

HowTo BUILD Direwolf (iGATE APRS) RX ONLY

IW9HHF – ALBERTO DRAGO
iw9hhf@gmail.com

Rimuoviamo pulseaudio con tutta la sua configurazione (--purge)

```
$sudo apt-get remove --purge pulseaudio
$sudo apt-get autoremove
$rm -rf /home/pi/.pulse
$sudo reboot
```

Installiamo il necessario per la compilazione e l'installazione:

```
$sudo apt-get install vim build-essential sigc++ gcc g++ make cmake
libcrypt-dev libgsm1-dev libsigc++-2.0-dev tcl-dev libspeex-dev
libasound2-dev libpopt-dev libssl-dev libopus-dev groff tcl8.5-dev tk8.5-
dev git
```

questa operazione impiegherà qualche minuto, al termine sarà necessario riavviare il sistema.

```
$sudo shutdown -r now
```

installiamo direwolf:

```
$sudo apt-get install libasound2-dev
$cd
$git clone https://www.github.com/wb2osz/direwolf
$cd direwolf
```

A questo punto possiamo decidere quale versione installare. Il Consiglio è sempre quello di installare la versione recente più stabile. Con il comando successivo andremo a visualizzare tutte le versioni disponibili:

```
$git tag
```

Queste le versioni disponibili al 09 Giugno2020

```
1.0
1.1
1.2
1.3
1.3-beta
1.3-dev-F
1.3-dev-I
1.3-dev-K
1.4
1.4-beta
1.4-dev-D
1.4-dev-E
1.5
1.5-beta
1.5-beta2
1.5-beta3
1.5-beta4
```

per comodità decidiamo di installare la versione 1.5

```
$git checkout 1.5
$make
```

Impiegherà qualche minuto... al termine verrà suggerito, a terminale, il comando successivo, eseguiamo:

```
$sudo make install
```

Fatto! Concludiamo l'installazione con il comando:

```
$make install-conf
```

Se tutto è andato a buon fine possiamo effettuare una prima configurazione minimale, questo ci consentirà di vedere i pacchetti aprs ricevuti dal file di log.

Torniamo nella nostra home digitando cd e invio.

Facciamo subito una copia del file di configurazione, utile riferimento in caso di errori accidentali nella modifica dello stesso.

```
cp direwolf.conf direwolf.conf.old
```

editiamo il file `direwolf.conf` e modifichiamo le parti evidenziate in giallo

```
root@direwolf:/home/pi# cat direwolf.conf
#####
#
#           Configuration file for Dire Wolf           #
#
#           Linux version                             #
#
#####
#
# Consult the User Guide for more details on configuration options.
#
#
# These are the most likely settings you might change:
#
# (1) MYCALL - call sign and SSID for your station.
#
#           Look for lines starting with MYCALL and
#           change NOCALL to your own.
#
# (2) PBEACON - enable position beaconing.
#
#           Look for lines starting with PBEACON and
#           modify for your call, location, etc.
#
# (3) DIGIPEATER - configure digipeating rules.
#
#           Look for lines starting with DIGIPEATER.
#           Most people will probably use the given example.
#           Just remove the "#" from the start of the line
#           to enable it.
#
# (4) IGSERVER, IGLOGIN - IGate server and login
#
#           Configure an IGate client to relay messages between
#           radio and internet servers.
#
#
```

```
# The default location is "direwolf.conf" in the current working directory.
# On Linux, the user's home directory will also be searched.
# An alternate configuration file location can be specified with the "-c" command line option.
#
# As you probably guessed by now, # indicates a comment line.
#
# Remove the # at the beginning of a line if you want to use a sample
# configuration that is currently commented out.
#
# Commands are a keyword followed by parameters.
#
# Command key words are case insensitive. i.e. upper and lower case are equivalent.
#
# Command parameters are generally case sensitive. i.e. upper and lower case are different.
#
```

```
#####
#
#           FIRST AUDIO DEVICE PROPERTIES           #
#           (Channel 0 + 1 if in stereo)           #
#
#####
```

```
#
# Many people will simply use the default sound device.
# Some might want to use an alternative device by choosing it here.
#
# Linux ALSA is complicated. See User Guide for discussion.
# To use something other than the default, generally use plughw
# and a card number reported by "arecord -l" command. Example:
```

```
ADEVICE plughw:1,0
```

```
# Starting with version 1.0, you can also use "-" or "stdin" to
# pipe stdout from some other application such as a software defined
# radio. You can also specify "UDP:" and an optional port for input.
# Something different must be specified for output.
```

```
# ADEVICE - plughw:1,0
# ADEVICE UDP:7355 default
```

```
#
# Number of audio channels for this soundcard: 1 or 2.
#
```

```
ACHANNELS 1
#ACHANNELS 2
```

```
#####
#
#           SECOND AUDIO DEVICE PROPERTIES           #
#           (Channel 2 + 3 if in stereo)           #
#
#####
```

```
#ADEVICE1 ...
```

```
#####
#
#           THIRD AUDIO DEVICE PROPERTIES           #
#           (Channel 4 + 5 if in stereo)           #
#
#####
```

```
#ADEVICE2 ...
```

```
#####
#
#           CHANNEL 0 PROPERTIES                     #
#
#####
```

