

Setup the software for the intercom base station

Requirements:

You have setup the server (Mumble, Mosquitto, Node.js) according to the server setup guide

Raspberry Pi 3 with Ethernet (tested with 3B), required hardware and software

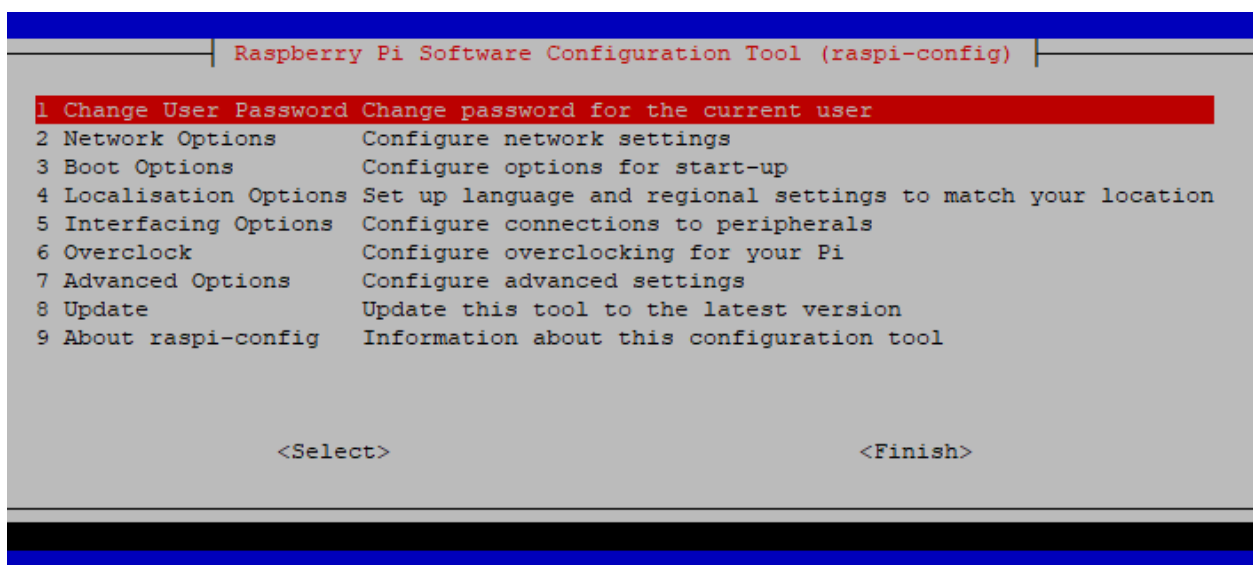
Download the latest Raspbian Stretch Lite image from <https://www.raspberrypi.org/downloads/raspbian/>

This guide assumes that you are using a Windows computer but both Mac and Linux can be used (you just have to find the tools needed). The setup is tested with Raspbian Stretch Lite image built on the 2018-06-27. When copying code from the blue fields remember to check that there are no whitespace characters (invisible spaces) at the end of the lines because that can break the code.

Etcher <https://etcher.io/> is recommended to write the image to the sd card. Insert the card into the Raspberry Pi, connect a keyboard and monitor and power up, after a few seconds the Raspberry will resize the partition to fill the card used and reboot. After the reboot you will see a login screen prompting for a username, use pi as username and raspberry as password. The first task is to setup the Raspberry Pi as needed. Write the command below and press enter

```
sudo raspi-config
```

You will now see a menu like this



1. Configure the Raspberry

- 1 Change User Password (to something you remember)
- 2 Network Options
 - N1 Hostname (base stations should be named icbs1, icbs2... use icbs1 to follow this guide)
- 4 Localisation Options
 - I2 Timezone
 - I3 Change Keyboard-Layout
- 5 Interfacing options
 - P2 SSH Enable
 - P5 I2C Enable
- 7 Advanced options
 - A5 Resolution -> 1280x720 60Hz 16:9

Do NOT reboot on exit

2. Change I2C bus speed

```
sudo pico /boot/config.txt
```

Change line

```
dtoverlay=i2c-arms
```

to

```
dtoverlay=i2c-arms,i2c-arms-baudrate=400000
```

Save file (CTRL+x)

3. Set a static IP address

```
sudo pico /etc/dhcpd.conf
```

Uncomment the example static IP configuration and use a IP that you are allowed to use

```
interface eth0
static ip_address=192.168.10.71/24
static routers=192.168.10.254
static domain_name_servers=8.8.8.8
```

Save file (CTRL+x) and reboot

```
sudo reboot
```

You should now have a IP address (look for the line My IP address is:xxxx)

You can now decide if you prefer to work with the keyboard and screen or continue the setup over SSH (using for example putty <https://www.chiark.greenend.org.uk/~sgtatham/putty/>)

4. Update the system

```
sudo apt update
sudo apt dist-upgrade
```

5. Install xfce4 (a minimal graphical interface)

```
sudo apt install --no-install-recommends xserver-xorg
sudo apt install --no-install-recommends xinit
sudo apt install xfce4 xfce4-terminal tango-icon-theme gnome-icon-theme lightdm
```

6. Setup auto-login

```
sudo raspi-config
```

3 Boot Options

B1 Desktop / CLI

B4 Desktop Autologin

Finish and do NOT reboot

7. Install X11VNC

This gives a possibility to access the desktop from another computer

```
sudo apt install x11vnc
```

Set VNC password

```
x11vnc -storepasswd
```

enter desired password and save to /home/pi/.vnc/passwd

8. Autostart X11VNC

```
sudo pico /etc/xdg/autostart/x11vnc.desktop
```

```
[Desktop Entry]
Type=Application
Name=X11VNC
Comment=VNC
Icon=preferences-desktop-screensaver
Exec=x11vnc -bg -forever -noxdamage -ncache_cr -rfbauth /home/pi/.vnc/passwd
TryExec=x11vnc
OnlyShowIn=XFCE;
```

Save file (CTRL+x)

9. Redirect error messages

Do not log error messages from the graphical interface to file to extend the sd card lifetime.

```
sudo pico /etc/X11/Xsession
```

point your ERRFILE to /dev/null IE. Line 77

comment out this line (add # before the line)

```
#exec >>"$ERRFILE" 2>&1
```

and add this below

```
exec >> /dev/null 2>&1
```

Save file (CTRL+x) and reboot (before this you should attach your usb sound card)

```
sudo reboot
```

10. Install Tightvnc

Download tightvnc on your windows computer from <https://www.tightvnc.com/> -> Download, when installing choose only client.

11. Start Tightvnc

Start tightvnc and enter the IP number of your Raspberry Pi, you will be asked for your VNC password (set in step 7). When logged in choose "Use default config" in the popup window.

12. Setup audio devices

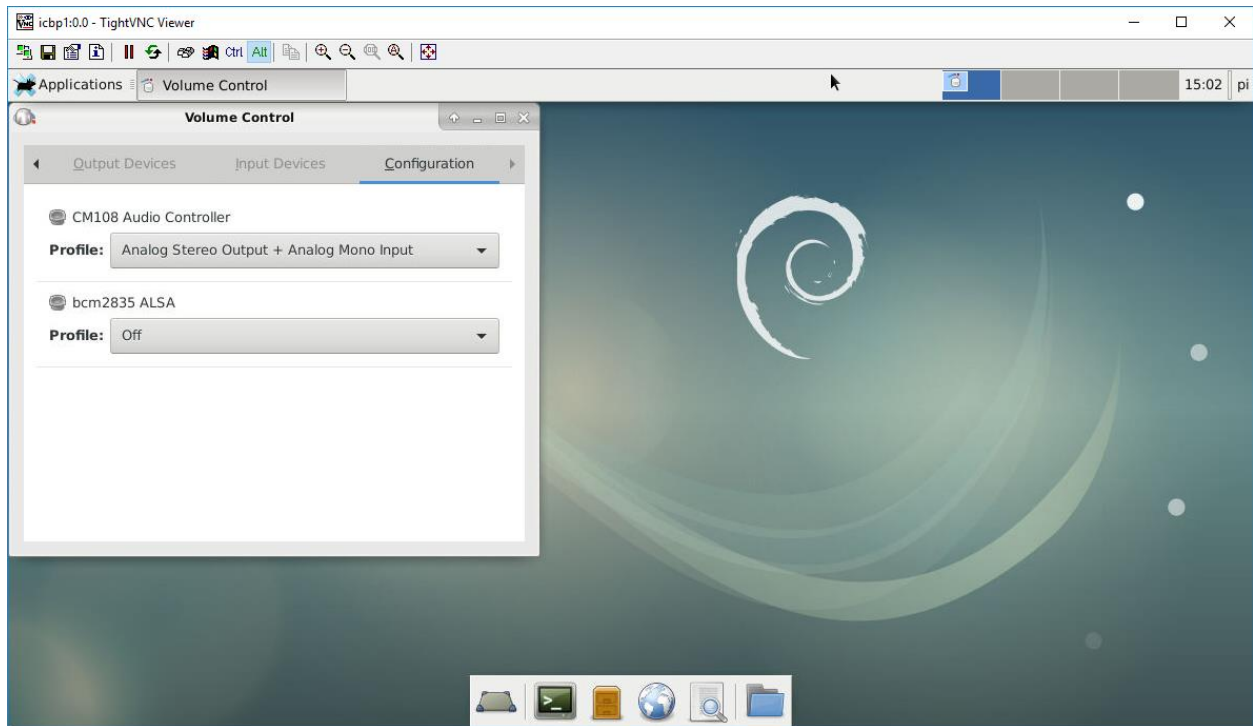
Click on the Applications button (upper left)

