

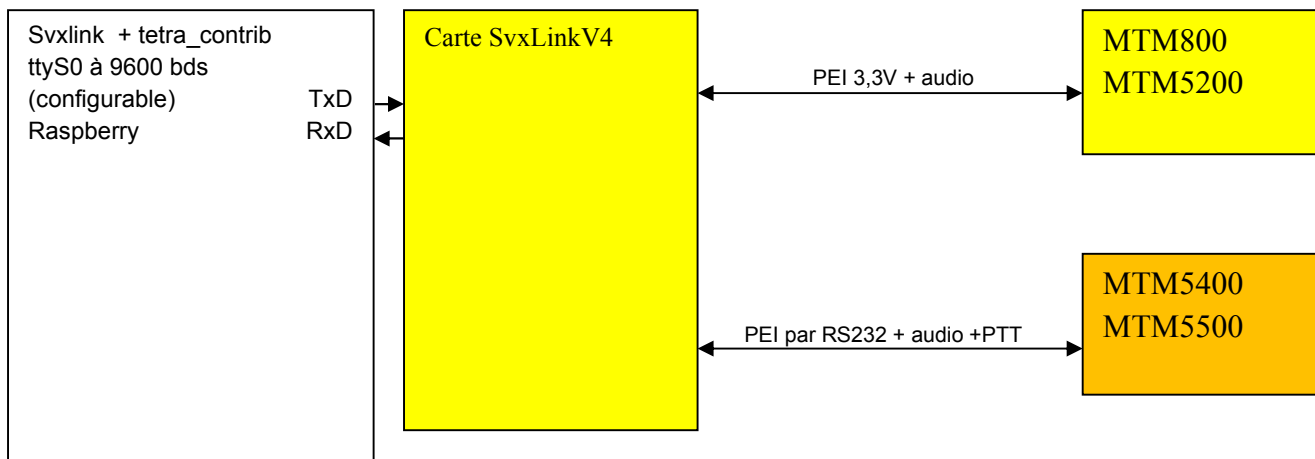
Téléchargement, installation et configuration de la branche TETRALOGIC pour SVXLINK

D'après F5UII et adapté par F1IWQ pour la Branche Tetra_contrib
V1.9 + librairies (avril 2021)

Table des matières

Schéma d'interfaçage de la branche Tetra_contrib.....	3
Première installation logicielle	3
Installation des voix françaises	4
Installation de la carte son USB.....	5
Configuration de SvxLink.....	6
Configuration de TetraLogic :	6
Configuration de ReflectorLogic	8
Echolink.conf	10
Configuration de l'audio	10
Traitements des évènements du TETRA (Tetralogic.tcl)	10
Traitement des évènements vers le GPIO	11
Paramètres du codeplug pour utiliser l'interface PEI	11
Configuration des GPIO	12
Script de libération de l'UART.....	13
Lancer Svxlink pour test.....	14
Lancement de SVXLINK au boot.....	16
Ouverture des ports du routeur en fonction des paramètres du réflecteur	17
Réglages avec alsamixer :.....	18
Menus à intégrer au menu principal du pi :	19
Module MetarInfo.....	19
Commandes DTMF	20
Mise à jour de svxlink/tetra.....	20
Diffuser le flux audio	21
Liste des erreurs	21
SERVEUR SVXLINK + TETRA GATEWAY	21
TETRA_GATEWAY.....	21
TETRA_SERVER software.....	22
Console SVXLINK.....	22
Méthode 1: par un socket.....	22

Schéma d'interfaçage de la branche Tetra_contrib



Première installation logicielle

Dans la configuration du pi, activer l'interface « Serial Port », et désactiver « serial console ». Désactiver le bluetooth (Pi3B+). Cela nécessite de redémarrer le pi.

Se mettre dans le répertoire `/home/pi` :

```
cd /home/pi
```

Clonage de la branche *tetra-contrib* pour disposer de la fonction TetraLogic qui interface le MTM via l'interface PEI :

```
sudo git clone -b tetra-contrib https://github.com/dllhrc/svxlink
```

Cela crée le répertoire `svxlink`.

Pour information, les instructions d'installation sont présentées dans le fichier `INSTALL.adoc`.

```
cd svxlink
```

Installer les paquets du compilateur et autres librairies (à installer une seule fois)

```
sudo apt-get install cmake libsigc++-2.0-dev libasound2-dev libpopt-dev libgcrypt20-dev  
tk-dev libgsm1-dev libspeex-dev libopus-dev groff librtlsdr-dev libqt4-dev libcurl4-  
openssl-dev doxygen libjsoncpp-dev
```

La ligne `sudo apt-get` doit être envoyée en une seule ligne

A la question (Y/n) taper entrée et créer un utilisateur `svxlink`, et l'intégrer au group `daemon`. Le mot de passe est au choix : (enter)

```

sudo adduser svxlink
sudo usermod -a -G daemon svxlink

Adding user `svxlink' ...
Adding new group `svxlink' (1001) ...
Adding new user `svxlink' (1001) with group `svxlink' ...
Creating home directory `/home/svxlink' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for svxlink
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y

```

Compiler le code source pour le Raspberry Pi. Avec les paramètres de la ligne `cmake`, `svxlink` sera installé dans le répertoire `/usr/bin` (paramètre `DCMAKE_INSTALL_PREFIX`). La compilation dure environ 15mn.

```

cd src
sudo mkdir build
cd build
sudo cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DUSE_OSS=NO -DUSE_QT=NO ..
sudo make
sudo make doc
sudo ldconfig
sudo make install

```

La ligne bleue `sudo cmake` doit être envoyée en une seule ligne. Notez qu'elle positionne la variable `DUSE_QT` à non, ce qui ne compile pas Qtel, qui est le client Echolink.