CHANNEL 0

```
#
# The following MYCALL, MODEM, PTT, etc. configuration items
# apply to the most recent CHANNEL.
#
#
# Station identifier for this channel.
# Multiple channels can have the same or different names.
#
# It can be up to 6 letters and digits with an optional ssid.
# The APRS specification requires that it be upper case.
#
# Example (don't use this unless you are me): MYCALL      WB2OSZ-5
#
```

MYCALL IW9HHF-10

```
#
# Pick a suitable modem speed based on your situation.
#   1200   Most common for VHF/UHF. Default if not specified.
#   300    Low speed for HF SSB.
#   9600   High speed - Can't use Microphone and Speaker connections.
#
# In the simplest form, just specify the speed.
#
```

MODEM 1200

```
#MODEM 300
#MODEM 9600
```

```
#
# These are the defaults should be fine for most cases. In special situations,
# you might want to specify different AFSK tones or the baseband mode which does
# not use AFSK.
#
```

```
#MODEM 1200 1200:2200
#MODEM 300 1600:1800
#MODEM 9600 0:0
```

```
#
#
# On HF SSB, you might want to use multiple demodulators on slightly different
# frequencies to compensate for stations off frequency. Here we have 7 different
# demodulators at 30 Hz intervals. This takes a lot of CPU power so you will
# probably need to reduce the audio sampling rate with the /n option.
```

```
#MODEM 300 1600:1800 7@30 /4
```

```
#
# Uncomment line below to enable the DTMF decoder for this channel.
#
```

```
#DTMF
```

```
#
# If not using a VOX circuit, the transmitter Push to Talk (PTT)
# control is usually wired to a serial port with a suitable interface circuit.
# DON'T connect it directly!
#
```

```
# For the PTT command, specify the device and either RTS or DTR.
# RTS or DTR may be preceded by "-" to invert the signal.
# Both can be used for interfaces that want them driven with opposite polarity.
#
# COM1 can be used instead of /dev/ttyS0, COM2 for /dev/ttyS1, and so on.
#
```

```
#PTT COM1 RTS
#PTT COM1 RTS -DTR
#PTT /dev/ttyUSB0 RTS
```

```
#
# On Linux, you can also use general purpose I/O pins if
# your system is configured for user access to them.
# This would apply mostly to microprocessor boards, not a regular PC.
```

```

# See separate Raspberry Pi document for more details.
# The number may be preceded by "-" to invert the signal.
#

#PTT GPIO 25

# The Data Carrier Detect (DCD) signal can be sent to the same places
# as the PTT signal. This could be used to light up an LED like a normal TNC.

#DCD COM1 -DTR
#DCD GPIO 24

#####
#
#           CHANNEL 1 PROPERTIES
#
#
#####

#CHANNEL 1

#
# Specify MYCALL, MODEM, PTT, etc. configuration items for
# CHANNEL 1. Repeat for any other channels.

#####
#
#           TEXT TO SPEECH COMMAND FILE
#
#
#####

#SPEECH dwespeak.sh

#####
#
#           VIRTUAL TNC SERVER PROPERTIES
#
#
#####

#
# Dire Wolf acts as a virtual TNC and can communicate with
# client applications by different protocols:
#
# - the "AGW TCPIP Socket Interface" - default port 8000
# - KISS protocol over TCP socket - default port 8001
# - KISS TNC via pseudo terminal (-p command line option)
#

AGWPORT 8000
KISSPORT 8001

#
# It is sometimes possible to recover frames with a bad FCS.
# This applies to all channels.
#
# 0 [NONE] - Don't try to repair.
# 1 [SINGLE] - Attempt to fix single bit error. (default)
# 2 [DOUBLE] - Also attempt to fix two adjacent bits.
# ... see User Guide for more values and in-depth discussion.
#

#FIX_BITS 0

#
#####
#
#           BEACONING PROPERTIES
#
#
#####

#
# Beaconing is configured with these two commands:
#
# PBEACON - for a position report (usually yourself)

