

Raspberry Pi APRS

Version 1.0 -- May 2014

The Raspberry Pi (RPI) is a good platform for running a Linux APRS TNC, digipeater, and IGate.

This document is a Quick Start guide for running Dire Wolf on the Raspberry Pi and describes special considerations where it may differ from other Linux systems.

After completing the steps here, refer to the User Guide for more details on the Linux version.

Select Appropriate Operating System

Select a version of Linux that has hardware floating point support. Dire Wolf makes extensive use of floating point calculations. Trying to use the slower software floating point will probably result in disappointment.

The Raspbian “wheezy” distribution from <http://www.raspberrypi.org/downloads> is suitable.

This operating system distribution comes with the **gcc** compiler and most required libraries pre-installed. If you use a different operating system version, you might need to install a suitable compiler and/or additional libraries.

Verify that gcc is configured to generate hardware floating point code. Enter the “gcc -v” command and observe the result. Make sure that “--with-fpu=vfp --with-float=hard” appears in the configuration.

```
pi@raspberrypi:~$ gcc -v

Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/arm-linux-gnueabi/4.6/lto-wrapper
Target: arm-linux-gnueabi
Configured with: ../src/configure -v --with-pkgversion='Debian 4.6.3-12+rpi1' -
--with-bugurl=file:///usr/share/doc/gcc-4.6/README.Bugs --enable-
languages=c,c++,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.6 --
enable-shared --enable-linker-build-id --with-system-zlib --libexecdir=/usr/lib
--without-included-gettext --enable-threads=posix --with-gxx-include-
dir=/usr/include/c++/4.6 --libdir=/usr/lib --enable-nls --with-sysroot=/ --
enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --
enable-gnu-unique-object --enable-plugin --enable-objc-gc --disable-sjlj-
exceptions --with-arch=armv6 --with-fpu=vfp --with-float=hard --enable-
checking=release --build=arm-linux-gnueabi --host=arm-linux-gnueabi --
target=arm-linux-gnueabi
Thread model: posix
gcc version 4.6.3 (Debian 4.6.3-12+rpi1)
```

System Setup & Configuration

The lessons here: <http://learn.adafruit.com/> are easy to follow and will get you running quickly.

The audio system on the Raspberry Pi has a history of many problems. Every time I did a software update, the behavior changed. Before October 13, 2013, the only way I could get it to work was with pulseaudio. After the most recent software update and firmware upgrade, pulseaudio stopped working (http://elinux.org/R-Pi_Troubleshooting#Removal_of_installed_pulseaudio) but the preferred method, which never worked properly before, is now fine.

You might be dooming your efforts to failure if you skip this software & firmware update step.

```
sudo apt-get install libasound2-dev
sudo apt-get update
sudo apt-get dist-upgrade
sudo rpi-update
sudo reboot
```

Check the firmware version with this command. I'm getting good results with the Oct 12 2013 version. Earlier versions were not satisfactory.

```
/opt/vc/bin/vcgencmd version
```

Pulseaudio was never right and it's worse now, so remove it just to be safe. I believe it is no longer included in the current version of Raspian.

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

At this point reboot to make sure you are running the latest versions.

```
sudo reboot
```

You might find it convenient to access the desktop remotely. In my case, the Raspberry Pi is in my unheated "shack" with the radio equipment. By installing VNC, I'm able to access the desktop from a more comfortable location. Here is how to install the VNC server.

<http://learn.adafruit.com/adafruit-raspberry-pi-lesson-7-remote-control-with-vnc/installing-vnc>

Optionally enable IPv6.

You can enable IPv6 immediately by typing: “sudo modprobe ipv6”.

To start it up automatically after each reboot, edit /etc/modules and add a new line at the end containing “ipv6” (without the quotes).

Audio interface for radio

APRS, or other packet radio, operation requires connections between your transceiver and computer.

- (a) Received audio from receiver.
- (b) Transmit audio to transmitter.
- (c) Push to Talk (PTT) signal to activate transmitter.

The Raspberry Pi has built-in audio output but no audio input.

You can get a list of audio **output** devices with the “aplay -l” command.

