

Igate APRS Partie 14



Sommaire :

Partie 14 : Igate APRS

Prérequis : Partie 4 : Utilisation du système d'exploitation Raspbian Lite sans interface graphique.

14.1 Rappel du système d'exploitation Raspbian Lite	P. 2
14.2 Mise à jour du système d'exploitation	P. 3
14.3 Installation de la clé RTL-SDR	P. 4
14.4 Installation de Direwolf	P. 8
14.5 Configuration de Direwolf avec la clé RTL-SDR	P. 13
14.6 Test de la configuration	P. 15
14.7 Exécuter automatiquement Direwolf au démarrage du RPI	P. 16
14.8 Configuration de Direwolf avec une carte son USB externe	P. 20
14.9 Conclusion	P. 23

Le créateur du système APRS est <u>Bob Bruninga</u>, WB4APR.

Le tutoriel décrit avec des copies d'écrans additionnelles les documentations suivantes :

https://github.com/wb2osz/direwolf/blob/master/doc/Raspberry-Pi-SDR-IGate.pdf https://github.com/wb2osz/direwolf/blob/master/doc/User-Guide.pdf

Version du 24/04/2022 V1.0 sur RPI4

RPI partie 14-V1.0.docx

14.1 Rappel du système d'exploitation Raspbian Lite

Télécharger la dernière version de Pi Imager https://www.raspberrypi.com/software/

Download for macOS Download for Ubuntu for x86 https://downloads.raspberrypi.org/imager/imager_latest.exe				
Raspberry Pi OS Lite (32-bit) A port of Debian Bullseye with no desktop environment Sorti le : 2022-04-04 Mis en cache sur votre ordinateur	Choisir Raspberry PI OS Lite			
Respherry PI Imager v1.7.2 Advanced options X Image customization options for this session only Set hostname: raspberrypi Set hostname: raspberrypi Base SSH O Use password authentication Allow public-key authentication only Set authorized_keys for 'p': Set username and password Username: Pi Pessword: O Configure wireless LAN SSID: Hidden SSID Pessword Wretess LAN country: FR Set locale settings Time zone: Europe/Paris Keybard layout: fr Persistent settincs	 Il est recommandé de préconfigurer l'image disque : Activer le SSH Username : pi Password : raspberry Configurer l'accès WIFI Changer le fuseau horaire et la configuration du clavier en fonction du pays. 			

Sauvegarder la configuration, puis lancer l'écriture sur la carte mémoire mSd.

14.2 Mise à jour du système d'exploitation

Une fois la carte insérée dans le Raspberry Pi et mis sous tension, identifier l'adresse IP du Raspberry PI connecté sur votre réseau en Wifi (encadré en rouge).

Rappel : Dans la fenêtre console :

ifconfig

```
pi@raspberrypi:~
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
       ether dc:a6:32:e6:dc:dc txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 10 bytes 1600 (1.5 KiB)
        RX errors 0 dropped 0 overruns 0
                                             frame 0
        TX packets 10 bytes 1600 (1.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.44 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7bed:6545:85b3:3ef9 prefixlen 64 scopeid 0x20<link>
        ether dc:a6:32:e6:dc:dd txqueuelen 1000
                                                   (Ethernet)
        RX packets 11530 bytes 15366617 (14.6 MiB)
        RX errors 0 dropped 55 overruns 0 frame 0
        TX packets 5973 bytes 594867 (580.9 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Effectuer une mise à jour des paquets :

sudo apt update

```
pi@raspberrypi:~ $ sudo apt update
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye InRelease [23.7 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.
2 MB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [279 kB]
Fetched 13.5 MB in 17s (782 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Rappel : Pour copier-coller une ligne depuis le fichier pdf dans putty :

- Sélectionner la ligne avec la souris ;
- Faire clic droit, puis copier, ou Ctrl+c ;
- Sélectionner la fenêtre Putty avec un clic gauche ;
- Faire ensuite un clic droit avec la souris.

sudo apt upgrade

```
pi@raspberrypi:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  gzip liblzma5 libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc
  libraspberrypi0 pi-bluetooth raspi-config userconf-pi xz-utils
```