La compilation va créer les fichiers config dans `/etc/svxlink`. S'ils sont déjà existants, ils ne seront pas écrasés.

La commande suivante évite le message d'erreur `TclVoiceMail` à l'exécution de `svxlink`:

```
*** ERROR: The spool directory (/var/spool/svxlink/voice_mail) is not writable by the current user or does not exist."
```

Corriger avec la commande:

```
sudo chmod -R 777 /var/spool/svxlink/voice_mail
```

Installation des voix françaises

Pour donner voix françaises au relais, installer les fichiers sons (format WAV 16k). Choisir une des voix françaises disponibles sur le dépôt GitHub de F8ASB.

```
cd /usr/share/svxlink/sounds/
```

```
sudo git clone https://github.com/F8ASB/fr_FR_Agnes
ls
```

Le répertoire `fr_FR_Agnes` contient à présent tous les répertoires et fichiers wav permettant de donner voix au relais.

Installation de la carte son USB

Sur une configuration Raspberry Pi 3 et Jessie, il est nécessaire d'ajouter `dwc_otg.fiq_split_enable=0` en fin de ligne du fichier `/boot/cmdline.txt`

Ajouter la ligne `snd-usb-audio` dans le fichier `/etc/modules`.

Modifier `/lib/modprobe.d/aliases.conf` en changeant `snd-usb-audio` index de -2 à 0. Ajouter également en fin de fichier options `snd-usb-audio nrpacks=1`

Rebooter le pi.

Connectez la carte son USB sur votre Raspberry Pi. La LED s'allume. Vérifier que le périphérique USB est reconnu :

```
Lsusb

Bus 001 Device 004: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controller
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

La carte USB est ici le Device 004 CM108 Audio Controller.

Vérifier que l'entrée audio microphone est présente:

```
arecord -l
```

```
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Pour la voie audio d'enregistrement, repérer le numéro de carte (ici 'card 1') et le numéro de device (ici 'device 0'), car ils vont servir dans la configuration de SvxLink

Vérifier la voie de sortie audio (haut-parleur). Il apparaît également le périphérique de lecture qui équipe le Raspberry Pi (bcm2835)

```
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
```

Configuration de SvxLink

Toute la configuration (hors modules) se trouve dans le fichier `/etc/svxlink/svxlink.conf`.

Le fichier d'origine non configuré (brut) se trouve dans
`/home/pi/svxlink/src/svxlink/svxlink/svxlink.conf`

Configuration de TetraLogic :

TetraLogic est la logique utilisée par la branche `TETRA_CONTRIB` de SvxLink.

TetraLogic utilise par défaut la liaison `/dev/ttyUSB0` à 115200 bauds pour se connecter au MTM et renvoyer les informations vers svxlink.

Modifier dans la rubrique [TetraLogic]

`dev/ttyUSB0` par `/dev/ttyS0` sur un pi3 pour utiliser l'uart série du pi
pour un pi2, `/dev/ttyAMA0`

Changer Bauds par 9600, qui est la vitesse par défaut de l'interface PEI dans le codeplug.

Désactiver le Bluetooth.

AU démarrage, TetraLogic positionne le poste Tetra en mode DMO par la commande `AT+CTOM=1`. Il utilise la liaison PEI pour récupérer l'information de Squelch (COS) et le passage en émission (PTT).

Il dialogue également le GPS pour envoyer la position sur APRS.

Deux modes d'accès au Squelch/PTT sont possibles en fonction de la logique utilisée : `TETRALOGIC` pour utiliser le squelch et PTT via interface PEI ou traditionnellement `SimplexLogic` ou `RepeaterLogic` pour squelch et PTT via GPIO (mais il faut prélever le signal SQ dans le MTM800)

Pour utiliser un MTM800/MTM5x00 en mode DMO Direct, positionner `AT+CTOM=1`.

Pour utiliser un MTM5x00 en mode DMO relais, positionner `AT+CTOM=6`. Dans ce cas il faut utiliser le PTT câblé (sinon les transmissions du réseau ne seront pas transmises).

Configuration avec la branche TETRA :

[GLOBAL]

```
(...)  
LOGICS=TetraLogic,SimplexLogic  
LINKS=ALLlinks, avec ALLlinks qui lie les deux logiques  
(...)
```

