

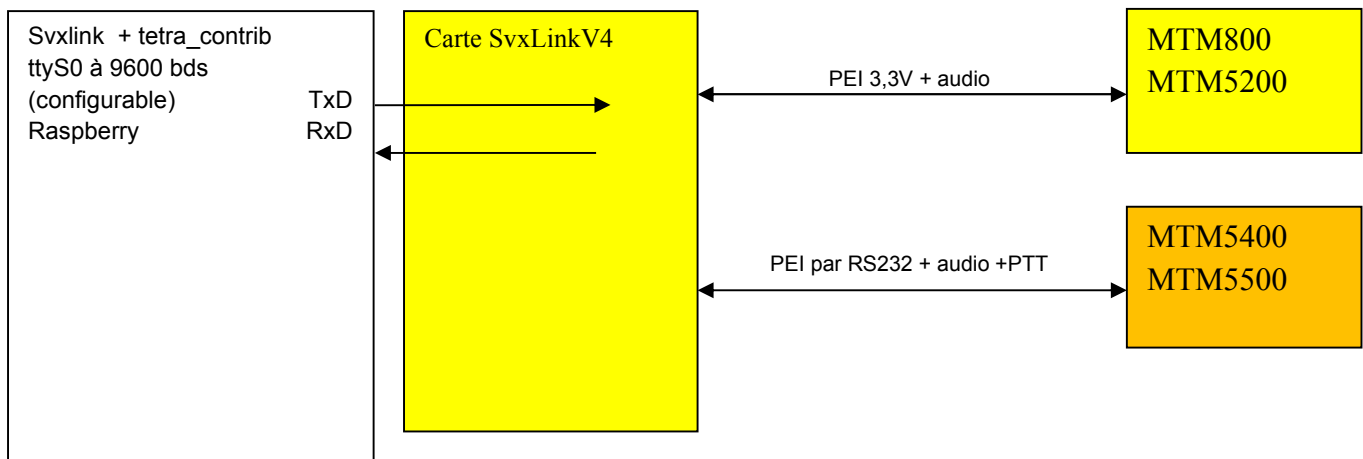
Download, install and configure the TETRALOGIC branch for SVXLINK

Based on F5UII and adapted by F1IWQ for the Tetra_contrib Branch
+ libraries (April 2025)

TABLE

Tetra_contrib intercoms	2
First software installation.....	2
French voices installation	4
Installing the USB sound card	4
SVXLINK configuration	6
TetraLogic configuration	6
Tetralogic.conf file configuration	7
ReflectorLogic configuration.....	9
Echolink.conf.....	10
Audio setup	10
TETRA events (Tetralogic.tcl)	11
Handling events to GPIO	11
Codeplug parameters to use the PEI interface	11
GPIO configuration.....	12
Restriction	13
UART release script.....	13
Run Svmlink for testing	14
SVXLINK launch at boot	17
Open the router ports according to the reflector settings:.....	18
IPV6 only networks	18
ALSAMIXER SETTINGS	19
Menu bar	20
DTMF controls.....	20
Update svmlink/tetra	21
SVXLINK Console.....	21
Using a socket.....	22
DTMF simulator	22

Tetra_contrib intercoms



First software installation

You can use the 32 bits or 64 bits version of Raspbian.

In the pi configuration, activate the "Serial Port" interface, and desactivate "serial console". Disable bluetooth (for pi3B+). This requires restarting the pi

Go to the /home/pi directory :

```
cd /home/pi
```

Clone the tetra-contrib branch to get the TetraLogic function which interfaces the MTM via the PEI interface:

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

This creates the svxlink directory.

For information, installation instructions are provided in the INSTALL.adoc file.