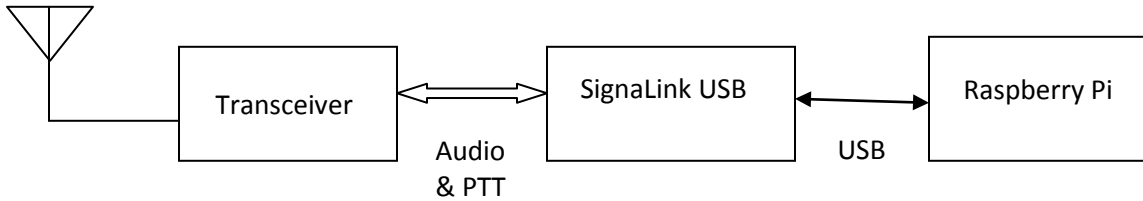
```
pi@raspberrypi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
    Subdevice #4: subdevice #4
    Subdevice #5: subdevice #5
    Subdevice #6: subdevice #6
    Subdevice #7: subdevice #7
```

You can get a list of audio **input** devices with the “arecord -l” command:

```
pi@raspberrypi:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
```

There aren't any!

It will be necessary to add some sort of sound input device. The **Tigertronics SignalLink USB** provides an ideal solution.



Some cheap USB audio adapters can also be used. Others have resulted in only frustration.

I've read where Raspberry Pi users reported good results with the CMedia CM119A or CM108AH chip.

I'm currently using this <http://www.amazon.com/gp/product/B001MSS6CS> Syba adapter successfully. The sticker on the package indicates it has a CMedia HS100 chip which is in the same family as the others mentioned.

Reference: <http://www.cmedia.com.tw/ApplicationDetail/C1Serno-1/C2Serno-2/C3Serno-4.html>

Here is a list of others reported to work with the Raspberry Pi:

http://elinux.org/RPi_VerifiedPeripherals#USB_Sound_Cards

After plugging in a suitable USB audio interface, you should see something like this, in response to the "aplay -l" and "arecord -l" commands:

```

pi@raspberrypi:~$ 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 1: CODEC [USB Audio CODEC], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
  
```

```

pi@raspberrypi:~$ arecord -l

**** List of CAPTURE Hardware Devices ****
  
```



```

card 1: CODEC [USB Audio CODEC], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
  
```

The built-in (output only) interface should be card 0. The USB audio device should be card 1. If you see a different number, you will have to modify a later step when we put the card number in the configuration file.

Transmitter PTT

If you want to transmit, some method is needed to activate the transmitter push-to-talk (PTT) function. Traditionally, this has been done with the RTS or DTR signal from a serial port.

Possible alternatives include:

- Reconfigure the built in serial port to have the RTS control.

The Raspberry Pi has a built in serial port which is normally configured as a console. It is possible to reconfigure this to be available for application use. In the default configuration, it does not have the usual RTS/CTS flow control signals. As described here, http://elinux.org/RPi_Serial_Connection#Handshaking_lines, it is possible to reconfigure some of the GPIO lines for the RTS/CTS signals. I haven't tried this.

- Use a USB to RS-232 converter cable to supply a full RS-232 port. You will need to put something like this in the configuration file.

```
PTT /dev/ttyUSB0 RTS
```

