone is registered.

11. Dial *43 (echo test) to verify that audio is properly being transmitted. Video should be echoed also (wait, it takes a short time).

12. Make a video call from one Linphone to the other using the extension number.

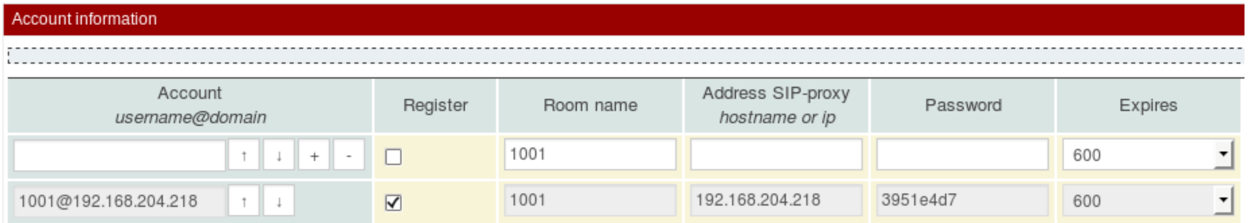
Creating FreePBX extension for Conference Room 1001

13. Make a SIP extension in FreePBX for the conference room 1001.
    - Extension: **1001**
    - Display name: **Conference 1001**
    - Secret: *******
    - Canreinvite: **YES**

Configure Account Information for Conference Room 1001

14. In OpenMCU-ru, go to Settings – SIP – Account Information:

    - Account: **1001@<PBX IP address>**
    - **Check** Register box
    - Room name: **1001**
    - Address SIP-proxy: **<PBX IP address>**
    - Password: **<extension secret>**
    - Expires: **600**



15. Make a phone call to extension 1001 from both your Linphones. You should connect into conference room 1001 with excellent video quality.

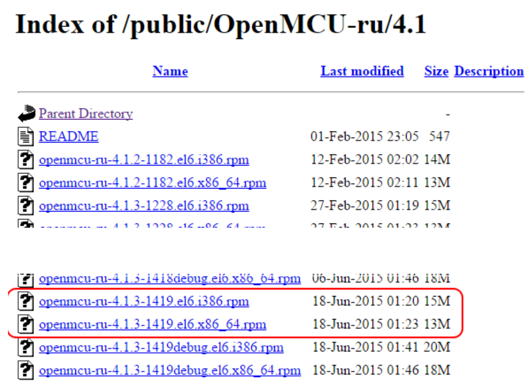
25. Installation on FreePBX Distribution:

This procedure installs OpenMCU-ru on an existing ***64 bit*** FreePBX distribution.

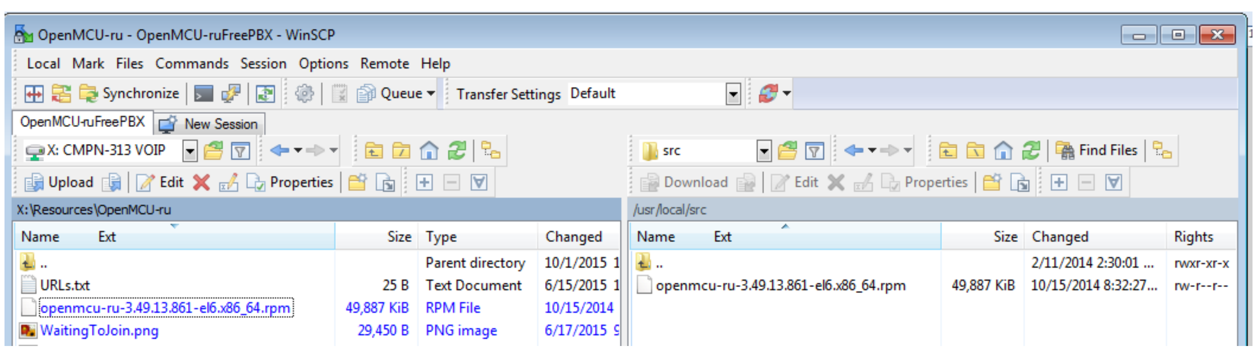
Important

Please consult the section of QoS and be aware that voice traffic has priority over video traffic and this can affect performance and the quality of video from OpenMCU-ru.

1. Open a web browser on a PC and go to http://openmcu.ru/public/OpenMCU-ru/ and find the latest version of OpenMCU-ru. At the time of writing, 4.1.3 was the latest:



2. I used the free WinSCP to copy from my PC to the FreePBX /usr/local/src directory. I used the root user to connect.



3. I used the PuTTY program to SSH into the FreePBX server and then moved to the /usr/local/src directory. Then installed the rpm:

```
[root@localhost ~] cd /usr/local/src
[root@localhost  src]  ls
openmcu-ru-4.1.6-1430.el6.x86_64.rpm
[root@localhost  src]  rpm -Uvh  openmcu-ru-4.1.6-1430.el6.x86_64.rpm
```

```
[root@localhost ~]# cd /usr/local/src
[root@localhost src]# ls
openmcu-ru-4.1.6-1430.el6.x86_64.rpm
[root@localhost src]# rpm -Uvh openmcu-ru-4.1.6-1430.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:openmcu-ru             ########################################### [100%]
Starting openmcu-ru:
Message from syslogd@localhost at Oct  2 15:36:21 ...
 OpenMCU-ru[21413]: Starting service process "OpenMCU-ru" v4.1.6
[  OK  ]
[root@localhost src]#
```

Note: If the SSHD daemon is not running,

- check if it has been installed by typing: "ls /etc/init.d" Look for sshd to be present.
- If it is not installed, install it by typing: "yum –y install openssh-server openssh-clients"
- To run sshd, type "service sshd start" or "/etc/init.d sshd start".
- To make sshd start automatically each time the server boots, type "chkconfig sshd on".

4. A successful installation of OpenMCU-ru will end with the following message:

   OpenMCU-ru[9846]: Starting service process "OpenMCU-ru" 4.1.6

5. OpenMCU-ru is installed in /opt/openmcu-ru directory:

```
[root@localhost src]# ls /opt/openmcu-ru/
AUTHORS  config   font     lib  NEWS  README   resource  share
bin      COPYING  include  log  pipe  records  scripts   ssl
```

6. Once installed, check that OpenMCU-ru is running. As the root user, from a terminal window type:

   service openmcu-ru status
   Or
   /etc/init.d/openmcu-ru status

```
[root@localhost student]# service openmcu-ru status
openmcu-ru (pid  1467) is running...
```

7. Starting and stopping the server (rarely ever need to):

   service openmcu-ru start/stop
   Or
   /etc/init.d/openmcu-ru start/stop

Network Interfaces (dual and virtual)

There is a problem with having both FreePBX and OpenMCU-ru on the same server. Both services will respond to SIP requests on the server IP address. For example, if a SIP request comes in on the server's IP address 192.168.202.251:5060 and both servers are listening to the standard SIP port of 5060, which server will respond? The answer is one or the other or both! Not a stable system.

A "solution" is to change the SIP port to a non-standard port for one of the servers but this can cause more problems with incompatibilities from end devices attempting to connect.

Two better solutions:

a. If your server has two network cards, assign one for FreePBX and the other for OpenMCU-ru. You can connect both network cards to the same network with separate IP addresses. The following information concentrates on the second solution.
b. If you have only one network card, create a virtual network interface (eth0:0) in addition to the existing network interface (eth0). This allows the servers to listen on separate IP addresses for the SIP requests while still using the standard SIP port of 5060.

8. Set your eth0 interface to a static IP address. Edit /etc/sysconfig/network-scripts/ifcfg-eth0 using your favorite editor. My preference is to SSH in and use the nano text editor.

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# nano ifcfg-eth0
```

9. Set ifcfg-eth0 to a static IP address:

DEVICE ="eth0"
BOOTPRO =" "
IPV6INIT ="no"
IPV6_AUTOCONF="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR=192.168.204.251
NETMASK=255.255.255.0
GATEWAY=192.168.204.1

10. ***If*** you have ***2*** network cards, configure ifcfg-eth1 for a second IP address, OTHERWISE skip this step:

DEVICE ="eth1"
BOOTPRO =" "
IPV6INIT ="no"
IPV6_AUTOCONF="no"

ONBOOT="yes"
TYPE="Ethernet"
IPADDR=192.168.204.252
NETMASK=255.255.255.0
GATEWAY=192.168.204.1

11. Restart your network services from the Linux command prompt. To stop and start the network interfaces:

    "service network stop/start/restart"

    If you SSH'd in, you will ***lose*** your network connection because the IP address changed. SSH into the new static IP address.

    To check that the network interfaces have accepted the new configuration, type "ifconfig"

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 1E:4B:6B:A5:4C:63
          inet addr:192.168.204.251  Bcast:192.168.204.255  Mask:255.255.255.0
          inet6 addr: fe80::1c4b:6bff:fea5:4c63/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344894 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206489871 (196.9 MiB)  TX bytes:27671985 (26.3 MiB)
```

    If you configured two network cards and they are connected to the network, then you should see eth0 and eth1 present. Skip ahead to Step 24.

Creating a Virtual Interface

If you do ***NOT*** have a second network interface, create a virtual interface using the following steps:

12. Create virtual interface eth0:0 by copying /etc/sysconfig/network-scripts/ifcfg-eth0 to ifcfg-eth0:0 and modifying it with a new device of eth0:0 and new IP address.

    For example, my FreePBX server's interface is eth0 with an IP address of 192.168.204.251, OpenMCU-ru's interface is eth0:0 with an IP address of 192.168.204.252. I copied from the command line by ssh'ing in using Putty:

```
[root@localhost src]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# cp ifcfg-eth0 ifcfg-eth0:0
[root@localhost network-scripts]# nano ifcfg-eth0:0
```

13. I used the text editor nano to modify the new ifcfg-eth0:0 file:

    DEVICE =”eth0:0”
    BOOTPRO =” ”
    IPV6INIT =”no”
    IPV6_AUTOCONF=”no”
    ONBOOT=”yes”
    TYPE=”Ethernet”
    IPADDR=192.168.204.252
    NETMASK=255.255.255.0
    GATEWAY=192.168.204.1

14. Restart your network services from the Linux command prompt.

    TIP: *If you have PuTTY on your PC, you can integrate it with WinSCP Command menu and run it from WinSCP which is real nice.*

    To stop and start the network interfaces: “service network stop/start/restart”

    To check that the network interfaces have accepted the new configuration, type “ifconfig”

```
[root@localhost ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:  Determining if ip address 192.168.204.251 is already in use for device eth0...
Determining if ip address 192.168.204.252 is already in use for device eth0...
                                                           [  OK  ]
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 1E:4B:6B:A5:4C:63
          inet addr:192.168.204.251  Bcast:192.168.204.255  Mask:255.255.255.0
          inet6 addr: fe80::1c4b:6bff:fea5:4c63/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344894 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206489871 (196.9 MiB)  TX bytes:27671985 (26.3 MiB)

eth0:0    Link encap:Ethernet  HWaddr 1E:4B:6B:A5:4C:63
          inet addr:192.168.204.252  Bcast:192.168.204.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

15. When you log into the server command line, you should see your two interfaces:

```
FreePBX

NOTICE! You have 6 notifications! Please log into the UI to see them!