Give permission to the svxlink directory:

```
sudo chmod 777 -R /home/pi/svxlink
git config -global -add safe.directory /home/pi/svxlink
```

(this last command tells git commands that svxlink directory is safe, otherwise you won't be able to update).

```
cd svxlink
```

Install the compiler packages and other libraries (to be installed only once) :

```
sudo apt-get -yq install gcc g++ make cmake libgcrypt-dev libgsm1-dev libsigc++-2.0-dev
tcl-dev libspeex-dev libasound2-dev libpopt-dev libssl-dev libopus-dev groff libcurl4-
openssl-dev git mc libjsoncpp-dev doxygen gpiod libasound2-dev
```

This *sudo apt-get* line must be sent as a single line.

Install pjproject :

```
cd .. (go back to /home/pi)
wget https://github.com/pjsip/pjproject/archive/refs/tags/2.12.1.tar.gz
mv 2.12.1.tar.gz pjProject-2.12.1.tar.gz
tar -zxvf pjProject-2.12.1.tar.gz
cd pjproject-2.12.1
./configure --disable-video --disable-libwebrtc
make dep
make
sudo make install
cd /home/pi/svxlink
```

For next commands :

To the question (Y / n) type enter and create a svxlink user, and integrate it into the group daemon. The password is your choice: (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
```

Compile the source code for the Raspberry Pi. With the parameters in the cmake line, svxlink will be installed in the /usr/bin directory (parameter DCMMAKE_INSTALL_PREFIX). The compilation takes about 15 mn.

```
cd src
sudo mkdir build
cd build
sudo cmake -DUSE_QT=OFF -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DCMAKE_BUILD_TYPE=Release -DWITH_CONTRIB_TETRA_LOGIC=ON ..
sudo make
sudo make doc
sudo ldconfig
```

```
sudo make install
```

The sudo cmake blue line should be sent as a single line. Note that it sets the DUSE_QT variable to no, which does not compile Qtel, which is the Echolink client.

The compilation will create the config files in /etc/svxlink. If they are already existing, they will not be overwritten. The tetralogic.conf file is located in /etc/svxlink/svxlink.d

The following command avoids the TclVoiceMail error message when running svxlink:

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

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

French voices installation

To give French voices to the repeater, install the sound files (16k WAV format). Choose one of the French voices available on the F8ASB GitHub repository.

```
cd /usr/share/svxlink/sounds/  
sudo wget https://github.com/F8ASB/fr_FR_Agnes/archive/fr_FR_Agnes.zip  
sudo unzip fr_FR_Agnes.zip  
ls
```

The fr_FR_Agnes directory now contains all the directories and wav files used to give voice to the relay. Rename the directory to fr_FR (This is the name that must be configured in the svxlink.conf file in the default language)

```
sudo mv fr_FR_Agnes-master fr_FR
```

Installing the USB sound card

On a Raspberry Pi 3 and Jessie configuration, it is necessary to add dwc_otg.fiq_split_enable = 0 at the end of the line of the /boot/cmdline.txt file (long line)



```
cmdline.txt ✕  
1 k.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles dwc_otg.fiq_split_enable=0  
2
```

Add the line snd-usb-audio in /etc/modules file.

Modify /lib/modprobe.d/aliases.conf and change. snd-usb-audio index from -2 to 0. Add at the end of the file options snd-usb-audio nrpacks=1

Reboot the pi.

Connect the USB sound card to your Raspberry Pi. The LED will light up. Check that the USB device is recognized:

```
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
```

The USB card here is Device 004 CM108 Audio Controller.
Check if the microphone audio input is present:

```
arecord -l          (Letter l not 1)
```

```
**** 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
```

For the audio recording channel, locate the card number (here 'card 1') and the device number (here 'device 0'), because they will be used in the configuration of SvxLink
Check the audio output channel (speaker). It also appears the reading device that equips the Raspberry Pi (bcm2835)