[TetraLogic]

```
TYPE=Tetra  
RX=RxTetra  
TX=Tx1  
MODULES=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleTclVoiceMail  
CALLSIGN=MYCALL  
DEBUG=0  
DEFAULT_LANG=en_US  
BAUD=9600  
PORT=/dev/ttyUSB0  
ISSI=23401  
GSSI=1  
MNC=16383  
MCC=901  
# AT+CTOM=1 = mode DMO  
# AT+CTOM=6 = relais DMO  
# AT+CTSP=1,3,131 = activation du GPS vers PEI. Renvoie une erreur si pas de GPS connecté  
# AT+CTSP=1,3,130 = activation des SDS vers PEI  
INIT_PEI=AT+CTOM=6;AT+CTSP=1,3,131;AT+CTSP=1,3,130;AT+CTSP=1,3,138;AT+CTSP=1,2,20;AT+CTSP=2,0,0;AT+CTSP=1,3,2  
4;AT+CTSP=1,3,25;AT+CTSP=1,3,3;AT+CTSP=1,3,10;AT+CTSP=1,1,11;AT+CTSDC=0,0,0,1,1,0,1,1,0,0  
SHORT_IDENT_INTERVAL=60  
LONG_IDENT_INTERVAL=60  
#IDENT_ONLY_AFTER_TX=4  
EVENT_HANDLER=@SVX_SHARE_INSTALL_DIR@/events.tcl  
RGR_SOUND_DELAY=0  
#RGR_SOUND_ALWAYS=0  
MACROS=Macros  
FX_GAIN_NORMAL=0  
FX_GAIN_LOW=-12  
#APRSPATH=APRS,qAR,MYCALL-10:  
#ACTIVATE_MODULE_ON_LONG_CMD=4:EchoLink  
#QSO_RECORDER=8:QsoRecorder  
#ONLINE_CMD=998877  
#STATE_PTY=/var/run/svxlink/state  
#DTMF_CTRL_PTY=/dev/shm/simplex_dtmf_ctrl  
#CTCSS_TO_TG=77.0:999,123.0:9990,146.2:9992  
PROXIMITY_WARNING=3.1  
TIME_BETWEEN_SDS=3600  
INFO_SDS=Welcome new user  
TETRA_USERS=Tetra_Users  
STATUS=Tetra_Status  
SDS_ON_USERACTIVITY=SdsOnUserActivity  
SDS_TO_OTHERS_ON_ACTIVITY=DMO_ON,DMO_OFF,PROXIMITY  
SDS_TO_COMMAND=SdsToCommand  
END_CMD=ATH
```

[Tetra_Users]

```
# ISSI = Call,Name,comment  
90116383000023401=DM0SVX,DMO-Repeater,\\r,Halle/Saale  
90116383000023404=DL1HRC,Adi,\\E,SvxLink operator  
90116383000023428=DL1HQN,Martin,\\E,Groebers
```

[Tetra_Status]

```
# status=message  
#The section name that contain a list of status codes in decimal notation and  
#descriptions. Example:  
# 32768=$8000  
32768=at home  
32769=on the road
```

[SdsOnUserActivity]

```

0=Hello user you have powered on
1=Hello user you have powered off
2=Hello user you want to declare the state of emergency?
3=Hello user Push-to-talk condition is detected
4=Hello user Status
5=Hello user TXI=on

```

[SdsToCommand]

```

#The section name that contain a list of status codes in decimal notation and the
#according command that will be executed when the state sds has been received.
#The idea behind is the possibility to connect/disconntct links.
#32768=$8000
32768=721
32769=72

```

[RxTetra]

```

TYPE=Local
#RX_ID=?
# numéro de carte son
AUDIO_DEV=alsa:plughw:0
# numéro de device
AUDIO_CHANNEL=0
#AUDIO_DEV_KEEP_OPEN=0
SQL_DET=TETRA_SQL
SQL_START_DELAY=0
SQL_DELAY=0
(...)

```

Configuration de ReflectorLogic

[ReflectorLogic]

```

TYPE=Reflector
# Nom d'hôte ou Adresse ip
HOST=rlf-87.dyndns.org
PORT=5387
CALLSIGN="xxxxxxx"
AUTH_KEY="xxxxxxx"
JITTER_BUFFER_DELAY=2
EVENT_HANDLER=/usr/share/svxlink/events.tcl
DEFAULT_TG=20887
MONITOR_TGS=20887
TG_SELECT_TIMEOUT=4000000000
Non décrites :
MUTE_FIRST_TX_LOC
MUTE_FIRST_TX_REM
TMP_MONITOR_TIMEOUT
#ANNOUNCE_REMOTE_MIN_INTERVAL=300
NODE_INFO_FILE

```

DEFAULT_TG

Le nœud sélectionnera ce TG sur le trafic entrant local si aucun autre TG n'est actuellement sélectionné. Par défaut: 0 (pas de TG).

MONITOR_TGS

Liste de TGs séparés par des virgules dont le nœud surveillera l'activité lorsqu'aucun autre TG n'est sélectionné. Il est également possible de marquer les TG comme étant plus prioritaires que les autres en ajoutant un ou plusieurs signes + après le numéro du TG. Davantage de signes + signifient une priorité plus élevée. Lorsqu'un TG est sélectionné et qu'il y a une activité sur un autre TG avec une priorité plus élevée, le TG de priorité supérieure sera sélectionné à moins qu'il y ait eu une activité locale sur le nœud.

Exemple: MONITOR_TGS=112++, 240, 2403+, 2403123

Surveillera les TG pour la Suède, le district 3 de Suède et un TG spécifique 2403123. Le trafic sur TG 2403 sera prioritaire et le 112 aura la priorité la plus élevée.

TG_SELECT_TIMEOUT

Nombre de secondes après lequel un TG sélectionné sera désélectionné. Le nœud retournera au groupe de conversation 0 (pas de TG) et recommencera à surveiller les TG configurés. Par défaut: 30 secondes.

ANNOUNCE_REMOTE_MIN_INTERVAL

Nombre minimum de secondes entre les annonces du même TG pour les activations de TG à distance. Si le même TG est activé à distance à plusieurs reprises, il ne sera pas annoncé avant qu'au moins le nombre de secondes spécifié dans cette variable de configuration ne se soit écoulé.