14.3 Installation de la clé RTL-SDR (déjà décrit dans la partie 3)

Commencer par Installer git :

sudo apt-get install git



git clone https://github.com/osmocom/rtl-sdr.git

pi@raspberrypi:~ \$ git clone https://github.com/osmocom/rtl-sdr.git						
Clonage dans 'rtl-sdr'						
remote: Enumerating objects: 65, done.						
remote: Counting objects: 100% (65/65), done.						
remote: Compressing objects: 100% (38/38), done.						
remote: Total 1863 (delta 25), reused 55 (delta 23), pack-reused 1798						
Réception d'objets: 100% (1863/1863), 448.80 KiB 1.11 MiB/s, fait.						
Résolution des deltas: 100% (1317/1317), fait.						
pi@raspberrypi:~ \$ ls						
audio Downloads Music <u>qsstv</u> Templates Videos						
Desktop GridTracker Pictures rtl-sdr tx_drm						
Documents MagPi Public templates tx_stock						
pi@raspberrypi:~ \$						

RPI partie 14-V1.0.docx

Installer les bibliothèques.

sudo apt install build-essential cmake usbutils libusb-1.0-0-dev

```
pi@raspberrypi:~/rtl-sdr/build $ sudo apt install build-essential cmake usbutils
libusb-1.0-0-dev
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
build-essential est déjà la version la plus récente (12.6).
usbutils est déjà la version la plus récente (1:010-3).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessair
es :
 libmicrodns0 libqt5charts5 xlog-data
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
 cmake-data libjsoncpp1 librhash0 libusb-1.0-doc libuv1
Paquets suggérés :
 cmake-doc ninja-build
Les NOUVEAUX paquets suivants seront installés :
 cmake cmake-data libjsoncpp1 librhash0 libusb-1.0-dev libusb-1.0-doc
 libuv1
0 mis à jour, 7 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 40578 ko dans les archives.
Après cette opération, 23,6 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] [
```

Taper ces commandes dans l'ordre :

cd rtl-sdr
mkdir build
cd build
cmake -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON/
pi@raspberrypi:~ \$ cd rtl-sdr/
pi@raspberrypi:~/rtl-sdr \$ mkair build
pi@raspberrypi:~/rtl-sdr/build \$ cmake -DINSTALL UDEV RULES=ON -DDETACH KERNEL D
RIVER=ON/
The C compiler identification is GNU 8.3.0
Check for working C compiler: /usr/bin/cc
Check for working C compiler: /usr/bin/cc works

-- Detecting C compiler ABI info - done

-- Detecting C compile features

Looking for pthread.h - found

- -- Detecting C compile features done
- -- Build type not specified: defaulting to release.
- -- Extracting version information from git describe... -- Looking for pthread.h

sudo make install



sudo ldconfig

	Set runtime	path of "/usr/local/bin/rtl_fm" to ""
	Installing:	/usr/local/bin/rtl_eeprom
	Set runtime	path of "/usr/local/bin/rtl_eeprom" to ""
	Installing:	/usr/local/bin/rtl_adsb
	Set runtime	<pre>path of "/usr/local/bin/rtl_adsb" to ""</pre>
	Installing:	/usr/local/bin/rtl_power
	Set runtime	<pre>path of "/usr/local/bin/rtl_power" to ""</pre>
pi(raspberrypi	~/rtl-sdr/build \$ sudo ldconfig
pi(]raspberrypi	-/rtl-sdr/build \$

sudo nano /etc/modprobe.d/rtlsdr-blacklist.conf

pi@raspberrypi:~/rtl-sdr/build \$ sudo nano /etc/modprobe.d/rtlsdr-blacklist.conf

```
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
blacklist dvb_usb_rtl2832u
blacklist dvb_usb_v2
blacklist dvb_core
```

Copier-coller la liste ci-dessus dans l'éditeur nano

Rappel clic droit pour coller le texte dans l'éditeur nano





Redémarrer le Raspberry PI

pi@raspberrypi:~/rtl-sdr/build \$ sudo reboot

sudo reboot

Vérification : Brancher la clé rtl-sdr sur un port USB de la Raspberry Pi. Avec LXTerminal ou Putty, taper la ligne de commande suivante :

lsusb

Localiser la clé rtl-sdr : RTL238 DVB-T

Cas de la Raspberry Pi 3 :

pi@raspberrypi:~ \$ lsusb Bus 001 Device 004: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T 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. SMC9514 Hub Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub F4GOH – KF4GOH