Applications -> Multimedia -> PulseAudio Volume Control

In the Configuration tab (press right arrow on keyboard to get to it)

bcm2835 ALSA Profile -> Off

USB Audio Device (CM108 Audio Controller) -> Analog Stereo Output + Analog Mono Input



Close the PulseAudio Window

Enable echo cancellation for the sound card

```
sudo pico /etc/pulse/default.pa
```

add this line to the end of the file

```
load-module module-echo-cancel aec_method=webrtc
```

CTRL+x to save and reboot

```
sudo reboot
```

13. Create intercom folders

Go to Applications -> Terminal Emulator

```
mkdir ~/intercom  
mkdir ~/intercom/certs  
cd ~/intercom
```

14. Copy certificate files to the Raspberry Pi

The certificate was generated in the server setup guide

Download WinSCP (<https://winscp.net/>) on your Windows computer, install and start.

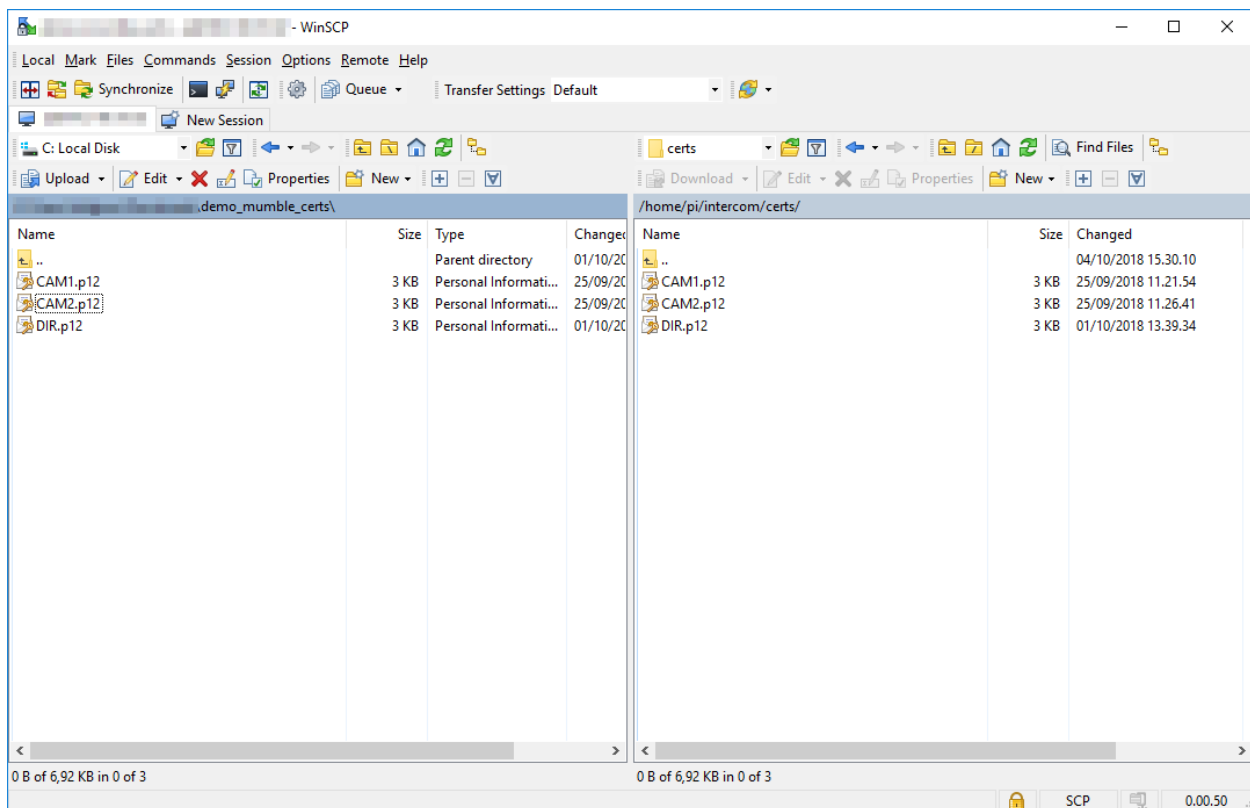
Under session choose SCP host name is your base station IP,

User name: pi, Password: (set in step 1)

When you are logged into the base station, on the left side you go to the folder where you exported the certs to in the server setup guide

On the right side you have your base station, double click intercom and then certs

Select all your certificates on the left side, right click and select Upload and OK



15. Install Mumble

Back to the Raspberry Pi and the terminal emulator

```
sudo apt install mumble
```

Close the terminal (click X in right corner)

16. Setup Mumble

When Mumble starts for the first time the Audio Tuning Wizard will appear, click next on the first introduction screen

