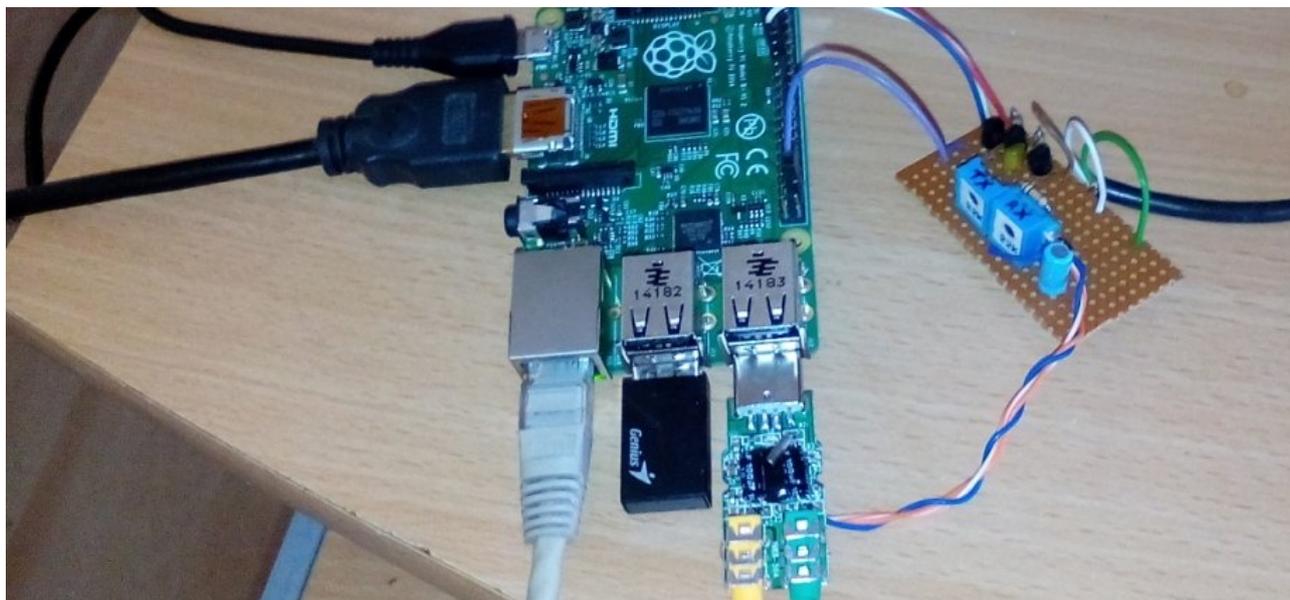


Installation de svxlink (EchoLink) sur Debian (raspberry B+ ou PC)



Préambule :

Il est intéressant d'installer Svxlink en lieu et place de Echolink dans un système radio , pour différentes raisons listé ci-dessous :

- Pas d'interface graphique nécessaire , ce qui permet d'utiliser un vieux PC ou un raspberry
- Le système peut fonctionner en autonome , même sans clavier /souris / écran
- Svxlink ,n'est pas juste un remplaçant sous Linux d'Echolink, il dispose d'autres fonctionnalités ainsi qu'une logique complète de relais et peut donc fonctionner en autonome sans internet (ceci sort du cadre de cet article) .
- facilement adaptable ou modifiable grâce à un langage de programmation simple (TCL)

Les parties spécifiques à chaque architecture seront identifiées par **PC** ou **Raspberry** en début de paragraphe qui les concerne, les parties communes seront identifiées par **Commun**.

Installation du système d'exploitation PC :

J'ai choisi pour la version PC d'installer une distribution optimisée pour les 486 , il s'agit de Wattos. disponible à cette adresse : <http://www.planetwatt.com/>

la version choisie pour ma part est wattOS-LXDE - R8 -32 bit disponible sur la page de téléchargement du site c'est un compromis entre la version light Microwatt et la version MATE-Desktop qui est la plus complète avec le bureau MATE (fork de Gnome) mais pose des problèmes avec le serveur de son PULSEAUDIO.

un noyau 32 bits est requis si vous ne voulez pas compiler le logiciel, un noyau PAE , prenant en charge + de 3Go de ram est disponible en 32bits dans la distrib si besoin.

N'importe quelle distribution à base de Debian est utilisable , (Ubuntu n'est pas debian ...)

