APRS – Direwolf - Xastir

Doug Kingston – KD7DK

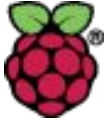## Dave Basham – AE7RD

SEAPAC

APRS – Direwolf - Xastir

- This session will consists of three parts
  - Overview of an **APRS** capability
  - **Direwolf** installation and configuration
  - **Xastir** installation and configuration and usage

Overview of APRS Capability

APRS
(Automated Packet Reporting System)

- Send and Receive "Packets"

- Packets may contain
  - Call Sign
  - Position information
  - Messages (similar to text message)
  - Weather Reports
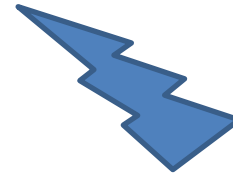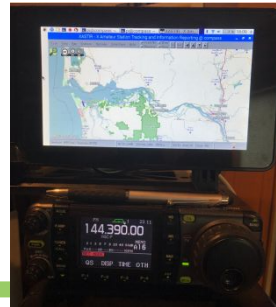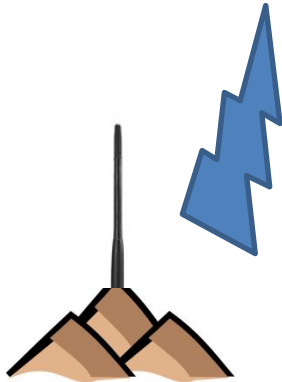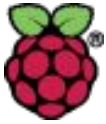 - Telemetry information

APRS.FI Website.

Internet

IGate

Cell Phone

**APRS Packets Containing
Position Reports, Messages, Status,
Weather Reports, Telemetry**

River Gauge

DigiPeater

APRS on a Yaesu VX-8DR

Many mobile radios when connected to a GPS can send packets with position information

STATION LIST   5/50
1 E W6QRZ - 7   10:03
2 W WB6QRZ   10:00
3 P W6QSP - 7   09:55

STATION LIST CHARACTER

DIRECTION TO THE RECEIVED STATION
STATION LIST CHARACTER
STATUS TEXT "FLAG"

APRS MESSAGE   6/30
1 RX W6QRZ - 7   10:03
2 RX WB6QRZ   10:00
3 RX W6QSP - 7   09:55

RX   W6QRZ - 7   12/31
MSG:11   10:03
Hello!...............
....

W6QRZ - 7
Mic-E   12/31
    3.2mi   10:03
Speed   32mph

RECEIVED DATE & TIME
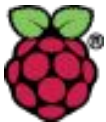**DISTANCE TO THE RECEIVED STATION**

APRS with a RPi and ICOM7000

Radios connected to a small computer or laptop and a GPS can send and receive APRS Packets and display stations on a map. If the computer has a keyboard, messages can be easily typed as well.
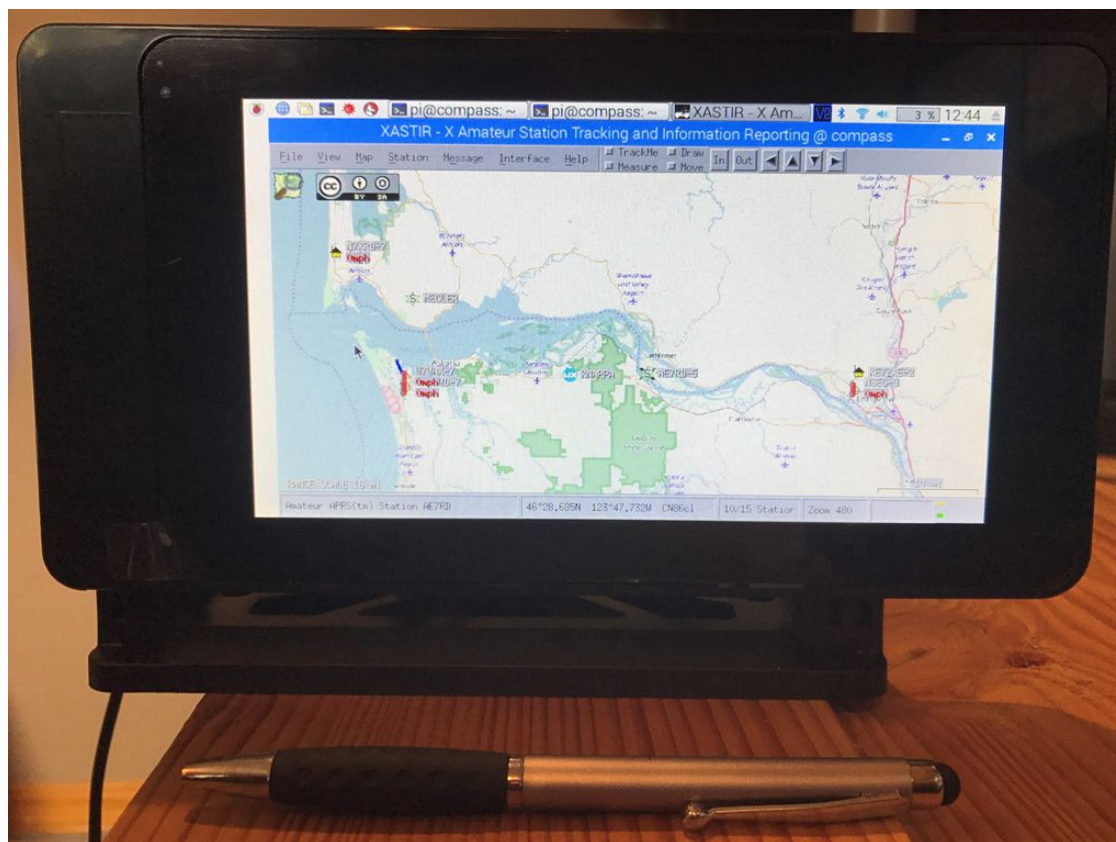