NODE_INFO_FILE

Fichier de configuration à utiliser pour envoyer des informations concernant ce client au serveur réflecteur. Ce n'est pas une configuration obligatoire. Le fichier a principalement une structure JSON de forme libre, mais la structure générale doit être conservée afin que SvxFLink et le serveur réflecteur puissent remplir des informations dynamiques sur le nœud, comme la force du signal pour les récepteurs. Utilisez le node_info.json par défaut comme modèle. Vous pouvez ajouter plus d'informations assez librement mais ne modifiez pas la structure globale.

Il est également possible de définir les paramètres du codec audio en utilisant les mêmes variables de configuration que celles documentées pour les récepteurs et les émetteurs en réseau. Par exemple, pour alléger la charge CPU de l'encodeur pour l'encodeur Opus, définissez OPUS_ENC_COMPLEXITY à inférieur à 9 par exemple.

MUTE_FIRST_TX_LOC

Coupe le son de la première transmission après avoir sélectionné un TG en raison d'une activité locale. Il est recommandé d'avoir activé cette fonctionnalité pour un certain nombre de raisons. Une des raisons est de supprimer les ouvertures courtes d'un groupe de discussion lorsque quelqu'un effectue juste une seule transmission pour tester le nœud local. Une autre raison est de permettre à quelqu'un de soumettre des commandes DTMF au nœud sans perturber le réseau de réflecteurs. Un exemple est que quelqu'un active un TG à l'aide de CTCSS, mais sélectionne immédiatement un autre groupe de conversation à l'aide du DTMF. Dans ce cas, aucune transmission ne sera effectuée sur le premier groupe de conversation. Cette fonction est activée par défaut.

MUTE_FIRST_TX_REM

Mise en sourdine de la première transmission après avoir sélectionné un groupe de conversation en raison d'une activité à distance. Cette fonction peut être activée pour permettre aux utilisateurs de nœuds locaux d'entrer des commandes DTMF sans perturber un groupe de conversation actif. Par exemple, le nœud local surveille un groupe de conversation actif. Cependant, personne sur le nœud local ne participe au QSO et un utilisateur local souhaite sélectionner un autre groupe de conversation. Avec cette fonction activée, il est possible de faire cela sans transmettre vers le réseau de réflecteurs tout en entrant des commandes DTMF. Cette fonctionnalité n'est pas activée par défaut car elle n'est pas intuitive. Si un utilisateur local entend un appel et veut y répondre, il doit d'abord faire un bref PTT pour «ouvrir» le nœud local. C'est facile à oublier.

TMP_MONITOR_TIMEOUT

Cette variable de configuration détermine après combien de secondes un moniteur de TG temporaire ajouté manuellement expirera. Réglez sur 0 pour désactiver cette fonction. La valeur par défaut est 3600, une heure.

Echolink.conf

Vérifiez si la variable MUTE_LOGIC_LINKING est positionnée :

```
[ModuleEchoLink]
NAME=EchoLink
ID=2
#timeout de déconnexion si pas d'activité
#TIMEOUT=60
# si 0 alors echolink transmet vers le gateway sinon non!
MUTE_LOGIC_LINKING=0
```

Configuration de l'audio

Pour que le son passe correctement, il faut déconnecter le micro du MTM800.

Configurer le périphérique Audio à utiliser par Svxlink. Plus haut il a été identifié que l'enregistrement (donc la voie de réception Rx) est card 1, device 0. Il faut donc avoir ceci dans le fichier de configuration :

```
[RxXXX]
TYPE=Local
AUDIO_DEV=alsa:plughw:1
AUDIO_CHANNEL=0
```

Pour la partie émission, la voie de sortie audio (haut-parleur) est également card 1, device 0 donc dans le fichier de configuration :

```
[TxXXX]
TYPE=Local
AUDIO_DEV=alsa:plughw:1
AUDIO_CHANNEL=0
```

Pour améliorer la qualité de l'audio transmis par le relais, il est conseillé d'installer un égaliseur logiciel en suivant ce descriptif : <http://svxcard.f5uii.net/doku.php?id=fr:equalizer>

Traitements des évènements du TETRA (TetraLogic.tcl)

Si le fichier **TetraLogic.tcl** est absent dans /usr/share/svxlink/event.d

Copier le fichier /home/pi/svmlink/src/svmlink/svmlink/**TetraLogic.tcl**
Dans /usr/share/svmlink/event.d

Traitement des évènements vers le GPIO

Il est possible d'exporter les évènements vers l'interface GPIO. La syntaxe TCL doit être utilisée dans les fichiers TCL. Par exemple, le code ci-dessous montre comment exporter le signal squelch (qu'il vienne du GPIO ou de l'interface PEI).

Fichier /usr/share/svmlink/events.d/TetraLogic.tcl

```
(...)  
# Executed each time the squelch is opened or closed  
#  
proc squelch_open {rx_id is_open} {  
    Logic::squelch_open $rx_id $is_open;  
    puts "Le squelch change: $is_open";  
    if {$is_open==0} { exec echo 0 >/sys/class/gpio/gpio18/value; }  
    if {$is_open==1} { exec echo 1 >/sys/class/gpio/gpio18/value; }  
}
```

Ecrit 1 (3,3V) au GPIO18 quand le squelch s'ouvre. *Exec* est une commande TCL qui exécute un script. Vous pouvez utiliser n'importe quelle commande raspbian qui n'est pas autorisée par la syntaxe TCL. N'oubliez pas de déclarer l'utilisation de GPIO18 dans /etc/rc.local (voir ci-dessous).

Cela peut être utilisé pour piloter des leds en fonction de l'état de svmlink (exécution/arrêt), commander un PTT vers un émetteur etc...