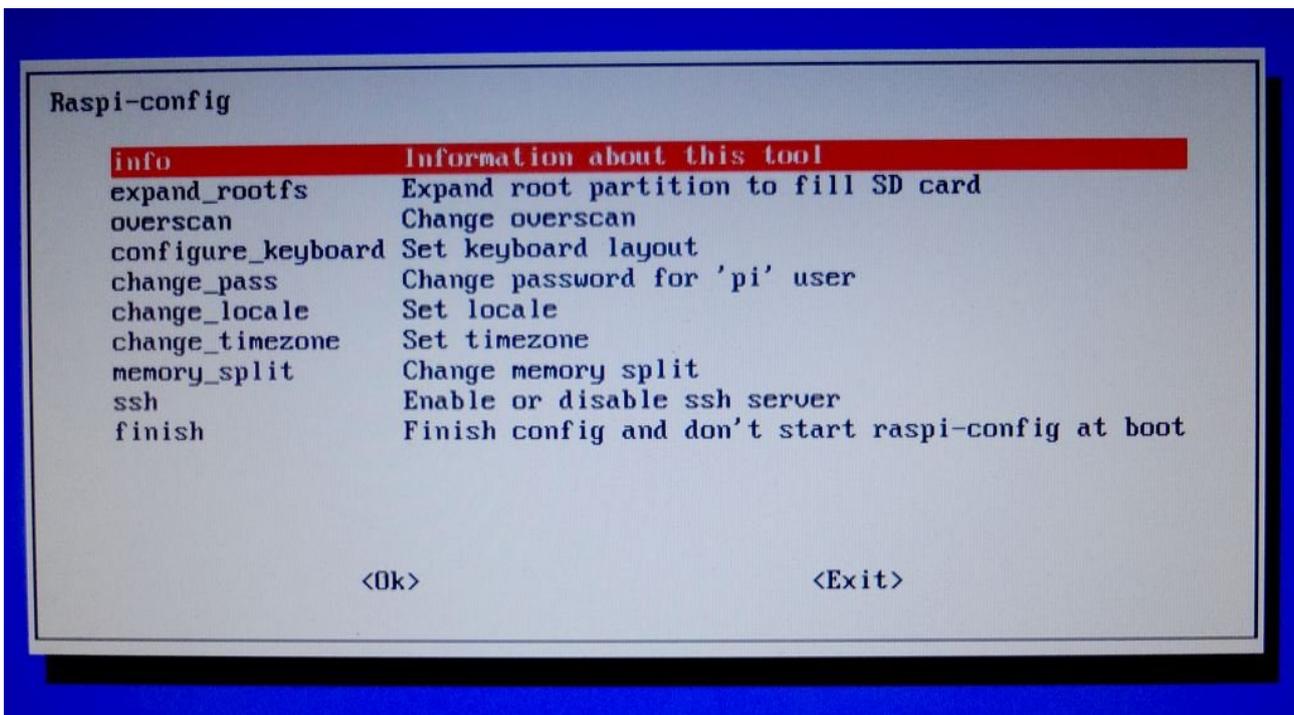
Pour l'installation , c'est un «live Cd» il faut graver l'image ISO et démarrer dessus , ou sur une clef USB je vous reporte à la documentation d'installation disponible sur le site , c'est en Anglais , mais Google translate peut aider les non anglophone. suivez les instructions d'installation , si tout se passe bien vous avez un système Linux fonctionnel sur votre PC auquel on peut rajouter des logiciels graphiques , comme prédiction de passage satellite, carnet de trafic , etc ..., mais le but est de monter Svxlink...

Installation du système d'exploitation sur le Raspberry B :

il faut installer le système d'exploitation sur la micro SD , on part sur une Debian. l'installation proprement parlé se fait sur le lecteur de carte d'un PC équipé de Linux ou Windows.
allez sur <http://www.raspberrypi.org/downloads/> et prenez l'image RASPBIAN en torrent ou le zip.
allez ensuite sur <http://www.raspberrypi.org/documentation/installation/installing-images/> et choisissez depuis quelle plateforme vous installé l'image sur la carte micro SD, suivez les instructions.