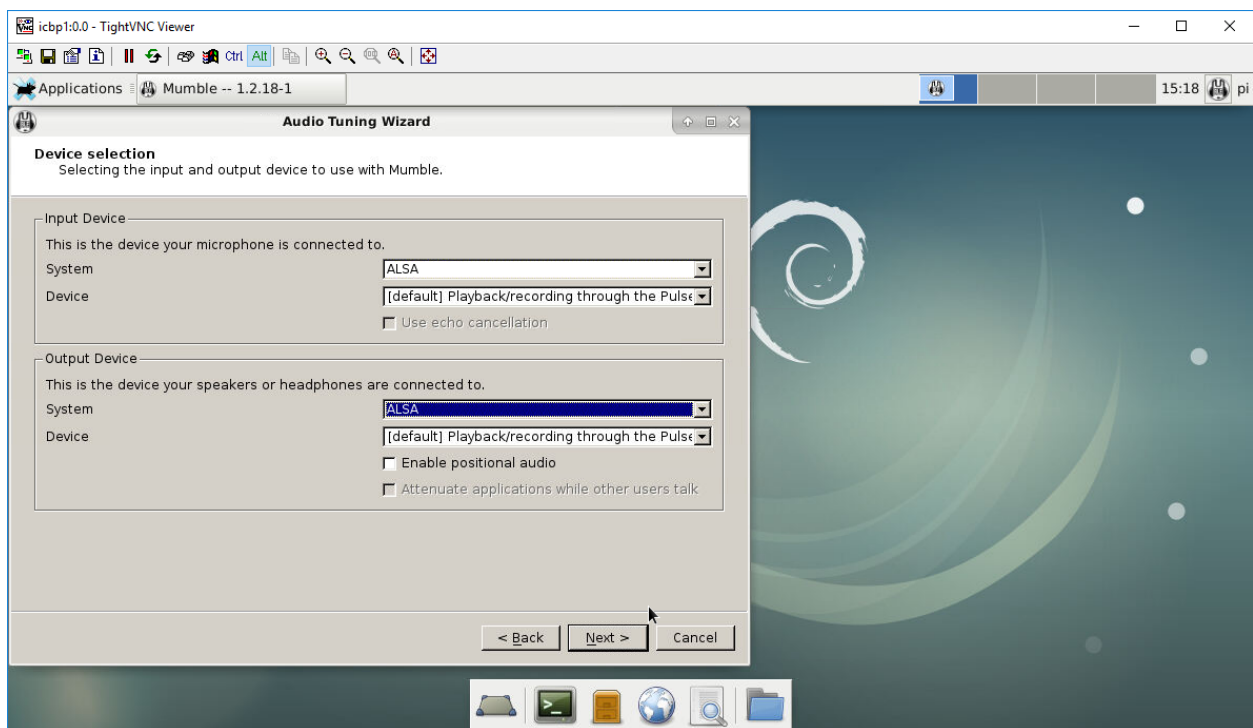
Device selection

System: ALSA

Device [default] Playback/recording through the PulseAudio sound server

on both Input and Output device,

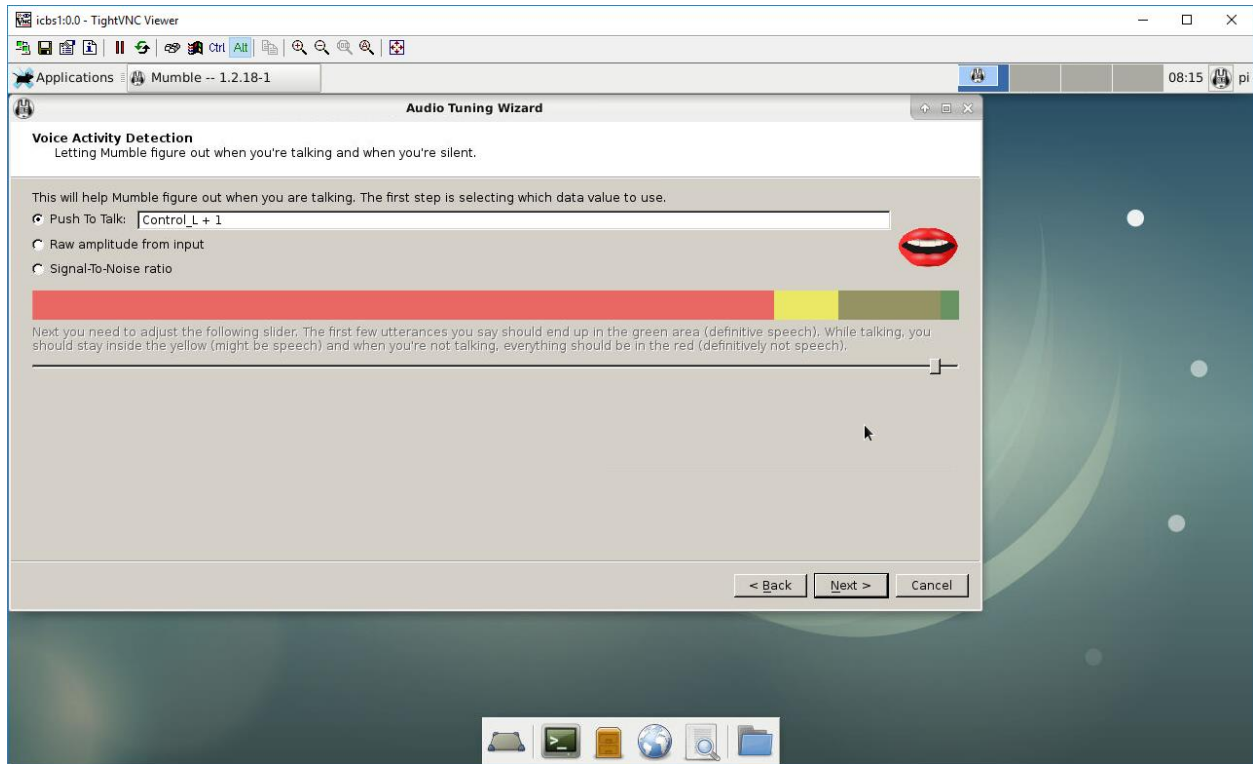
Uncheck positional audio -> Next



Set the buffer to 10ms (you will not hear anything here) -> Next

Let the microphone tuning be at its preset -> Next

Voice Activity Detection, choose Push To Talk and click the white box next to it and press CTRL + 1 -> Next



Quality and notifications, Quality settings -> High,

Uncheck Text-To-Speech and use sound instead -> Next

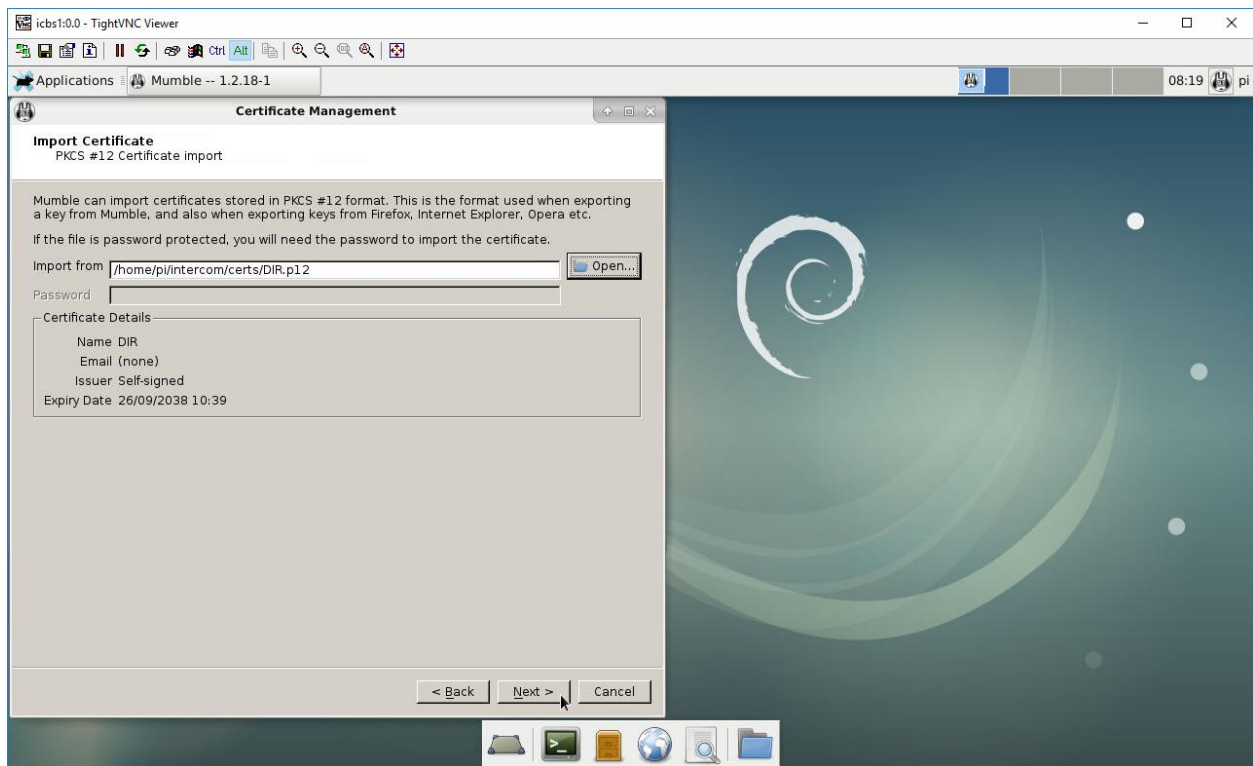
Choose if you want to submit anonymous statistics -> Finish

17. Import Mumble certificate

The Certificate Management should popup,

choose Import a certificate -> Next (you might have to scale the window to get to the next button)

Click Open and select the DIR certificate from /home/pi/intercom/certs/DIR.p12 -> Open -> Next



Click Finish

18. Connect to Mumble server

The Mumble Server Connect screen should have popped up, if not go to Server -> Connect

