

Bonjour,

Je souhaite mettre en œuvre une vidéo de qualité dans mes émissions DATV, et avec de l'audio en H264

Pas de mystère, il faut un bon TS, donc avec les bons timings rythmant le flux. La « suite » logicielle ci-dessous permet de diffuser une excellente vidéo (dans un premier temps) via un Raspberry et sa caméra et un Lime mini (Lime USB je ne sais pas) et ensuite de rajouter l'audio (via un convertisseur USB). Je n'ai rien inventé, je joue le rôle d'un intégrateur seulement.

Je présume que les personnes voulant se lancer, connaissent les bases de l'utilisation et des commandes linux utilisées, les logiciels de type Putty / WinSCP, ainsi que la manière de charger l'OS Raspbian sur la carte SD pour démarrer.

Je m'appuie sur les logiciels d'Evariste F5OEO que l'on trouve sur son github :

<https://github.com/F5OEO?tab=repositories>

Jean-Pierre, sur le site <http://www.vivadatv.org/> a donné des éléments pour démarrer.

Plutôt que de faire une installation manuelle, j'ai créé un fichier batch qui facilite l'installation ou la réinstallation en cas de modification/évolution. Je ne suis pas un habitué de la programmation, les instructions utilisées sont basiques, et j'utilise l'adressage direct et non relatif.

Il nous faut donc charger et compiler sur raspberry, l'OS est stretch light :

<https://raspbrian-france.fr/telechargements/>

les logiciels : avc2ts / dvb2iq / Limesdr_send depuis le site github

Installation :

Mettre le fichier install.sh (fourni sur Vivadatv.org) sous /home/pi

Rendre le fichier exécutable : `chmod +x install.sh`

Exécuter ! : `./install.sh`

Une fois la compilation exécutée (attention comptez 15 minutes) on se place dans le répertoire « bin » créé et on peut rentrer en exemple la commande suivante, qui enchaîne les trois fonctions et qui va générer une émission par le Lime à 437Mhz / SR 250 / DBB-S2 / QPSK (entrée vidéo dans ce cas : la Plcam)

```
../avc2ts/avc2ts -m 403200 -b 220000 -s F1FAQ -x 768 -y 576 -f 25 -d 800 -o /dev/stdout  
|../libdvbmod/DvbTsToIQ/dvb2iq -s 250 -f 5/6 -r 4 -m DVBS2 d-c QPSK -p| sudo ./limesdr_send -f 437e6 -b 2.5e6 -s  
250000 -g 0.8 -p 0.05 -a BAND2 -r 4 -l 102400
```

A vous bien sûr de modifier les paramètres de la chaîne pour l'adapter à vos besoins et essais !

Pour aller plus loin et rajouter l'audio :

Cette partie pose encore problème chez moi et ne fonctionne pas totalement, mais je donne des pistes que j'ai glanées à droite et à gauche. Taper les commandes suivantes :

`mkfifo audioin.wav "création fifo son"`

`arecord -f S16_LE -r 48000 -c 2 -B 100000 -D plughw:1,0 > audioin.wav & "lancement acquisition son via USB"`

et ensuite lancer la commande précédente pour la vidéo avec des paramètres supplémentaires :

```
../avc2ts/avc2ts -m 403200 -b 220000 -s F1FAQ -x 768 -y 576 -f 25 -d 800 -o /dev/stdout  
|../libdvbmod/DvbTsToIQ/dvb2iq -s 250 -f 5/6 -r 4 -m DVBS2 d-c QPSK -p| sudo ./limesdr_send -f 437e6 -b 2.5e6 -s  
250000 -g 0.8 -p 0.05 -a BAND2 -r 4 -l 102400 -a audioin.wav -z 48000 > /dev/null &
```

Voilà ! Fait pour être amélioré ! Communiquez sur le site !

73 – Christian

F1FAQ

Ci- après les paramètres acceptés par ces logiciels

avc2ts pour la capture, l'encodage et fabrication du TS

Usage:

```
rpi-avc2ts -o OutputFile -b BitrateVideo -m BitrateMux -x VideoWidth -y VideoHeight -f Framerate -n MulticastGroup [-d PTS/PCR][ -v ][ -h ]
```

- o path to Transport File Output
- b VideoBitrate in bit/s
- m Multiplex Bitrate (should be around 1.4 VideoBitrate)
- x VideoWidth (should be 16 pixel aligned)
- y VideoHeight (should be 16 pixel aligned)
- f Framerate (25 for example)
- n Multicast group (optionnal) example 230.0.0.1:10000
- d Delay PTS/PCR in ms
- v Enable Motion vectors
- i IDR Period
- t TypeInput {0=Picamera,1=InternalPatern,2=USB Camera,3=Rpi Display,4=VNC,5=ffmpeg}
- e Extra Arg:
 - For usb camera name of device (/dev/video0)
 - For VNC : IP address of VNC Server. Password must be datv
 - For ffmpeg : url or file to stream
- p Set the PidStart: Set PMT=PIDStart,Pidvideo=PidStart+1,PidAudio=PidStart+2
- s Set Servicename : Typically CALL
- a Raw PCM audio(48Khz stereo) Filename
- z Set AAC audio bitrate (32000 by default)
- h help (print this help).