Cas de la Raspberry Pi 4 :

pi@	pi@raspberrypi:~ \$ lsusb					
Bus	002	Device	001:	ID	1d6b:0003	Linux Foundation 3.0 root hub
Bus	001	Device	003:	ID	0bda:2838	Realtek Semiconductor Corp. RTL2838 DVB-T
Bus	001	Device	002:	ID	2109:3431	VIA Labs, Inc. Hub
Bus	001	Device	001:	ID	1d6b:0002	Linux Foundation 2.0 root hub

Avec LXTerminal ou Putty, taper la ligne de commande suivante : (attention tiret du bas)

rtl test

La clé rtl-sdr devrait être reconnue, si ce n'est pas le cas, débrancher et rebrancher la clé et recommencer le test.



Quitter le programme en appuyant simultanément sur les touches Ctrl et la touche c (Ctrl+c)

Si **rtl_test** affiche en boucle ce message **lost at least 112 bytes**, réinstaller le programme en utilisant la méthode manuelle.

14.4 Installation de direwolf

Le meilleur logiciel linux pour gérer l'APRS est <u>Direwolf</u>, conçu par John, wb2osz. Celui-ci fonctionne très bien avec une clé RTL-SDR.

Vérification des prérequis : Normalement, ces utilitaires sont déjà installés dans le système d'exploitation, mais il est quand même plus prudent de vérifier.

sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install make
sudo apt-get install cmake

i@raspberrypi:~ \$ sudo apt-get install gcc Lecture des listes de paquets... Fait Construction de l'arbre des dépendances Lecture des informations d'état... Fait gcc est déjà la version la plus récente (4:8.3.0-1+rpi2). gcc passé en « installé manuellement ». Le paquet suivant a été installé automatiquement et n'est plus nécessaire : python-colorzero Veuillez utiliser « sudo apt autoremove » pour le supprimer. 0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour. pi@raspberrypi:~ \$ sudo apt-get install g++ Lecture des listes de paquets... Fait Construction de l'arbre des dépendances Lecture des informations d'état... Fait g++ est déjà la version la plus récente (4:8.3.0-1+rpi2). g++ passé en « installé manuellement ». Le paquet suivant a été installé automatiquement et n'est plus nécessaire : python-colorzero /euillez utiliser « sudo apt autoremove » pour le supprimer. 0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour. pi@raspberrypi:~ \$ sudo apt-get install make Lecture des listes de paquets... Fait Construction de l'arbre des dépendances Lecture des informations d'état... Fait make est déjà la version la plus récente (4.2.1-1.2). make passé en « installé manuellement ». Le paquet suivant a été installé automatiquement et n'est plus nécessaire : python-colorzero Veuillez utiliser « sudo apt autoremove » pour le supprimer. 0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour. pi@raspberrypi:~ \$ sudo apt-get install cmake Lecture des listes de paquets... Fait Construction de l'arbre des dépendances Lecture des informations d'état... Fait cmake est déjà la version la plus récente (3.16.3-3~bpo10+1). Le paquet suivant a été installé automatiquement et n'est plus nécessaire : python-colorzero Veuillez utiliser « sudo apt autoremove » pour le supprimer. O mis à jour, O nouvellement installés, O à enlever et O non mis à jour. i@raspberrypi:~ \$

Installer les bibliothèques sonores

sudo apt-get install libasound2-dev





Télécharger le code source de Direwolf en utilisant git

git clone https://www.github.com/wb2osz/direwolf



Aller dans le répertoire Direwolf, puis créer un répertoire build.

cd direwolf git checkout dev mkdir build && cd build



Compiler et installer le programme

cmake ..