```

```

# OBEACON - for an object report (usually some other entity)
#
# Each has a series of keywords and values for options.
# See User Guide for details.
#
# Example:
#
# This results in a broadcast once every 10 minutes.
# Every half hour, it can travel via two digipeater hops.
# The others are kept local.
#
#PBEACON delay=1 every=30 overlay=S symbol="digi" lat=42^37.14N long=071^20.83W power=50 height=20
gain=4 comment="Chelmsford MA" via=WIDE1-1,WIDE2-1
#PBEACON delay=11 every=30 overlay=S symbol="digi" lat=42^37.14N long=071^20.83W power=50 height=20
gain=4 comment="Chelmsford MA"
#PBEACON delay=21 every=30 overlay=S symbol="digi" lat=42^37.14N long=071^20.83W power=50 height=20
gain=4 comment="Chelmsford MA"

# With UTM coordinates instead of latitude and longitude.

#PBEACON delay=1 every=10 overlay=S symbol="digi" zone=19T easting=307477 northing=4720178

#
# When the destination field is set to "SPEECH" the information part is
# converted to speech rather than transmitted as a data frame.
#
#CBEACON dest="SPEECH" info="Club meeting tonight at 7 pm."

# Similar for Morse code. If SSID is specified, it is multiplied
# by 2 to get speed in words per minute (WPM).

#CBEACON dest="MORSE-6" info="de MYCALL"

#
# Modify for your particular situation before removing
# the # comment character from the beginning of appropriate lines above.
#

#####
# DIGIPEATER PROPERTIES #
# #
#####

#
# For most common situations, use something like this by removing
# the "#" from the beginning of the line below.
#

#DIGIPEAT 0 0 ^WIDE[3-7]-[1-7]$|^TEST$ ^WIDE[12]-[12]$ TRACE

# See User Guide for more explanation of what this means and how
# it can be customized for your particular needs.

# Filtering can be used to limit what is digipeated.
# For example, only weather reports, received on channel 0,
# will be retransmitted on channel 1.
#

#FILTER 0 1 t/wn

#####
# INTERNET GATEWAY #
# #
#####

# First you need to specify the name of a Tier 2 server.
# The current preferred way is to use one of these regional rotate addresses:

```

```
# noam.aprs2.net - for North America
# soam.aprs2.net - for South America
# euro.aprs2.net - for Europe and Africa
# asia.aprs2.net - for Asia
# aunz.aprs2.net - for Oceania
```