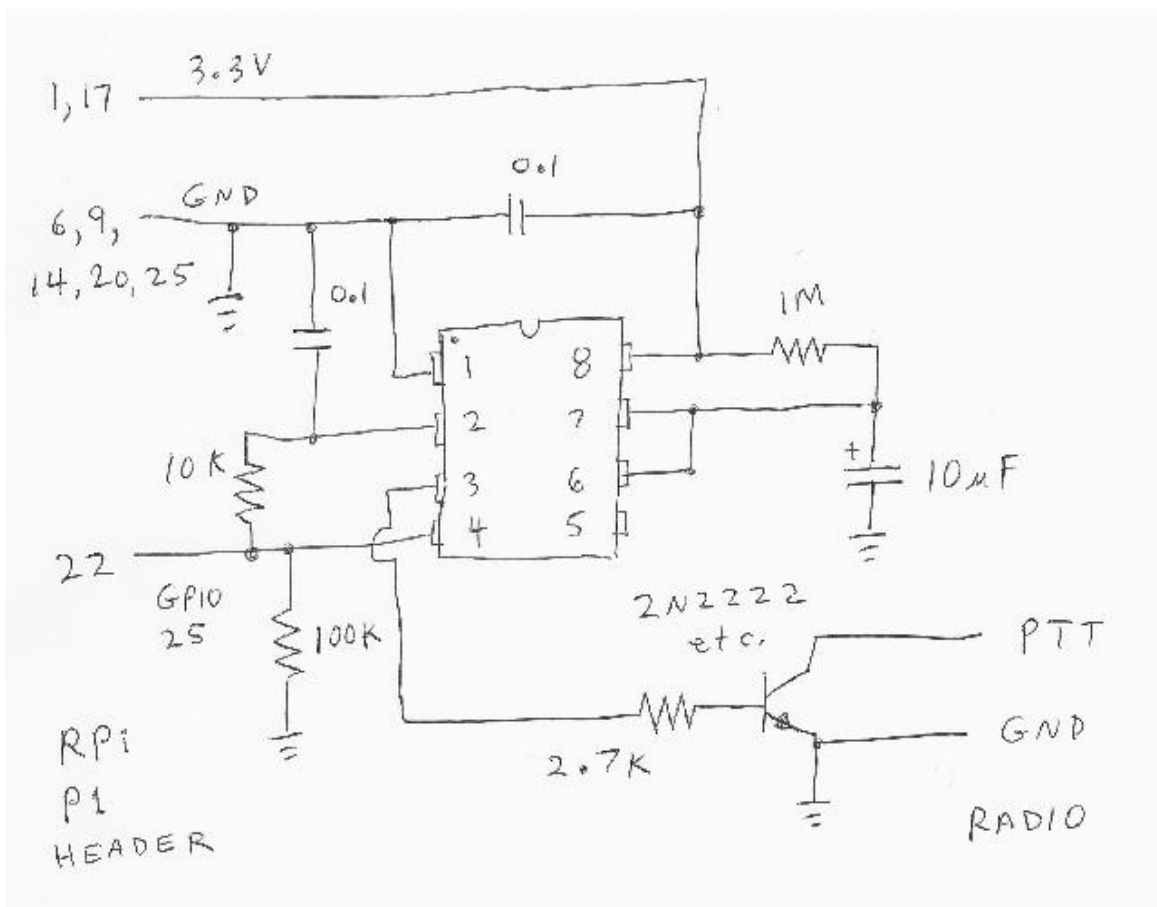
- VOX operation where transmitter is activated by transmit audio signal. The Signalink USB uses this technique. (Homebrew circuit example: <https://sites.google.com/site/kh6tyinterface/>)
- Use one of the General Purpose I/O (GPIO) pins. I'm currently using this along with the cheap USB audio adapter.

CAUTION! The general purpose input output (GPIO) pins are connected directly to the CPU chip. There is no buffering or other protection. The interface uses 3.3 volts and will not tolerate 5 volt signals. Static discharge, from careless handling, could destroy your Raspberry Pi.

These are my suggestions for the best choices. The others have special functions such as UART, SPI, PWM, or I²C. A few others changed positions between board revisions 1 & 2 which could cause confusion.

- P1-11 GPIO 17
- P1-15 GPIO 22
- P1-16 GPIO 23
- P1-18 GPIO 24
- P1-22 GPIO 25

Here is a recommended circuit using a CMOS 555 timer (LMC555, TLC555, ICM7555, etc.) to limit transmissions to about 10 seconds. Don't try using the original 555 because it needs a minimum of 4.5 volts and we have only 3.3 here. The time can be increased by making the 10 μ F capacitor larger. It's roughly 1 second for each μ F.



You could get by with only a resistor and transistor but a software failure could cause the transmitter to be stuck on, annoying other people, and possibly damaging the transmitter from overheating.

Finally, put a command like this in the configuration file with the actual I/O signal being used.

```
PTT GPIO 25
```

A different style of PTT interface might require the signal to be inverted. In this case, precede the pin number with "-".

Install Dire Wolf

First, install the "libasound2-dev" package with the following command:

```
sudo apt-get install libasound2-dev
```

Failure to install libasound2-dev step will result in a compile error resembling "audio.c:....: fatal error: alsa/asoundlib.h: No such file or directory"

Download the source version from <http://home.comcast.net/~wb2osz/site/> and copy it to your Raspberry Pi.

The file should be named something like direwolf-1.0-src.zip. /home/pi is the suggested location. If you put it somewhere else, you will need to make suitable adjustments to the commands shown.