SEA PAC
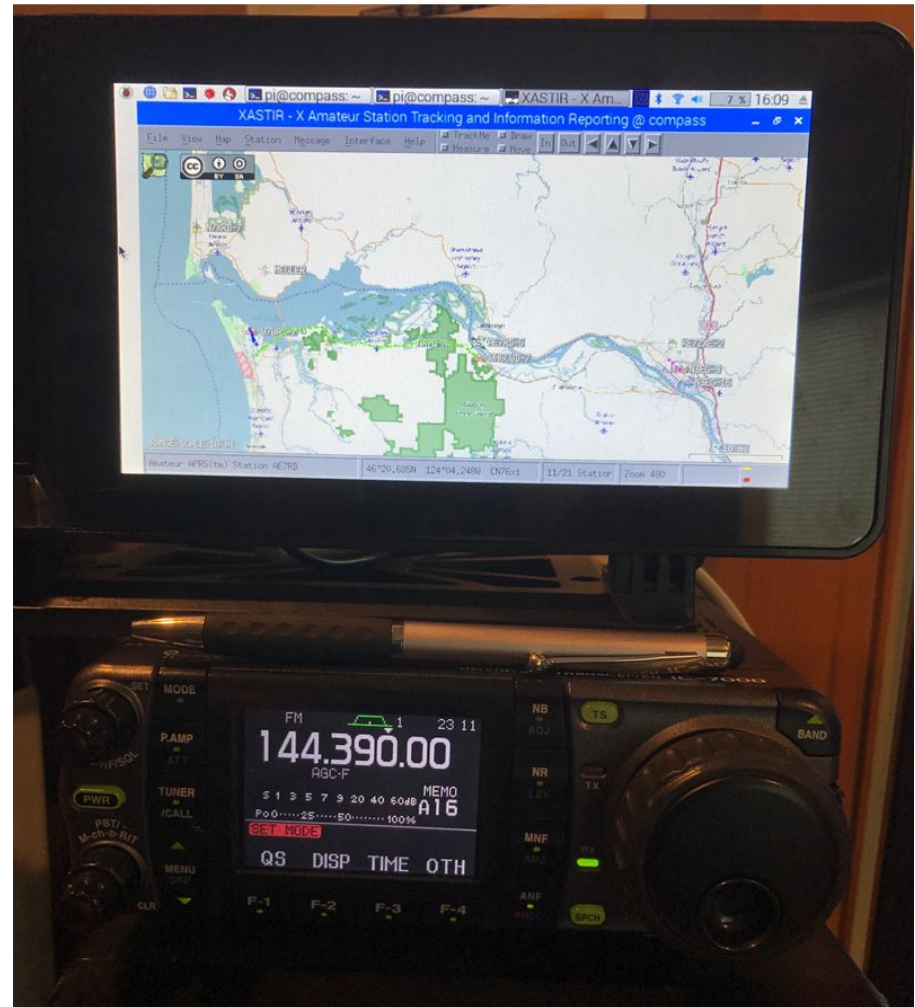
Raspberry Pi – Direwolf - Xastir

- Raspberry Pi – Compass Image

- UDRC II Sound Card

- Running Direwolf for TNC Functionality

- Running Xastir for APRS Display

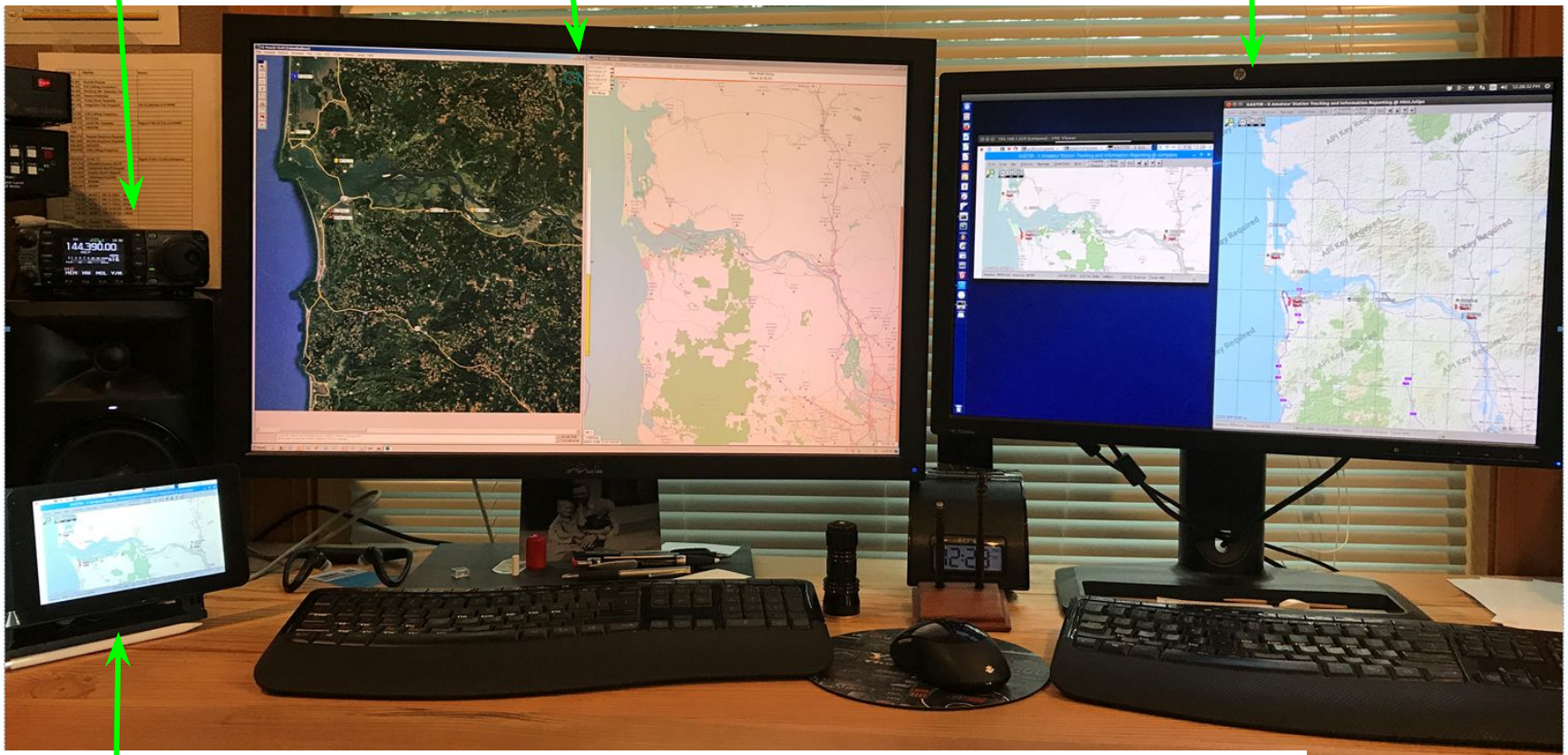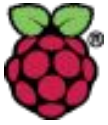- PI/Direwolf Can Also be a "Network Server" to other APRS applications on the network.