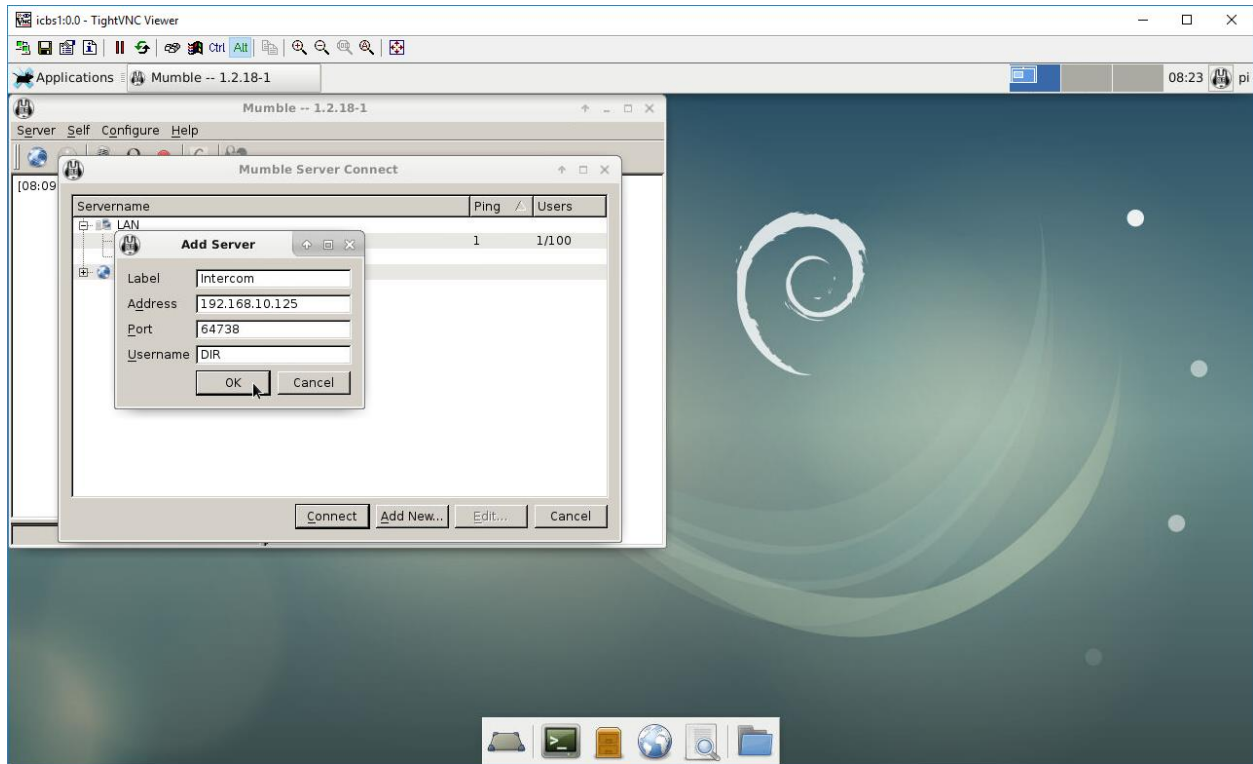
Click Add New,

Label: Intercom

Address: <YOURSERVERIP> in this example: 192.168.10.125

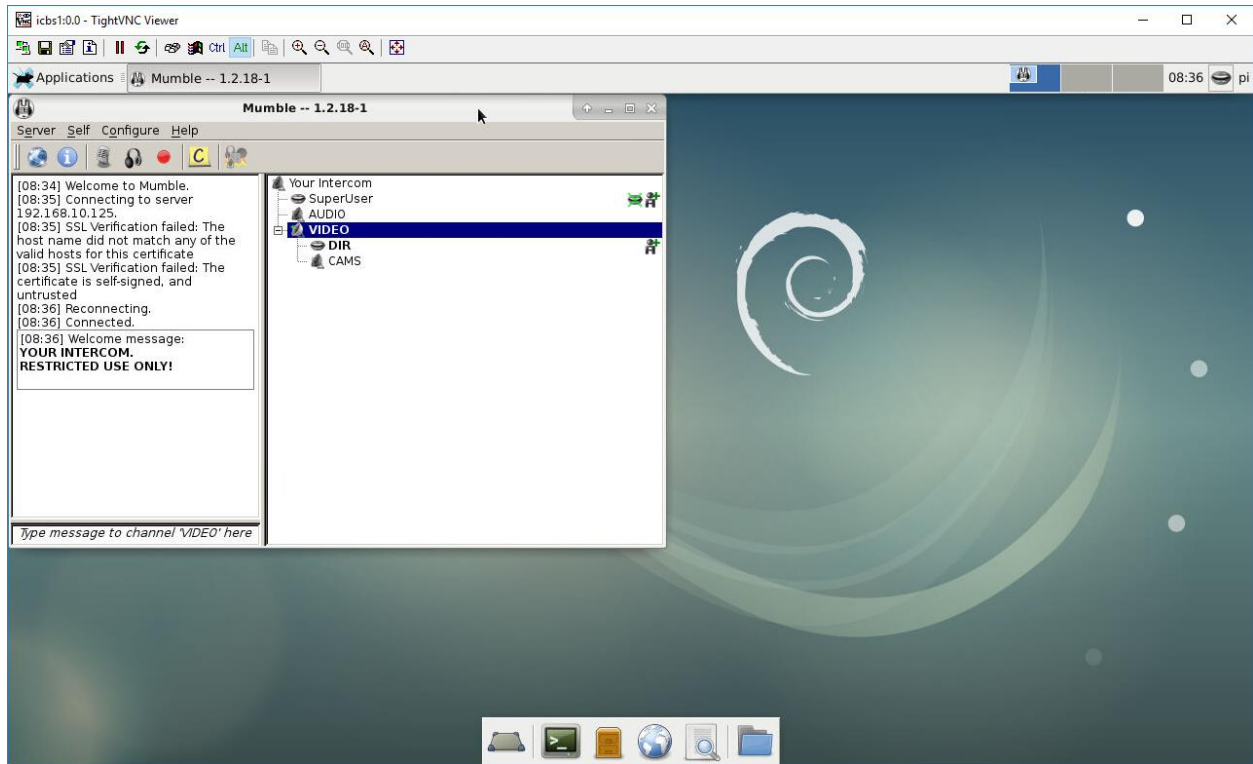
Port: 64738,

Username: DIR -> OK



Under Favorite you should now see the Intercom server you just added, click Cancel and quit Mumble (Remember to ALWAYS quit/restart Mumble when making changes, settings are not saved when using power off/restart from command line).

Start Mumble again and connect to the Intercom server you just made, when you login to your server for the first time (with a new certificate) you will get a warning about the self-signed certificate, choose Yes and continue. You should now be logged in as DIR and put into the Video channel.

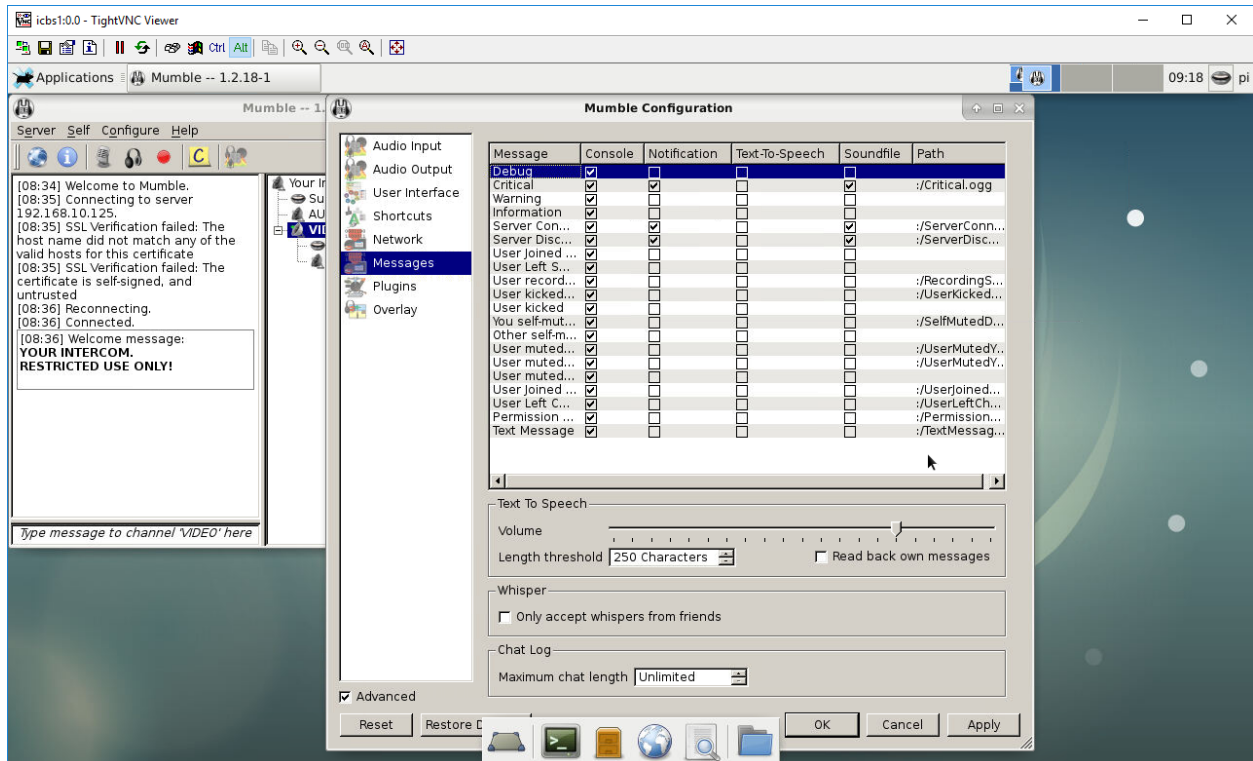


19. Configure Mumble, part 1

Under Configure->Settings->Click Advanced (down left)

Under Network, click Reconnect to last server on startup

Under Messages, set the matrix as in the picture, Notifications and Soundfile are the important ones, otherwise you will hear sounds every time a user joins/leaves a channel, etc.