pi@raspberrypi:~/direwolf/build \$ cmake
The C compiler identification is GNU 8.3.0
^[[D The CXX compiler identification is GNU 8.3.0
Check for working C compiler: /usr/bin/cc
Check for working C compiler: /usr/bin/cc works
Detecting C compiler ABI info
Detecting C compiler ABI info - done
Detecting C compile features
Detecting C compile features - done
Check for working CXX compiler: /usr/bin/c++
Check for working CXX compiler: /usr/bin/c++ works
Detecting CXX compiler ABI info
Detecting CXX compiler ABI info - done
Detecting CXX compile features
Detecting CXX compile features - done
Found Git: /usr/bin/git (found version "2.20.1")
Dire Wolf Version: 1.7.0-b66c21d
Build type set to: Release
CMake system: Linux
Target architecture: ARM
Use NEON SIMD instructions
Looking for strlcpy
Looking for strlcpy - not found
Looking for strlcat
Looking for strlcat - not found

make -j4

pi@raspberrypi:~/direwolf/build \$ make -j4
Scanning dependencies of target misc
Scanning dependencies of target geotranz
[1%] Building C object external/misc/CMakeFiles/misc.dir/strlcat.c.o
[1%] Building C object external/misc/CMakeFiles/misc.dir/strlcpy.c.o
[2%] Building C object external/geotranz/CMakeFiles/geotranz.dir/error_string.c.
[2%] Building C object external/geotranz/CMakeFiles/geotranz.dir/mgrs.c.o
[2%] Building C object external/geotranz/CMakeFiles/geotranz.dir/polarst.c.o
[3%] Linking C static library libmisc.a
[4%] Building C object external/geotranz/CMakeFiles/geotranz.dir/tranmerc.c.o
[4%] Built target misc
Scanning dependencies of target text2tt
[4%] Building C object src/CMakeFiles/text2tt.dir/tt_text.c.o
[4%] Building C object external/geotranz/CMakeFiles/geotranz.dir/ups.c.o
[5%] Building C object external/geotranz/CMakeFiles/geotranz.dir/usng.c.o
[5%] Building C object external/geotranz/CMakeFiles/geotranz.dir/utm.c.o
Scanning dependencies of target atest
[5%] Building C object src/CMakeFiles/atest.dir/atest.c.o
[6%] Building C object src/CMakeFiles/atest.dir/ais.c.o
[8%] Linking C executable text2tt
[96%] Building C object src/CMakeFiles/direwolf.dir/xmit.c.o
[97%] Building C object src/CMakeFiles/direwolf.dir/dwgpsd.c.o
[97%] Building C object src/CMakeFiles/direwolf.dir/dwgpsnmea.c.o
[98%] Building C object src/CMakeFiles/direwolf.dir/mheard.c.o
[98%] Building C object src/CMakeFiles/direwolf.dir/audio.c.o
[100%] Building C object src/CMakeFiles/direwolf.dir/cm108.c.o
[100%] Linking C executable direwolf
[100%] Built target direwolf
pi@raspberrypi:~/direwolf/build \$

pi@raspberrypi:~/direwolf/build \$ sudo make install
<pre>[4%] Built target geotranz</pre>
<pre>[6%] Built target misc</pre>
<pre>[9%] Built target ll2utm</pre>
[10%] Built target text2tt
[12%] Built target utm211
[28%] Built target atest
[31%] Built target aclients
[40%] Built target decode aprs
[44%] Built target kissutil
[45%] Built target log2gpx
[47%] Built target tt2text
[55%] Built target gen packets
[91%] Built target direwolf
[93%] Built target cm108
[95%] Built target ttcalc
[100%] Built target appserver
Install the project
Install configuration: "Release"
Installing: /usr/local/share/doc/direwolf/CHANGES.md
Installing: /usr/local/share/doc/direwolf/LICENSE
Installing: /usr/local/share/doc/direwolf/external/LICENSE
Installing: /usr/local/share/applications/direwolf.desktop
Installing: /usr/local/share/pixmaps/direwolf_icon.png

sudo make install

make install-conf



14.5 Configuration de Direwolf avec la clé RTL-SDR

Retourner dans le répertoire /home/pi

On peut constater que des exemples de configurations de Direwolf ont été créés. Ces fichiers se terminent par l'extension .conf.