Windows 7
UI-View32 on Left
APRSIS32 on Right

Ubuntu Linux
VNC Viewer of PI Desktop On Left
Xastir on Right

ICOM 7000

Raspberry Pi – Running Direwolf and Xastir and providing APRS services
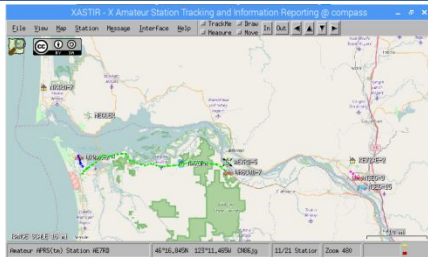via the network to both the Windows and Linux Machines

Xastir – Direwolf –Sound Card – Radio

Raspberry Pi Computer and Display

Direwolf – Decodes & Encodes digital audio from/to sound card and makes data available to external applications
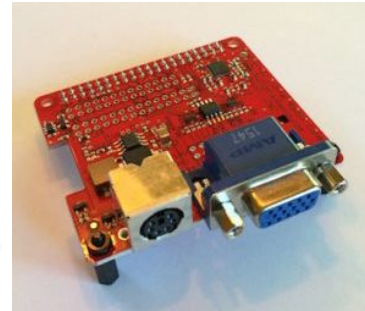
Network Port

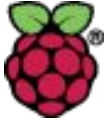XASTIR – Displays APRS Packet Information on Map

USB

Direct Connection

UDRC-II or Signalink Sound Card

6-Pin Cable

Radio

Direwolf
# Installation and Configuration

Goals of this Presentation

- Provide an overview of Direwolf and software TNCs
- Show how to do a manual install of direwolf
- Show how to start and stop direwolf when run as a daemon
- Show how to run direwolf from the command line
- Explain the configuration file format and a basic configuration
- Explain how to debug and test your direwolf configuration
- Demonstrate how to adjust the audio levels
- Show how Winlink Express on a PC can use your Pi as a TNC
- Provide references for additional information
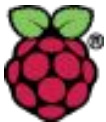
What is a TNC?

A terminal node controller (**TNC**) is a device used by amateur radio operators to participate in AX.25 packet radio networks. It includes Packet Assembler/Disassemblers and a modem to convert baseband digital signals to audio tones.
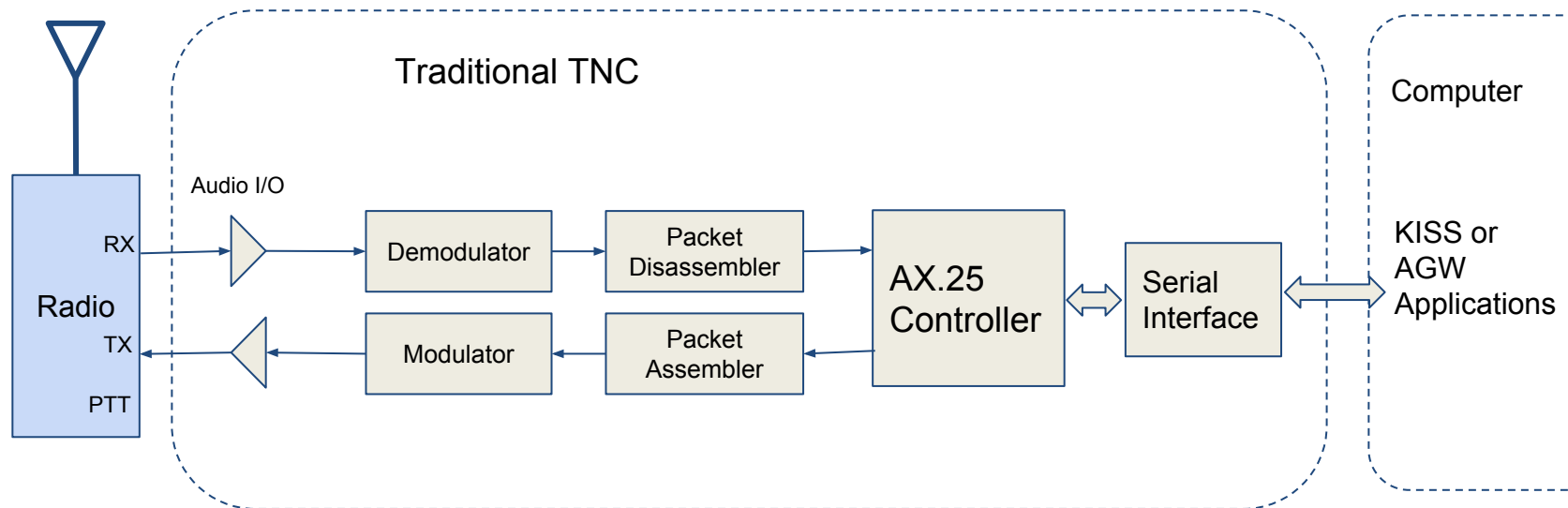
What is Direwolf?

direwolf is a software "soundcard" TNC and APRS encoder/decoder which:
- provides a virtual TNC for other applications*
- displays APRS messages, and generates APRS beacons
- acts as a digipeater, APRStt gateway, or Internet Gateway (IGate)
- supports KISS & AGWPE interfaces over TCP and virtual serial ports

* For example: Linux AX25 implementation, PacLink, APRSIS32, UI-View32, Xastir, APRS-TW, YAAC, UISS, SARTrack, Winlink Express, and others.

Direwolf - History and Status

Developed starting in 2011 by John Langner (WB2OSZ)

First major release in 2014. Current release 1.4 (April 2017).

Open source software, actively maintained, developed and documented

Ported to Windows, MacOS, and Linux (both x86 and ARM)

Recent changes:
- Added 2400 & 4800 bps PSK modems
- Top speed of increased from 9600 to 38400. (w/appropriate sound card)
- Better decoder performance for 9600+
- AX.25 v2.2 connected mode
- Improved GPS device support … and lots of bug fixes!
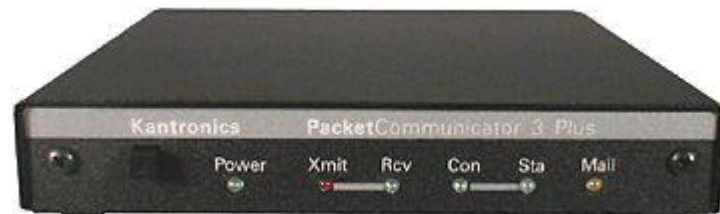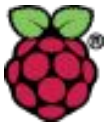
Direwolf - Resources

GitHub Project:

https://github.com/wb2osz/direwolf

Dire Wolf User Guide (143 pages of detailed documentation)

https://github.com/wb2osz/direwolf/blob/master/doc/User-Guide.pdf

SEA☰PAC

So this is our traditional TNC...