Mettez la carte dans le raspberry en y ayant branché un clavier USB , un câble réseau , et un écran HDMI (éventuellement un écran a vidéo composite voir ici pour le branchement sur le jack 3.5 <http://www.raspberrypi-spy.co.uk/2014/07/raspberry-pi-model-b-3-5mm-audiovideo-jack/#more-3718>)

lancer raspi-config et régler le fuseau horaire , le clavier , activer le ssh au démarrage et ex pendez rootfs (la carte SD) si elle fait plus de 4Go le mot de passe de l'utilisateur pi par défaut est raspberry



Commun au 2 plateformes:

pour ma part je fait toute la configuration en root ce qui est plus simple
login root , pasword : votre mot de passe

on commence par mettre a jour le système :

```
apt-get update  
apt-get upgrade
```

on installe les paquets suivants :

en une ligne, la magie de Apt ;)

```
apt-get install postfix procmail sox mutt alsa-utils alsa-plugins alsa-lib alsa-firmware mc tzdata ca-certificates git-core  
subversion libsigt++-2.0-dev g++ make libsigt++-1.2-dev libgsm1-dev libpopt-dev tcl8.5-dev libcrypt-dev libspeex-  
dev libasound2-dev
```

on reboot le système histoire que tout ça se remette d'aplomb et que les librairies soient relues et active.
reboot.

Note sur MC , mc est un clone de Norton commander sous linux qui permet de manipuler le système très facilement , un éditeur interne plus convivial que VI est inclus.

paquet Debian et installation PC:

on télécharge le paquet svxlink_13.12-1_i386.deb depuis cette adresse :
https://www.dropbox.com/s/1vfrzw6hpqbuxka/svxlink_13.12-1_i386.deb?dl=0

ainsi que les fichiers tuning_svxlink_sons.tar.gz a cette adresse :
https://www.dropbox.com/s/f1q32p5v0d16suq/tuning_svxlink_sons.tar.gz?dl=0

on installe le paquet svxlink : dpkg -i svxlink_13.12-1_i386.deb

(si vous voulez compiler votre version allez sur le site de svxlink et reportez vous aux instructions de compilation, installation.)11

paquet Debian et installation Raspberry:

on télécharge le paquet svxlink_armhf.deb depuis cette adresse :
https://www.dropbox.com/s/czn5askmavm8b81/svxlink_armhf.deb?dl=0

ainsi que les fichiers tuning_svxlink_sons.tar.gz a cette adresse :
https://www.dropbox.com/s/f1q32p5v0d16suq/tuning_svxlink_sons.tar.gz?dl=0

(vous pouvez compiler vous même le svxlink sur le raspberry ça prends environ une demi heure ...)
on l'installe avec dpkg -i svxlink_armhf.debian

Installation des voix françaises et tuning du logiciel Commun :

toutes les manipulations sont faites avec MC , reportez vous aux commandes Unix si vous voulez faire du mkdir , mv, cp et autre., «mc» est semi graphique à double fenêtre et bien plus intuitif que la ligne de commande pure et dure , enfin chacun fait comme il veut mais je ne donne ni un cours Linux , ni le mode d'emploi de «mc» reportez vous aux documentations sur le net.

on décompresse le fichier téléchargé dans root par exemple (il suffit d'appuyer sur entrée sur le fichier tar.gz , pour visualisé son contenu et F5 pour en copier le contenu vers l'autre fenêtre ..)
, on a une arborescence type «Linux» /etc /var /usr dans le répertoire Export contenu dans le tar.gz

les voix sont a copié dans /usr/share/svxlink pour avoir la synthèse vocale française.

le fichier /etc/svxlink/svxlink.conf et le fichier /etc/svxlink/svxlink.d/ModuleEcholink.conf sont des exemples fonctionnel (au call et mot de passe pres bien évidemment) .

ils prennent en charge , dans les exemples l'interface décrite plus bas (sur les ports GPIO du raspberry) si vous utilisez un autre hardware il faudra adapté en fonction du votre, voir plus bas explication sur le fichier svxlink.conf

les fichiers suivants sont a modifié ou créer, ils sont dans le tar.gz en exemple :

/boot/cmdline.txt , ce fichier est propre au **raspberry** on désactive la «console sur le GPIO»

/etc/inittab , ce fichier est propre au **raspberry** la console sur le GPIO est aussi désactivé