Paramètres du codeplug pour utiliser l'interface PEI

Les paramètres de cette interface sont réglés dans le CPS dans **Data_services > AT commands** et cocher les 3 options ETSI group format , ETSI AT SDS/Status format et Extended ETSI adresssing.

Attention, suivant les versions de firmware ou de radio, les paramètres ci-dessus ne sont pas forcément existants. Dans ce cas, ils sont validés de façon interne, et il n'y a rien à faire. Ces paramètres sont existants pour les MTM5x00.

En mode LAB : **cp_ergo_block / ergo_data / ro / rui_parameters / limited_access_services**
Changer la valeur 7 (pei) de 0 à 1 pour activer l'interface pei

Pour tester la liaison série avec le MTM, installer le terminal minicom :
sudo apt-get install minicom

Lancer le terminal avec
minicom -b 9600 -o -D /dev/ttyS0

Pour quitter minicom, taper CTRL-A Q

Configuration des GPIO

Vous n'avez pas besoin d'utiliser le GPIO avec l'interface TETRA PEI (sauf en mode relais DMO pour le PTT). Pour l'utilisation de l'interface classique filaire, on utilise le GPIO du PI vers le poste local. Ouvrir pour édition le fichier /etc/rc.local et le modifier en fonction du câblage du PTT et du COS (Squelch). Exemple ici GPIO16=PTT=sortie=out ; GPIO19=SQL=entrée=in et actif à 1.

```
#!/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

#GPIO16=PTT
echo "16" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio16/direction

#GPIO19=SQL
echo "19" >/sys/class/gpio/export
sleep 2
echo "in" >/sys/class/gpio/gpio19/direction

#GPIO18=utilitaire
echo "18" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio18/direction

#GPIO23=bouton stop géré directement dans un script python

# vide le tampon du MTM en envoyant un CR
/home/pi/serial_write-CR.py
(...)

exit 0
```

Les E/S du GPIO peuvent être testés avec l'instruction :
gpio readall

Restrictions

1. Sur les MTM5x00 si on utilise le mode DMO Repeater, il est impossible d'utiliser l'interface PEI pour passer le poste en émission. En conséquence, une transmission du réseau ne sera pas transmise sur le MTM5x00. Il faut utiliser le PTT câblé via GPIO16 et modifier la section Rx en conséquence.
2. Sur les MTM en mode DMO direct, il n'est pas possible de recevoir un signal radio et d'émettre simultanément. Cette fonctionnalité est réservée aux MTM5x00 en mode DMO répéteur.

Script de libération de l'UART

Afin que TetraLogic communique correctement avec l'interface PEI à la première mise sous tension, il faut envoyer un retour chariot (CR) au poste pour en vider son tampon. Cette macro doit être lancée avant l'exécution de svxlink, dans un fichier appelé « serial_write_CR » dans /home/pi :

```
#!/usr/bin/env python
import serial

ser = serial.Serial(port='/dev/ttyS0', ← adresse de l'UART
                    baudrate = 9600, ← vitesse à adapter en fonction du PEI
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    bytesize=serial.EIGHTBITS,
                    timeout=1
)

ser.write(13)
ser.close
```

Rendre exécutable le script par
chmod 744 serial_write-CR.py

Ce script est à lancer au démarrage du pi, il faudra bien sûr que le poste MTM soit sous tension à la mise sous tension du pi :
le rajouter dans /etc/rc.local :

```
#!/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

#GPIO16=PTT
echo "16" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio16/direction

#GPIO19=SQL
echo "19" >/sys/class/gpio/export
sleep 2
echo "in" >/sys/class/gpio/gpio19/direction

# vide le tampon du MTM en envoyant un CR
/home/pi/serial_write-CR.py

exit 0
```

Lancer Svxlink pour test

Avant de lancer svxlink, libérer le tampon du MTM en lançant la commande

```
./serial_write-CR.py
```

Lancer svxlink en tapant la commande suivante:

```
svxlink
```

La led de la carte son USB devrait clignoter si elle en est équipée. Au lancement il est possible d'obtenir des erreurs du PEI.

Suivant la variable MUTE_FIRST_TX_LOC, le premier coup de PTT sélectionne le TG (décrit dans TETRALOGIC). Le 2ème coup de PTT lance le talker qui aiguille l'audio vers le TG sur le réseau.