Exemples : ./avc2ts -o result.ts -b 1000000 -m 1400000 -x 640 -y 480 -f 25 -n 230.0.0.1:1000
ou

./avc2ts/avc2ts -m 403200 -b 220000 -x 640 -y 480 -f 25 -d 800 -i 30 -s F6DZP -o /dev/stdout

dvb2iq pour la génération de IQ:

Usage:

```
dvb2iq -s SymbolRate [-i File Input] [-o File Output] [-f Fec] [-r upsample] [-m Modulation Type] [-c Constellation Type] [-p] [-h]
```

- i Input Transport stream File (default stdin)
- o OutputIQFile (default stdout)
- s SymbolRate in KS (10-4000)
- f Fec : {1/2,3/4,5/6,7/8} for DVBS {1/4,1/3,2/5,1/2,3/5,2/3,3/4,5/6,7/8,8/9,9/10} for DVBS2
- m Modulation Type {DVBS,DVBS2}
- c Constellation mapping (DVBS2) : {QPSK,8PSK,16APSK,32APSK}\
- p Pilots on(DVBS2)\
- r upsample (1,2,4) Better MER for low SR(<1M) choose 4
- h help (print this help).\

Exemple : `./dwb2iq -s 1000 -f 7/8 -r 4 -m DVBS2 -c 8PSK -p`
ou
`./dwb2iq -s 250 -f 5/6 -r 4 -m DVBS2 -c QPSK -p`

limesdr_send pour la génération de la modulation par lime SDR mini:

Usage: `limesdr_send <OPTIONS>`
`-f <FREQUENCY>`
`-b <BANDWIDTH_CALIBRATING>` (default: 200e3)
`-s <SAMPLE_RATE>` (default: 2e6)
`-g <GAIN_NORMALIZED>` (default: 1)
`-l <BUFFER_SIZE>` (default: 1024*1024)
`-p <POSTPONE_EMITTING_SEC>` (default: 3)
`-d <DEVICE_INDEX>` (default: 0)
`-c <CHANNEL_INDEX>` (default: 0)
`-a <ANTENNA>` (BAND1 | BAND2) (default: BAND1)
`-i <INPUT_FILENAME>` (default: stdin)
`-r upsample (1,2,4)` Better MER for low SR(<1M) choose 4

Exemple :

```
sudo ./limesdr_send -f 437e6 -b 2.5e6 -s 250000 -g 0.8 -p 0.05 -a BAND2 -r 4 -l 102400
```

logiciel acquisition du son : arecord

Tapez `arecord -l`, pour voir si le logiciel « voit » votre clef USB son
Chez moi je vois :

```
pi@tst:~ $ arecord -l
```

```
**** List of CAPTURE Hardware Devices ****
```

```
card 1: Device [USB Audio Device], device 0: USB Audio [USB Audio]
```

```
Subdevices: 1/1
```

```
Subdevice #0: subdevice #0
```

```
pi@tst:~ $
```

Paramètres de arecord [OPTION]... [FILE]...

`-h, --help` help
`--version` print current version
`-l, --list-devices` list all soundcards and digital audio devices
`-L, --list-pcms` list device names
`-D, --device=NAME` select PCM by name
`-q, --quiet` quiet mode

-t, --file-type TYPE file type (voc, wav, raw or au)
 -c, --channels=# channels
 -f, --format=FORMAT sample format (case insensitive)
 -r, --rate=# sample rate
 -d, --duration=# interrupt after # seconds
 -M, --mmap mmap stream
 -N, --nonblock nonblocking mode
 -F, --period-time=# distance between interrupts is # microseconds
 -B, --buffer-time=# buffer duration is # microseconds
 --period-size=# distance between interrupts is # frames
 --buffer-size=# buffer duration is # frames
 -A, --avail-min=# min available space for wakeup is # microseconds
 -R, --start-delay=# delay for automatic PCM start is # microseconds
 (relative to buffer size if <= 0)
 -T, --stop-delay=# delay for automatic PCM stop is # microseconds from xrun
 -v, --verbose show PCM structure and setup (accumulative)
 -V, --vumeter=TYPE enable VU meter (TYPE: mono or stereo)
 -l, --separate-channels one file for each channel
 -i, --interactive allow interactive operation from stdin
 -m, --chmap=ch1,ch2,.. Give the channel map to override or follow
 --disable-resample disable automatic rate resample
 --disable-channels disable automatic channel conversions
 --disable-format disable automatic format conversions
 --disable-softvol disable software volume control (softvol)
 --test-position test ring buffer position
 --test-coef=# test coefficient for ring buffer position (default 8)
 expression for validation is: $\text{coef} * (\text{buffer_size} / 2)$
 --test-nowait do not wait for ring buffer - eats whole CPU
 --max-file-time=# start another output file when the old file has recorded
 for this many seconds
 --process-id-file write the process ID here
 --use-strftime apply the strftime facility to the output file name
 --dump-hw-params dump hw_params of the device
 --fatal-errors treat all errors as fatal

Recognized sample formats are: S8 U8 S16_LE S16_BE U16_LE U16_BE S24_LE S24_BE U24_LE U24_BE S32_LE S32_BE U32_LE U32_BE FLOAT_LE FLOAT_BE FLOAT64_LE FLOAT64_BE IEC958_SUBFRAME_LE IEC958_SUBFRAME_BE MU_LAW A_LAW IMA_ADPCM MPEG GSM SPECIAL S24_3LE S24_3BE U24_3LE U24_3BE S20_3LE S20_3BE U20_3LE U20_3BE S18_3LE S18_3BE U18_3LE U18_3BE G723_24 G723_24_1B G723_40 G723_40_1B DSD_U8 DSD_U16_LE DSD_U32_LE DSD_U16_BE DSD_U32_BE

Some of these may not be available on selected hardware

The available format shortcuts are:

-f cd (16 bit little endian, 44100, stereo)

-f cdr (16 bit big endian, 44100, stereo)

-f dat (16 bit little endian, 48000, stereo)