/etc/rc.local , ce fichier est lui aussi propre au **raspberry** on initialise les broches du GPIO pour les signaux RTS et CTS il permet en outre de démarrer le système (modif du 24 oct 2014)

voici son contenu :

```

-----
/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

# configuration des broches GPIO et démarrage de SVXLINK

# Radio "VHF" GPIO 17:PTT - GPIO 16:SQULECH
echo "17" > /sys/class/gpio/export &
echo out > /sys/class/gpio/gpio17/direction

echo "16" > /sys/class/gpio/export &
echo in > /sys/class/gpio/gpio16/direction

# Radio "UHF" GPIO 24:PTT - GPIO 23:SQUELCH
echo "24" > /sys/class/gpio/export &
echo out > /sys/class/gpio/gpio24/direction

echo "23" > /sys/class/gpio/export &
echo in > /sys/class/gpio/gpio23/direction

sleep 2

# Lancement de SvxLink
svxlink --pidfile=/var/run/svxlink.pid --logfile=/var/log/svxlink --runasuser=root --config=/etc/svxlink/svxlink.conf
--daemon

exit 0

```

les autres fichiers :

/etc/asound.conf , ce fichier initialise la carte son **Commun** (la première)
/etc/modprobe.d/alsa-base.conf ,ce fichier initialise la carte son USB comme étant la première carte son **Commun**
/etc/network/interface ,adresse IP fixe impérative et ouverture des ports sur la «BOX» **Commun**
/lib/lsb/init-functions.d/30-svxlink, paramètres nécessaire au fichier de démarrage ci-dessus **Commun**
/etc/svxlink, répertoire contenant les fichiers de configuration de svxlink ils sont décrit plus bas **Commun**
/usr/share/svxlink/events.tcl , fichier qui gère les événements **Commun**
/usr/share/svxlink/sounds , répertoire contenant les fichiers sons à copié tel quel **Commun**
/usr/share/svxlink/modules.d répertoire contenant les fichiers des modules ils doivent déjà existé **ne pas copié Commun**
/usr/share/svxlink/events.d répertoire contenant les scripts événements j'ai fait des modifs dans certains pour enlever les
«beep» dans EchoLink.tcl , et dans Logic.tcl , j'ai également rajouté un 1750 pour déclenché le relais local. **ne pas copié**
c'est a titre d'exemple.

Commun Paramétrage à proprement parler du logiciel :

Fichier /etc/svxlink/svxlink.d/ModuleEcholink.conf

ce fichier s'occupe de la connexion au réseau Echolink c'est le plus simple à éditer.

CALLSIGN= votre call (-L ou -R) reportez vous à la doc d'Echolink en général L pour un lien , R pour un relais
PASSWORD= votre password Echolink

le reste du fichier est à adapter comme SYSOPNAME, LOCATION, DESCRIPTION

Fichier /etc/svxlink/svxlink.conf

l'exemple donné ici est fonctionnel avec un raspberry dont l'interface radio est décrite plus bas , et les broches utilisées pour le ptt et la détection de squelch éventuelle sont la gpio 16 et gpio 17.

je vous reporte à la documentation de svxlink , 2 solutions internet , ou man svxlink.conf en anglais.

tel quel en changeant juste le call , le système est opérationnel , pensez à changer également la description pour l'APRS à la fin du fichier. un certain nombre de macros disponibles en DTMF depuis la voie radio sont définies principalement des raccourcis pour appeler d'autres digi echolink , par exemple la séquence DTMF D22# connecte le digi F5NLG-L.

sur un pc pour que le fichier soit directement utilisable avec l'interface «PC» décrite plus bas , il faut modifier dans la section [TX1] la façon dont on actionne le PTT

```
PTT_TYPE=SERIAL
```

```
PTT_PORT=/dev/ttyUSB0 pour un adaptateur usb-serie ou /dev/ttyS0 pour un port com 1
```

```
SERIAL_SET_PINS=DTR!RTS
```

Fichier /etc/svxlink/svxlink.d/ModuleMetarinfo.conf

ce fichier récupère la météo des aéroports , adaptez en fonction de votre localité si vous souhaitez la météo sur le système.

pour ma part j'ai changé l'ordre des Modules par rapport à ce qui est défini par défaut , le «numéro» de module est identifié par la variable ID dans les fichiers de configuration de /etc/svxlink/svxlink.d/

pour les autres modules je vous invite à lire la doc de svxlink.