Current Network Configuration
+-----------+-------------------+--------------------------+
| Interface | MAC Address       | IP Addresses             |
+-----------+-------------------+--------------------------+
| eth0      | EE:F2:65:E1:91:84 | 192.168.204.251          |
|           |                   | 192.168.204.252          |
|           |                   | fe80::ecf2:65ff:fee1:9184 |
+-----------+-------------------+--------------------------+
```

16. To access FreePBX in my system, I use the 192.168.204.251 IP address

17. To access OpenMCU-ru in my system, I use the 192.168.204.252 IP address

Configure FreePBX to identify SIP local networks

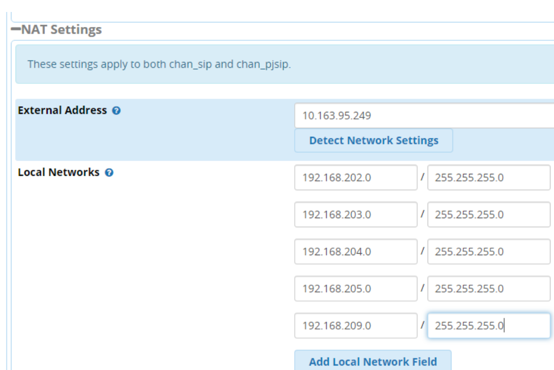
This is required so that FreePBX can recognize which local networks are behind NAT and which are public networks. FreePBX will modify the SIP headers to include the WAN/Public IP address in a STUN like fashion. If FreePBX does not know which networks are local and which are public you may have one of the following symptoms:

- one-way audio: call connects but one caller can't hear the other
- no audio: call connects but no audio
- connection disconnects after 7 second
- connection disconnects after 30 seconds
- call connects, audio present but no video

The solution is to inform FreePBX of the WAN/pubic IP address and which networks are local (private).

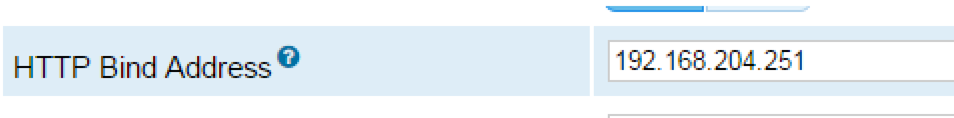
To access FreePBX, point your web browser to http://192.168.204.251, replace with an IP address that is appropriate for your configuration. You will be running FreePBX's initial setup. Consult the FreePBX documentation to configure it as that is beyond the scope of this guide.

18. Go to FreePBX – Settings – Asterisk SIP Settings, enter your public IP address and local network IP addresses. You may use "Detect Network Settings" to automatically detect. I prefer to manually set them.

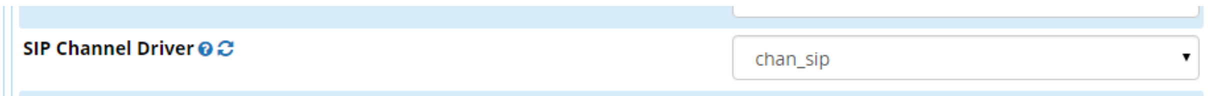


Configure FreePBX to bind to eth0's IP address only (ex. 192.168.204.251)

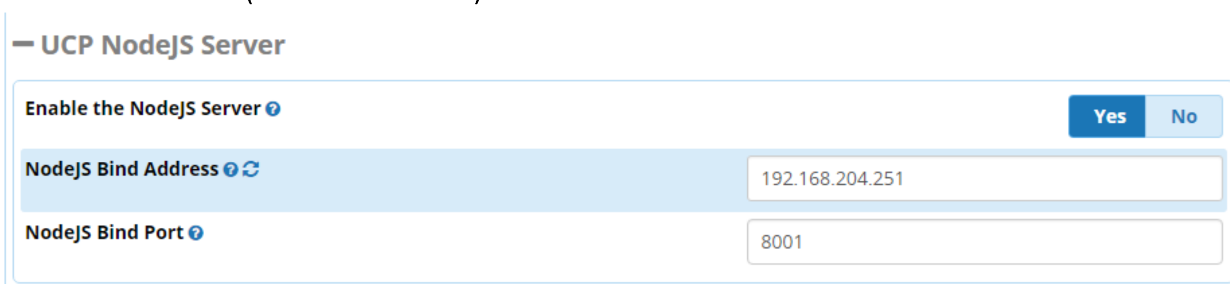
19. FreePBX – Settings – Advanced Settings:



20. Set the SIP channel driver to only chan_sip (at this time pjsip is experimental):



21. Bind the UCP server (User control Panel):



22. FreePBX – Settings – Asterisk IAX Settings (Advanced Setting tab):

23. FreePBX – Settings – Asterisk SIP Settings – Chan SIP:

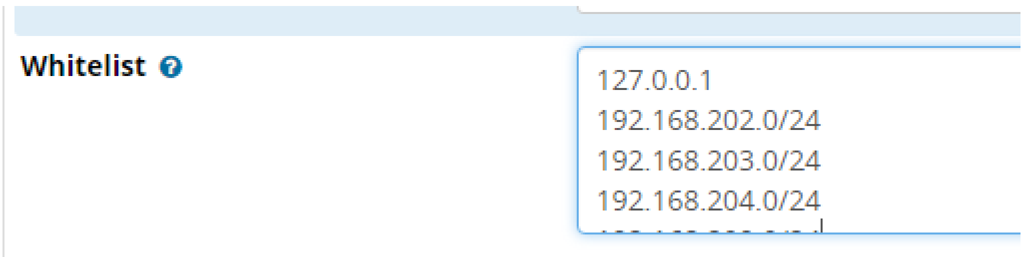
Troubleshooting: If a pop-up window appears when connecting to OpenMCU-ru asking for your username and password then FreePBX's SIP or PJSIP channel is listening (binding) to all interfaces. Go back , check the previous bindings listed and make sure that you restart the server using fwconsole or "shutdown –r now" for a complete reboot. You can also monitor from the Asterisk CLI> to see if it is intercepting the call:

FreePBX server uses the Fail2ban for intrusion detection. It is a program that monitors the logs of various services on your server. For example, if a user makes 3 failed login attempts within a set period of time, the user's IP address is banned for preset amount of time such as 30 minutes. *The following instruction will prevent the fail2ban from unintentionally banning the IP addresses of local devices on the Voice VLAN while attempting future labs.*

24. Go Admin – System Admin – Intrusion Detection and add all of your LAN ***networks*** in the Whitelist in this format:

<IP address><Mask Prefix>

for example: 192.168.204.0/24

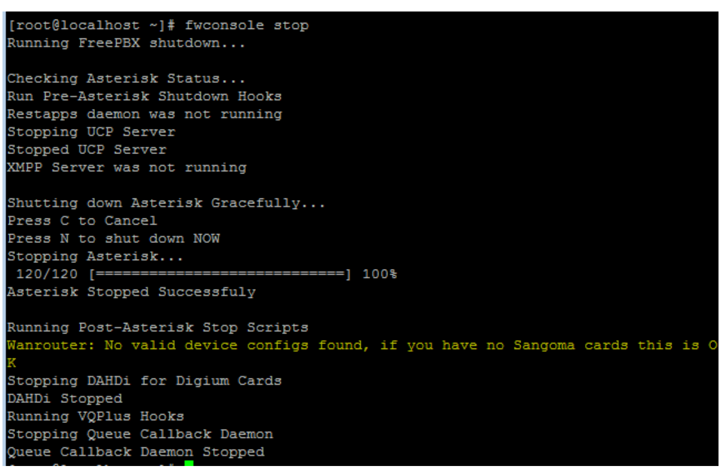


25. Enter your contact email.

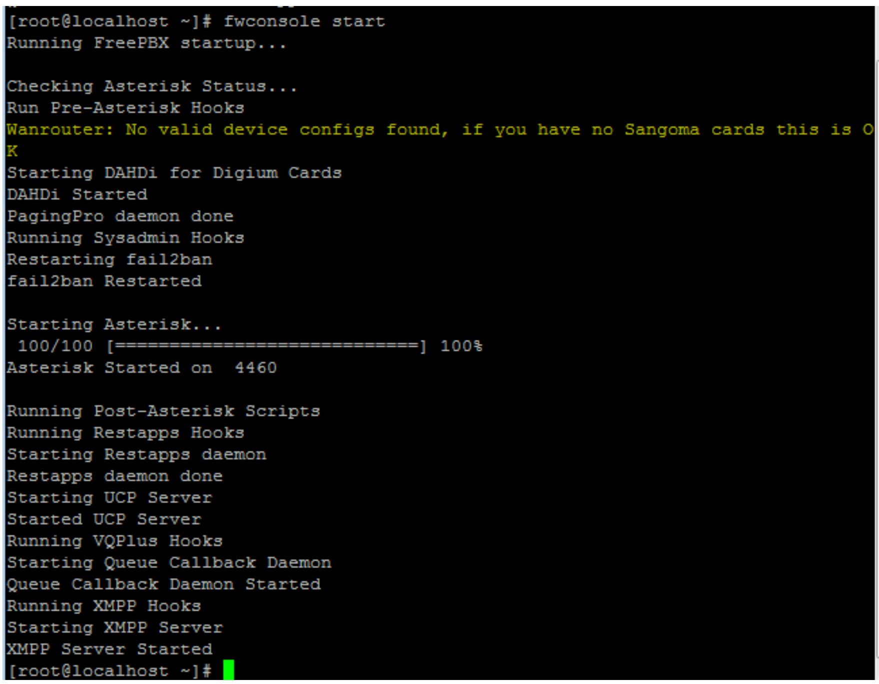
26. Submit Query then Restart the Intrusion Detection.

27. The Intrusion Detection menu allows you to check if an IP address has been banned. It is good to routinely check to see if someone is trying to brute hack your system. Email alerts can be sent also.

28. The Asterisk PBX engine must be shut down and restarted for the new protocol bindings to take effect. From the Linux command prompt, type “fwconsole stop”:



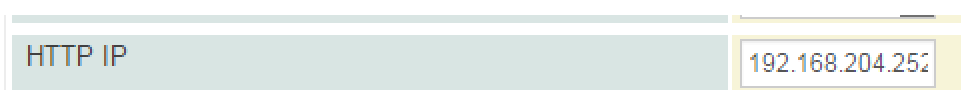
29. Then "fwconsole start":



```
[root@localhost ~]# fwconsole start
Running FreePBX startup...