```
aplay -l          (Letter l not 1)
```

```
**** 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
```

SVXLINK configuration

All the configuration (except modules) is in the file `/etc/svxlink/svxlink.conf`.

The original unconfigured (raw) file is located in

`/home/pi/svxlink/src/svxlink/svxlink/svxlink.conf`

TetraLogic configuration

Tetralongic.conf is the config file of the tetra configuration since 2022 oct. It is located in `/etc/svxlink/svxlink.d`

TetraLogic is the logic used by the TETRA_CONTRIB branch of SvxLink.

TetraLogic uses the 115200 baud `/dev/ttyUSB0` link by default to connect to the MTM and send information back to svxlink. This has to be changed if you want to use the pi internal UART.

Edit in the [TetraLogic] section

Change `dev/ttyUSB0` by `/dev/ttyS0` on a pi3

And for a pi2, `/dev/ttyAM0`

Change Bauds to 9600, which is the default speed of the PEI interface in the codeplug.

Turn off Bluetooth.

At start, TetraLogic sets the Tetra station in DMO mode using `AT+CTOM=1` command. It use the PEI link to retrieve information from Squelch (COS) and send PTT.

It also fetch the GPS to send the position to APRS. If the GPS board is not in the radio, you will get an error on the specific AT command if the DEBUG is set to 3.

Two access modes to squelch / PTT are possible depending on the logic used: TETRALOGIC to use squelch and PTT via software PEI interface or traditionally via hardware : SimplexLogic or RepeaterLogic for squelch and PTT via GPIO (but the SQ signal must be taken from the MTM800)

The operating mode is set in `TETRA_MODE` in the TetraLogic.conf file. It is used as an entry point to parse the file `/etc/svxlink/pei-init.json`

<code>TETRA_MODE=TMO</code>	Enables mode TMO (<code>AT+CTOM=0</code>)
<code>TETRA_MODE=DMO-RPT</code>	Enables mode Dmo repeater (<code>AT+CTOM=6</code>)
<code>TETRA_MODE=DMO-MS</code>	Enables mode TMO (<code>AT+CTOM=1</code>)
<code>TETRA_MODE=GATEWAY</code>	Enables mode Gateway (<code>AT+CTOM=5</code>)

To use the MTM800/MTM5x00 in Direct DMO mode, set DMO-MS.

To use the MTM5x00 in DMO Repeater, set DMO-RPT. In this case you must use the wired PTT (otherwise broadcasts from network won't be transmitted).

TETRA branch configuration in svxlink.conf

```
[GLOBAL]
LOGICS=ReflectorLogic,TetraLogic
CFG_DIR=svxlink.d
TIMESTAMP_FORMAT="%c"
CARD_SAMPLE_RATE=48000
```

```

CARD_CHANNELS=1
LOCATION_INFO=LocationInfo
# Pour RLF en TETRAlogic
LINKS=ALLlinkTetra

[SimplexLogic]
TYPE=Simplex
RX=Rx1
TX=Tx1
#MODULES=ModuleHelp,ModuleParrot,ModuleMetarInfo
MODULES=ModuleEchoLink
CALLSIGN=xxxxxx
SHORT_IDENT_INTERVAL=60
LONG_IDENT_INTERVAL=60
#IDENT_ONLY_AFTER_TX=4
#EXEC_CMD_ON_SQL_CLOSE=500
EVENT_HANDLER=/usr/share/svxlink/events.tcl
DEFAULT_LANG=fr_FR_Agnes

[ReflectorLogic]
TYPE=Reflector
HOSTS=your tetra host
HOST_PORT=your port
CALLSIGN=your call
AUTH_KEY=your key
DEFAULT_TG=2080
MONITOR_TGS=2080,208,9

[ALLlinkTetra]
CONNECT_LOGICS=TetraLogic:9:NET,ReflectorLogic
DEFAULT_ACTIVE=1
TIMEOUT=300
ACTIVATE_ON_ACTIVITY=TetraLogic

```

TetraLogic.conf file configuration