à partir d'ici le système doit être fonctionnel, en le lançant par svxlink dans une invite de commande.

Sans la carte son USB ça ne démarre pas , si la carte son est mal définie sur un système pc , c'est pareil.

Note sur le réglage du «voix» en réception et la partie RX en général:

il est à mon avis préférable d'utiliser une détection squelch , la led RX par exemple du transceiver fournit une tension positive qu'il est sûrement possible d'utiliser ou une détection par CTCSS.

pour les essais ou en fonctionnement sur un relais dont l'émission est permanente pendant la période d'ouverture du squelch de celui-ci , c'est le cas ici , il n'y a pas d'autres alternatives, à moins d'avoir accès au dit relais et de pouvoir modifier certaines choses comme d'envoyer un ctcss pendant les périodes d'ouverture du squelch du relais.

le squelch logiciel et les tempos sont réglés au cas par cas .

Note sur un mini réseau maillé de digi , exemple du système en place actuellement

Un fonctionnement en étoile dans un réseau de plusieurs digi Echolink est préférable à un fonctionnement en série, les temps de latences se multiplient en fonctionnement «série» tout est encore en test sur ce point .

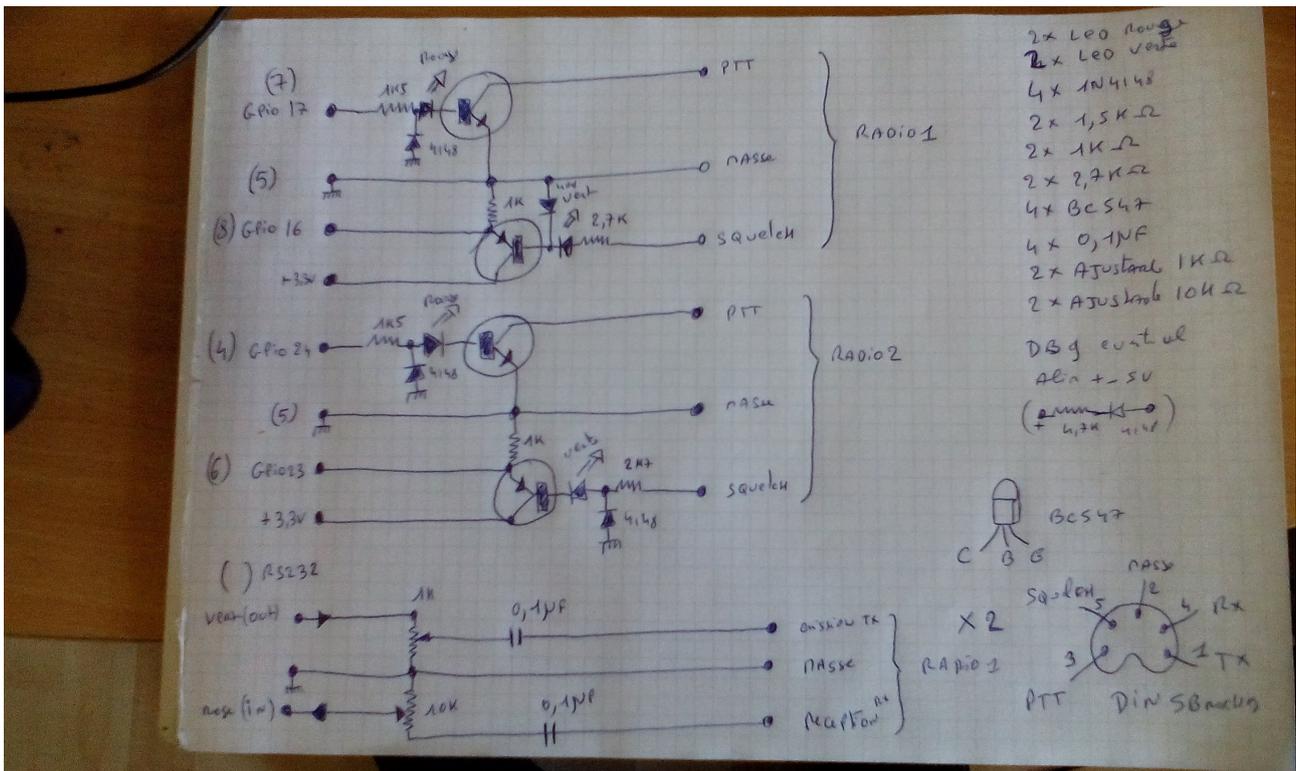
Annexe : exemples d'interfaces.