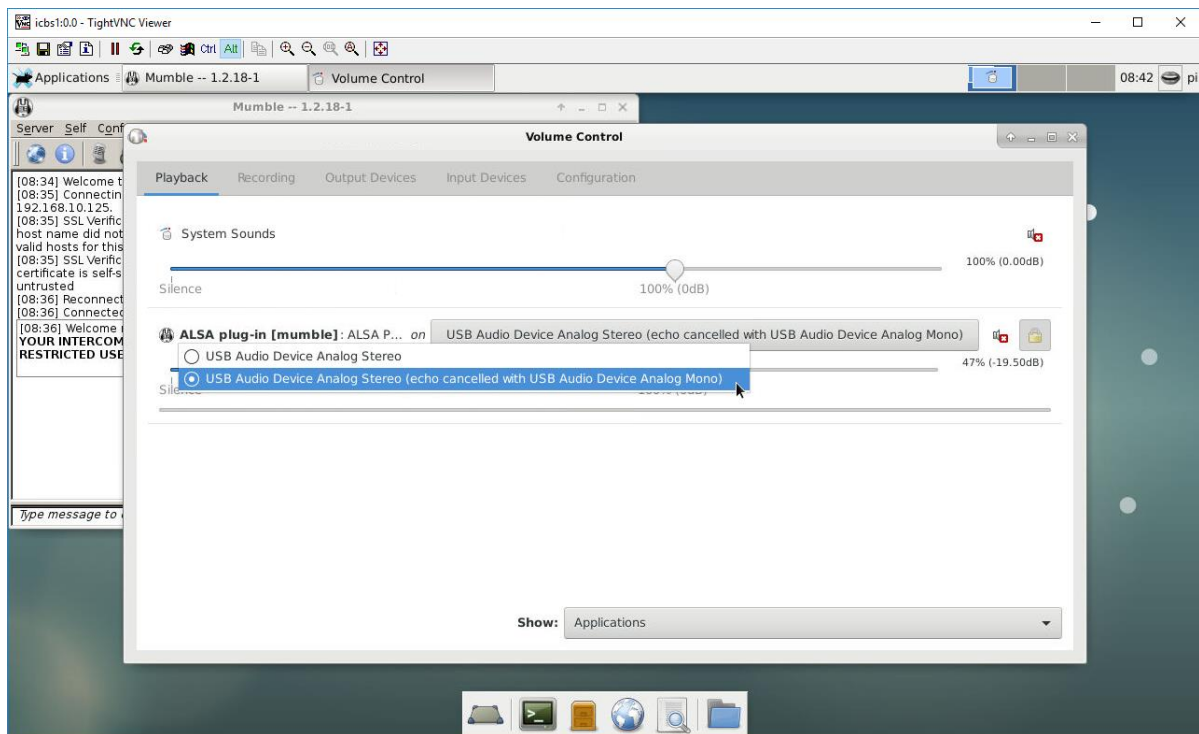
Apply->OK (and remember to quit/restart Mumble after this)

When Mumble is running we need to enable the echo cancellation (without a working echo cancellation the base station will get audio feedback and users talking to the base station will hear themselves if the base station is talking to them). This was setup in step 12.

Applications -> Multimedia -> PulseAudio Volume Control

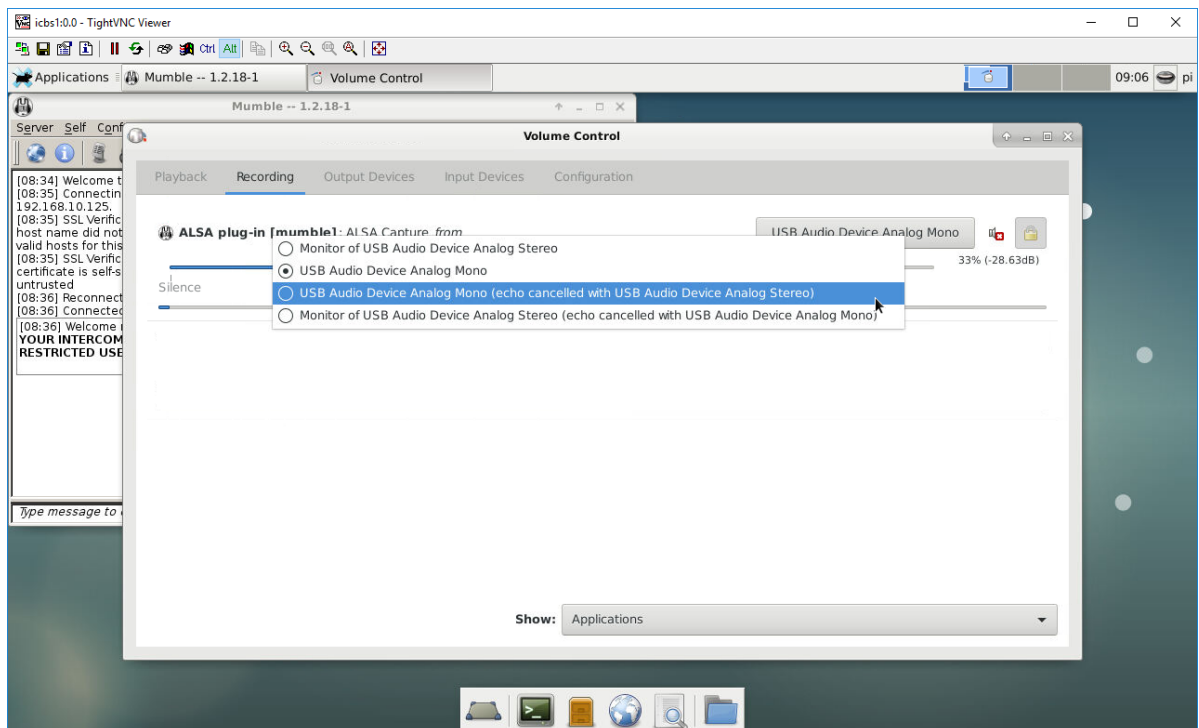
Playback tab

Click USB Audio Device Analog Stereo button next to ALSA plug-in [mumble] Change dropdown ALSA Plugin [mumble] to the option that has echo cancelled.



Click Recording tab

Click USB Audio Device Analog Mono and choose the echo cancelled option of USB Audio (not Monitor of)



Close the Volume Control window with the X in the corner (there is no save button)

20. Autostart Mumble

Using Application->Terminal emulator (or using the terminal over ssh) enter the following lines

```
sudo pico /etc/xdg/autostart/start_mumble.desktop
```

```
[Desktop Entry]
Type=Application
Name=Mumble
Comment=Start Mumble client
Icon=preferences-desktop-screensaver
Exec=mumble
TryExec=mumble
OnlyShowIn=XFCE;
```

Save file (CTRL+x) and reboot

```
sudo reboot
```

Check (with tightvnc or screen) that Mumble starts and connects to server as DIR after reboot.