```

[TetraLogic]
DTMF_CTRL_PTY=/tmp/dtmf
TYPE=Tetra
RX=RxTetra
TX=TxTetra
MODULES=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleTclVoiceMail
#MODULES=ModuleHelp,ModuleTclVoiceMail
CALLSIGN=xxxxxx
DEFAULT_LANG=fr_FR_Agnes
BAUD=9600
#0=pas de debug 1=LOGWARNING 2=LOGINFO 3=LOGDEBUG
DEBUG=0
PORT=/dev/ttyS0
ISSI=123456
GSSI=1
MNC=1995
MCC=901
TETRA_MODE=DMO-RPT
PEI_INIT_FILE=/etc/svxlink/pei-init.json
#balise courte toutes les xx mn doit être un multiple de LONG_IDENT_INTERVAL
SHORT_IDENT_INTERVAL=0
#balise longue toutes les xx mn
LONG_IDENT_INTERVAL=60
TIME_FORMAT=24
#IDENT_ONLY_AFTER_TX=4
EVENT_HANDLER=/usr/share/svxlink/events.tcl
RGR_SOUND_DELAY=0
#RGR_SOUND_ALWAYS=0
MACROS=Macros
FX_GAIN_NORMAL=0
FX_GAIN_LOW=-12

```

```

PROXIMITY_WARNING=3.1
TIME_BETWEEN_SDS=3600
INFO_SDS=Welcome new user
TETRA_USER_INFOFILE=/etc/svxlink/tetra_users.json
TETRA_STATUS=Tetra_Status
SDS_ON_USERACTIVITY=SdsOnUserActivity
SDS_TO_OTHERS_ON_ACTIVITY=DMO_ON,DMO_OFF,PROXIMITY
SDS_TO_COMMAND=SdsToCommand
END_CMD=ATH
#SHARE_USERINFO=0
DAPNET_SERVER=dapnet.afu.rwth-aachen.de
DAPNET_PORT=43434
DAPNET_CALLSIGN=xxxxxxx
DAPNET_KEY=xxxxxxxxxxx
DAPNET_RIC2ISSI=Ric2ISSI
DAPNET_RUBRIC_REGISTRATION=RicRegistration

[Ric2ISSI]
# RIC=ISSI
#1234=23401
2089144=2089144

[RicRegistration]
# RIC=Rubric1,Rubric2,Rubric54 (in decimal)
1234=1024,1051
23451=1028,1051
2089144=1024

[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
61004=Parrot:
61005=Parrot:#
61010=MetarInfo:1

[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
61000=9*
33009=919
33391=9191
33040=91204
33080=91208
33140=91214
33220=91222
33260=91226
33280=91228
33320=91232
33350=91235
33620=91262
33629=912629
40990=919990

```


ReflectorLogic configuration