IGSERVER euro.aprs2.net

```
# You also need to specify your login name and passcode.
# Contact the author if you can't figure out how to generate the passcode.
```

```
#al posto del XXXXXXXX inserire il codice generato (http://apps.magicbug.co.uk/passcode/)
```

```
IGLOGIN IW9XXX XXXXXXXX
```

```
# That's all you need for a receive only IGate which relays
# messages from the local radio channel to the global servers.
```

```
# Some might want to send an IGate client position directly to a server
# without sending it over the air and relying on someone else to
# forward it to an IGate server. This is done by using sendto=IG rather
# than a radio channel number. Overlay R for receive only, T for two way.
```

```
#per identificare la corretta posizione fate riferimento al link: https://www.geoplaner.com/
```

```
PBEACON sendto=IG delay=0:40 every=20 symbol=R& alt=40 lat=45.52062N long=09.32947E comment="iGATE
AAPRS rx-only"
```

```
#PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=R lat=42^37.14N long=071^20.83W
#PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=T lat=42^37.14N long=071^20.83W
```

```
# To relay messages from the Internet to radio, you need to add
# one more option with the transmit channel number and a VIA path.
```

```
#IGTXVIA 0 WIDE1-1
```

```
# You might want to apply a filter for what packets will be obtained from the server.
# Read about filters here: http://www.aprs-is.net/javaprfilter.aspx
# Example, positions and objects within 50 km of my location:
```

```
#IGFILTER m/50
```

```
# That is known as a server-side filter. It is processed by the IGate server.
# You can also apply local filtering to limit what will be transmitted on the
# RF side. For example, transmit only "messages" on channel 0 and weather
# reports on channel 1.
```

```
#FILTER IG 0 t/m
#FILTER IG 1 t/wn
```

```
# Finally, we don't want to flood the radio channel.
# The IGate function will limit the number of packets transmitted
# during 1 minute and 5 minute intervals. If a limit would
# be exceeded, the packet is dropped and message is displayed in red.
```

```
IGTXLIMIT 6 10
```

```
#####
#                                     #
#           APRStt GATEWAY             #
#                                     #
#####
```

```
#
# Dire Wolf can receive DTMF (commonly known as Touch Tone)
# messages and convert them to packet objects.
#
# See separate "APRStt-Implementation-Notes" document for details.
#
```

```
#
# Sample gateway configuration based on:
#
# http://www.aprs.org/aprstt/aprstt-coding24.txt
# http://www.aprs.org/aprs-jamboree-2013.html
```

```

#
# Define specific points.

TTPOINT B01 37^55.37N 81^7.86W
TTPOINT B7495088 42.605237 -71.34456
TTPOINT B934 42.605237 -71.34456

TTPOINT B901 42.661279 -71.364452
TTPOINT B902 42.660411 -71.364419
TTPOINT B903 42.659046 -71.364452
TTPOINT B904 42.657578 -71.364602

# For location at given bearing and distance from starting point.
TTVECTOR B5bbbbdd 37^55.37N 81^7.86W 0.01 mi

# For location specified by x, y coordinates.
TTGRID Byyyxxx 37^50.00N 81^00.00W 37^59.99N 81^09.99W

# UTM location for Lowell-Dracut-Tyngsborough State Forest.
TTUTM B6xxxxyy 19T 10 300000 4720000

# Location for the corral.
TTCORRAL 37^55.50N 81^7.00W 0^0.02N

# Compact messages - Fixed locations xx and object yyy where
# Object numbers 100 - 199 = bicycle
# Object numbers 200 - 299 = fire truck
# Others = dog

TTMACRO xx1yy B9xx*AB166*AA2B4C5B3B0A1yy
TTMACRO xx2yy B9xx*AB170*AA3C4C7C3B0A2yy
TTMACRO xxyyy B9xx*AB180*AA3A6C4A0Ayyy

TTMACRO z Cz

# Receive on channel 0, Transmit object reports on channel 1 with optional via path.
# You probably want to put in a transmit delay on the APRStt channel so it
# it doesn't start sending a response before the user releases PTT.
# This is in 10 ms units so 100 means 1000 ms = 1 second.

#TTOBJ 0 1 WIDE1-1
#CHANNEL 0
#DWAIT 100

# Advertise gateway position with beacon.

# OBEACON DELAY=0:15 EVERY=10:00 VIA=WIDE1-1 OBJNAME=WB20SZ-tt SYMBOL=APRStt LAT=42^37.14N
LONG=71^20.83W COMMENT="APRStt Gateway"

# Sample speech responses.
# Default is Morse code "R" for received OK and "?" for all errors.

#TTERR OK SPEECH Message Received.
#TTERR D_MSG SPEECH D not implemented.
#TTERR INTERNAL SPEECH Internal error.
#TTERR MACRO_NOMATCH SPEECH No definition for digit sequence.
#TTERR BAD_CHECKSUM SPEECH Bad checksum on call.
#TTERR INVALID_CALL SPEECH Invalid callsign.
#TTERR INVALID_OBJNAME SPEECH Invalid object name.
#TTERR INVALID_SYMBOL SPEECH Invalid symbol.
#TTERR INVALID_LOC SPEECH Invalid location.
#TTERR NO_CALL SPEECH No call or object name.
#TTERR SATSQ SPEECH Satellite square must be 4 digits.
#TTERR SUFFIX_NO_CALL SPEECH Send full call before using suffix.