Unzip it and build:

```
cd /home/pi
unzip direwolf-1.0-src.zip
cd direwolf-1.0
make -f Makefile.linux
make -f Makefile.linux install
```

You should now have files in these locations:

/usr/local/bin/direwolf	The application.
/usr/local/bin/decode_aprs	Utility to interpret "raw" data you might find on http://aprs.fi or http://findu.com
/usr/local/bin/tt2text /usr/local/bin/text2tt /usr/local/bin/ll2utm /usr/local/bin/utm2ll	Utilities related to APRStt gateway.
/usr/share/applications/direwolf.desktop	Application definition with icon, command to execute, etc.
/home/pi/Desktop/direwolf.desktop	Symbolic link to above. This causes an icon to be displayed on the desktop.
/usr/local/share/direwolf/tocalls.txt	Mapping from destination address to system type. Search order for tocalls.txt is first the current working directory and then /usr/share/direwolf. You might want to get a newer copy from: http://www.aprs.org/aprs11/tocalls.txt
/usr/local/share/direwolf/symbolsX.txt /usr/local/share/direwolf/symbols-new.txt	Descriptions and codes for APRS symbols.
/usr/share/direwolf/dw-icon.png	Icon for the desktop.
/usr/local/share/doc/direwolf/*	Various documentation.
/home/pi/direwolf.conf	Configuration file. Search order is current working directory then the user's home directory.
/home/pi/dw-start.sh	Script to start Dire Wolf if it is not running already.

Select Desired Audio Device

In a previous version, I recommended using pulseaudio and provided a script to change the default device. This stopped working with the latest Raspbian update. I now recommend using the plughw device, same as with other versions of Linux. Put this in the direwolf.conf configuration file.

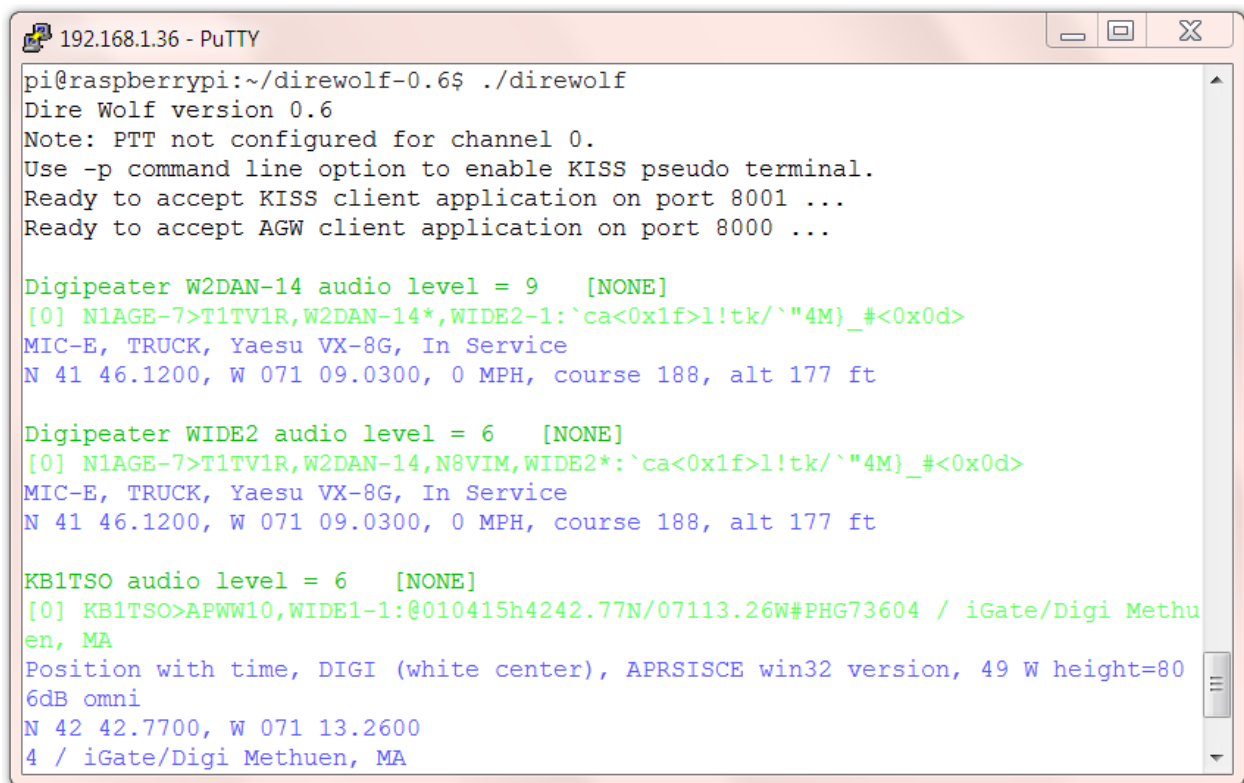
```
ADEVICE plughw:1,0
```

Run Dire Wolf application

Start up the application with the desktop icon or from the command line:

```
direwolf
```