```
[ReflectorLogic]
TYPE=Reflector
# host or ip address
HOST=rlf-87.dyndns.org
PORT=5387
CALLSIGN="xxxxxx"
AUTH_KEY="xxxxxxx"
JITTER_BUFFER_DELAY=2
EVENT_HANDLER=/usr/share/svxlink/events.tcl
DEFAULT_TG=20887
MONITOR_TGS=20887
TG_SELECT_TIMEOUT=4000000000
```

DEFAULT_TG

The node will select this TG on local inbound traffic if no other TG is currently selected. Default: 0 (no TG).

MONITOR_TGS

Comma separated list of TGs whose node will monitor activity when no other TG is selected.

It is also possible to mark TGs as being higher priority than others by adding one or more + signs after the TG number. More + signs mean higher priority. When a TG is selected and there is activity on another TG with a higher priority, the higher priority TG will be selected unless there has been local activity on the node.

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

Will monitor TGs for Sweden, District 3 of Sweden and a specific TG 2403123. Traffic on TG 2403 will be given priority and 112 will have the highest priority.

TG_SELECT_TIMEOUT

Number of seconds after which a selected TG will be deselected. The node will return to talkgroup 0 (no TG) and resume monitoring configured TGs. Default: 30 seconds.

ANNOUNCE_REMOTE_MIN_INTERVAL

Minimum number of seconds between announcements of the same TG for remote TG activations. If the same TG is activated remotely multiple times, it will not be announced until at least the number of seconds specified in this configuration variable has elapsed.

NODE_INFO_FILE

Configuration file to use to send information about this client to the reflector server. This is not a mandatory configuration. The file has mostly a free form JSON structure, but the general structure should be kept so that SvxLink and the reflector server can populate dynamic information about the node, like signal strength for receivers. Use the default node_info.json as a template. You can add more information quite freely, but don't change the overall structure.

It is also possible to set the audio codec parameters using the same configuration variables as documented for networked receivers and transmitters. For example, to lighten the CPU load on the encoder for the Opus encoder, set OPUS_ENC_COMPLEXITY to less than 9 for example.

MUTE_FIRST_TX_LOC

Mutes the sound of the first transmission after selecting a TG due to local activity. It is recommended that you have this feature enabled for a number of reasons. One reason is to suppress short newsgroup openings when someone is doing just one push to test the local node. Another reason is to allow someone to submit DTMF commands to the node without disturbing the reflector array. An example is that someone activates a TG using CTCSS, but immediately selects another talkgroup using DTMF. In this case, no transmission will be made on the first talkgroup. This feature is enabled by default.

MUTE_FIRST_TX_REM

Mute the first transmission after selecting a talkgroup due to remote activity. This feature can be enabled to allow local node users to enter DTMF commands without disrupting an active talkgroup. For example, the local node monitors an active talkgroup. However, no one on the local node is participating in the QSO, and a local user wants to select another talkgroup. With this feature enabled, it is possible to do this without transmitting to the reflector array while entering DTMF commands. This feature is not enabled by default because it is not intuitive. If a local user hears a call and wants to answer it, he must first do a short PTT to “open” the local node. It's easy to forget.

TMP_MONITOR_TIMEOUT

This configuration variable determines after how many seconds a manually added temporary TG monitor will expire. Set to 0 to disable this function. The default is 3600, one hour.

Echolink.conf

Be careful to check if the MUTE_LOGIC_LINKING variable is set to 0 :

```
[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
```

Audio setup

Configure the Audio device to be used by Svxlink. Above it was identified that the recording (therefore the receiving channel Rx) is card 1, device 0. You must therefore have this in the configuration file:

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

For the transmission part, the audio output channel (speaker) is also card 1, device 0 therefore in the configuration file:

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

TETRA events (TetraLogic.tcl)

If the TetraLogic.tcl file is missing in /usr/share/svxlink/event.d

Copy the file /home/pi/svxlink/src/svxlink/svxlink/**TetraLogic.tcl**
to /usr/share/svxlink/event.d

Handling events to GPIO

It is possible to output event states to the GPIO interface. TCL syntax must be used in the TCL files. For instance, the code shows below how to export the squelch signal (whatever it comes from the GPIO or the PEI interface). Beware : for recent raspbian versions, GPIO range pins number have to be shifted to 512. For instance to use GPIO18, use $(512+18) = \text{GPIO530}$.

File /usr/share/svxlink/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/gpio530/value; }  
    if {$is_open==1} { exec echo 1 >/sys/class/gpio/gpio530/value; }  
}
```

Write 1 (3,3V) to the GPIO18 (530) when squelch opens. *Exec* command is the TCL script execution command. You can then use any raspbian command which is not supported by TCL syntax. Don't forget to declare the use of GPIO530 in /etc/rc.local (see below).

It can be used to drive leds according to svxlink state (running/idle), drive a PTT keyer for another transceiver etc...

Codeplug parameters to use the PEI interface

The parameters of this interface are set in the CPS in Data_services> AT commands and check the 3 options ETSI group format, ETSI AT SDS / Status format and Extended ETSI addressing.

