How to build a Translation system

Table of Contents

Purpose

The scope of this project is to build a system to easily spread a translation, made by a human, over the air. There are plenty of professional system out in the world. But they can be fairly expensive. So the initial requirement came to me some months ago when in the Church where I usually go I was asked to evaluate how to do it. After a short research I found out some very nice howtos regarding the software part. Essentially two main software are used: icecast to create an shareable stream and darkice to encode the audio input and pass to icecast. The Hardware part is a Raspberry PI 2 with a Wifi Access Point. The users will listen the translation, connecting their smartphone to the local wifi provided, using a media stream software like VLC or chrome that gives out of the box such functionality.

What do you need?

1 x Raspberry PI – I used the *v2 because I had that one. But the PI 3 would be even better*

1 x cf memory card >= 16 Gb

1 x Wifi Access point – I used the *TP-Link Archer C59 AC1350 Dual Band WLAN Router*

1 x Usb audio dongle – I used this one: *Sabrent USB External Aluminum Stereo Sound Adapter*

1 x Power Socket Strip - I used this: *Brennenstuhl Eco-Line Power Socket Strip with Switch Triple 1,5 m*

1 x Wide Self Adhesive Sticky Back – I used this: *TRIXES 1M Long 20mm Wide Self Adhesive Sticky Back Black Strips Hook and Loop Fastener*

1 x wooden board ~ 25 x 25 cm

Tools needed

1 x wood saw

1 x screw driver

1 x sd reader

Description

Software

First of all you have to set up your raspberry PI. I used the Ubuntu version because it contains all the software I needed. You can download an image here: https://wiki.ubuntu.com/ARM/RaspberryPi

Select the correct version for you RP version.

Follow the nice info provided to install it: https://wiki.ubuntu.com/ARM/RaspberryPi#Installation

I think I did the initial setup connecting the RP to an HDMI video with a usb keyboard and mouse but may be you can avoid this step and connect directly via ssh.

However I enabled the ssh daemon and then I started to hack it remotely.

At this point you need to configure the software:

1. Connect to the raspberry

   ```
   ssh raspberry
   ```
2. Check out the repo

   ```
   mkdir /workdir ; cd /workdir ; git clone
   https://github.com/zioalex/Raspberry-PI.git
   ```
3. Install the needed packages:

   ```
   apt-get install icecast2 darkice
   ```
4. Copy the configuration files

   ```
   cp -b -S ".bak" /workdir/Raspberry-PI/Translation_System/configs/etc/*
   /etc
   ```
5. Added a new systemd service to start darkice; it is called darkice2

   ```
   systemctl start darkice2 ; systemctl enable darkice2
   ```
6. Restart icecast2; it is managed through rc files. It should be already configured to autostart

   ```
   /etc/init.d/icecast2 restart
   ```
7. To support the point **(5)** you must disable udisk2.service to manage the mount/umount of the usb key via script

   ```
   systemctl stop udisk2.service; systemctl disable udisk2.service
   ```

Audio notes

The additional usb dongle, if unique, should be at the address **hw:1,0.** If not verify with the follow command all the Audio device recognized by the system:

```
aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

When you have identified your card try to record from it:

```
arecord -d hw:1.0 -t wav test.wav
```

And try to play it:

```
aplay test.wav
```

Automation details

Because the raspberry cannot be switched of with a soft button I added some automation to recreate such kind of behavior. I created an udev rule that switchoff the RPII when the audio usb dongle is removed. To identify the correct usb dongle refer to the “Udev Custom rules” provided

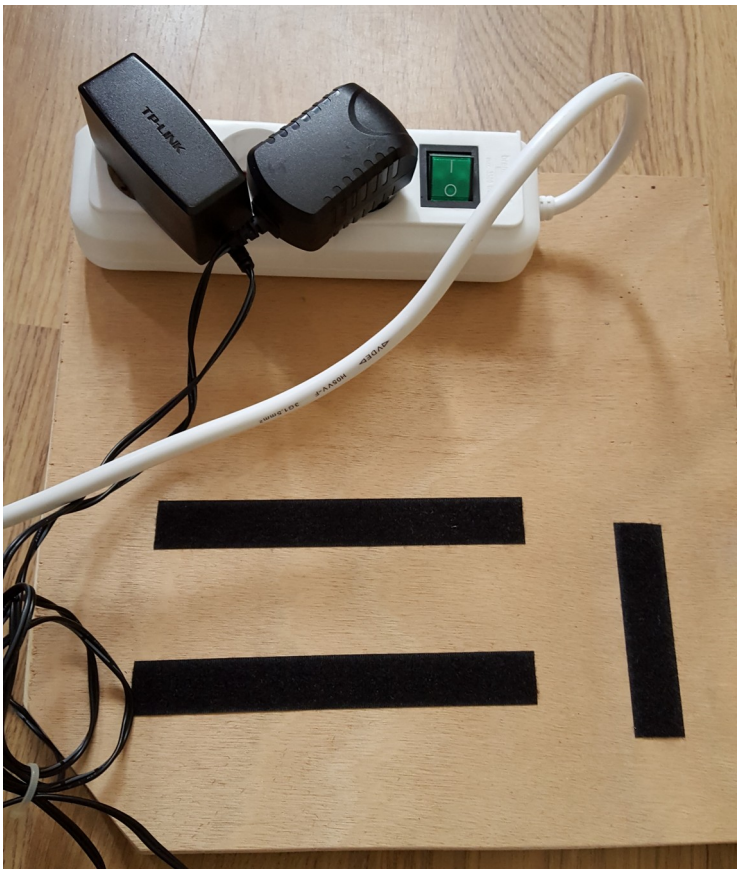
```
> cd Raspberry-PI/Translation_System/configs/etc/udev/rules.d
> cat 00-audio_off_poweroff.rules
ACTION=="remove", KERNEL=="pcmC1D0p", RUN+="/bin/systemctl poweroff"
```

The second automation copy the saved audio files to a usb key inserted, taking care of the numeration and of shutdown the RPII when the usb key is removed:

```
> cat 01-USBKEY_copy.rules
ACTION=="add", DRIVERS=="usb-storage", KERNEL=="sda1",
RUN+="/usr/local/sbin/copy_recording.sh"
```

Assembling

After all the software part you I assembled everything as you can see in the follow pictures:



Conclusion

This project has been interesting! Not so incredible difficult but tricky in some part. The Wifi connectivity problem was a major cause of lost time but then I realized that I can use radio transmitter and then the RPII is attached directly to the WIFI access point via cable and therefore maximum performance and minimum delay.

Was a bit difficult also to understand which was the correct audio device to use.

The system is now used by several months without any problem.

Regarding the performance we have between 0.5 – 1.5 seconds of delay not a big issue for the listeners.

Any feedback or improvement is always well accepted: https://github.com/zioalex/Raspberry-PI/issues/new

Goodies

In the Docs dir you can find some documents I created for the support team (OPERATIONAL_GUIDE.odt) and the user guide (User_guide.odt).

Improvements

Software installation and configuration via puppet / ansible or other.

Add a second audio usb dongle to translate multiple language contemporary.

Links

Type vs Range

https://www.geckoandfly.com/10041/wireless-wifi-802-11-abgn-router-range-and-distance-comparison/

USB Audio Dongle problem

https://www.raspberrypi.org/forums/viewtopic.php?f=45&t=132046

Realtime clock

Not present on the Raspberry Pi2. An external module is needed:

http://www.hobbytronics.co.uk/raspberry-pi-real-time-clock

Misc

http://raspberrypi.stackexchange.com/questions/639/how-to-get-pulseaudio-running

http://askubuntu.com/questions/426831/lxde-auto-login

http://raspberrypi.stackexchange.com/questions/32677/setup-microphone-stream-and-turn-your-raspberry-pi-into-a-baby-phone

Real time audio config with jack

http://wiki.linuxaudio.org/wiki/raspberrypi

Full project with icecast and darkice

https://stmllr.net/blog/live-mp3-streaming-from-audio-in-with-darkice-and-icecast2-on-raspberry-pi/

Network configuration in raspberry

https://www.raspberrypi.org/forums/viewtopic.php?t=44044

https://www.raspberrypi.org/forums/viewtopic.php?t=7592

Custom systemd configs

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sect-Managing_Services_with_systemd-Unit_Files.html

Ubuntu and RaspberryPi

https://wiki.ubuntu.com/ARM/RaspberryPi

Wifi problem with RT5370

https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=40474&start=25

Disable console screensaver

http://askubuntu.com/questions/138918/how-do-i-disable-the-blank-console-screensaver-on-ubuntu-server

Udev custom rules

http://askubuntu.com/questions/284224/autorun-a-script-after-i-plugged-or-unplugged-a-usb-device

https://wiki.archlinux.org/index.php/udev#Testing_rules_before_loading

Raspberry babyphone

http://raspberrypi.stackexchange.com/questions/32677/setup-microphone-stream-and-turn-your-raspberry-pi-into-a-baby-phone

Wireless/Usb Radio Microphone

http://www.bax-shop.nl/usb-microfoon/samson-stage-xpd1-headset-draadloze-usb-microfoon-2-4-ghz?gclid=CMS_pcOm1tACFUmeGwodqkUAMg

Network problem with WIFI DONGLE. Not solved

https://www.raspberrypi.org/forums/viewtopic.php?t=61665

CHROOTED ICECAST

https://forum.sourcefabric.org/discussion/13883/icecast-on-port-80/p1