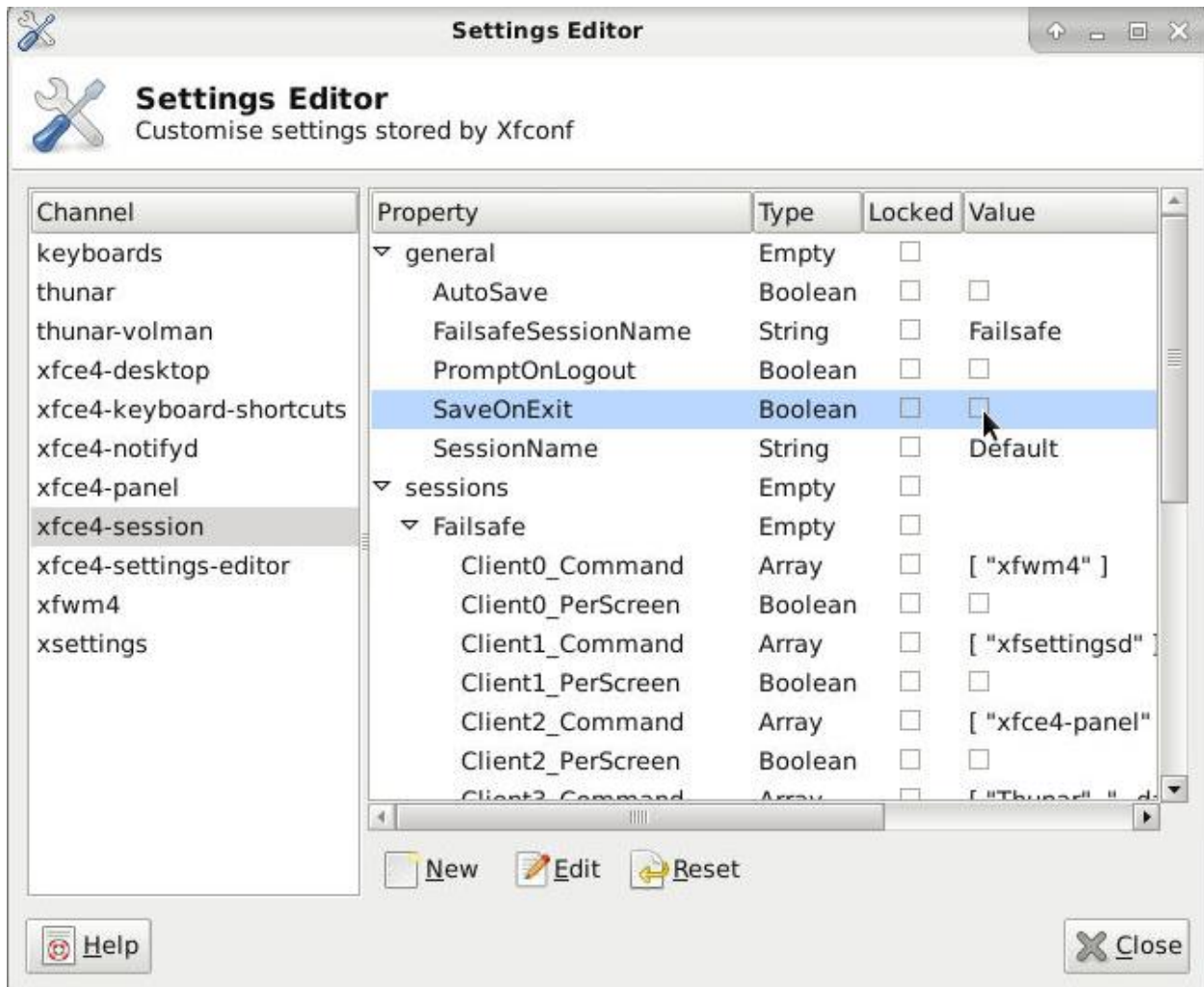
21. Correct session settings

Go to Applications->Settings->Session and Startup

Uncheck Automatically save session on logout under Logout Settings, and click Close

Go to Application->Terminal Emulator and write the following command

```
xfce4-settings-editor
```



Under xfce4-session->Uncheck SaveOnExit if it is checked. Without this step, two Mumble clients will be running (one that is remembered from the last session and the new one that autostarts) and they will disconnect each other as only one user with that certificate can be logged in at the same time. Click Close and reboot once more to check that only one Mumble is running

```
sudo reboot
```

If everything is working, turn off the Raspberry Pi.

```
sudo poweroff
```

22. Install the Raspberry Pi into the enclosure

Check the base station hardware installation guide

23. Install python libraries

After the last step you can't access the screen, as the Raspberry Pi is mounted in the base station enclosure. Instead you have to connect to it with putty (under hostname enter your base station IP) and click Open, login with pi and your password.

Install the OLED library

```
sudo apt install python3-dev python3-pip libfreetype6-dev libjpeg-dev
sudo pip3 install --upgrade pip
sudo pip3 install --upgrade setuptools
sudo pip3 install --upgrade luma.oled
```

Install GUI automation libraries

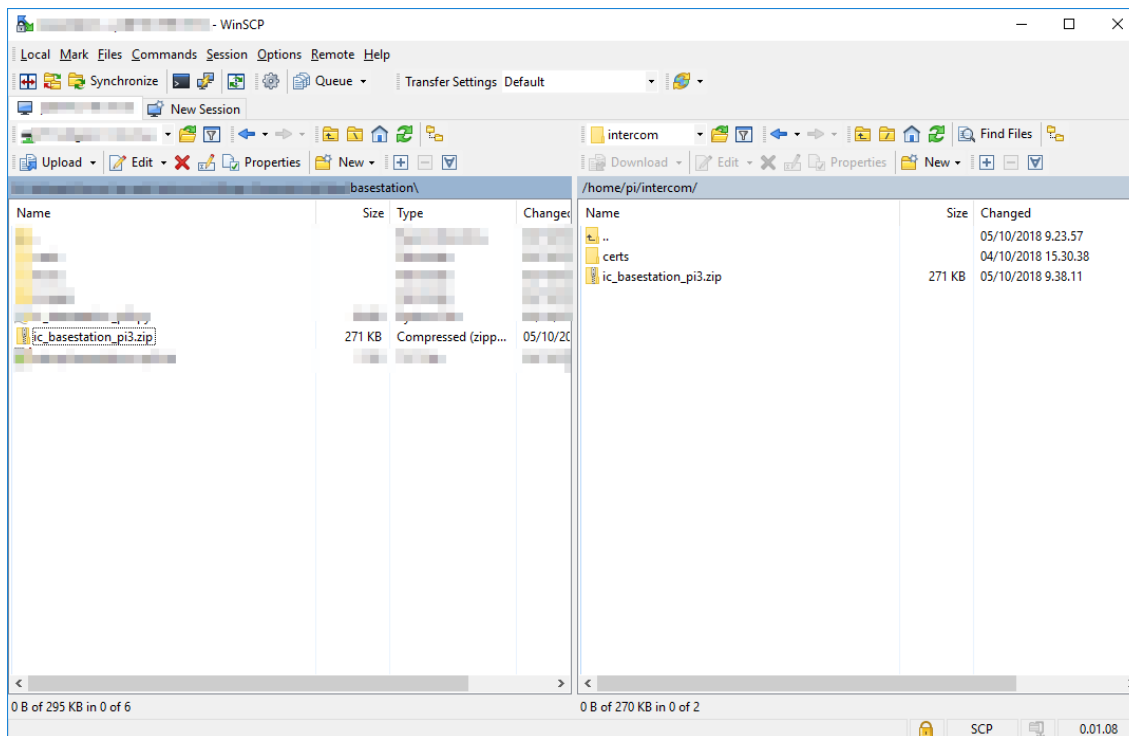
```
sudo apt install libx11-dev libxtst-dev
sudo pip3 install autopy3
```

Install MQTT library

```
sudo pip3 install paho-mqtt
```

24. Copy and extract base station script on Raspberry Pi

Use WinSCP on your Windows computer to copy the ic_basestation_pi3.zip file to the base station (/home/pi/intercom/ic_basestation_pi3.zip).



Back in putty

```
cd ~/intercom
unzip ic_beltpack_pi3.zip
rm ic_beltpack_pi3.zip
```

25. Copy MQTT server certificate

login in to your server (the example server is at 192.168.10.125)

```
ssh pi@192.168.10.125
```

(answer Yes)

Copy server cert to base station

```
scp /etc/mosquitto/certs/ca.crt pi@192.168.10.52:/home/pi/intercom
(replace IP with your base stations, answer yes and enter the base station password)
```

Disconnect from server CTRL+d and list the folder to check that the ca.crt is there

```
ls -la
```

26. Test the intercom script

Before you start the script you must edit it to connect to the right IP

```
pico ~/intercom/ ic_basestation_pi3.py
```

edit the following lines to match your setup:

mqttServer

mqttUser

mqttPass

Save file (CTRL+x) and start script

```
python3 ic_basestation_pi3.py
```

You should now see your IP and SRV UNAVAIL on the display. CTRL+c to quit

28. Make intercom startup script

```
pico ~/start_intercom.sh
```

```
#!/bin/bash
#START INTERCOM
DISPLAY=:1.0 XAUTHORITY=/home/pi/.Xauthority sudo /usr/bin/python3
/home/pi/intercom/ic_basestation_pi3.py &
#WAIT 30s AND SET MIC LEVEL
sleep 30
export PULSE_RUNTIME_PATH="/run/user/$(id -u)/pulse/"
# EXPERIMENT WITH THE LAST NUMBER TO GET THE RIGHT MIC LEVEL (CAN BE DONE FROM CMD LINE)
/usr/bin/pacmd set-source-volume 1 100000
```

observe that there should not be a line break after python3, everything should be on the same line like
.../python3 /home/...