Checking Asterisk Status...
Run Pre-Asterisk Hooks
Wanrouter: No valid device configs found, if you have no Sangoma cards this is O
K
Starting DAHDi for Digium Cards
DAHDi Started
PagingPro daemon done
Running Sysadmin Hooks
Restarting fail2ban
fail2ban Restarted

Starting Asterisk...
 100/100 [============================] 100%
Asterisk Started on  4460

Running Post-Asterisk Scripts
Running Restapps Hooks
Starting Restapps daemon
Restapps daemon done
Starting UCP Server
Started UCP Server
Running VQPlus Hooks
Starting Queue Callback Daemon
Queue Callback Daemon Started
Running XMPP Hooks
Starting XMPP Server
XMPP Server Started
[root@localhost ~]#
```

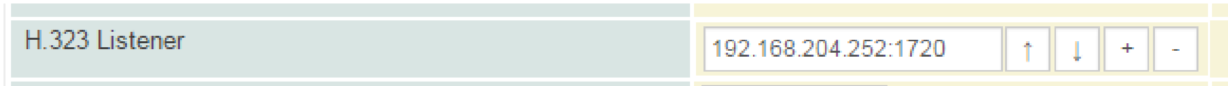
Configure OpenMCU-ru to bind to eth0:0's IP address only (ex. 192.168.204.252)

30. OpenMCU-ru – Settings – General, set HTTP IP:



31. OpenMCU-ru – Settings – Advanced – Telnet Server, set Telnet Listener:

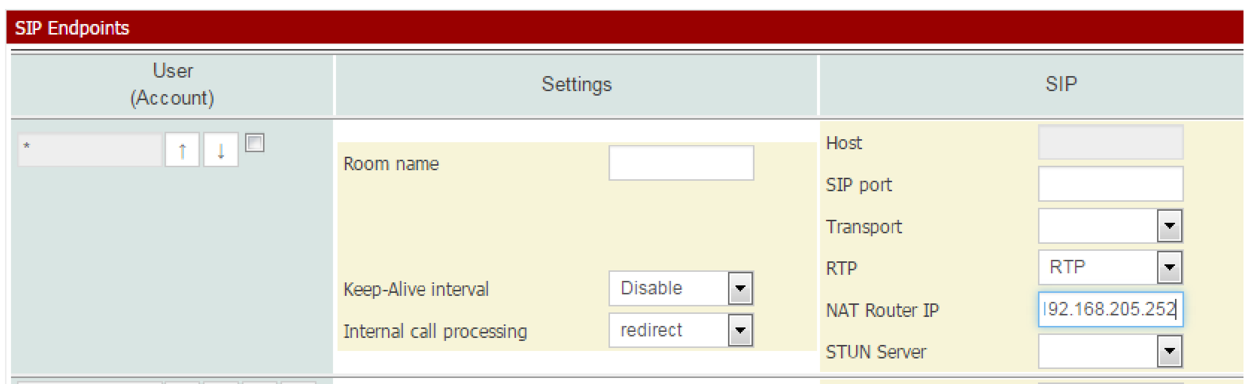
32. OpenMCU-ru – Settings – H.323 – H.323 Parameters, set H.323 Listener:



33. OpenMCU-ru – Settings – RTSP – RTSP Parameters, set RTSP Listener:

34. OpenMCU-ru – Settings – SIP – SIP Parameters:, set SIP Listener

35. OpenMCU-ru – Settings – SIP – Endpoints, set NAT Router IP. This really should be the public IP address of your network, not the private IP but we're playing games to get FreePBX and OpenMCU-ru to work together. May need to explore this further as it may cause problems down the road:



36. If you are doing web streaming then you should modify /opt/openmcu-ru/config/ffserver.conf to listen to the OpenMCU-ru interface. Modify this line:

BindAddress 0.0.0.0

Ex. BindAddress 192.168.204.252

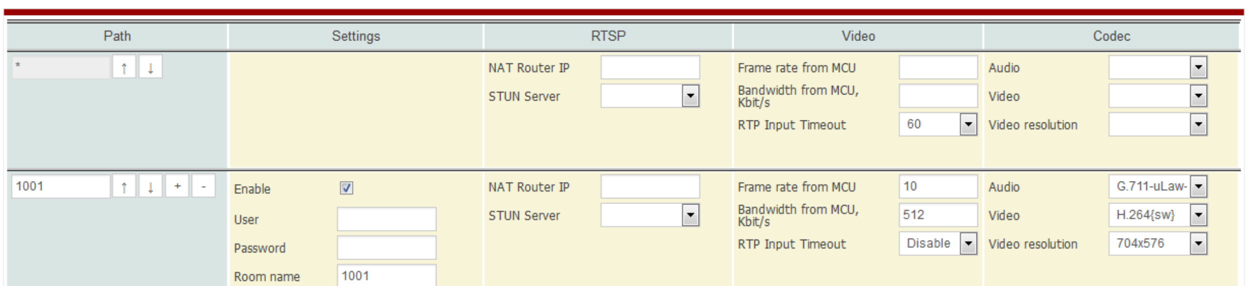
This should separate the two services running on the same physical server.

26. RTSP Streaming a Conference Room

OpenMCU-ru can stream a conference room (***display #0 only***) to the Internet. To view it, you need a media player such as VLC player to watch the stream. The quality is very good and this method is the preferred streaming method.

To start, you must create an RTSP server (by default this will automatically be created).

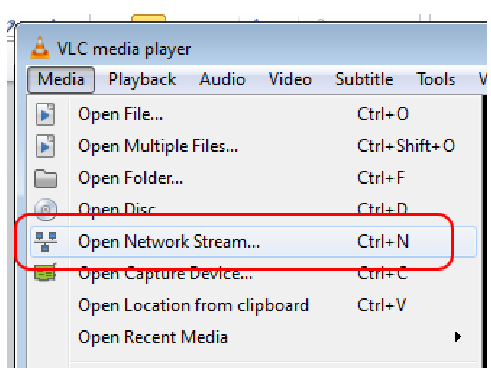
1. Go to Settings – RTSP – RTSP Servers:
    - Check the Enable box
    - Give it a "Path", in the example the path is "1001" (make it the same as the room name)
    - Assign it to a room through the "Room name". Ex. Room name 1001
    - Set the frame rate. Ex. Is conservatively set for 10 fps
    - Set the Bandwidth. 512 kbps seems to work quite well
    - Disable RTP Input Timeout.
    - Set the Audio coded to G.711-ulaw because I'm in N.A. G.711-alaw if anywhere else.
    - Set the Video codec to H.264(sw)
    - Set the Video resolution to 704x576

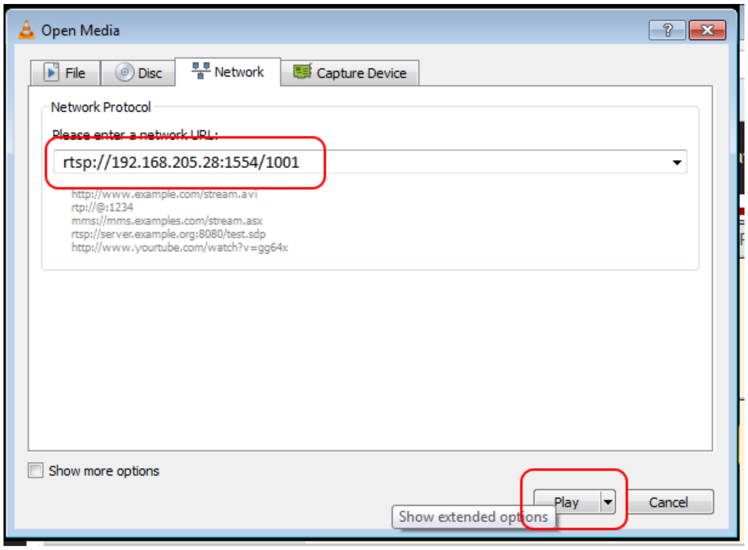


2. To view the network stream from a media player such as VLC, point the network stream to

rtsp://<OpenMCU-ru IP address>:1554/<path>

ex. rtsp://192.168.205.28:1554/1001





Observation: the web streamed video will show more vertical information then a normal conference room which is cropped.

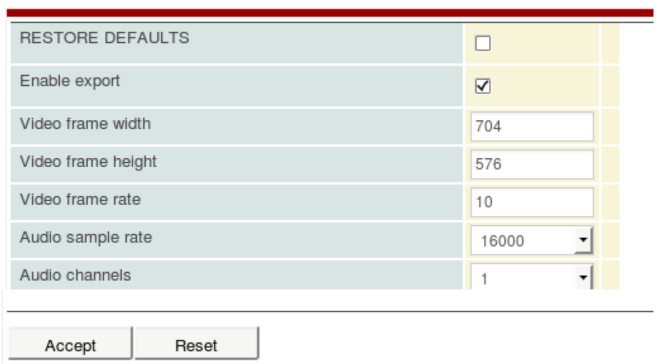
27. Web Streaming a Conference Room

OpenMCU-ru can stream a conference room (***display #0 only***) to the Internet. To view it, you only need a web browser to watch the stream. The quality is not very good compared to RTSP streaming. RTSP streaming is the preferred method.

Before the Conference room is created, you must allow creation of video and audio files to be streamed. These files are called pipes and reside in the /opt/openmcu-ru/pipe directory.

1. Go to Advanced –Export (named pipe), Enable export and set the desired video frame width, height and rate.

   Advanced – Export (named pipe)



I've used the default settings and found that it worked okay. Currently the only stream size available is 704x576 (4cif).

2. Create a new conference room (ex. 1001) and check in /opt/openmcu-ru/pipe for two files to be present: sound.1001 and video.1001. They will have a file size of "0"

```
[root@localhost openmcu-ru]# ls -l pipe
total 0
prw-r--r-- 1 mcu mcu 0 Nov 26 12:57 sound.1001
prw-r--r-- 1 mcu mcu 0 Nov 26 12:57 video.1001
```

3. Start the web stream for room ***1001*** by issuing the command at the Linux command line prompt :

```
/opt/openmcu-ru/scripts/web_stream_start 1001
```

4. You should see the stream start:

```
[root@localhost openmcu-ru]# ./scripts/web_stream_start 1001
ffserver version 0.10.4 Copyright (c) 2000-2012 the FFmpeg developers
  built on Jun  6 2015 00:12:49 with gcc 4.4.7 20120313 (Red Hat 4.4.7-4)
  configuration: --prefix=/opt/openmcu-ru --extra-cflags=-I/opt/openmcu-ru/inclu
de/ --extra-ldflags=-L/opt/openmcu-ru/lib/ --enable-libx264 --enable-libvpx --en
able-gpl --disable-static --enable-shared
  libavutil      51. 35.100 / 51. 35.100
  libavcodec     53. 61.100 / 53. 61.100
  libavformat    53. 32.100 / 53. 32.100
  libavdevice    53.  4.100 / 53.  4.100
  libavfilter     2. 61.100 /  2. 61.100
  libswscale      2.  1.100 /  2.  1.100
  libswresample   0.  6.100 /  0.  6.100
  libpostproc    52.  0.100 / 52.  0.100
ffmpeg version 0.10.4 Copyright (c) 2000-2012 the FFmpeg developers
  built on Jun  6 2015 00:12:49 with gcc 4.4.7 20120313 (Red Hat 4.4.7-4)
  configuration: --prefix=/opt/openmcu-ru --extra-cflags=-I/opt/openmcu-ru/inclu
de/ --extra-ldflags=-L/opt/openmcu-ru/lib/ --enable-libx264 --enable-libvpx --en
able-gpl --disable-static --enable-shared
  libavutil      51. 35.100 / 51. 35.100
  libavcodec     53. 61.100 / 53. 61.100
  libavformat    53. 32.100 / 53. 32.100
  libavdevice    53.  4.100 / 53.  4.100
  libavfilter     2. 61.100 /  2. 61.100
  libswscale      2.  1.100 /  2.  1.100
  libswresample   0.  6.100 /  0.  6.100
  libpostproc    52.  0.100 / 52.  0.100
Thu Nov 26 12:59:46 2015 FFserver started.
[rawvideo @ 0x18de5e0] Estimating duration from bitrate, this may be inaccurate
Input #0, rawvideo, from '/opt/openmcu-ru/pipe/video.1001':
  Duration: N/A, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: rawvideo (I420 / 0x30323449), yuv420p, 704x576, 10 tbr,
10 tbn, 10 tbc
[s16le @ 0x18ed4c0] Estimating duration from bitrate, this may be inaccurate
Input #1, s16le, from '/opt/openmcu-ru/pipe/sound.1001':
  Duration: N/A, start: 0.000000, bitrate: N/A
    Stream #1:0: Audio: pcm_s16le, 16000 Hz, 1 channels, s16, 256 kb/s
Thu Nov 26 12:59:47 2015 127.0.0.1 - - [GET] "/feed.ffm HTTP/1.1" 200 4175
[buffer @ 0x18fcca0] w:704 h:576 pixfmt:yuv420p tb:1/1000000 sar:0/1 sws_param:
Output #0, ffm, to 'http://127.0.0.1:8090/feed.ffm':
  Metadata:
    encoder         : Lavf53.32.100
    Stream #0:0: Audio: mp2, 16000 Hz, 1 channels, s16, 64 kb/s
    Stream #0:1: Video: msmpeg4v2, yuv420p, 704x576, q=2-31, 256 kb/s, 1000k tbn
, 10 tbc
Stream mapping:
  Stream #1:0 -> #0:0 (pcm_s16le -> mp2)
```

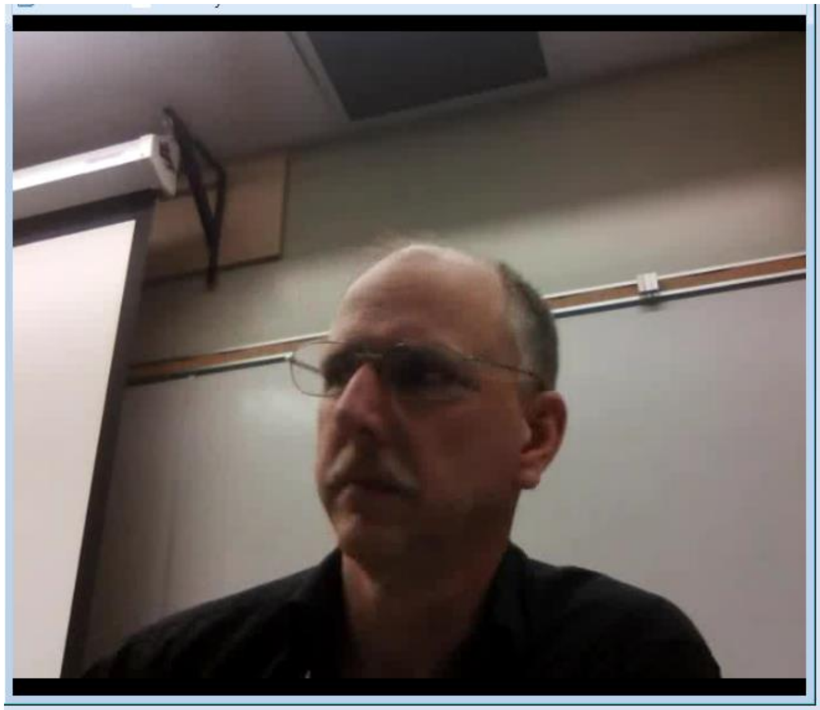
5. Once the stream is running, you should see the stream:

```
Stream mapping:
  Stream #1:0 -> #0:0 (pcm_s16le -> mp2)
  Stream #0:0 -> #0:1 (rawvideo -> msmpeg4v2)
Press [q] to stop, [?] for help
frame=    6 fps=  0 q=4.0 size=      68kB time=00:00:00.60 bitrate= 928.4kbits/s
frame=   12 fps= 11 q=3.0 size=      92kB time=00:00:01.15 bitrate= 654.2kbits/s
frame=   17 fps= 11 q=5.1 size=     136kB time=00:00:01.65 bitrate= 672.8kbits/s
frame=   22 fps= 11 q=4.4 size=     168kB time=00:00:02.16 bitrate= 637.2kbits/s
frame=   27 fps= 10 q=4.7 size=     208kB time=00:00:02.66 bitrate= 639.6kbits/s
frame=   32 fps= 10 q=4.0 size=     240kB time=00:00:03.16 bitrate= 620.6kbits/s
frame=   37 fps= 10 q=16.5 size=     264kB time=00:00:03.67 bitrate= 589.0kbits/
frame=   42 fps= 10 q=5.6 size=     308kB time=00:00:04.20 bitrate= 600.7kbits/s
frame=   47 fps= 10 q=5.6 size=     340kB time=00:00:04.70 bitrate= 592.6kbits/s
frame=   52 fps= 10 q=6.4 size=     380kB time=00:00:05.20 bitrate= 598.6kbits/s
frame=   58 fps= 10 q=4.0 size=     404kB time=00:00:05.76 bitrate= 574.6kbits/s
frame=   63 fps= 10 q=6.6 size=     464kB time=00:00:06.26 bitrate= 606.8kbits/s
```

6. To view the network stream from a web browser, point the browser to:

http://<OpenMCU-ru IP address>:8090/stream.asx

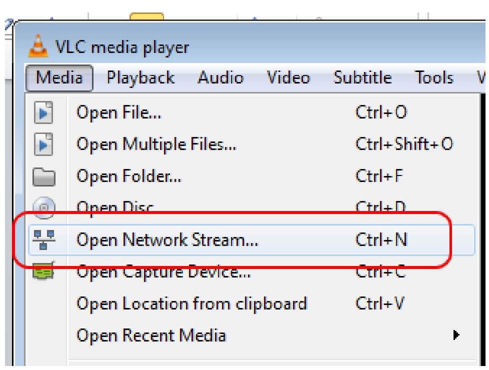
ex. http://192.168.205.28:8090/stream.asx

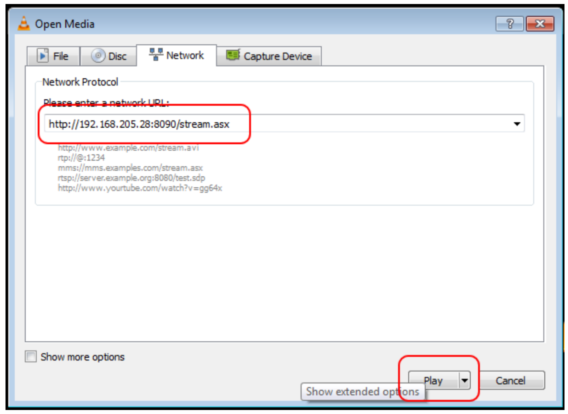


7. To view the network stream from a media player such as VLC, point the network stream to

http://<OpenMCU-ru IP address>:8090/stream.asx

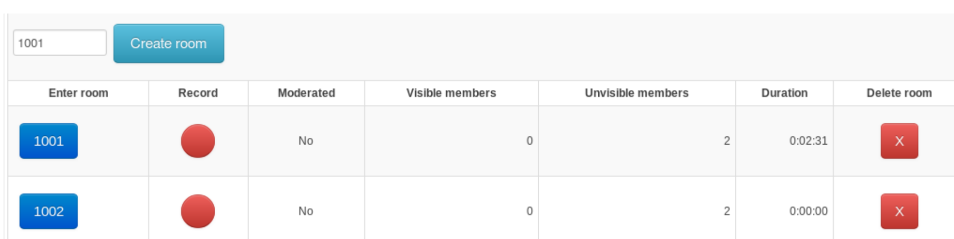
ex. http://192.168.205.28:8090/stream.asx





Observation: the web streamed video will show more vertical information then a normal conference room which is cropped. The stream uses the HTTP protocol running on the TCP protocol.

8. To ***stop*** the web streaming, you must ***delete*** the conference room in the Control menu.



9. To delete a room, press the delete room button , any participants who are connected will be disconnected and the web stream will stop.

Web Streaming Configuration

The following files configure the web streaming video size.

10. The configuration file for the web streaming is located at /opt/openmcu-ru/config/ffserver.conf. You can change the streaming port from 8090 in this file.

    A good resource for understanding the settings is located at

    https://www.ffmpeg.org/ffmpeg-utils.html

11. The configuration file /opt/openmcu-ru/config/openmcu.ini contains the information ***changed*** through the web gui. Do not manually configure this file.

12. The script /opt/openmcu-ru/scripts/web_stream_start contains the input format for streaming. The line with" ffmpeg –s 4cif" determines the video format. 4cif is the default 704x576.

Web Streaming Performance

Web streaming adds a load on the server compared to regular conference room creation. In testing, the server was run at the following states and performance markers were noted on the CPU, Network Traffic and Disk I/O graphs:

Point A: Server is idle, no conferences running. Running about 4% CPU

Point B: Created a conference room 1001 with no pipes and no participants. No change - 4% CPU

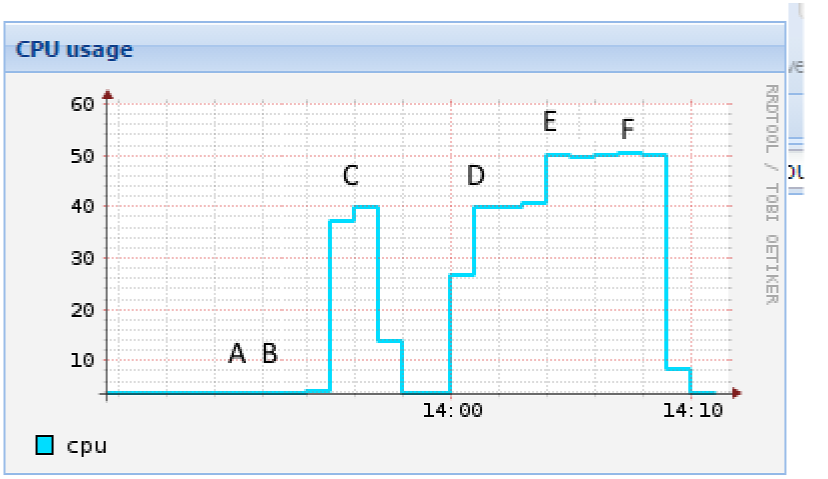
Point C: A single participant joined the conference room 1001. CPU utilization rose to 40%

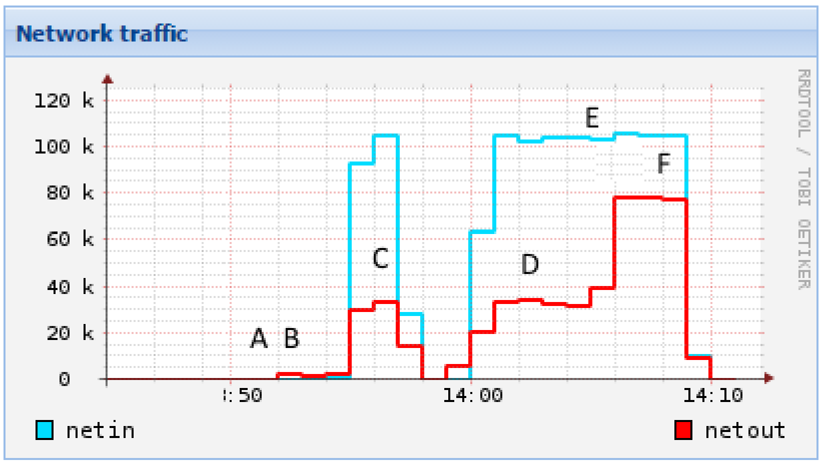
The conference room was closed and then Export Pipes was enabled.

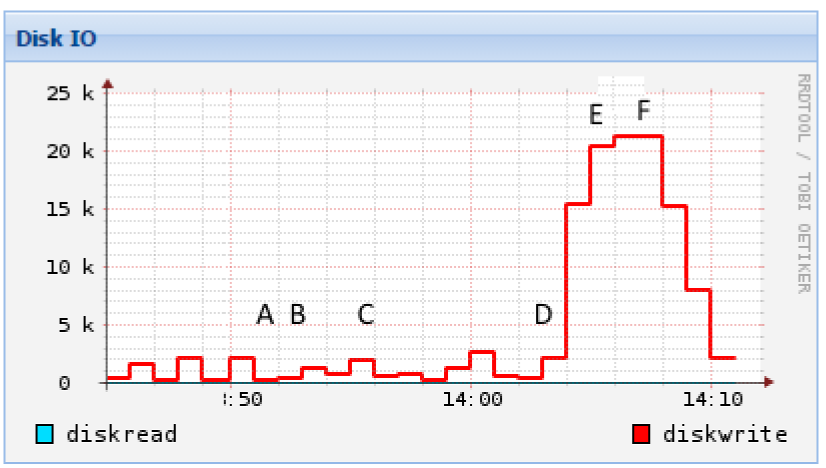
Point D: Created a conference room 1001 with pipes enabled and participant. CPU – 40%

Point E: The web stream for 1001 was created. CPU – 50%

Point F: A VLC media player played the web stream. CPU – 50%







A surprise was the amount of disk activity that appears when the webstream was started.

28. Directory Structure

For the Linux installation, the following directory structure is used:

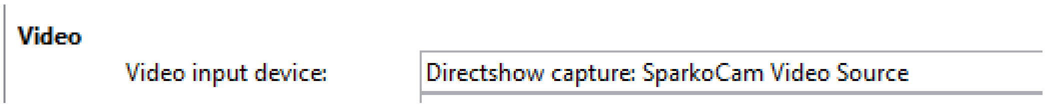
/opt/openmcu-ru

- bin
  - contains the executables:
    - ffmpeg, ffprobe, ffserver, openmcu-ru, vp8_scalable patterns, vpxdec, vpxenc, x264
- config
  - contains the configuration files:
    - daemon.conf, ffserver.conf, layouts.conf, openmcu.ini
  - and the images used:
    - background.jpg, background2.jpg, emptySubframe.jpg, logo.png, noVideoFrame.jpg, offlineFrame.jpg
- font
  - contains the font Russo_One.ttf
- include
  - contains the following includes:
    - libavcodec, libavdevice, libavfilter, libaformat, libavutil lipostporc, libswresample, libswscale, libyasm, vpx, x.264
- lib
  - contains the following libraries:
    - pkconfig, ptlib, livavcodec, libavdevice, libavfilter, libeavformat, libavutils, libopus, libpostproc, libsofia-sip-ua, libspeex, libsrtp, libswresample, libswscale, libvpx, libx264, libyux

- log
    - contains the follwing log files:
        - muc_log.txt, trace.txt
- pipe
    - if pipes are enabled, it will contain:
        - audio.1001 and video.1001 where the number is the room designation
- records
    - contains the recordings of the video conferences if enabled.
- resource
    - contains all of the images used for the menus and logos
    - contains the language translation files
    - contains the javascripts for the control and status menus
    - contains the sounds for connecting, entering and leaving a room
    - contains the template.html file
- scripts
    - contains the following scripts:
        - tls_info.sh, tls_make.sh, tls_make_client.sh, web_stream_start, web_stream_start.cmd
- share
    - contains the following directories
        - ffmpeg and man (empty)
- ssl
    - It is empty

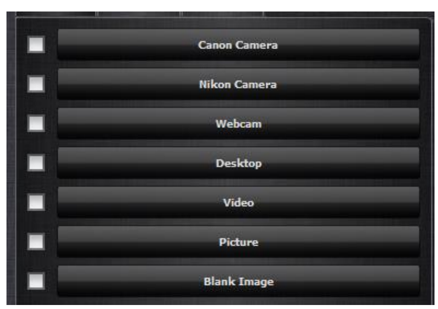
29. SparkoCam

SparkoCam (http://sparkosoft.com/sparkocam) is an inexpensive Windows based program that emulates a webcam source for your IP phone. Your IP phone recognizes the SparkoCam program as a video source.



SparkoCam sits between your IP phone and your video sources and allows you to choose different video source for your IP phone "on the fly" while in a conference call. You can live select from the following video sources by checking their checkbox:

- Canon Camera – you can connect a Canon camera and use it as a video source
- Nikon Camera – ditto
- Webcam - no explanation needed
- Desktop – you can display the complete desktop, partial, follow the mouse or an application window
- Video – you can have several videos ready and select which one to display
- Picture – you can have many images ready and select which one to display
- Blank Image



This allows the main presenter to live switch between his webcam, videos, pictures or anything on his desktop to the conference. It is a very useful program. There is a free version that has all the functionality of the paid version but adds a watermark to the video stream. The free version allows you to test drive the many features. SparkoCam's main screen has a preview pane on the right hand side. In addition to switching video sources, you can do many special effects if needed.

Webcam Selection (preview screen on right):

Desktop Selection:

Video Selection:

Picture Selection:

30. Firewall Settings

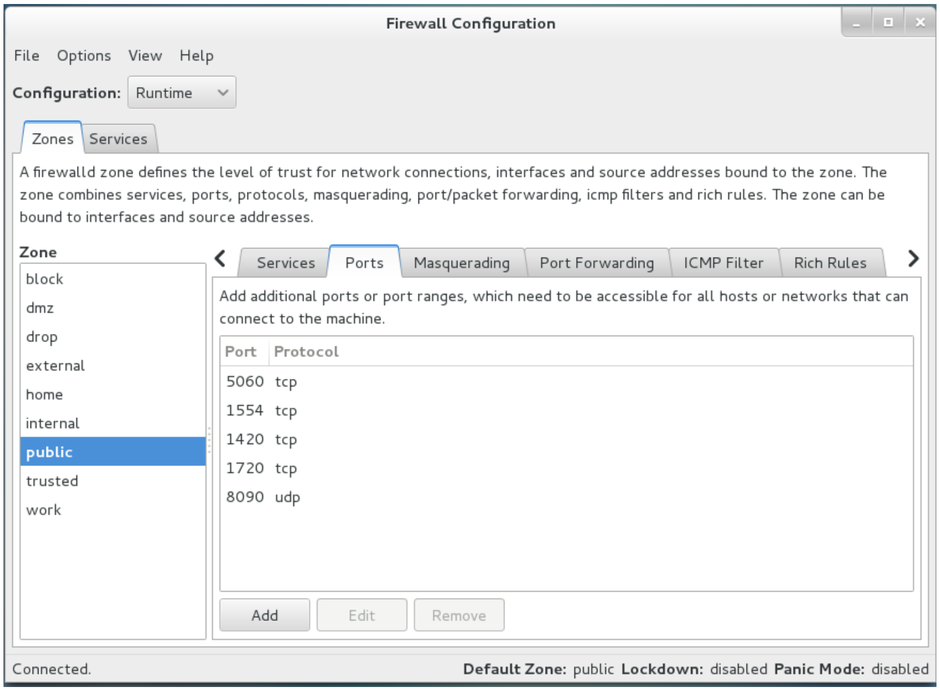
In order for Conference participants to access OpenMCU-ru, the following firewall ports should be open on the server that is running OpenMCU-ru. During testing in a controlled environment, the firewall may be disabled. When OpenMCU-ru is brought into production, then the firewall should be enabled and only the basic services required should be open to the public.

These are the basic ports that should be open for the services covered in this guide. Your configuration may require more ports to be open.

- Port 1420 – TCP: for accessing OpenMCU-ru administration menu
    - Port can be changed at Settings – General – HTTP Port
- Port 1423 – TCP: for accessing OpenMCU-ru Telnet server (not recommended)
    - Port can be changed at Settings – Advanced – Telnet server – Telnet Listener
- Port 1554 – TCP: for RTSP streaming
    - Port can be changed at Settings – RTSP – RTSP Parameters – RTSP Listener
- Port 1720 – TCP: for H.323 connections
    - Port can be changed at Settings – H.323 – H323 Parameters – H.323 Listener