cd ~	
pwd	
ls	

```
pi@raspberrypi:~/direwolf/build $ cd ~
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls
direwolf dw-start.sh sdr.conf telem-m0xer-3.txt
direwolf.conf rtl-sdr telem-balloon.conf telem-volts.conf
pi@raspberrypi:~ $
```

Obtenir un mot de passe chiffré pour le serveur APRS en utilisant l'adresse suivante : <u>https://aprs.do3sww.de/</u>

APRS Passcode Generator				
Enter your amature radio callsign to get a Passocde for the APRS-IS network.				
Callsign: F4GOH	Get Passcode!			
Your Passcode: 15001				
Source Code on Github				

Editer le fichier sdr.conf

nano	sdr.	conf
------	------	------

Modifier les lignes conformément à votre indicatif et situation géographique.

🛃 pi@raspberrypi: ~		
GNU nano 5.4	sdr.conf	
‡ ‡ Sample configuration fo ‡	for SDR read-only IGate.	
# We might not have an au	audio output device so set to null.	
# We will override the in	input half on the command line.	
ADEVICE null null		
CHANNEL 0		
MYCALL F4GOH		
<pre># First you need to speci # The current preferred w # noam.aprs2.net # soam.aprs2.net # euro.aprs2.net # asia.aprs2.net # aunz.aprs2.net</pre>	<pre>:ify the name of a Tier 2 server. way is to use one of these regional rotate addresses:</pre>	
IGSERVER euro.aprs2.net		
<pre># You also need to specif # Contact the author if y # https://aprs.do3sww.de/ IGLOGIN F4GOH 15001</pre>	fy your login name and passcode. you can't figure out how to generate the passcode. z/	
PBEACON sendto=IG delav=0	=0:30 every=60:00 symbol="igate" overlay=R lat=47753.41N long=000716.60E COMMENT="https://hamprojects.word	dpress.com/"

Il est possible d'ajouter la commande PBEACON qui signalera la position de l'igate sur https://aprs.fi



PBEACON doit être sur une même ligne !

14.6 Test de la configuration

Exécuter la commande suivante :

rtl fm -f 144.80M - | direwolf -c sdr.conf -r 24000 -D 1 -



Vérifier ensuite que l'igate apparait sur https://aprs.fi



Si la ligne de commande est mal tapée ou le copier-coller est mauvais, le programme ne s'exécute pas correctement comme le montre l'exemple suivant :

pi@raspberrypi:~ \$ rtl_fm -f 144.80M - | direwolf -c sdr.conf -r 24000 -D 1 -Dire Wolf DEVELOPMENT version 1.7 E (Apr 23 2022) Includes optional support for: cm108-ptt Reading config file sdr.conf Audio input device for receive: - (channel 0) Audio out device for transmit: null (channel 0) ALSA lib pcm.c:2660:(snd pcm open noupdate) Unknown PCM -Could not open audio device - for input No such file or directory Pointless to continue without audio device. Found 1 device(s): 0: Realtek, RTL2838UHIDIR, SN: 00000001 Using device 0: Generic RTL2832U OEM Found Rafael Micro R820T tuner Tuner gain set to automatic. Tuned to 145052000 Hz. Oversampling input by: 42x. Oversampling output by: 1x. Buffer size: 8.13ms Exact sample rate is: 1008000.009613 Hz Sampling at 1008000 S/s. Output at 24000 Hz. Signal caught, exiting! User cancel, exiting...

L'autre solution est de télécharger un script tout prêt, changer les droits en exécution puis de l'exécuter.



Utiliser un tracker APRS pour envoyer une trame d'essai, afin de confirmer le bon décodage.

Mon tracker F4GOH-12 apparait bien dans la console et sur https://aprs.fi





14.7 Exécuter automatiquement Direwolf au démarrage du RPI

Quitter le programme à l'aide des touches Ctrl + c.