Vous devriez voir ceci (avec TetraLogic, et DEBUG=1)

```
pi@raspberrypi:~ $ svxlink
SvxLink v1.7.99.24 Copyright (C) 2003-2020 Tobias Blomberg / SM0SVX

SvxLink comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it in accordance with the terms and conditions in the
GNU GPL (General Public License) version 2 or later.
Using configuration file: /etc/svxlink/svxlink.conf
--- Using sample rate 48000Hz
Starting logic: ReflectorLogic
ReflectorLogic: Connecting to rlf-87.dyndns.org:5387
Starting logic: TetraLogic
Loading RX: RxTetra
Loading TX: TxTetra
Loading module "ModuleHelp" into logic "TetraLogic"
Found /usr/lib/arm-linux-gnueabi/hf/svxlink/ModuleHelp.so
```

```

Module Help v1.0.0 starting...
Loading module "ModuleTclVoiceMail" into logic "TetraLogic"
Found /usr/lib/arm-linux-gnueabi/hf/svxlinc/ModuleTcl.so
Module Tcl v1.0.1 starting...
TetraLogic: Event handler script successfully loaded.
8200=121
8201=120
Creating tetra specific Sql ok
To PEI:
To PEI:AT+CTOM=1
Activating link ALLlinkTetra
ReflectorLogic: Connection established to 82.64.55.4:5387
From PEI:+CME ERROR: 35
35 - Syntax error. The syntax of the command is incorrect e.g. mandatory
parameters are missing or are exceeding Data received without command
From PEI:+CME ERROR: 35
35 - Syntax error. The syntax of the command is incorrect e.g. mandatory
parameters are missing or are exceeding Data received without command
ReflectorLogic: Authentication OK
ReflectorLogic: Connected nodes: (19)-F8KHC, (87)-F1ZJA, (59)-F1IWQ
----- Opus encoder parameters -----
Frame size          = 320
Complexity           = 9
Bitrate              = 20000
VBR                  = YES
Constrained VBR      = YES
Maximum audio bw     = MEDIUMBAND
Audio bw             = FULLBAND
Signal type          = VOICE
Application type      = AUDIO
Inband FEC           = NO
Expected Packet Loss = 0%
DTX                  = NO
LSB depth            = 16
-----
----- Opus decoder parameters -----
Gain                 = 0dB
-----
ReflectorLogic: Using audio codec "OPUS"
From PEI:+CTOM: 1
+++ New Tetra mode: 1 - DMO
From PEI:OK
Connected to APRS server 217.160.179.143 on port 14580
To PEI:AT+CTSP=1,3,131
From PEI:+CME ERROR: 3    <-- Erreur causée par l'absence de GPS sur la commande 131
3 - This is a general error report code which indicates that the MT supports
the command but not in its current state. This code shall be used when no
other code is more appropriate for the specific context
To PEI:AT+CTSP=1,3,130
From PEI:OK
To PEI:AT+CTSP=1,3,138
From PEI:OK
To PEI:AT+CTSP=1,2,20
From PEI:OK
To PEI:AT+CTSP=2,0,0
From PEI:OK
To PEI:AT+CTSP=1,3,24
From PEI:OK
To PEI:AT+CTSP=1,3,25
From PEI:OK
To PEI:AT+CTSP=1,3,3
From PEI:OK
To PEI:AT+CTSP=1,3,10
From PEI:OK
To PEI:AT+CTSP=1,1,11
From PEI:OK
To PEI:AT+CTSDC=0,0,0,1,1,0,1,1,0,0
From PEI:OK
To PEI:AT+CNUMF?
From PEI:+CNUMF: 0,9010000102089144

```

```

<num type> is 0 (0 - Individual (ISSI or ITSI))
From PEI:2,16777184
From PEI:3,16777184
From PEI:OK
From PEI:+CTICN: 1,0,0,,,1,1,0,1,1,1,9011638300000001,0
*** No valid +CTICN response, message to short
From PEI:+CTCC: 1,1,1,0,0,1,1
From PEI:+CTXG: 1,3,0,0
TetraLogic: The squelch is OPEN
RxTetra: The squelch is OPEN (5.89658)
TetraLogic: The squelch is OPEN
ReflectorLogic: Selecting TG #20887 ←-- sélection du TG sur le 1er coup de PTT
From PEI:+CTICN: 1,0,0,1,7530236102089144,1,1,0,1,1,1,9011638300000001,0
*** No valid +CTICN response, message to short
From PEI:+CDTXC: 1,0
TetraLogic: The squelch is CLOSED
RxTetra: The squelch is CLOSED (5.89336)
TetraLogic: The squelch is CLOSED
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
From PEI:+CTXG: 1,3,0,0,1,7530236102089144
TetraLogic: The squelch is OPEN
RxTetra: The squelch is OPEN (5.955)
TetraLogic: The squelch is OPEN
ReflectorLogic: Talker start on TG #20887: (59)-F1IWQ ←-- 2eme coup de PTT
From PEI:+CDTXC: 1,0
TetraLogic: The squelch is CLOSED
ReflectorLogic: Talker stop on TG #20887: (59)-F1IWQ
RxTetra: The squelch is CLOSED (5.67421)
TetraLogic: The squelch is CLOSED
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
RxTetra: Distorsion detected! Please lower the input volume!
From PEI:+CTCR: 1,13
APRS,qAR,F1IWQ-10:Transmission ended
  To PEI:AT
From PEI:OK
  To PEI:AT
From PEI:OK
  To PEI:AT

```

Lancement de SVXLINK au boot

Il faut créer un service qui démarre svxlink après que le réseau soit disponible et la carte son (svxlink.service).

Un autre service sera créé pour démarrer svxlink 60s après le boot (svxlink.timer)

Dans le répertoire /lib/systemd/system

Créer le fichier svxlink.service qui contient :

```

[Unit]
Description=démarre SVXlink
After=syslog.target network-online.target sound.target

```



```
[Service]
User=root
ExecStart=sudo /usr/bin/svxlink --config=/etc/svxlink/svxlink.conf --
logfile=/var/log/svxlink
```

```
[Install]
WantedBy=multi-user.target
```

Et créer un fichier `svxlink.timer` qui contient

```
[Timer]
OnStartupSec=60
```

```
[Install]
WantedBy=multi-user.target
```

Recharger les nouveaux services services pour qu'ils soient pris en compte dans le système:

```
sudo systemctl daemon-reload
```

Démarrer le service timer avec

```
sudo systemctl start svxlink.timer : démarre le service timer
sudo systemctl enable svxlink.timer : valide le service timer à chaque boot
```

Ne pas démarrer le service `svxlink.service`, c'est `svxlink timer` qui le démarre.