- Port 5060 – UDP: for SIP registrations
    - Port can be changed at Settings – SIP – SIP Parameters – SIP Listener
    - Format: <IP addres>:<Port number>
    - Ex. 0.0.0.0:5070 (all ports) or 192.168.25.34:5078 (specific interface)
- Port 5061 – TCP: for SIP secure (SIPs).
    - Port can be changed at Settings – SIP – SIP Parameters – SIP Listener
    - Format: <IP addres>:<Port number>
    - Ex. 0.0.0.0:5070 (all ports) or 192.168.25.34:5078 (specific interface)
- Port 8090 – TCP: for http Web streaming
    - Port can be changed at file /opt/openmcu-ru/config/ffserver.conf

CentOS 7 Full ISO Firewall:

This is the firewall configuration for CentOS 7 full ISO:



CentOS 7 Minimal ISO Firewall:

CentOS 7 Minimal ISO does not come with a firewall loaded. So the first thing to do is to install the firewall:

1. Install iptables services:

   # yum install iptables-services

2. Enable the service at boot-time:

   # systemctl enable iptables

3. Managing the service:

   # systemctl [stop|start|restart] iptables

4. Saving your firewall rules can be done as follows:

   # service iptables save

5. Sample /etc/sysconfig/iptables file:

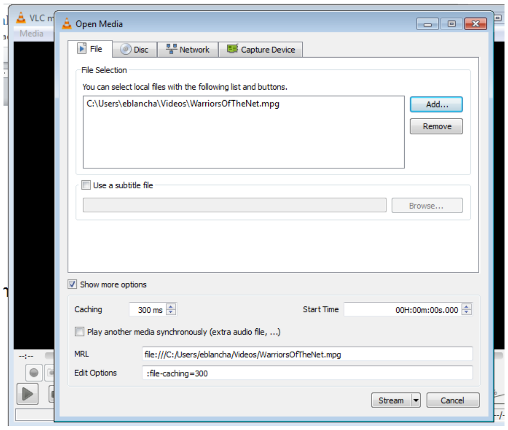
```
# sample configuration for iptables service
*filter
:FORWARD ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
# TCP 3 Way Handshake
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Allow Pings
-A INPUT -p icmp -j ACCEPT
# Allow all Loopback connections
-A INPUT -i lo -j ACCEPT
# Allow SSH
-A INPUT -p tcp -m tcp -m state --dport 22 --state NEW -j ACCEPT
# OpenMCU-ru Administration Port 1420
-A INPUT -p tcp -m tcp --dport 1420 -j ACCEPT
# Allow SIP connections for UDP Port 5060
-A INPUT -p udp -m udp --dport 5060 -j ACCEPT
# Allow H.323 connections to Port 1720
-A INPUT -p tcp -m tcp --dport 1720 -j ACCEPT
# Allow Telnet connections to OpenMCU-ru Port 1423
-A INPUT -p tcp -m tcp --dport 1423 -j ACCEPT
# Allow HTTP web streaming to Port 8090
-A INPUT -p tcp -m tcp --dport 8090 -j ACCEPT
# Allow RTSP Streaming Port 1554
-A INPUT -p tcp -m tcp --dport 1554 -j ACCEPT
# Reject everything else
-A INPUT -j REJECT --reject-with icmp-host-prohibited
# Allows SIPs connections to TCP Port 5061
-A INPUT -p tcp -m tcp --dport 5061 -j ACCEPT
COMMIT
# Completed
*mangle
:FORWARD ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed
*nat
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
# Completed
```

Note: SIPs (SIP secure) TCP Port 5061 is after the “Reject everything else” rule in this example. If you wanted to use SIPs, move the comment line and SIPs rule before the “Reject everything else” rule.

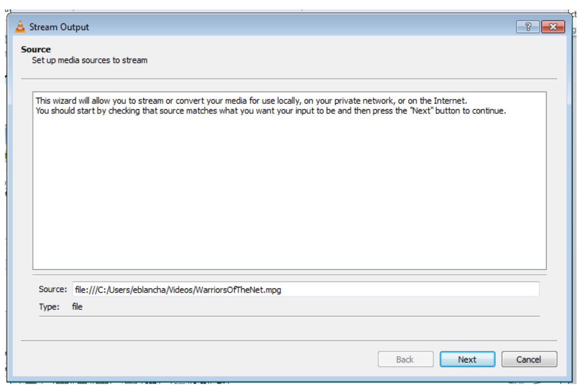
31. VLC RTSP Streaming a file

This section describes how to use VLC player to create and test an RTSP stream:

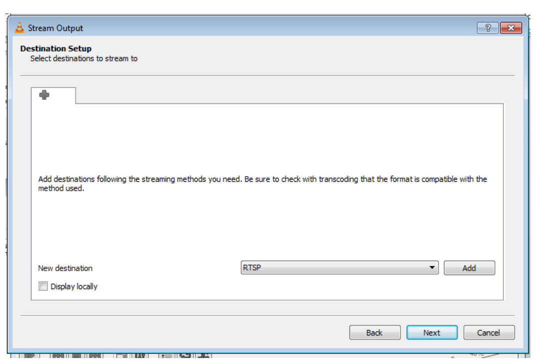
1. Open VLC player and select Media – Stream – Add to select video to stream
2. Select more options



3. Then select Stream and see Output.

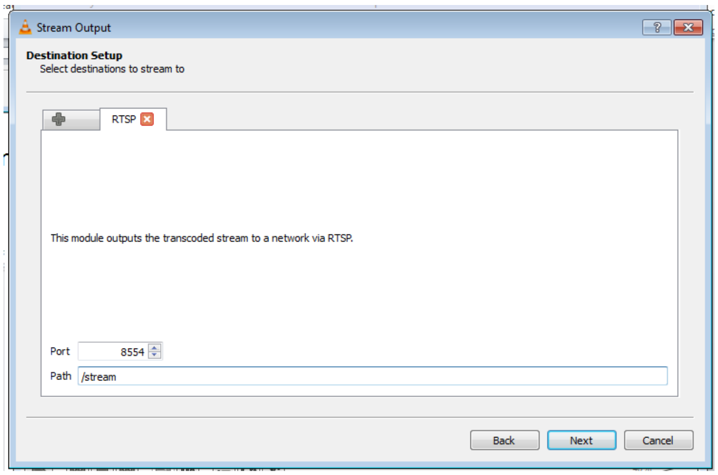


4. Next, Select RTSP and Add. Display locally is an option to see what you are streaming, normally you don't see.

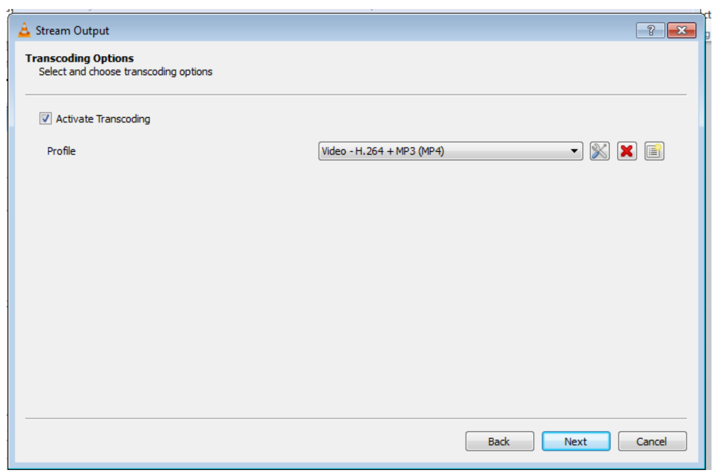


5. You'll see a new tab for RTSP, note port number, it should match the default OpenMCU-ru RTSP listening port 1554 configured under Settings – RTSP – RTSP Parameters – RTSP Listener.

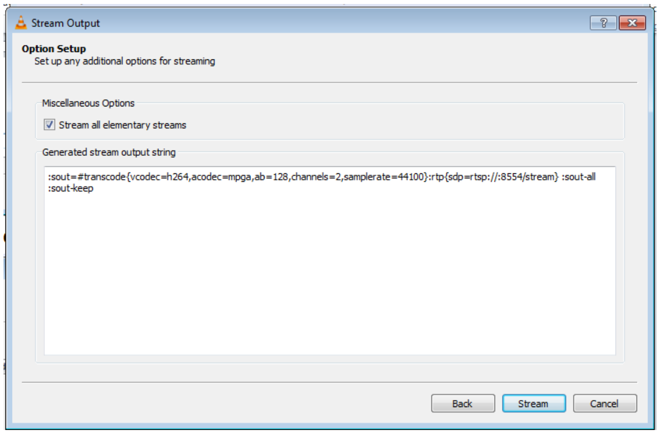
   This example uses non standard port 8554. Add a stream name, it can be anything, I chose /stream



6. Activate transcoding if necessary (not needed) and select default codecs



7. Stream all elementary elements (not needed) then Stream



8. From another VLC player, select open network stream, url: rtsp://localhost:8554/stream. Voila

9. For webcam use Sparkocam as source as webcam doesn't work directly.

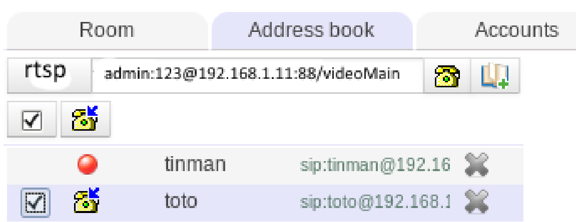
32. RTSP IP Camera Client

OpenMCU-ru can use an IP camera as a participant for a conference if it has RTSP streaming capabilities. This example is based on the Foscam HD FI9853EP as reported in the OpenMCU-ru English forums. It is noted here as a guide for other models of IP cameras. So basically there may be some experimenting required to get it to work for your IP camera.

1. Read your IP camera manual to find out what URLs (Universal Resource Locators) are available to connect to your IP camera RTSP stream. This is the section from the Foscam HD FI9853EP manual:

For example:
IP: 192.168.1.11
HTTP Port number: 88
User name: admin
Password: 123
Here I can enter one of the following URLs in the VLC.
1) rtsp://admin:123@192.168.1.11:88/videoMain
2) rtsp:// @192.168.1.11:88/videoMain
3) rtsp://:123@192.168.1.11:88/videoMain
4) rtsp://admin@192.168.1.11:88/videoMain

2. The URLs numbered 1) to 4) are of interest and should be tested by sending an invite from the conference room:



3. The format for the URL is

<protocol>://<username>:<password>@<server>:<port>/<share name>

<protocol> - RTSP
<username> - admin (username for IP camera authentication, may be optional)
<password> - 123 (password for username for IP camera authentication may be optional)
<server> - 192.168.1.11 (server domain name or IP address)
<port> - 88 (RTSP port)
<share name> - videoMain (may be configured or preconfigured resource name)

33. Troubleshooting

1. **Problem:** Installed OpenMCU-ru version 3.48 or version 4.0 or earlier and something doesn't work.

   Answer: There are problems with versions earlier then 4.1 and there is a reason that OpenMCU-ru gets regularly updated. Quit wasting your time and update to the latest version.

2. **Problem:** I'm having problems compiling the source code onto the following platforms: Ubuntu, Debian, Slackware, Suse, etc..

   Answer: There are too many distributions and variables to support all of the Linux distributions. In order to concentrate on developing OpenMCU-ru and to reduce the amount of work required to support all of the Linux distributions, OpenMCU-ru is supported on the CentOS distribution only. That's the one that the developers use and test. Makes sense to use the same platform?

3. **Network Connectivity:** Do you have network connectivity from your end user device to the server? Can you ping the server and can the server ping back? Sometimes, you can ping in one direction but not the other. If you can ping in one direction only, it might be Windows firewall blocking incoming pings. Things that affect connectivity:

   - Firewalls: software and hardware. See Section 30 to see which ports must be open in order for OpenMCU-ru to work properly.
   - Routes: do you have a network path between the two devices?

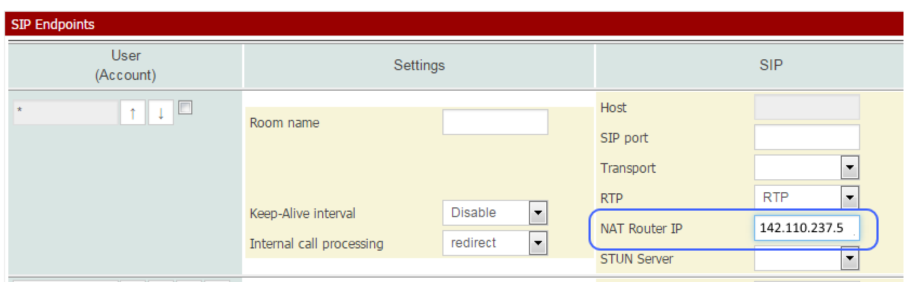
4. **Understand the communication flow**:

   - When an end device like a SIP video soft phone connects to OpenMCU-ru, it will connect to OpenMCU-ru's SIP port (5060). This is the ***standard*** port used for connecting to SIP based servers. This is the port that OpenMCU-ru by default listens to. If you use port 5061 or any other port, OpenMCU-ru is not listening to that port and will not respond. SIP is an example of a call connect protocol. Its job is to dial and connect.
   - The IP phone will introduce itself with an arbitrary client port number (for example: 64,123)
   - The end device will talk to OpenMCU-ru at port ***5060***, OpenMCU-ru will talk back to the end device on port ***64,123***.
   - Once the call is connected (dialed and answered), separate audio and video RTP streams are created in the range of 10,000 to 20,000. This example range is arbitrary and can be configured at OpenMCU-ru - Settings - General tab. I use 10,000 to 20,000 for compatibility with FreePBX.

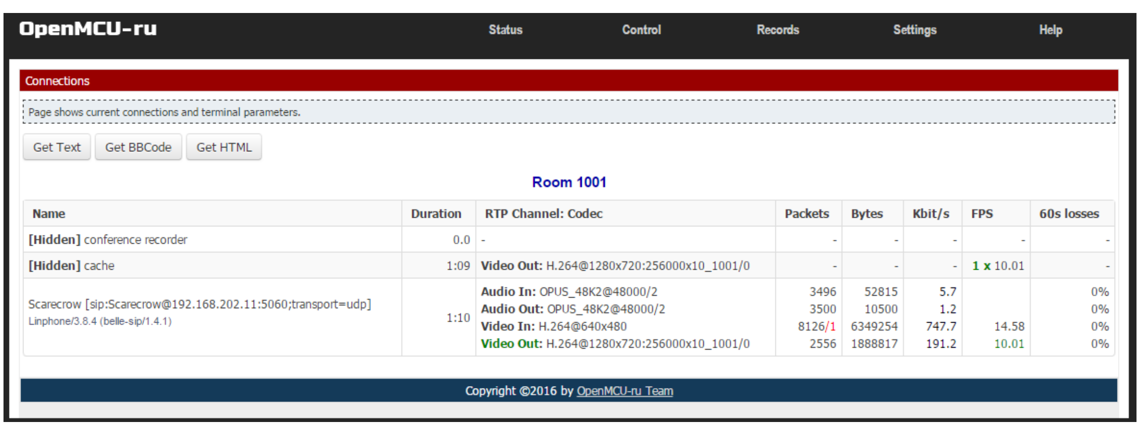
- If the end device identifies itself as port 25,000 which is outside of the 10,000 to 20,000 range that OpenMCU-ru expects, then OpenMCU-ru will ignore any video or audio from that end device.

5. **IMPORTANT:** the setup of the call uses the SIP or H.323 call connect protocols, the actual transmission of voice and audio uses the RTP (Real Time Protocol) on different ports. This can cause problems for SIP connections going through a firewall with NAT services. NAT may change the port numbers and IP addresses to something that SIP protocol doesn't expect.