Traditional TNC

Computer

KISS or AGW Applications

Audio I/O

RX

Radio

TX

PTT

Demodulator → Packet Disassembler → AX.25 Controller ⇄ Serial Interface

Modulator ← Packet Assembler ← AX.25 Controller
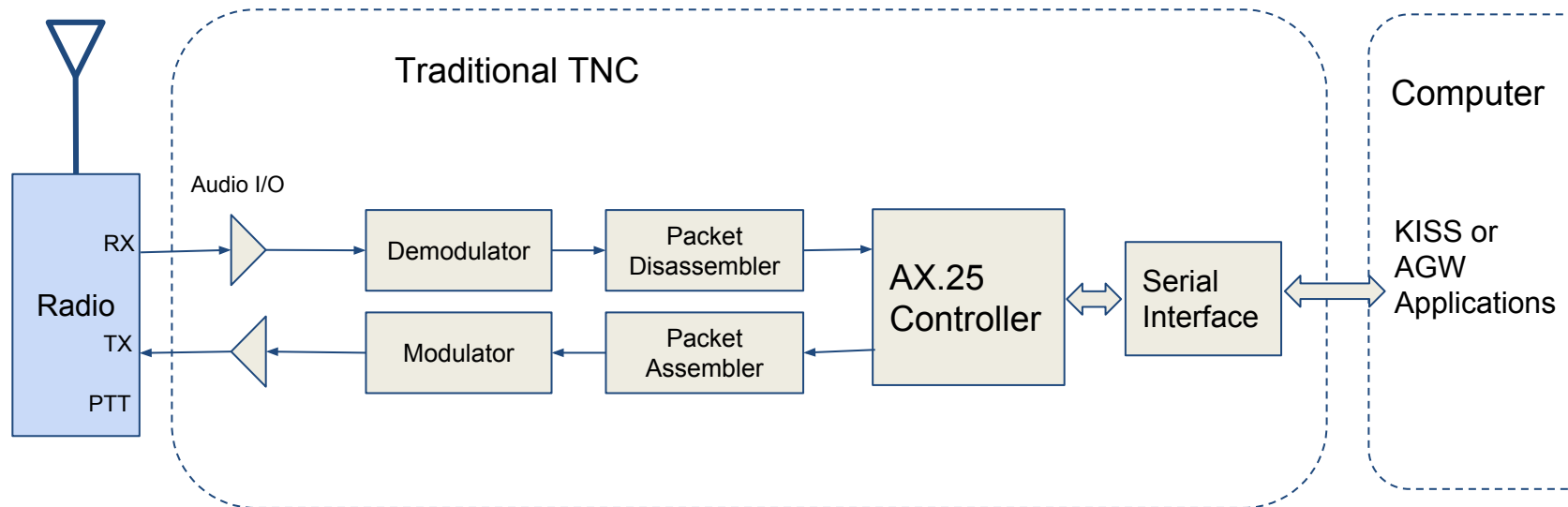
Direwolf System Overview

Most of the TNC moves into the computer and it uses either the standard or an auxiliary audio interface to communicate to the radio.
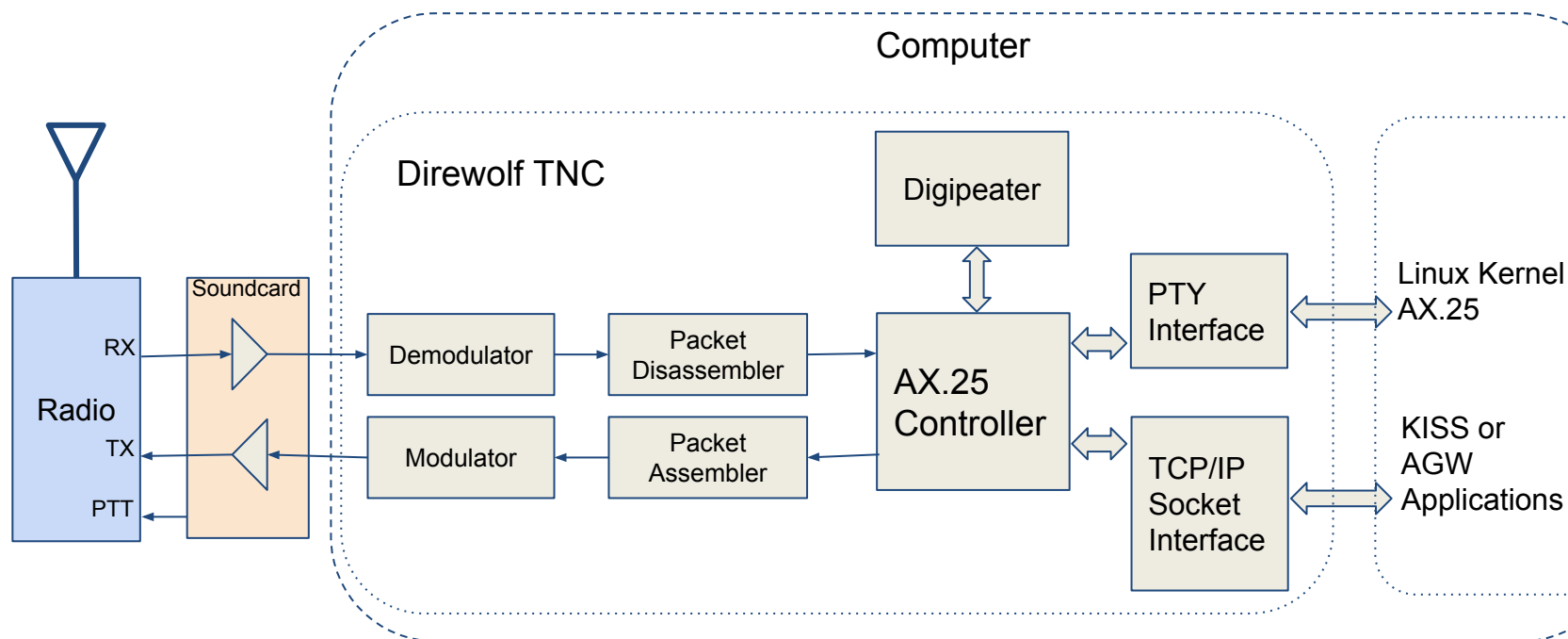
Installation (manual)