Vérifier si les services ont démarré avec

```
sudo systemctl status svxlink.timer
(ctrl c pour quitter)
```

Le service est démarré pour toujours, au prochain démarrage du pi, `svxlink` démarrera après que le réseau soit disponible, et après 60s.

Le fichier de log de `svxlink` se trouve dans `/var/log` fichier `svxlink`

Pour arrêter `SVXLINK`, taper `sudo systemctl stop svxlink.service`
Ce qui arrête le service.

Ouverture des ports du routeur en fonction des paramètres du réflecteur

Pour Echolink :

```
5198-5199 udp
5200 tcp
```

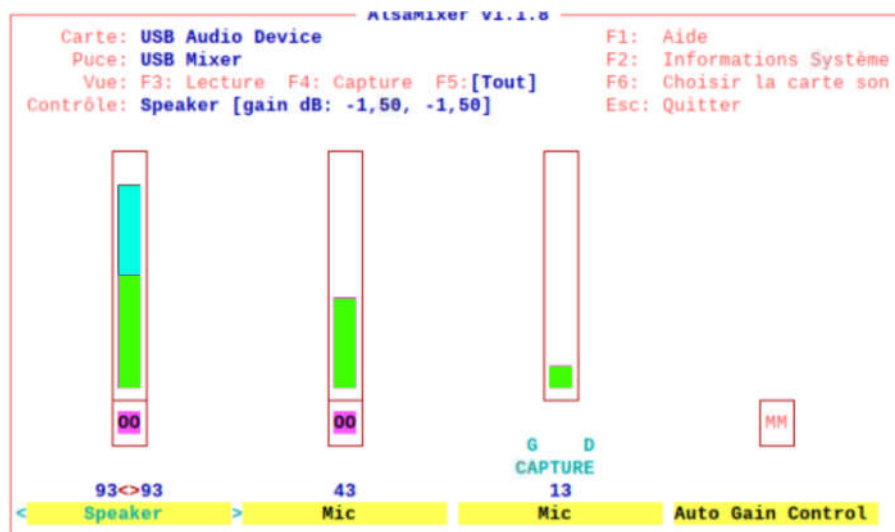
Pour le RLF :

5387 udp/tcp

Pour Leipzig:

5300 udp/tcp

Réglages avec alsamixer :



Faire F5 pour afficher tout.

Speaker permet de régler la BF venant du réseau vers la radio. (50%)

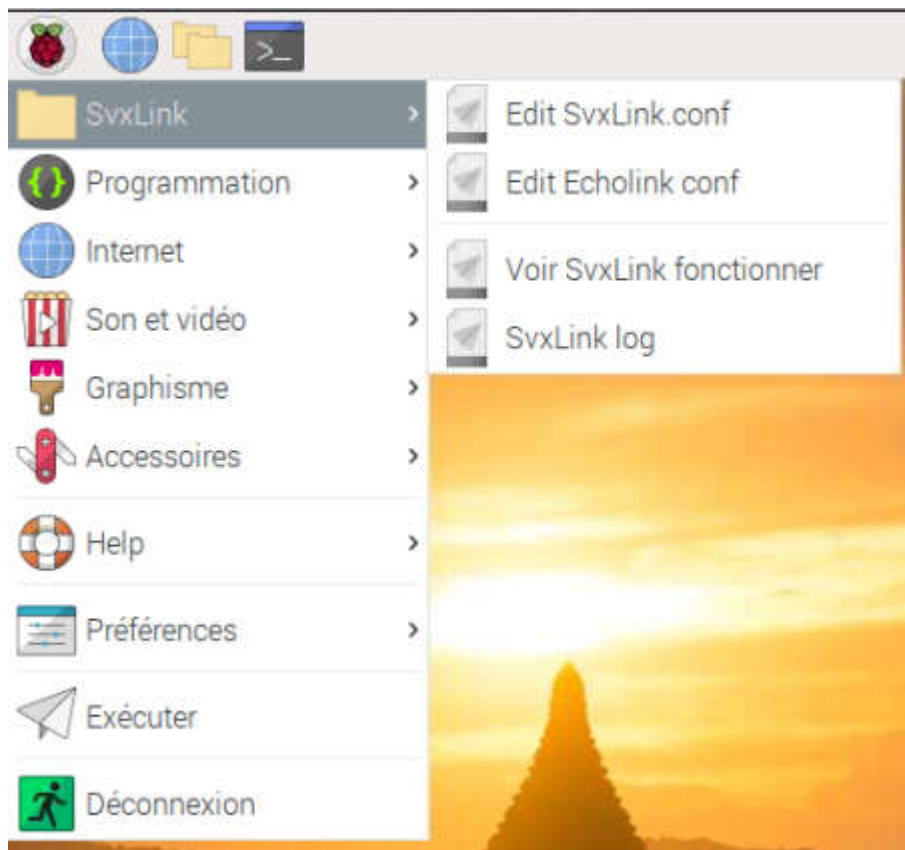
Capture permet de régler la BF de la radio vers le réseau. (13% vers Leipzig)

Mic n'a aucun effet.

Noter que Auto Gain control est muté (MM) taper sur m.