   To help SIP transvers NAT, configure the public/WAN IP address in OpenMCU-ru. Go to OpenMCU-ru – Settings – SIP – Endpoints, set NAT Router IP to the public IP address (WAN IP address) of your network.



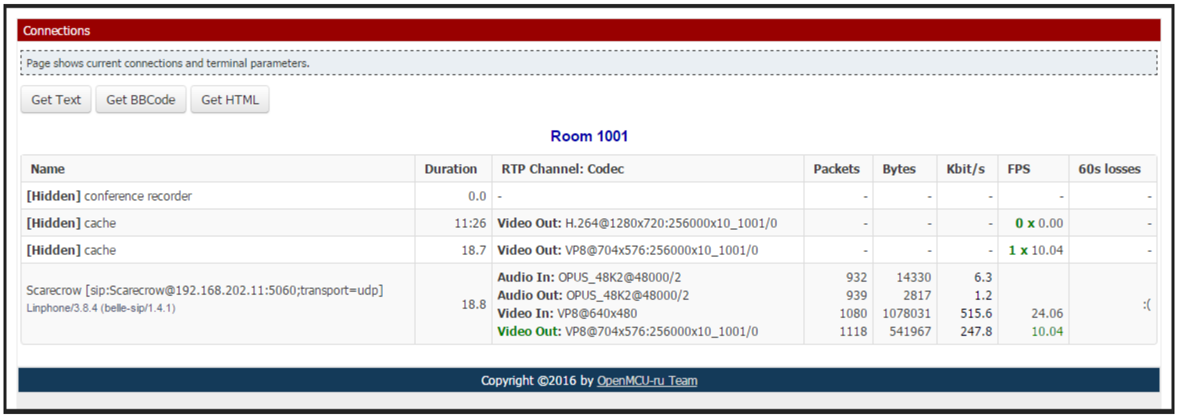
6. Look at the OpenMCU-ru Status menu. It will give you clues as to what is not working and what is:



- This shows that the end-user named Scarecrow is connected from IP address 192.168.202.11 using UDP on port 5060. The user is using a Linphone.

- Incoming video from the Linphone is H.264 with a resolution of 640x480, 747.7 kbps
- Outgoing video from OpenMCU-ru is H.264 with a resolution of 1280x720, 191.2 kbps, 10 frames per second.
- Incoming audio from the Linphone is Opus 48k2, 5.1 kbps
- Outgoing audio from OpenMCU-ru is Opus 48k2, 1.2 kbps

7. Here's another example where the Linphone is using a VP8 codec. OpenMCU-ru is transcoding (translating from VP8 to H.264). The input and output video codecs to the Linphone are VP8 and there is a second output stream using H.264:


Connections

Page shows current connections and terminal parameters.

Get Text | Get BBCode | Get HTML

Room 1001

Name	Duration	RTP Channel: Codec	Packets	Bytes	Kbit/s	FPS	60s losses
[Hidden] conference recorder	0.0	-	-	-	-	-	-
[Hidden] cache	11:26	**Video Out:** H.264@1280x720:256000x10_1001/0	-	-	-	**0 x** 0.00	-
[Hidden] cache	18.7	**Video Out:** VP8@704x576:256000x10_1001/0	-	-	-	**1 x** 10.04	-
Scarecrow [sip:Scarecrow@192.168.202.11:5060;transport=udp] Linphone/3.8.4 (belle-sip/1.4.1)	18.8	**Audio In:** OPUS_48K2@48000/2 **Audio Out:** OPUS_48K2@48000/2 **Video In:** VP8@640x480 **Video Out:** VP8@704x576:256000x10_1001/0	932 939 1080 1118	14330 2817 1078031 541967	6.3 1.2 515.6 247.8	 24.06 10.04	:(

Copyright ©2016 by OpenMCU-ru Team

8. Video Codec compatibility:

- OpenMCU-ru can ***output*** only the following video codecs:
    - H.263,
    - H.264
    - VP8 .
    - Does the end device support these codecs? If not then no video will be returned to the end device
- OpenMCU-ru can transcode from the following SIP video codecs from the end device:
    - H.263
    - H.264
    - MP4V
    - VP8

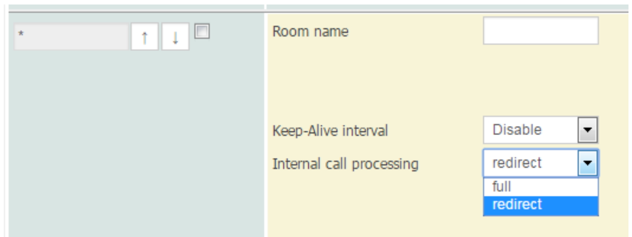
- Can the end device send video using one of these codecs? If not then OpenMCU-ru will not be able to display the video that it receives.

- OpenMCU-ru can transcode from the following H.323 video codecs from the end device:
    - H.263
    - H.264
    - VP8
    - Can the end device send video using one of these codecs? If not then OpenMCU-ru will not be able to display the video that it receives.

9. **Audio Codec compatibility:**
    - OpenMCU-ru can transcode between the following audio codecs
        - iLBC
        - G.711 alaw/ulaw
        - G.722
        - G7231-6
        - G729a
        - G.729
        - G.726
        - Opus
        - Silk
        - Speex
        - Does the end device support these codecs and one of the supported bit rates. Bit rates for the codecs are shown in the OpenMCU-ru – Settings section under the appropriate call connect protocol: H.323 or SIP.
10. **Room Control Mode:** Is the conference room that you are trying to connect to in auto mode or has the admin taken control of the room? A user can only join a conference room if it is in auto mode. If the admin has taken control of the room, the admin must add the user to the room. See Section 15: Taking Control of a Conference Room.

11. **Problem:** Call to OpenMCU-ru is redirected to another PBX with the SIP response "302 Moved Temporarily". May occur if you are attempting to set up a SIP trunk or SIP endpoint.

    Solution: Go to Settings – SIP – SIP Endpoints and set the "Internal call processing" to "full" and "Accept". Then set the "Internal call processing" back to "full" and "Accept"



12. **Problem:** Call disconnects after 30-60 seconds. Works fine then disconnects.

    Solution: This is a typical symptom of a SIP NAT configuration problem. Step 5 documents how to set the public IP address for the router NAT configuration. Also, if the end device (video phone) is being routed through a PBX, it could be the PBX's configuration of NAT.

    Background: A SIP call consists of two parts:

    - The call connect process using the SIP protocol. This dials and establishes the call
    - The audio/video stream using the RTP protocol.

    The problem is that the IP and Port number information about the RTP audio/video stream is encoded in the SIP protocol's SDP (Session Description Protocol) information (yes a 3rd protocol!). When the SIP and RTP stream go through NAT, the IP address and Port number become translated to the public IP address and a new port number is assigned. This doesn't match what is encoded in the SDP information. For audio, a typical symptom is one way audio, after the call is connected, you can hear them but they can't hear you.

    A solution is to modify the SIP header by inserting information which keeps track of the public IP address/port number and the original IP address/port number of the device behind the NAT router. This is what adding the external IP address to the NAT router setting does as documented in Step 5.

    Usually this requires documenting all local private network addresses on the PBX or MCU if there are more than one network address. OpenMCU-ru only identifies the network that it is connected to.

    NAT problems are one of the most difficult problems to track down and solve. It gets worse if you are communicating through double NAT: a local NAT router on your network and a remote NAT router on the endpoints network.

    One solution is to provide a VPN between the two networks which takes NAT out of the picture.

34. Connecting to H.320 ISDN

Videoconferencing is something that has evolved greatly over the years. In the beginning, videoconference systems were connecting primarily over ISDN. There are some advantages with ISDN over other connections, the main one being a circuit switched connection through the PSTN with guaranteed bandwidth. Unlike ISDN, IP based videoconferencing operates over an IP connection, which is a packet switched network which is best effort.

The bandwidth available with each ISDN channel is quite low at 64kps per channel but multiple channels can be bonded together to increase throughput. Bonding happens when the call is first established, then the codec will request additional lines as required.

You cannot connect OpenMCU-ru directly to H.320 ISDN. You will need a H.323 gateway that supports H.320 ISDN. OpenMCU-ru supports H.323 as endpoint connections (video clients) but can't act as a H.323 gateway. However, there is a configuration menu (Settings – H.323 – H.323 Parameters) that may allow you to connect to a H.323 gateway (which I haven't tried).

An alternative is to use FreePBX/Asterisk to connect as a H323 gateway. Unfortunately Asterisk doesn't support H.323 out of the box and you would have to recompile Asterisk with H.323 support. Then manually configure/program the H.323 trunk to the H.323 gateway.

This seems to be the most current documentation for installing the ooh323 H.323 libraries and recompiling Asterisk (it is located after the Avaya configuration information):

http://cyril-constantin.blogspot.ca/2011/07/asterisk-1844-connected-to-avaya-with.html

License

OpenMCU-ru is distributed under the following license:

MOZILLA PUBLIC LICENSE

Version 1.0

1. Definitions.

1.1. ``Contributor'' means each entity that creates or contributes to the creation of Modifications.

1.2. ``Contributor Version'' means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. ``Covered Code'' means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. ``Electronic Distribution Mechanism'' means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. ``Executable'' means Covered Code in any form other than Source Code.