Save file (CTRL+x)

```
chmod +x start_intercom.sh
```

Make the intercom script start automatically

```
crontab -e
```

(choose 2 nano if asked)

add the following line to the end of the file

```
@reboot /home/pi/start_intercom.sh
```

CTRL+x to exit and save changes (do not change filename)

```
sudo reboot
```

After boot you should see a signal meter, the text UNKNOWN and your IP on the display

29. Configure the base station role

On your Windows computer open a browser and go to http://YOURSERVER_IP:3000

You now need to configure the base station, click the Wrench (the connection symbol on the left should now be green because the base station is online) on icbs1

DEVICE:
icbs1

CURRENT ROLE:

DIR

ASSIGN	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8	DESCRIPTION
CAMS									ALL CAMERAS
CAM1									CAMERA 1
CAM2									CAMERA 2
DIR									DIRECTOR BASESTATION
[VIDEO]									VIDEO CHANNEL + SUB CHANNELS

OBSERVE THAT CHANGES WILL BE STORED BUT ONLY APPLIED WHEN DEVICE IS ONLINE

Set config

Select DIR as Role and the rest as the picture, click Set config

Your base station display should now be updated accordingly, it is now time to setup Mumble to match the script role and who you can talk to.

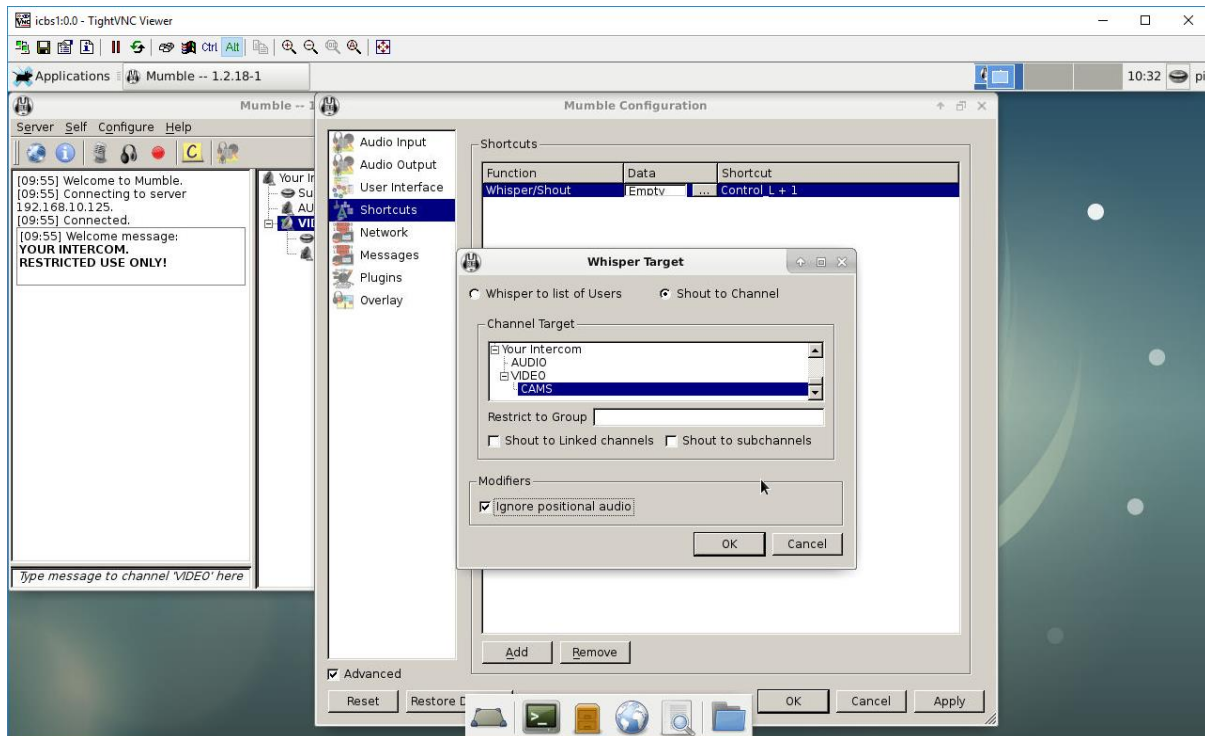
30. Continue to configure Mumble part 2

Connect with tightvnc to your base station, you want to talk to all cameras (CAMS) on channel 1, CAM1 on channel 2, CAM2 on channel 3 and the VIDEO channel with sub channels on 4-8. For this to work all users that you want to talk to needs to be connected to the server.

Let's start with the CAMS channel

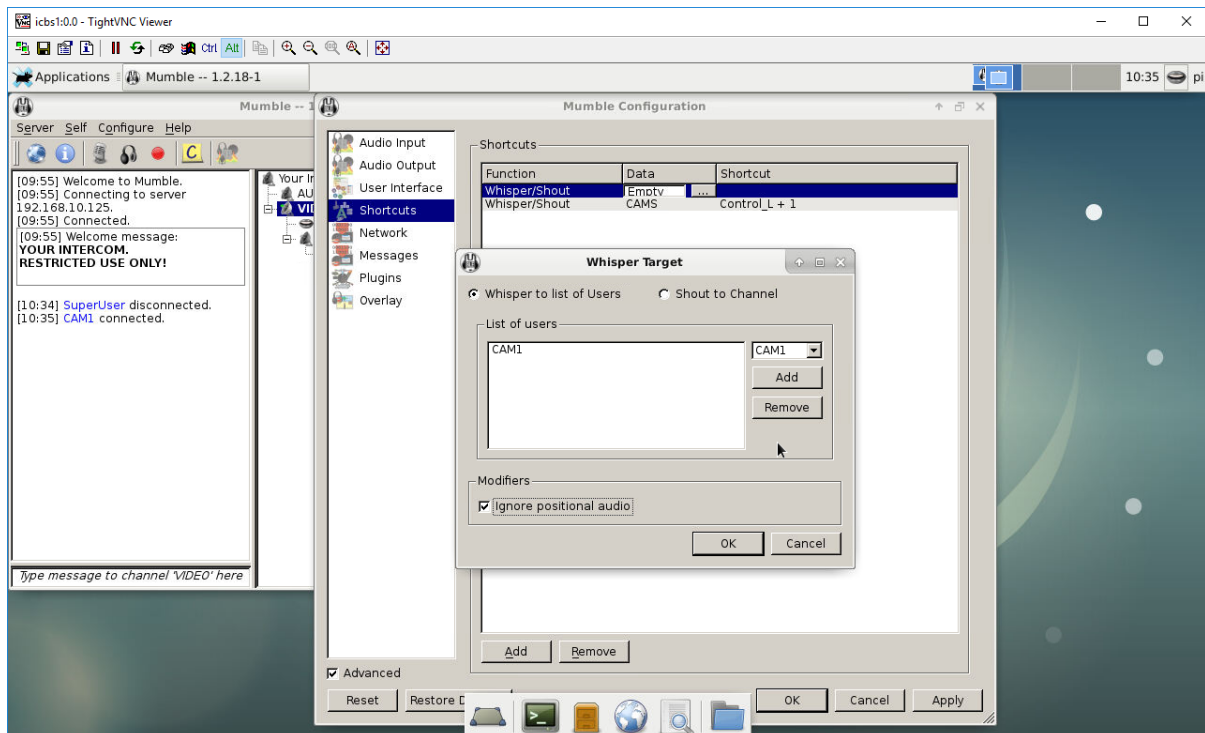
Under Shortcuts - > Change Push-to-Talk to Whisper/Shout (PTT transmits only the the same channel where the user is)

Under Shortcuts - > Click 3 dots under Data -> Choose Shout to channel -> Click on CAMS (UNDER VIDEO), ignore positional audio and OK



Click ADD -> Double click Unassigned -> Select Whisper/shout -> Under Data click the three dots -> Choose Whisper to list of Users -> Choose from dropdown on right CAM1, ignore positional audio and OK (Whisper to list of Users requires the user to be logged on to the server)

Click under Shortcut to assign keyboard shortcut (This should also be CTRL + 2) and OK