You should see something resembling this:



```
192.168.136 - PuTTY
pi@raspberrypi:~/direwolf-0.6$ ./direwolf
Dire Wolf version 0.6
Note: PTT not configured for channel 0.
Use -p command line option to enable KISS pseudo terminal.
Ready to accept KISS client application on port 8001 ...
Ready to accept AGW client application on port 8000 ...

Digipeater W2DAN-14 audio level = 9    [NONE]
[0] N1AGE-7>T1TV1R,W2DAN-14*,WIDE2-1:`ca<0x1f>!tk/`"4M}_#<0x0d>
MIC-E, TRUCK, Yaesu VX-8G, In Service
N 41 46.1200, W 071 09.0300, 0 MPH, course 188, alt 177 ft

Digipeater WIDE2 audio level = 6    [NONE]
[0] N1AGE-7>T1TV1R,W2DAN-14,N8VIM,WIDE2*:`ca<0x1f>!tk/`"4M}_#<0x0d>
MIC-E, TRUCK, Yaesu VX-8G, In Service
N 41 46.1200, W 071 09.0300, 0 MPH, course 188, alt 177 ft

KB1TSO audio level = 6    [NONE]
[0] KB1TSO>APWW10,WIDE1-1:@010415h4242.77N/07113.26W#PHG73604 / iGate/Digi Methu
en, MA
Position with time, DIGI (white center), APRSISCE win32 version, 49 W height=80
6dB omni
N 42 42.7700, W 071 13.2600
4 / iGate/Digi Methuen, MA
```

The first line reveals the version number.

Next there is a reminder that push-to-talk (PTT) has not been configured. In this case, the interface automatically supplies PTT when transmit audio is present so we don't need a separate signal.

The KISS protocol is available thru a fake serial port (“pseudo terminal”) for older applications that only know how to talk to a TNC thru a serial port. This feature is turned off by default and can be activated by using the “-p” option on the command line.

The final part of the greeting contains the network ports for the KISS and AGW protocols for use by client applications. You will need to know these numbers for the next step. If you don’t like these default values, they can be changed in a configuration file.

The audio input level can be set with **alsamixer** or other audio configuration application available for your system.

Automatic Startup

You probably want your TNC / application server / digipeater to start up automatically after a power interruption.

If you followed the installation steps above, you should have a file named dw-start.sh in your home directory. Run the “crontab -e” command and add a line like this:

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

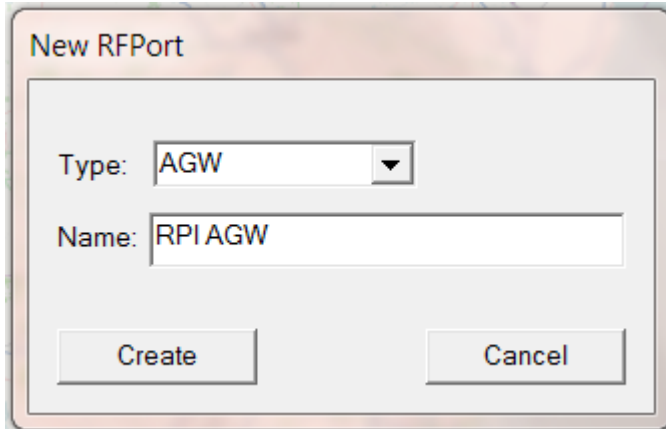
This script will run once per minute. Dire Wolf is started automatically if not running already. If it crashes, or is terminated for any other reason, it will be restarted. A log of restarts can be found in /tmp/dw-start.log.

Use with client applications

Client applications can run on different computers and communicate with Dire Wolf over your local network. For example, you might have a Raspberry Pi in your “shack” in a basement connected to your radio equipment. You might want to use a client application (such as APRSIS32, YAAC, or Xastir) on a laptop in a more comfortable location. Here is an example of how you could configure APRSIS32 in this situation.