1.6. ``Initial Developer'' means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. ``Larger Work'' means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. ``License'' means this document.

1.9. ``Modifications'' means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. ``Original Code'' means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.11. ``Source Code'' means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or a list of source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. ``You'' means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, ``You'' includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, ``control'' means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell (``Utilize'') the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

2.2. Contributor Grant.

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

3. Distribution Obligations.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which you contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.

If You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you must include a text file with the source code distribution titled ``LEGAL'' which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after You make Your Modification available as described in Section 3.2, You shall promptly modify the LEGAL file in all copies You make available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.

If Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients' rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the notice described in Exhibit A. If it is not possible to put such notice in a particular Source Code file due to its structure, then you must include such notice in a location (such as a relevant directory file) where a user would be likely to look for such a notice. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A, and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Netscape Communications Corporation (``Netscape'') may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If you create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), you must (a) rename Your license so that the phrases ``Mozilla'', ``MOZILLAPL'', ``MOZPL'', ``Netscape'', ``NPL'' or any confusingly similar phrase do not appear anywhere in your license and (b) otherwise make it clear that your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN ``AS IS'' BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR

OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a ``commercial item,'' as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ``commercial computer software'' and ``commercial computer software documentation,'' as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in, the United States of America: (a) unless otherwise agreed in writing, all disputes relating to this License (excepting any dispute relating to intellectual property rights) shall be subject to final and binding arbitration, with the losing party paying all costs of arbitration; (b) any arbitration relating to this Agreement shall be held in Santa Clara County, California, under the auspices of JAMS/EndDispute; and (c) any litigation relating to this Agreement shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

Except in cases where another Contributor has failed to comply with Section 3.4, You are responsible for damages arising, directly or indirectly, out of Your utilization of rights under this License, based on the number of copies of Covered Code you made available, the revenues you received from utilizing such rights, and other relevant factors. You agree to work with affected parties to distribute responsibility on an equitable basis.

EXHIBIT A.

``The contents of this file are subject to the Mozilla Public License Version 1.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ________________________. Portions created by ______________________ are Copyright (C) ______ _______________________. All Rights Reserved.

Contributor(s): ______________________________________.''

Revision Log

1.00 Initial guide

1.01 - Minor changes and corrections
- Added section on installing OpenMCU-ru on to existing FreePBX distrobution using interface eth0 (FreePBX) and a virtual interface eth0:0 (OpenMCU-ru) and binding protocols to appropriate interfaces.
- Added security to Telnet server

1.02 - Minor corrections on spelling,better explanations and matching drawings to text.
- Added information on SSHD services
- Added information on using 2 network cards for installing on FreePBX
- Added QoS section
- Added OpenMCU-ru initialization file section
- Moved FreePBX/Asterisk to end of guide as this is an OpenMCU-ru guide not FreePBX
- Switched to pdf from Word so that more users can view it and not have formating screwed up.
- Added Version Log section

1.03 - Increased bit rate on FreePBX SIP settings for video from 384 kbps to 512 kbps and it improved the video quality dramatically.
- Added section on Web Streaming of Conference Rooms
- Clarified the purpose of Export (namd - pipe) menu

1.04 - Minor corrections and formating
- Indicate that only display #0 is streamed to the web
- Added web streaming performance information and that it uses TCP protocol
- Add information on the matrix display configuration file
- Clarify room naming is not limited to numbers
- Clarified the VAD configuration values
- Clarified limit of 16 participants viewed
- Clarified default audio sent to all participants
- Added a section on the Linux directory structure
- Added licensing agreement
- Added authors to introduction

1.05 - Added information on Advanced Settings – Conference Rooms
- Added information on auto starting recording of conference rooms
- FreePBX-OpenMCU-ru integration, added explanation on OpenMCU-ru – Settings – SIP – Endpoints, set NAT Router IP setting
- Adjust “Video resolution” settings in OpenMCU-ru – Settings – SIP – Endpoints

- Added setting public IP address through the NAT IP setting in Initial Settings section.
- Added layouts.conf and modifying display frame border width.
- Added section on SparkoCam
- Added section on RTSP streaming
- Added web browser to Web streaming
- Added Firewall section

1.06 - Added info on template locks
- Added the Status menu's Get codes
- Added port change settings to the firewall section
- Cleanup of server installation instructions
- Clarified checking SSHD service running
- Corrected /etc/inittab for disabling X windows
- Added info on no support for version 3.48

1.07 - Added 3 protocols (SIP, RTSP and H323) to participant Invite option for manual control of rooms
- Add information on using VLC player to create and test an RTSP stream and it is preferred over web streaming.
- Expanded the "1. Introduction" section to include "What is the OpenMCU-ru Video Conference Server?"
- Expanded the "3. Features" section
- Added information on VAD (thanks to marcelloc for the explanation!)
- Added information on "Recall last template" for permanently created conference rooms (thx marcelloc)
- Added Performance Test 2
- Added information on connecting a streaming IP camera client

1.08 - Added a Section 33 Troubleshooting
- Added disclaimer to Section 5 Command Line Options: "you will most likely never use these commands"
- Modified Section 6 Installation
    * Added a subsection 6.2 Windows OpenMCU-ru version
    * Added a subsection 6.1.1 CentOS 7 minimal installation
- Corrected section 8.21 removed line referring to private IP address.
- Restarted step numbering at 1 for each section.

1.09 - Clarified port 5061 as SIPs (SIP secure)
- Added firewall support for CentOS Minimal ISO configuration
- Added "check Enable box" for RTSP streaming
- Add Redirect problem/fix to troubleshooting.

- For section 24, connecting to FreePBX, to "Other SIP Settings" add "directrtpsetup = yes" to solve FreePBX 13 connection problem
-Expand on the Windows version Section 6.2
- Rename Version Log to Revision as it is more appropriate
- Advanced – Video: Outgoing Video Quality: provided explanation of settings
- Clean-up of grammar and typos throughout document
- Added section 34. Connecting to H.320 ISDN