RV1 est à 80% au delà des saturations apparaissent

Menus à intégrer au menu principal du pi :



Edit SvxLink.conf permet de lancer un éditeur pour modifier le fichier de configuration svxlink.conf

Edit EchoLink.conf permet de lancer un éditeur pour modifier le fichier de configuration EchoLink.conf

Voir SvxLink fonctionner : affiche la fenêtre des dernières 40 lignes du fichier log de svxlink

SvxLink.log permet de lancer un éditeur pour voir le fichier de log svxlink.log

Module MetarInfo

/etc/svxlink/svxlink.d/ModuleMetarInfo.conf

/usr/share/svxlink/events.d/locale.tcl (lignes 37-38 pour le @.wav, lignes 177-187 et 197 pour modifier l'heure au format 24 h "The time is 15 41" sans AM et PM)

/usr/share/svxlink/events.d/Logic.tcl (lignes 241-242, démarrer le script /home/svxlink/wetter.pl)

/usr/share/svxlink/events.d/Module.tcl (ligne 26, pour émettre un message "aide.wav" dès que le ModuleMetarInfo est activé.)

Commandes DTMF

Les commandes DTMF ne sont pas possibles depuis un poste TETRA.

*Présentation du relais

#déconnexion

0# activation du module aide

1# activation du parrot

2# active le module ECHOLINK

5# active le module metarInfo (météo)

01# Liste des stations météo disponibles

Mise à jour de svxlink/tetra

Se mettre dans /home/pi

Taper

```
sudo ./maj.sh
```

fichier maj.sh:

```
#!/bin/bash

cd /home/pi/svxlink
git fetch

LOCAL=$(git rev-parse @)
REMOTE=$(git rev-parse @{u})
BASE=$(git merge-base @ @{u})

if [ $LOCAL = $REMOTE ]; then
    echo "[UPDATE]$(date) Svxlk est à jour"
elif [ $LOCAL = $BASE ]; then
    echo "[UPDATE]$(date) ===== arrêt de Svxlk ====="
    sudo pkill -f svxlink
    sleep 1
    echo "[UPDATE]$(date) ===== mise à jour de Svxlk depuis Repository ====="
    git pull
    cd src
    cd build
    sudo cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -DLOCAL_STATE_DIR=/var
-DUSE_OSS=NO -DUSE_QT=NO ..
    sudo make
    sudo make doc
    sudo ldconfig
    sudo make install
    sleep 2
    sudo chmod -R 777 /var/spool/svxlink/voice_mail
    echo "[UPDATE]$(date) ===== Mise à jour réussie de Svxlk ====="
```

```
else
    echo "[UPDATE][$(date)] !!!!! Erreur de mise à jour de Svmlink !!!!!"
fi
```

Diffuser le flux audio

http://svxcard.f5uii.net/doku.php?id=fr:streaming_installation

Liste des erreurs

Avec Echolink :

Bind : cannot assign requested address

Mauvaise adresse ip dans le module Echolink.conf à la variable BIND_AADR

SERVEUR SVXLINK + TETRA GATEWAY

Cette partie décrit la partie serveur de svxlink.

Cela nécessite :

Raspbian

Svmlink

Tetra_gateway

Tetra server software

TETRA_GATEWAY

TetraGateway utilise la liaison série du pi pour se connecter au MTM et renvoyer les informations vers le gateway..

```
cd /opt
sudo mkdir tetra_gw
cd tetra_gw
    Copiez le fichier tetra_link_gw
    Copiez le fichier tetra_link_gw.conf
sudo chmod 755 tetra_link_gw
```

Ces fichiers se trouvent ici :

<http://srv08.oevsv.at/tetrawiki/doku.php?id=start:tetraaustria:software:downloads>

Modifier ensuite le fichier tetra_link_gw.conf :

```

[GATEWAY]
GATEWAY_CALL=indicatif du relais-L ou R
GATEWAY_COMMENT=Ville/430.4125
GATEWAY_ID=208XXX
GATEWAY_LAT=4816.82
GATEWAY_LON=01624.91
# GATEWAY =NO
# DMO-REPEATER=YES
DMO_REPEATER=YES
[SVXREFLECTOR]
# UDP svxreflector connection via INTERNET
REFLECTOR_DNS=tetra-reflector.dyndns.org
# UDP svxreflector connection via HAMNET WIENERBERG
# REFLECTOR_DNS=44.143.8.74
# LASTHEARD PORT
REFLECTOR_PORT=5220
[WEATHER]
CITY=Wien
ZIP=1210
COUNTRY=AT
[SYSTEM]
# MTM5400 connexion to RASPI
SYS_SERPORT=/dev/ttyS0
SYS_SERBAUD=9600
SYS_DEBUG=0

```

Lancer tetra gateway avec

```
./tetra_link_gw
```

TETRA_SERVER software

Pas d'information

Console SVXLINK

La console permet d'activer les macros de la section [Macros] de svxlink.conf.#

Il existe plusieurs façons de simuler les commandes DTMF sur SVXLINK

Méthode 1: par un socket

Lancer svxlink pour qu'il soit télécommandable depuis un client distant :

```
nc -lk port |sudo svxlink
```

et depuis le client taper :

```
echo -n "commande" |nc adresse_ip port
```

adresse_ip = 127.0.0.1 si le client est sur la même carte réseau que le serveur

port = numéro du port de communication
commande = commande à envoyer à svxlink (exemple : *0#)

Méthode 2: simulateur de dtmf

Dans TetraLogic :

```
DTMF_CTRL_PTY=/tmp/dtmf
```

Définit un dossier d'échange des commandes DTMF (exemple : /tmp/dtmf)

Et taper dans une fenêtre console:

```
sudo echo "commande" >/tmp/dtmf
```

Et taper les commandes à envoyer.