On Compass Linux:

```
cd
sudo apt-get install direwolf
nano direwolf.conf
direwolf -c direwolf.conf
```

Issues:
1) not a daemon under systemd (does not start automatically)
2) you need to create a configuration manually

There is a better way - Basil's scripts (n7nix/direwolf)

Installation (scripts with systemd support)

Basil's scripts automate most of the installation for Direwolf and much more.

https://github.com/nwdigitalradio/n7nix

- enable ax25 support in the linux kernel
- build, install and configure the ax25 software
- check if direwolf is installed, and if not, build it from source
- create /etc/direwolf.conf
- check for udrc-II and create and run configuration script if needed (set-udrc-din6.sh)
- configure systemd to start direwolf at system bootup

SEA PAC

Operation: Starting/Stopping with systemd

```
# service direwolf stop
# service direwolf status
  – direwolf.service - Direwolf Daemon
   Loaded: loaded (/etc/systemd/system/direwolf.service; enabled)
   Active: inactive (dead) since Sat 2017-05-13 17:56:49 PDT; 1min 22s ago
…
# service direwolf start
# service direwolf status
  – direwolf.service - Direwolf Daemon
   Loaded: loaded (/etc/systemd/system/direwolf.service; enabled)
   Active: active (running) since Sat 2017-05-13 18:00:43 PDT; 2s ago
 ...
```

Direwolf Configuration

The direwolf configuration lives in 2 places:

- direwolf.conf - edit **`/etc/direwolf.conf`**
  This controls all the modem and packet settings

- ALSA Sound System - configure with alsamixer or amixer
  This is how you control the sent and received audio levels

SEA◉PAC®

Determining sound card (UDRC example)

```
root@kd7dk-pi2:/home/pi# 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: udrc [udrc], device 0: Universal Digital Radio Controller
tlv320aic32x4-hifi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Default Pi Sound Interface

UDRC

SEA☻PAC

Determining sound card (SignalLink example)

```
root@kd7dk-pi2:/home/pi# 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: CODEC [USB Audio CODEC], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Default Pi Sound Interface

SignalLink

SEA*PAC

Configuration File Format (direwolf.conf)

```
# Comment text

KEYWORD VALUE [VALUE2 …]

ADEVICE plughw:CARD=udrc,DEV=0  plughw:CARD=udrc,DEV=0

ADEVICE1 plughw:1,0

IGSERVER noam.aprs2.net

IGLOGIN KD7DK 17634

PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=R
lat=47^40.19N long=122^24.14W
```

Optional output device

A Simple Config file (no Internet gateway)

ADEVICE plughw:CARD=udrc,DEV=0
ARATE 48000
ACHANNELS 2

#HD-15
CHANNEL 0
MYCALL KD7DK-6
MODEM 1200
PTT GPIO 12
#DIN-6
CHANNEL 1
MYCALL KD7DK-7
MODEM 1200
PTT GPIO 23

AGWPORT 8000
KISSPORT 8001

How do we handle Push-to-Talk?

There are 4 options:

1. Sound card handles PTT (e.g. SignalLink USB)
2. Use General Purpose I/O (GPIO) pins and potentially a relay/transistor (e.g. UDRC)
   `PTT GPIO 12`
3. Use USB serial port adapter and control RS-232 signal lines (DTR, RTS)
   `PTT /dev/ttyUSB0 RTS`
4. HamLib rig control library (open source)
   `PTT RIG 120 /dev/ttyUSB0`
   `PTT RIG 7 localhost:4532`

All of these option are covered in detail in the Dire Wolf User Guide section 9.2.8

A Simple Config file (adding Internet gateway)

IGSERVER noam.aprs2.net

IGLOGIN KD7DK 17634

```
# 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

# Stationary position beacon
PBEACON sendto=IG delay=0:30 every=60:00 symbol="igate" overlay=R
lat=47^40.19N long=122^24.14W
```

Digipeating

Format:

`DIGIPEAT` *from-chan* *to-chan* *aliases* *wide* *preemptive*

A good default:

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

See the Dire Wolf User Guide section 9.5 for a detailed explanation.

Configuration - Sound with amixer

Basil's script sets up this initialization file:
`/usr/local/src/udrc/set-udrc-din6.sh`

Here's a simple shell script that sets 3 key parameters:

```bash
#!/bin/bash

amixer -c udrc -s << EOF
#  Set input and output levels to 0dB
sset 'ADC Level' -2.0dB
sset 'LO Driver Gain' 0dB
sset 'PCM' 0.0dB
EOF
```
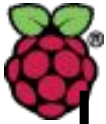
UDRC-II   AIC3204 Block Diagram
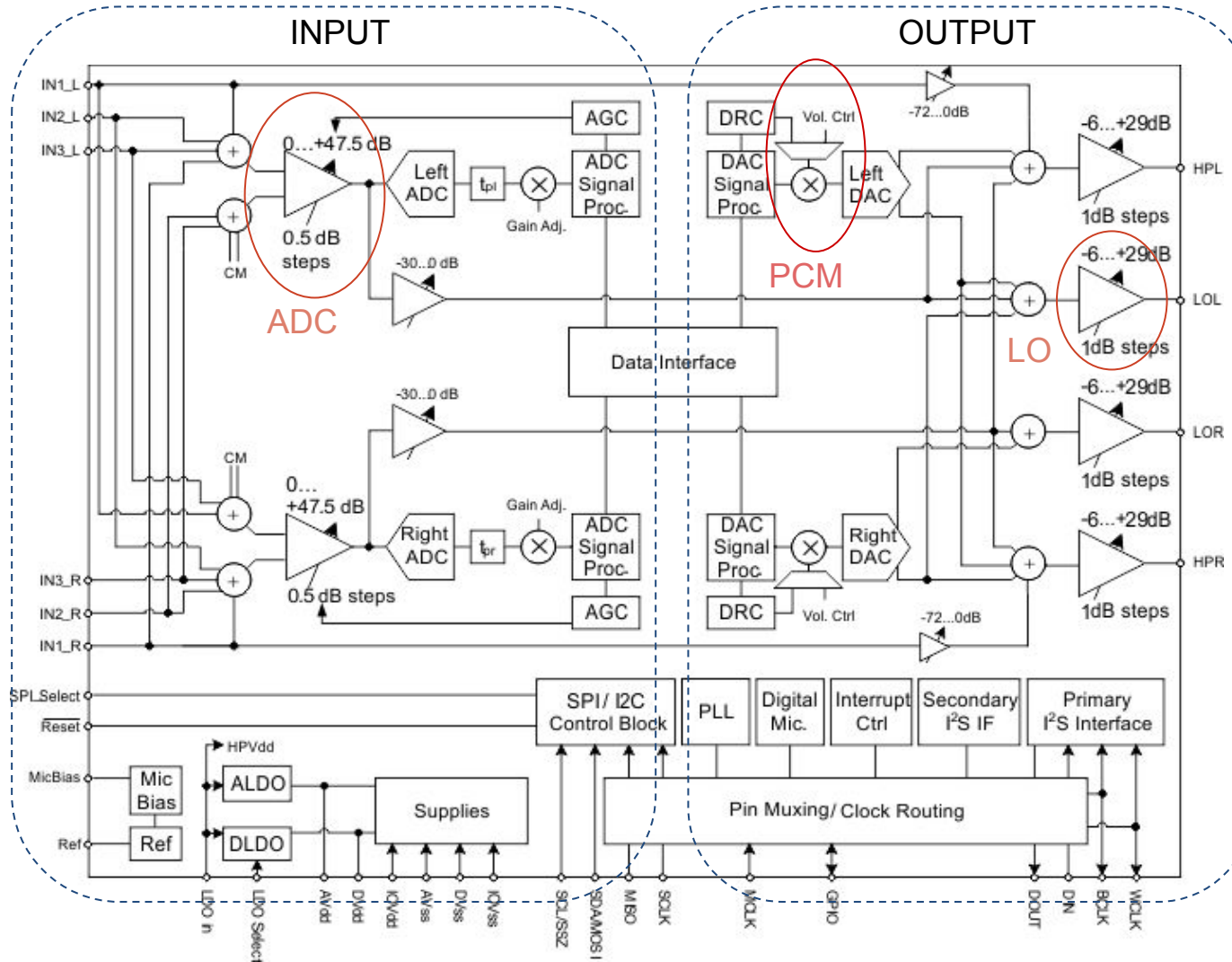
Configuration - Sound with Alsamixer