Interface PC et Raspberry :

si votre pc ne dispose pas de carte son , vous pouvez en mettre une en usb , le fichier /etc/modprobe.d/alsa-base.conf , dans les exemples défini ce type de carte , si votre pc dispose d'une carte son , le fichier généré à l'installation doit être ok.

elle est valable pour les 2 plateforme , elle prends en compte 2RTX pour la partie logique. dans le cas d'un PC il faut alimenter la partie détection de squelch avec environ + 5volts , dans le cas du Raspberry on utilise du **3,3V impérativement** issu du GPIO , toute autre interface plus sophistiqué doit fonctionner , avec transfo d'isolement etc

Il faut cependant que la détection de squelch **soit positive** malgré de nombreux shéma éronné ou elle est négative , ca ne fonctionne pas en negatif ...



Evolution 2 transceivers (24 octobre 2014) , un VHF et un UHF (en mode simplex)

L'évolution qui suit décrit le branchement de 2 transceivers sur la version Raspberry , ca doit être réalisable sur la version PC (non testé on utilise les signaux RTS CTS DSR DTR comme signaux de commutation avec un seul port série)

Aux ports GPIO 16 et 17 , on rajoute 2 GPIO pour commuter l'autre TX les GPIO 24 et 23.
et il faut dans mon exemple , 2 cartes Sons .

on copie le fichier /usr/share/svlink/events.d/SimplexLogic.tcl 2 fois :

VHFLogic.tcl

on modifie dans ce fichier : la 1^{er} ligne après les # :
namespace eval VHFLogic {

on fait pareil dans le 2eme Fichier renommer en
UHFLLogic.tcl

on modifie dans ce fichier : la 1^{er} ligne après les # :
namespace eval UHFLLogic {