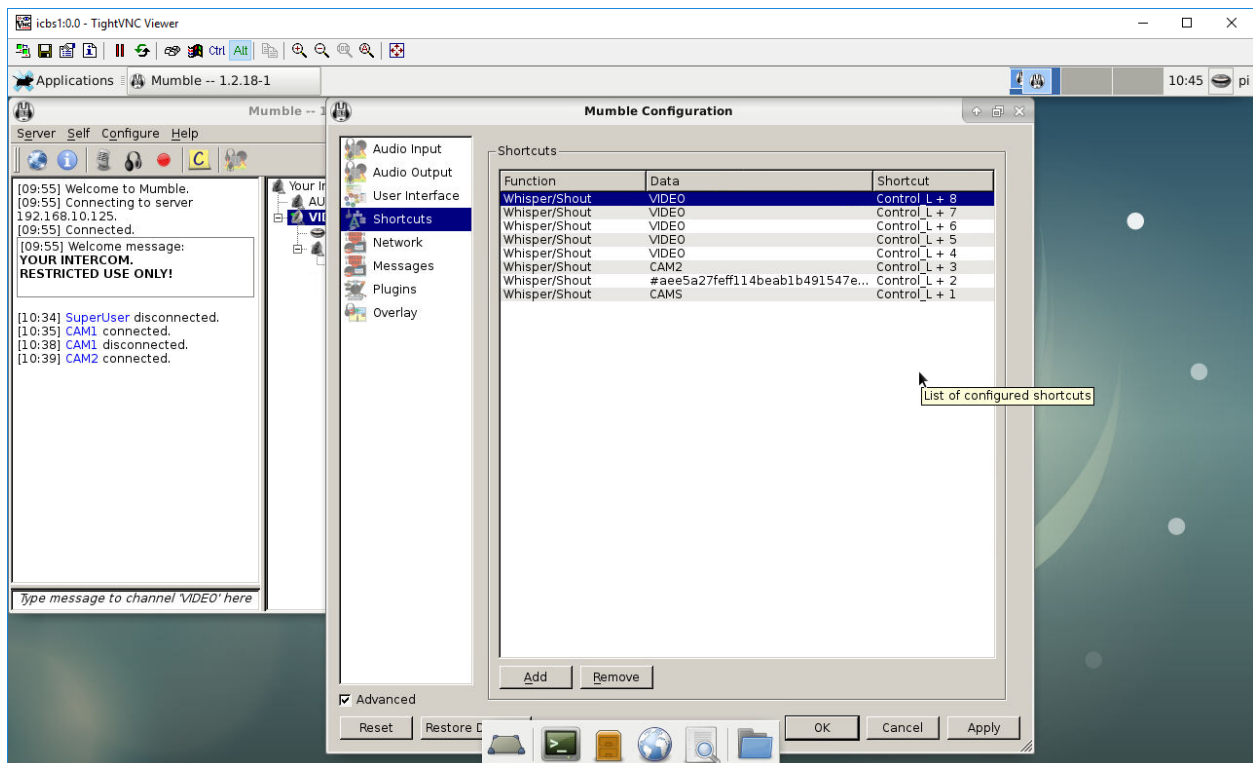
Click ADD -> Double click Unassigned -> Select Whisper/shout -> Under Data click the three dots -> Choose Whisper to list of Users -> Choose from dropdown on right CAM2, ignore positional audio and OK

Click under Shortcut to assign keyboard shortcut (This should also be CTRL + 3)

Next we add the VIDEO channel (to channels 4-8) repeat the next steps 5 times but remember to change the shortcut CTRL+4,5,6,7,8

Click ADD -> Double click Unassigned -> Select Whisper/shout -> Under Data click the three dots -> Choose Shout to channel -> Click on VIDEO, (if you select Shout to subchannels users in both VIDEO and CAMS can hear you) ignore positional audio and OK

Click under Shortcut to assign keyboard shortcut

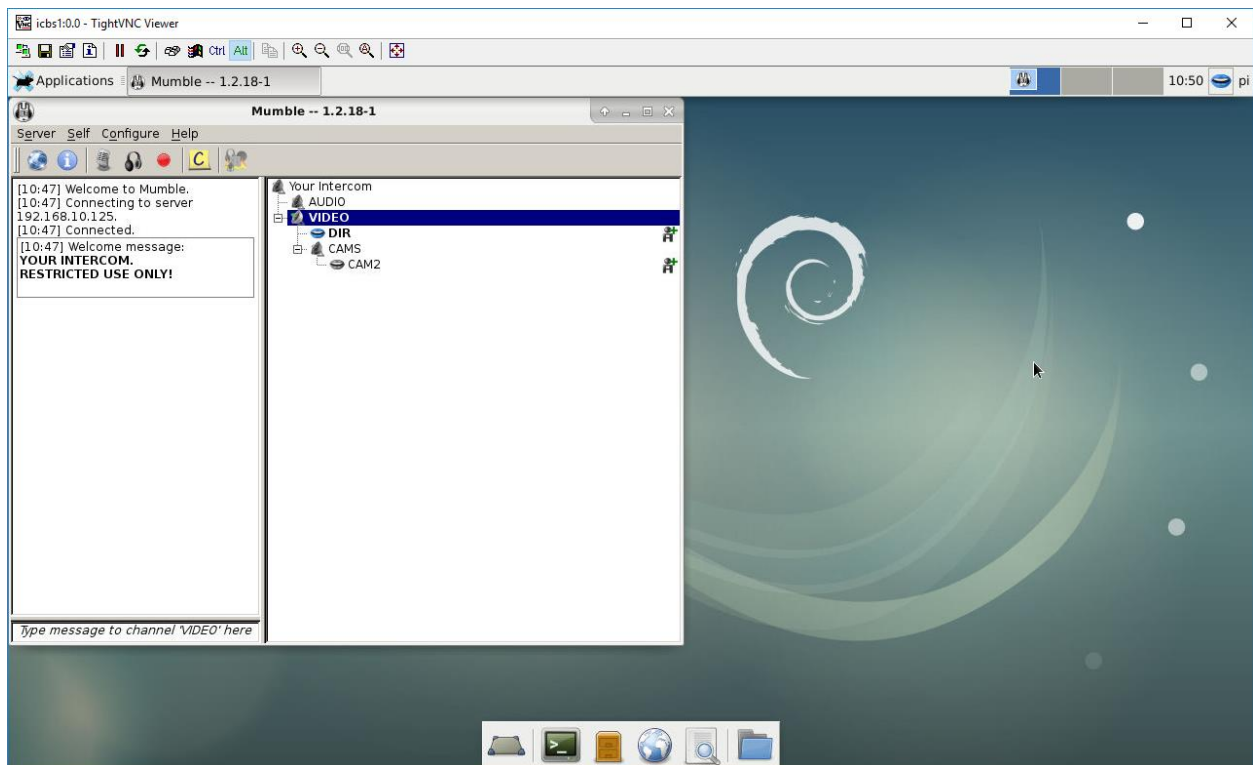


This is what it should look like after you are ready, in this picture CAM1 is no longer connected (had only one beltpack available when I did this guide so I had to change CAM1 to CAM2 after I assigned CAM1) and then you will see Mumbles unique id for the user. When the user is connected it will change to CAM1

Click Apply followed by OK (and remember to quit/start Mumble again)

It is possible to have own channels for users so you can use “Shout to channel” without needing the user to be online when doing the setup. Another good reason for having beltpack users in own channels are in case the beltpack user disconnects from Mumble. Mumble then stop broadcasting to that user. The python script that controls the talk buttons does not know if a user is disconnected from Mumble continues to send for example CTRL+2 (talk to CAM1), when the user connects to Mumble the users will not hear anything before the base station button is released and pressed again.

If you now hold down the first button above the display on the base station you can see that the icon before DIR turns blue (the director is talking)



32. Conclusions

If everything works you are now ready to start using the base station. Now you can make an image of the sdcard so that you can easily duplicate it to other base stations.

Before making an image of the card, consider changing the IP, hostname and Mumble certificate to something not used normally. If you must deploy a new device during production the device the image is based on will disconnect when the new device with the same name/IP/Mumble certificate joins the network.

If you have access to a linux machine, you can use ddrescue and pishrink (replace sdc with where your sd card is found, use sudo dmsg to find out)

sudo ddrescue /dev/sdc beltpack.img rescue.log

Download and read more here <https://github.com/Drewsif/PiShrink>

```
sudo pishrink.sh beltpack.img
```

Pishrink (without the -s) will automatically expand to use the whole sd card when started for the first time

When you deploy the image to new devices you must remember to do the following things after deployment:

Change hostname (step 1) and IP (step 3)

Reboot to use the new name and IP

Load a new certificate in Mumble and setup who to talk to

Setup the same information in the intercom setup webgui.