PCM - Transmit audio (fine)
ADC - Receive audio
LO - Transmit audio

Configuration - Sound with Alsamixer

PCM - Transmit audio (fine)
ADC - Receive audio
LO - Transmit audio

Operation: Diagnostic Output

```
pi@kd7dk-pi2:/var/log/direwolf $ more /var/log/direwolf/direwolf.log
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Dire Wolf version 1.3
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Reading config file /etc/direwolf.conf
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Audio device for both receive and transmit:
plughw:CARD=udrc,DEV=0  (channels 0 & 1)
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100
sample rate / 3.
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Channel 1: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100
sample rate / 3.
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Ready to accept AGW client application 0 on port 8000
...
Apr  6 18:45:10 kd7dk-pi2 direwolf[1440]: Ready to accept KISS client application on port 8001 ...
```

SEA⬤PAC

Operation: Diagnostic Output (simplified)

```
Dire Wolf version 1.3
Reading config file /etc/direwolf.conf
Audio device for both receive and transmit: plughw:CARD=udrc,DEV=0 (channels 0 & 1)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate / 3.
Channel 1: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate / 3.
Ready to accept AGW client application 0 on port 8000 ...
Ready to accept KISS client application on port 8001 ...
Virtual KISS TNC is available on /dev/pts/0
WARNING - Dire Wolf will hang eventually if nothing is reading from it.
Created symlink /tmp/kisstnc -> /dev/pts/0
KISS protocol set TXDELAY = 50 (*10mS units = 500 mS), port 1
KISS protocol set TXtail = 10 (*10mS units = 100 mS), port 1
KISS protocol set Persistence = 32, port 1
KISS protocol set SlotTime = 20 (*10mS units = 200 mS), port 1
KISS protocol set FullDuplex = 0, port 1
```

Testing your Direwolf Installation (Receive)

1. Check the UDRC is available: `amixer -c udrc`
2. Attach a radio tuned to APRS frequency (144.39 MHz)
3. Listen via speaker to confirm traffic, then attach to UDRC
4. Check for packet in Direwolf logs:
   `tail -f /var/log/direwolf/direwolf.log`
5. Check receive audio:
   `audio level = 53(20/20) [NONE]    ||||||___`
   Format: level(mark/space) displayRetries   spectrum
6. Adjust audio receive level (ADC for UDRC)

Testing your Direwolf Installation (Transmit)

There are three options:

1. For UDRC, generate a test tone with (instructions at top of file)
   `n7nix/deviation/measure_deviate.sh`
2. Others can generate test tones with direwolf
   `direwolf -x -c /etc/direwolf.conf`
3. Listen to yourself generate packets while listening on another radio and tailing log file

Measure with a service monitor or compare to other signals.

Target deviation is 3-3.5 KHz.

Using Winlink Express with your Raspberry Pi TNC

Direwolf is listening for KISS connections on TCP port 8001
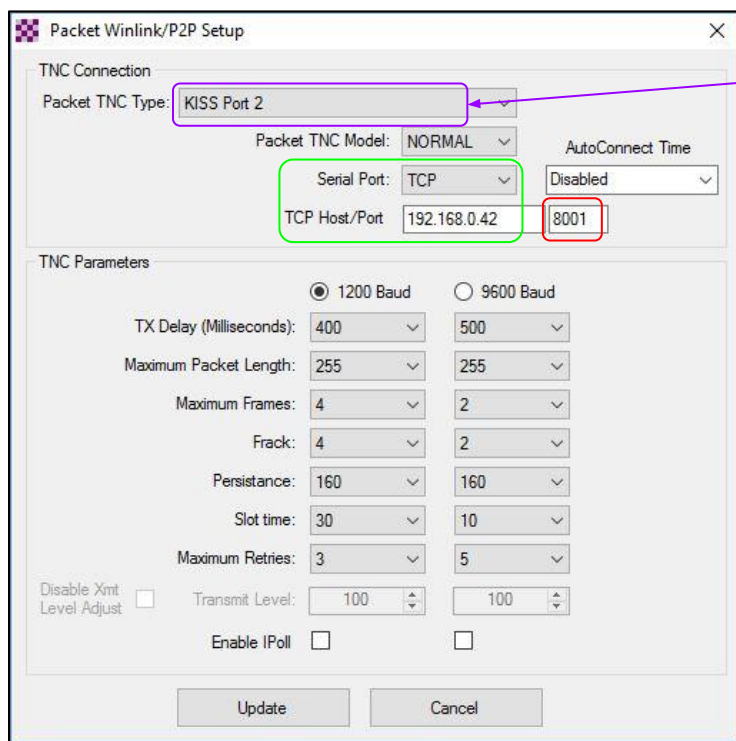
Could be KISS or KISS Port 2