```


Attiviamo direwolf all'avvio del raspberry.

Aggiungete in fondo al file /etc/rc.local il **comando per avviarlo**, prima di exit 0.

esempio:

```
#!/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
sleep 2
/usr/local/bin/direwolf -t 0 -c /home/pi/direwolf.conf >> /var/log/direwolf.log &
exit 0
root@direwolf:/home/pi#
```

Creiamo il file che si occuperà della rotazione:

```
sudo vim /etc/logrotate.d/logdw
```

Copiamo al suo interno le seguenti righe rispettando i ritorni a capo e la posizione delle parentesi graffe, potete fare copia incolla, se state usando il terminale, potete incollarlo tenendo premuto contemporaneamente Ctrl+Maiuscolo e v:

```
/home/pi/logdirewolf {
    daily
    rotate 4
    missingok
    copytruncate
    notifempty
    create 0644 pi pi
}
```

Così facendo, se il vostro raspberry sarà acceso alle 6:45 di ogni mattino, il file di log verrà ruotato da logdirewolf a logdirewolf.1 e così via fino a quattro per poi essere cancellato.

Bene effettuiamo un reboot e verifichiamo che tutto funzioni:

STEP 1. verifichiamo che il processo direwol sia stato avviato automaticamente al riavvio:

```
root@direwolf:/home/pi# ps -ef |grep direwolf
avahi    297      1   0 Jun09 ?        00:00:08 avahi-daemon: running [direwolf.local]
root     423      1  11 Jun09 ?        02:37:23 /usr/local/bin/direwolf -t 0 -c /home/pi/direwolf.conf
root     5139   4602  0 22:49 pts/0    00:00:00 grep direwolf
```

STEP 2. apriamo il file di log con `tail -f /var/log/direwolf.log`

Related links

DIREWOLF GITHUB

<https://github.com/wb2os/direwolf>

Raspberry Pi OS (32-bit) Lite

https://downloads.raspberrypi.org/raspbian_lite_armhf_latest

Geoplaner V3.0

<https://www.geoplaner.com/>

APRS Passcode Generator

<http://apps.magicbug.co.uk/passcode/>

RTX CTE1800 - Raspberry Pi3 - USB AUDIO CARD



questa procedura guidata è stata creata prendendo spunto dal collega IT9FDP Biagio La Fauci
<http://it9fdp.blogspot.com/2017/09/direwolf-sperimentiamo-un-igate-aprs.html>