Installer le logiciel screen





Renommer le fichier dw-start.sh existant en dw-start.sh.bak, afin de ne pas le perdre. Télécharger le script dw-start.sh configuré puis changer les droits en exécution.

mv dw-start.sh dw-start.sh.bak wget https://github.com/f4goh/DRAPRS/raw/master/config direwolf/sdr/dw-start.sh chmod +x dw-start.sh

Exécuter le fichier dw-start.sh, puis attendre 30 secondes

./dw-start.sh



Une fois le script actif, vérifier à nouveau sur <u>https://aprs.fi</u> l'envoi de la trame de balise.



Editer le planificateur de tâches crontab en choisissant l'option 1 (éditeur nano)

crontab -e



A la fin du fichier ajouter la ligne suivante

/home/pi/dw-start.sh >/dev/null 2>&1 * * * * *





Rappel:



Un message apparait une fois sorti de l'éditeur « installing new crontab »



Redémarrer le Raspberry

sudo reboot



Direwolf va s'exécuter en tâche de fond au redémarrage du Raspberry PI.



Attendre une minute, puis vérifier une nouvelle fois sur https://aprs.fi que l'envoi des trames de balise et du tracker fonctionne correctement.



2022-04-24 12:24:05 CEST: F4GOH-12>F4GOH,WIDE1-1,WIDE2-1,qAO,F4GOH:/102400h4753.41N/00016.60Eb/A=000276/73 Anthony 2022-04-24 12:24:35 CEST: F4GOH-12>F4GOH,WIDE1-1,WIDE2-1,qAO,F4GOH:/102430h4753.41N/00016.60Eb/A=000276/73 Anthony 2022-04-24 12:25:05 CEST: F4GOH-12>F4GOH,WIDE1-1,WIDE2-1,qAO,F4GOH:/102500h4753.41N/00016.60Eb/A=000276/73 Anthony

Pour arrêter le planificateur de tâches :

crontab -e

Ajouter un # au début de la ligne, puis redémarrer le RPI.

/home/pi/dw-start.sh >/dev/null 2>&1

sudo reboot



14.8 Configuration de Direwolf avec une carte son USB externe

Le matériel utilisé pour la suite sera la carte son USB cm108 modifiée décrite dans le tutoriel 11.



Plugger la carte son sur le Raspberry PI, puis vérifier la compatibilité de commande PTT (1) et noter le numéro de la carte son reconnue dans le RPI (2).

crontab -e

pi@raspberrypi:~ \$ cm108							
VID PID Product	Sound	ADEVICE 2	ADEVICE	HID [ptt]			
** 0d8c 000c C-Media USB Headphone Set	/dev/snd/pcmC3D0c	plughw:3,0	plughw:Set,0	/dev/hidraw0			
** 0d8c 000c C-Media USB Headphone Set	/dev/snd/pcmC3D0p	plughw:3,0	plughw:Set,0	/dev/hidraw0			
** 0d8c 000c C-Media USB Headphone Set	/dev/snd/controlC3			/dev/hidraw0			
** = Can use Audio Adapter GPIO for PTT. 1							
Notice that each USB Audio adapter is assigned a number and a name. These are not predictable so you could							
end up using the wrong adapter after adding or removing other USB devices or after rebooting. You can assign a							
name to each USB adapter so you can refer to the same one each time. This can be based on any characteristics							
that makes them unique such as product id or serial number. Unfortunately these devices don't have unique serial							
numbers so how can we tell them apart? A name can also be assigned based on the physical USB socket.							
Create a file like "/etc/udev/rules.d/85-mv-usb-audio.rules" with the following contents and then reboot.							
SUBSYSTEM!="gound" GOTO="mu ush audio end"							
CTIONI=#add#GOTO=#www.ush_audio_end#							
North at , or masdate_the							
pi@raspberrvpi:~ S							

Renommer le fichier direwolf.conf original, puis télécharger le fichier direwolf.conf préconfiguré

mv direwolf.conf direwolf.conf.bak

wget https://github.com/f4goh/DRAPRS/raw/master/config_direwolf/soundcard/direwolf.conf



Tutoriel

Configuration personnalisée du fichier direwolf.conf.