Radio

Wired or WiFi LAN

SEA◈PAC
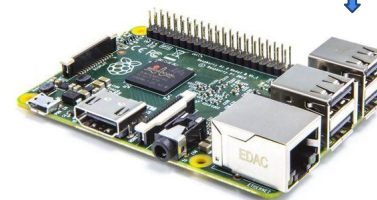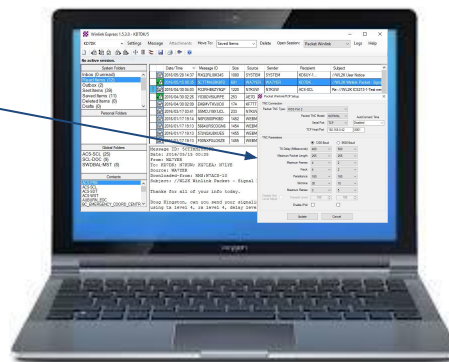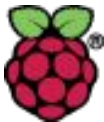
The Wireless TNC configuration

The Raspberry Pi can be a wireless client or a wireless access point.

If you are away from network infrastructure, configure your Pi as an access point and let your laptop and other devices connect to the Pi.
https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/

To login to your Pi over the network, use ssh connection with putty:
https://www.chiark.greenend.org.uk/~sgtatham/putty/

Pi & UDRC Resources

https://nw-digital-radio.groups.io/g/udrc/wiki/home
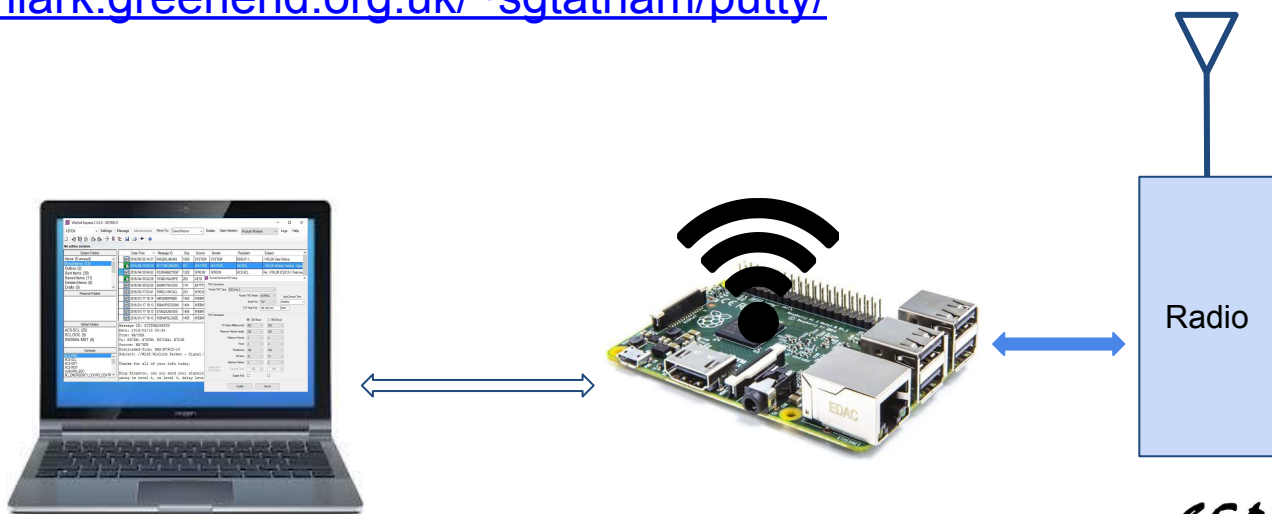
- Configuring ircDDBgateway
- Creating Virtual Sound Cards for the UDRC™
- Generate a Tone to Set Radio Deviation via the UDRC™
- Notes for the Kenwood TKR 750 850 Repeater
- Powering the UDRC™ and Raspberry Pi with 12 VDC
- PTT Keying Application Note UDRC II
- UDRC™ and Direwolf Packet Modem
- UDRC™ and fldigi Setup Page
- UDRC™ and Raspberry Pi 3 Addendum
- UDRC™ For Simplex Hotspots and Converted Analog Repeaters
- UDRC™ II Cabling For 2 radios
- UDRC™ Setup for the Yaesu DR 1X Repeater
- UDRC™ With Bridgecom Systems Repeaters

I found this flakey

Really useful!

SEA-PAC

Operation: Diagnostic Output Colors

`Black for information`

`Dark Green for the audio level`

`Green for received raw data`

`Blue for decoded version of raw data`

`Magenta for transmitted data`

`Red for errors`

Manual invocation:

`/usr/local/bin/direwolf -c /etc/direwolf.conf`

Direwolf manual startup with enhanced diagnostics


```
root@kd7dk-pi2:/home/pi# /usr/local/bin/direwolf -c /etc/direwolf.conf
Dire Wolf version 1.3

Reading config file /etc/direwolf.conf
Audio device for both receive and transmit: plughw:CARD=udrc,DEV=0  (channels 0 & 1)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate / 3.
Channel 1: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate / 3.
Ready to accept AGW client application 0 on port 8000 ...
Ready to accept KISS client application on port 8001 ...
Use -p command line option to enable KISS pseudo terminal.

Digipeater WIDE2 (probably ERINB) audio level = 60(22/22)   [NONE]   _|||||___
[1.3] AC7KP-1>T6UTXT,W7PFR-1,WIDE1,ERINB,WIDE2*: 21AL ;>/ 5c}+1%m rl!wN:!l3
MIC-E, normal car (side view), Byonics TinyTrack3, In Service
N 46 54.8449, W 122 21.3727, 0 MPH, course 231, alt 548 ft
Seq=926, A1=440, A2=627

Now connected to IGate server noam.aprs2.net (66.109.111.18)
Check server status here http://66.109.111.18:14501


Digipeater WIDE2 (probably ERINB) audio level = 55(21/21)   [NONE]   ||||||||__
[1.3] WA7VC-10>APRX28,SONTN,WIDE1,ERINB,WIDE2*:!4727.90N/12140.74W;Fill-in iGate & TX-Digi
Position, Portable operation (tent), >40 APRSmax
N 47 27.9000, W 121 40.7400
Fill-in iGate & TX-Digi
^C
QRT
root@kd7dk-pi2:/home/pi#
```

Detected hardware and config settings

Good, balanced audio levels!