Please note, depending on the firmware version or radio, the above parameters may not be available. In this case, they are validated internally, and there is nothing to be done.

In LAB mode: cp_ergo_block / ergo_data / ro / rui_parameters / limited_access_services
Change the value 7 (pei) from 0 to 1 to activate the pei interface

To test the serial link with the MTM, install the minicom terminal:

```
sudo apt-get install minicom
```

Start the terminal with

```
minicom -b 9600 -o -D /dev/ttyS0
```

To quit minicom, type CTRL-A Q

GPIO configuration

You don't have to use GPIO configuration with tetra PEI interface (except for DMO repeater mode for PTT). For the use of the classic wired interface, the GPIO from the PI to the local station is used. Open the `/etc/rc.local` file for editing and change it according to the wiring of the PTT and COS (Squelch). Example here GPIO16 = PTT = output = out; GPIO19 = SQL = input = active when 1.

For new raspbian versions (>2024) the `/etc/rc` file doesn't exist anymore. Just create it and give rights :

```
sudo chmod 755 /etc/rc.local
```

Again for raspbian version of you get :

```
bash: echo: write error: Invalid argument
```

Don't forget that GPIO range has changed and you have to use a +512 offset.

<https://arestless.rest/blog/lab--raspberry-pi-4b-gpio-debugging/>

```
sudo chmod ugo+rw /sys/class/gpio
```

Please also change PTT_PIN in `svxlink.conf` file if you use a wired ptt and add 512.

```
#!/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+512=PTT
echo "528" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio528/direction

#GPIO19+512=SQL
```

```

echo "531" >/sys/class/gpio/export
sleep 2
echo "in" >/sys/class/gpio/gpio531/direction

#GPIO18=utilitaire
echo "530" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio530/direction

#GPIO23=stop button directly handling in python script

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

exit 0

```

GPIO I/O can be tested with the instruction:

```
gpio readall
```

Restriction

With MTM5x00 devices, if the DMO Repeater mode is used, it is not possible to use the PEI interface to key the transmitter on ; a broadcast from the network will not be transmitted by the MTM5x00 on air. You **must** use the wired PTT via GPIO16 and modify the Tx section accordingly.

UART release script

In order for TetraLogic to communicate correctly with the PEI interface the first time it is powered on, a carriage return (CR) must be sent to the station to empty the buffer. This macro must be launched before running svxlink, in a file called "serial_write_CR" in /home/pi:

```

#!/usr/bin/env python
import serial

ser = serial.Serial(port='/dev/ttyS0',    ← adresse de l'UART for pi3
                    baudrate=9600,        ← speed baudrate à adapter en fonction du PEI
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    bytesize=serial.EIGHTBITS,
                    timeout=1
)

ser.write(13)
ser.close

```

Make the script executable with

```
chmod 744 serial_write-CR.py
```

This script is to be launched when starting the pi, it will of course be necessary for the MTM station to be powered on when the pi is powered on:

add it in /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+512=PTT
echo "16" >/sys/class/gpio/export
sleep 2
echo out >/sys/class/gpio/gpio16/direction

#GPIO19+512=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
```

Run Svxlink for testing

Before launching svxlink, free the MTM buffer by with the command

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

Give permissions to call-key file:

Xxx is the main call used in the svxlink.conf file (CALLSIGN=xxxx) in SimplexLogic, RepeaterLogic, ReflectorLogic ...) the one you actually use.