Tutoriel

Editer le fichier direwolf.conf.

```
nano direwolf.conf
```

Indiquer le numéro de la carte son. (Ici dans le RPI4, la carte son externe est en numéro 3)



Changer l'indicatif

MYCALL WB2OSZ-5 Example (don't use this unless you are me): MYCALL F4GOH

Activer la commande PTT via la carte son cm108



Envoyer un signal en transmission de la position de l'igate. (Facultatif et utile si le RPI n'est pas connecté à internet)



Activer le digipeater. (Retransmission radio des trames entendues)



Activer l'igate et envoyer toutes les 60 minutes la position de l'igate (PBEACON) sur internet.



Sauvegarder le fichier Ctrl+o, Ctrl+x

Exécuter le programme direwolf, celui-ci charge par défaut le fichier de configuration direwolf.conf

direwolf

```
pi@raspberrypi:~ $ direwolf
Dire Wolf DEVELOPMENT version 1.7 E (Apr 23 2022)
Includes optional support for: cm108-ptt
Reading config file direwolf.conf
Audio device for both receive and transmit: plughw:3,0 (channel 0)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, A+, 44100 sample rate / 3.
Using /dev/hidraw0 GPIO 3 for channel 0 PTT control.
Ready to accept AGW client application 0 on port 8000
                                                      . . .
Ready to accept KISS TCP client application 0 on port 8001 ...
Now connected to IGate server euro.aprs2.net (44.141.143.40)
Check server status here http://44.141.143.40:14501
[ig] # aprsc 2.1.5-g8af3cdc
[ig] # logresp F4GOH verified, server T2NORWAY
[ig] F4GOH>APDW17:!4753.42NR00016.60E&https://hamprojects.wordpress.com/
[0L] F4GOH>APDW17,WIDE1-1,WIDE2-1:!4753.42NS00016.60E#PHG7140Teloche
F4GOH-12 audio level = 24(6/4)
                                [NONE]
                                            [0.4] F4GOH-12>F4GOH,WIDE1-1,WIDE2-1:/123000h4753.41N/00016.61Eb/A=000229/73 Anthony
Position with time, BIKE
N 47 53.4100, E 000 16.6100, alt 70 m (229 ft)
 73 Anthony
 [0H] F4GOH-12>F4GOH, F4GOH*, WIDE2-1:/123000h4753.41N/00016.61Eb/A=000229/73 Anthony
```

Vérifier sur https://aprs.fi que l'envoi des trames de balise et du tracker fonctionne correctement.





Renommer le fichier dw-start.sh existant en dw-start.sh.bak, afin de ne pas le perdre. Télécharger le script dw-start.sh configuré puis changer les droits en exécution.

```
mv dw-start.sh dw-start.sh.bak
wget https://github.com/f4goh/DRAPRS/raw/master/config_direwolf/soundcard/dw-start.sh
chmod +x dw-start.sh
```

Exécuter le fichier dw-start.sh, puis attendre 30 secondes

```
./dw-start.sh
```

Une fois le script actif, vérifier à nouveau sur https://aprs.fi l'envoi de la trame de balise.



Reprendre la configuration du planificateur de tâches « crontab » décrit sur les pages 17,18 et 19.

14.9 Conclusion

Le protocole APRS est toujours très utilisé de nos jours. Sa mise en œuvre nécessite peu de composants et il existe de nombreux codes sources sur le net. Le programme Direwolf en est un bon exemple, même si sa configuration est parfois un peu complexe. Cependant les possibilités de configuration sont nombreuses. Avec une clé RTL-SDR ou une carte son modifiée, il sera facile de dédier un igate à moindre frais.

Depuis quelques années Direwolf supporte le FX25.

https://en.wikipedia.org/wiki/FX.25_Forward_Error_Correction

Le code source d'un tracker pour ballon sonde supportant le FX25 est disponible à l'adresse suivante :

https://github.com/PhilippeSimier/Radiocommunication/tree/master/projet_ballon

Ce tracker à base d'un (ESP32+DRA818) a été programmé par Philippe et Anthony (F4GOH)

Direwolf est capable de décoder le FX25 en ajoutant l'option -X 1 dans le fichier « dw-start.sh »

```
# 2. FX.25 Forward Error Correction (FEC) will allow your signal to
# go farther under poor radio conditions. Add "-X 1" to the command
line.
```

DWCMD="\$DIREWOLF -a 100 -X 1"