SEA PAC

RTL-SDR Receive-only Radio Input to Direwolf

1. Install the RTL Software
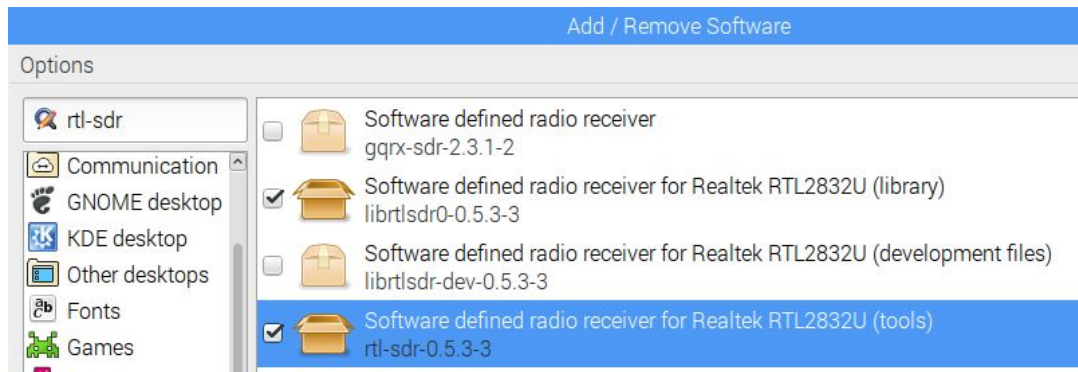
2.   From the command line type the following to invoke the rtl_fm application and pipe the data to direwolf:

```
rtl_fm -f 144.39M -o 4 - | direwolf -n 1 -r 24000 -b 16 -
```

Xastir

Xastir

- APRS Client Application

- Provides mapping, tracking, messaging, weather, weather alerts, and Search & Rescue features over radio or internet.

- Can be configured for Igate and digipeater functionality

- Can take GPS input for location
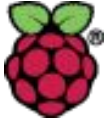
Xastir Installation

- Should already be installed on your Compass Image
- Invoked from a terminal window (click on the terminal icon on desktop upper right, then type xastir and return). Xastir will run and the terminal window will remain in the background.
- If not using the compass image, install as follows: sudo apt-get install xastir

SEA PAC
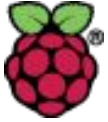
Xastir Configuration

- Connections
  - GPS
  - TNC
- Define Station Attributes
  - Location
  - Power output, Antenna Height/Gain
  - Timing
- Define what maps you would like to see
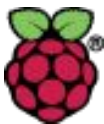
Xastir Screen

XASTIR - X Amateur Station Tracking and Information Reporting @ compass

File   View   Map   Station   Message   Interface   Help

TrackMe   Draw   In   Out
Measure   Move

N7DRD-4

N7XRD-7

MEGLER

W7WAX-7
0mph

KNAPPA   AE7RD-5   KE7ZKE-2

N7DRD-12
30mph

N3EG-9

RANGE SCALE 16 mi

Amateur APRS(tm) Station AE7RD | 46°21.845N   124°04.248W   CN76xi | 11/19 Station | Zoom 480

File/Configure/Station Page 1

File/Configure/Station (Page 2)

Adding an Interface (Page 1)
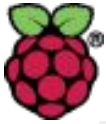
Adding and Interface (Page 2)

Configure AGWPE (Connection to Direwolf)

Adding GPS Interface

GPS Installation and Configuration

From NWDIGITALRADIO.COM web site:
https://nw-digital-radio.groups.io/g/udrc/wiki/UDRC%E2%84%A2-and-Direwolf-Packet-Modem

Plug the USB GPS into one of the 4 USB ports on your Raspberry Pi. If it is the only serial device it will typically be a **/dev/ttyUSB0**. If you have other USB serial devices, it may be at some other **/dev/ttyUSBx** (where x is a digit).

```
sudo apt-get install gpsd gpsd-clients
```

Note: Compass Image has taken care of all this. No need to do anything but refer to this chart and the next chart if you're installing on another image.

SEA PAC

GPS Installation and Configuration

Configure gpsd by editing **/etc/default/gpsd** –
Note: You need to use **sudo** nano /etc/default/gpsd to edit the file (need privileges)
For **/dev/ttyUSB0** the file will contain:

```
# Default settings for the gpsd init script and the hotplug wrapper.
# Start the gpsd daemon automatically at boot time
START_DAEMON="true"
# Use USB hotplugging to add new USB devices automatically to the daemon
USBAUTO="true"
# Devices gpsd should collect to at boot time.
# They need to be read/writeable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyUSB0"
# Other options you want to pass to gpsd
GPSD_OPTIONS=""
```

Then start gpsd with the following command:

```
sudo systemctl start gpsd
```

Finally, enable the service so it will start automatically:

```
sudo systemctl enable gpsd
```

Station Menu

Map/Map Chooser Menu

File/Configure/Timing

File View Map Station Message Interface Help
TrackMe Draw Measure Move In Out

Configure Timing

30.0
Posit TX Interval (min)

80
Station Ghosting Time (min)

30
Object/Item Max TX Interval (min)

12
Station Clear Time (hours)

60
GPS Check Interval (sec)

1
Station Delete Time (days)

10
Dead-Reckoning Timeout (min)

1
Serial Inter-Char Delay (ms)

45
New Track Time (min)

1
New Track Interval (degrees)

0
RINO -> Objects Interval (min), 0 = Disabled

120
Internet Map Timeout (sec)

5
Snapshot Interval (min)

OK   Cancel
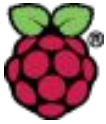
File/Configure/Defaults

🍓 File/Configure/Smart Beaconing

Help/Help Index

Xastir Scripts

Xastir installs with an assortment of useful scripts located here:

    /usr/share/xastir/scripts

See "Help/Help Index/Included Scripts" for more info

One useful script will download and install the FCC database of amateur radio operators which can be searched when clicking on a station. To download and install the FCC Database:

    cd /usr/share/xastir/scripts
    sudo ./get-fcc-rac.pl

Searching the FCC Database for Call Sign Information

Right click on a station on the map, pick "Station Info" then click "Search FCC Database"

Note: This function will be grayed out if you have not installed the database.

Questions?

For more information go to:
    xastir.org