```
sudo chmod ugo+rw /var/lib/svxlink/pki/xxxx.key
```

Start svxlink by typing the following command:

```
svxlink
```

The led of the USB sound card should blink if it is equipped. At launch it is possible to get errors from the PEI.

According to the MUTE_FIRST_TX_LOC variable, the first push of PTT selects the TG (described in TETRALOGIC). The 2nd PTT call starts the talker which leads the audio to the TG on the network.

You should see this (with TetraLogic, and DEBUG = 3)

```
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/svxlink/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 (c'est parce que j'ai enlevé le GPS de mon MTM)
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 1^{er} 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
```

SVXLINK launch at boot

You need to create a service that starts svxlink after the network is available and the sound card (svxlink.service).

Another service will be created to start svxlink 60s after booting (svxlink.timer)

In the `/lib/systemd/system` directory, create the `svxlink.service` file which contains:

```
[Unit]
Description=this start 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
```

And create `svxlink.timer` file which contains :

```
[Timer]
OnStartupSec=60

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

Reload the new services services in the system:

```
sudo systemctl daemon-reload
```

Start the timer service :

```
sudo systemctl start svxlink.timer : start timer service
sudo systemctl enable svxlink.timer: enable timer service on each boot
```

Do not start the `svxlink.service` service, `svxlink timer` starts it.

Check if the services started with

```
sudo systemctl status svxlink.timer
(ctrl c to leave)
```

The service is started forever, the next time the pi starts, svxlink will start after the network is available, and after 60s.

The svxlink log file is located in /var/log/svxlink

To stop SVXLINK, type `sudo systemctl stop svxlink.service`

This stops the service.

Open the router ports according to the reflector settings:

For Echolink:

5198-5199 udp

5200 tcp

5300 Tetra Leipzig

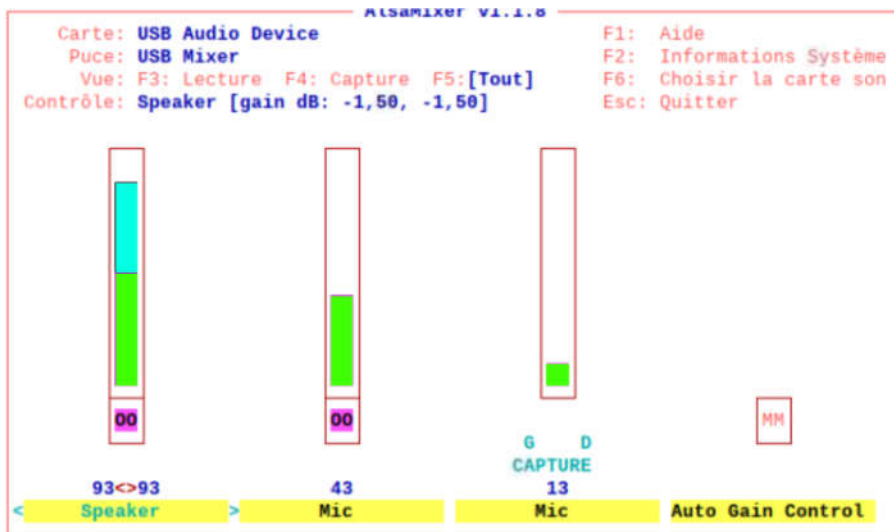
IPV6 only networks

More and more internet providers do not supply IPV4 addresses anymore, but only handles IPV6 only. Svxlink, Echolink, and TETRA Leipzig (leipzig2000.dyndns.org) will not work ☹ if you use a low level router.

You will need to use a V4/V6 router including a "2 stacks system" and a for example DNS64 system to translate addresses. Some routers have, some don't. I tested MR400 archer and it didn't work.

Note : The MMDVM system worked on Brandmeister servers (DMR) because you can reach BM servers with a IPV6 written in the config file (works on MR400 with IPv6 only)

ALSAMIXER SETTINGS



Start alsamixer

Press F5 to display all settings.