le fichier /etc/svxlink/svxlink.conf deviens celui-ci :

```
-----  
[GLOBAL]  
MODULE_PATH=/usr/lib/svxlink  
LOGICS=VHFLogic,UHFLogic  
CFG_DIR=svxlink.d  
TIMESTAMP_FORMAT="%c"  
CARD_SAMPLE_RATE=48000  
LOCATION_INFO=LocationInfo  
LINKS=VUHFLink  
  
[VHFLogic]  
TYPE=Simplex  
RX=Rx1  
TX=Tx1  
MODULES=ModuleEchoLink,ModuleHelp,ModuleMetarInfo  
CALLSIGN=F5XYZ  
SHORT_IDENT_INTERVAL=30  
LONG_IDENT_INTERVAL=60  
EVENT_HANDLER=/usr/share/svxlink/events.tcl  
DEFAULT_LANG=fr_FR  
RGR_SOUND_DELAY=-1  
MACROS=Macros  
FX_GAIN_NORMAL=0  
FX_GAIN_LOW=-12  
ACTIVATE_MODULE_ON_LONG_CMD=2:EchoLink  
ONLINE_CMD=123456789  
#MUTE_RX_ON_TX=1  
#MUTE_TX_ON_RX=1  
  
[UHFLogic]  
TYPE=Simplex  
RX=Rx2  
TX=Tx2  
MODULES=ModuleHelp,ModuleMetarInfo,ModuleParrot  
CALLSIGN=F5XYZ  
SHORT_IDENT_INTERVAL=15  
LONG_IDENT_INTERVAL=60  
EVENT_HANDLER=/usr/share/svxlink/events.tcl  
DEFAULT_LANG=fr_FR  
  
# ici on active les beep sur la partie uhf  
RGR_SOUND_DELAY=0  
MACROS=Macros  
FX_GAIN_NORMAL=0  
FX_GAIN_LOW=-12  
ONLINE_CMD=123456789  
#MUTE_RX_ON_TX=1  
#MUTE_TX_ON_RX=1  
  
# on relie les 2 Logic  
[VUHFLink]  
CONNECT_LOGICS=VHFLogic,UHFLogic:99  
DEFAULT_ACTIVE=1  
  
[Macros]
```

```
[Rx1]
TYPE=Local
AUDIO_DEV=alsa:plughw:0
AUDIO_CHANNEL=0
# type de detection VOX,CTCSS,SERIAL,EVDEV,SIGLEV,GPIO
SQL_DET=GPIO
GPIO_SQL_PIN=16
# début du squelch si par exemple le TX emmet un bruit a la fermeture en ms
SQL_START_DELAY=60
# retard de detection 20 -100ms
SQL_DELAY=0
# fin de detection squelch apres fermeture de celui ci en ms
SQL_HANGTIME=2000
# extension surtout utile sur signaux faible en général 1000ms
#SQL_EXTENDED_HANGTIME=1000
#
#SQL_EXTENDED_HANGTIME_THRESH=1000
# oblige le squelch a se fermé si il est ouvert trop longtemps en secondes
SQL_TIMEOUT=300
# petite valeur , plus rapide en ms
VOX_FILTER_DEPTH=30
# a ajuster pour éviter des déclenchements intempestif 1000 ms
VOX_THRESH=1200
DEEMPHASIS=0
PREAMP=0
PEAK_METER=1
DTMF_DEC_TYPE=INTERNAL
DTMF_MUTING=1
DTMF_HANGTIME=100
#1750_MUTING=0
```

```
[Tx1]
TYPE=Local
AUDIO_DEV=alsa:plughw:0
AUDIO_CHANNEL=0
PTT_TYPE=GPIO
PTT_PIN=gpio17
#PTT_HANGTIME=1000
# coupe l'emeteur en cas d'emission permanente
TIMEOUT=600
# temps avant l'envoi de la BF une fois le ptt activé
TX_DELAY=10
PREEMPHASIS=0
DTMF_TONE_LENGTH=100
DTMF_TONE_SPACING=50
DTMF_TONE_AMP=-18
```

```
[Rx2]
TYPE=Local
AUDIO_DEV=alsa:plughw:1
AUDIO_CHANNEL=0
SQL_DET=GPIO
GPIO_SQL_PIN=23
SQL_START_DELAY=60
SQL_DELAY=0
SQL_HANGTIME=2000
SQL_TIMEOUT=300
DEEMPHASIS=0
#SQL_TAIL_ELIM=0
```

```
PREAMP=0
PEAK_METER=1
DTMF_DEC_TYPE=INTERNAL
DTMF_MUTING=1
DTMF_HANGTIME=100
#1750_MUTING=0
```

```
[Tx2]
TYPE=Local
AUDIO_DEV=alsa:plughw:1
AUDIO_CHANNEL=0
PTT_TYPE=GPIO
PTT_PIN=gpio24
#PTT_HANGTIME=1000
TIMEOUT=600
TX_DELAY=10
PREEMPHASIS=0
DTMF_TONE_LENGTH=100
DTMF_TONE_SPACING=50
DTMF_TONE_AMP=-18
```

```
[LocationInfo]
APRS_SERVER_LIST=euro.aprs2.net:14580
STATUS_SERVER_LIST=aprs.echolink.org:5199
LON_POSITION=
LAT_POSITION=
CALLSIGN=ER-F5XYZ
FREQUENCY=145.600
TX_POWER=10
ANTENNA_GAIN=6
ANTENNA_HEIGHT=10m
ANTENNA_DIR=-1
PATH=WIDE1-1
BEACON_INTERVAL=10
TONE=1750
COMMENT=>>> Serveur Vocal Radio SvxLink - Rpt <<<
```

Conclusion :

L'intérêt d'un tel réseau maillé , est de continuer à faire de la **RADIO** avec des moyens moderne, d'augmenter la portée de nos installations de type «Relais ou Transpondeurs» de joindre nos amis pendant les périodes de déplacement ou de vacances, certains diront que les relais ne sont pas de la radio , et qu'Internet tue le radio amateurisme .ce à quoi je ré-ponds que l'expérimentation et l'évolution sont notre raison d'être . depuis que ce système est en place , le relais R0 local revis malgré une panne sur un aérien , un duplexeur VHF est en cours de fabrication.
Skype n'a pas tué le radio amateurisme , et pourtant il est présent chez bon nombres d'OM.

pour recruter des jeunes ou moins jeunes, il faut montrer des choses ayant un attrait, étant un peu «sexy» et ce type de système va dans ce sens la, de plus il est accessible avec un talkie a 30 €

73» de Jean-Philippe , F5NLG
F5NLG-R : digi 224090