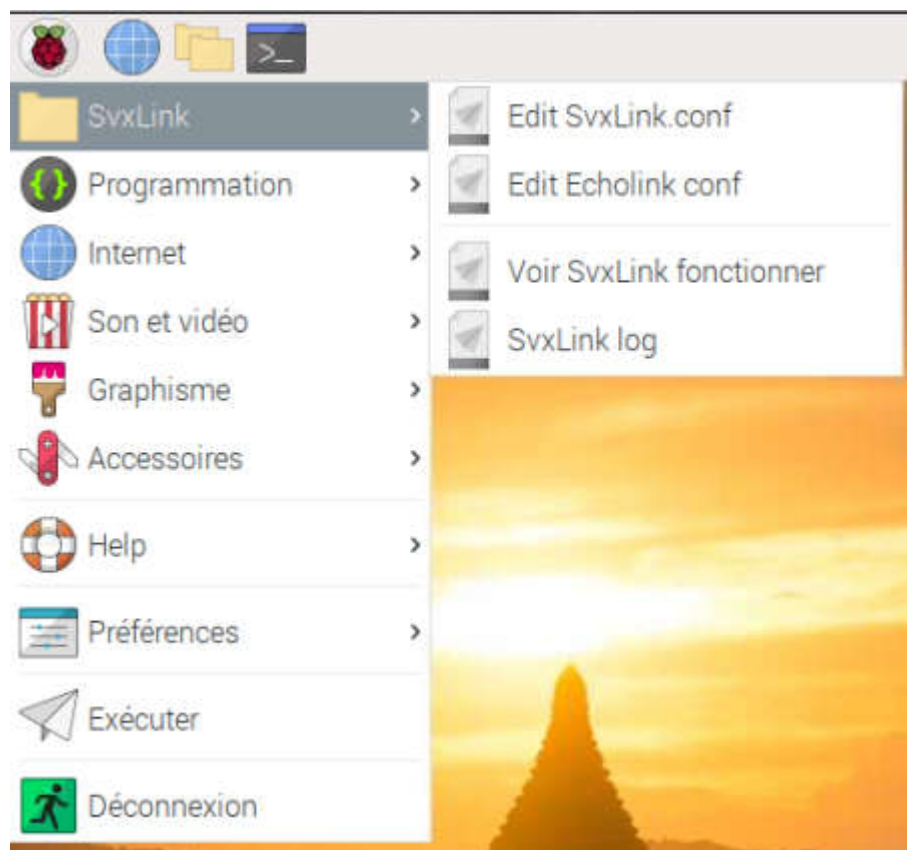
Speaker is used to adjust the audio coming from the network to the radio.

Capture allows to set the audio from the radio to the network.

Mic has no effect.

Note that Auto Gain control is muted (MM) press m.

Menu bar



Edit SvxLink.conf allows you to launch an editor to modify the svxlink.conf configuration file

Edit EchoLink.conf allows you to launch an editor to modify the EchoLink.conf configuration file

See SvxLink working: displays the window of the last 40 lines of the svxlink log file

SvxLink.log allows you to launch an editor to view the svxlink.log log file

DTMF controls

DTMF commands are not possible from a TETRA station.

* Repeater presentation

disconnection

0# help module activation

1# parrot activation

2# ECHOLINK module activation

5# metarInfo module module (weather forecast)

01# List of available weather stations

Update svxlink/tetra

Go to /home/pi

Sudo ./maj.sh

File 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 is up to date"
elif [ $LOCAL = $BASE ]; then
    echo "[UPDATE][$(date)] ===== Stop 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)] ===== Svxlk update success ====="
else
    echo "[UPDATE][$(date)] !!!!! Erreur de mise à jour de Svxlk !!!!!"
fi
```

SVXLINK Console

The console allows you to activate the macros of the [Macros] section of svxlink.conf. #
There are several ways to simulate DTMF commands on SVXLINK:

Using a socket

Launch svxlink so that it can be remotely controlled from a remote client:

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

And from the client, type:

```
echo -n "command" | nc ip_address port
```

ip_address = 127.0.0.1 if the client is on the same network as the server

port = communication port number

command = command to send to svxlink (example: *0#)

DTMF simulator

In TetraLogic section :

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

Defines a folder for exchanging DTMF commands (example: /tmp /dtmf)

And type in a console window:

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

Type in the command to send.