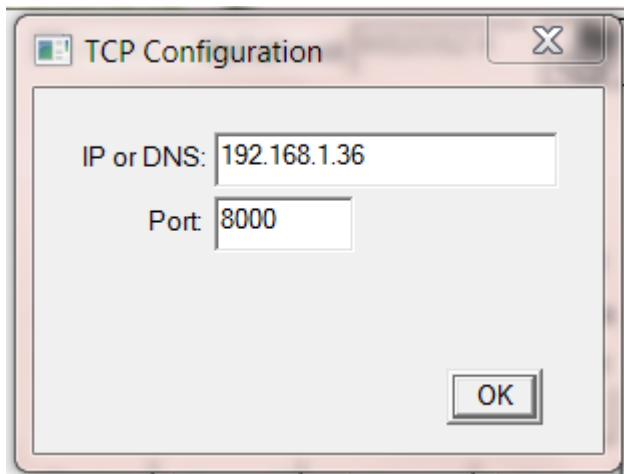
From the Configure menu, pick Ports → New Port...

Choose type of AGW and give it a meaningful name.



Click on Create and pick port type of TCP/IP.

Enter the address of the Raspberry Pi, and the default port of 8000.



Digipeater Operation

“cd” to /home/pi so the proper configuration file will be found.

Edit the **/home/pi/direwolf.conf** file and look for a section like this:

```
# Station identifier for this channel.
# Multiple channels can have the same or different names.
#
# Naturally it must be up to 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 NOCALL

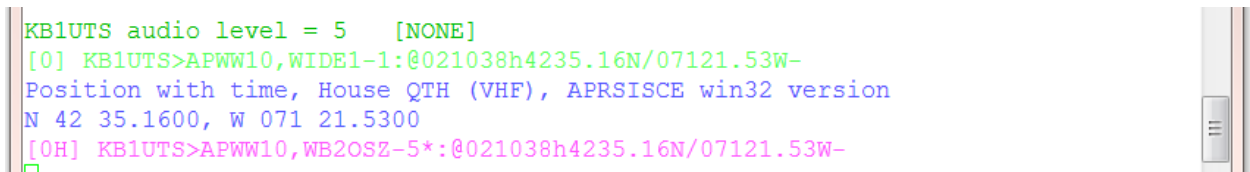
Change "NOCALL" to your ham radio call and optional SSID.

Next, look for a section like this:

```
#-----  
# ----- Example 1: Typical digipeater -----  
#-----  
  
#  
# For most common situations, use something like this by removing  
# the "#" from the beginning of the line.  
# To disable digipeating, put # at the beginning of the line.  
#  
  
#  
#DIGIPEAT 0 0 ^WIDE[3-7]-[1-7]$ ^WIDE[12]-[12]$
```

Remove the "#" character from the beginning of the last line shown above. It is necessary to stop and restart the application to notice configuration file changes.

Here is an example of what you should see:



```
KB1UTS audio level = 5    [NONE]  
[0] KB1UTS>APWW10,WIDE1-1:@021038h4235.16N/07121.53W-  
Position with time, House QTH (VHF), APRSISCE win32 version  
N 42 35.1600, W 071 21.5300  
[0H] KB1UTS>APWW10,WB2OSZ-5*:@021038h4235.16N/07121.53W-
```

Dark green: Information about the station we heard. Either the originating station or a digipeater.

Green: Raw received data. Notice that the digipeater field contains "WIDE1-1."

Blue: An explanation for troubleshooting. The destination (APWW10) is used to determine the type of system generating the signal.

Magenta: This is the re-transmitted packet. Notice that the digipeater field now contains "WB2OSZ-5*." The "*" indicates that it has been used up and won't be digipeated again.

Enable Beacons

Look for a section like this in direwolf.conf file.

```
# PBEACON delay=0:15 every=30 overlay=S symbol="digi" lat=42^37.14N lon=071^20.83W  
power=50 height=20 gain=3 comment="Chelmsford MA" via=WIDE1-1,WIDE2-1  
# PBEACON delay=10:15 every=30 overlay=S symbol="digi" lat=42^37.14N lon=071^20.83W  
power=50 height=20 gain=3 comment="Chelmsford MA"  
# PBEACON delay=20:15 every=30 overlay=S symbol="digi" lat=42^37.14N lon=071^20.83W  
power=50 height=20 gain=3 comment="Chelmsford MA"
```

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

Remove the “#” comment character from the beginning of the “PBEACON” lines. Make necessary adjustments to latitude, longitude, comment, etc.

Note that each position beacon command must be on a single line. Multiple lines are shown above due to page width limitations.

Internet Gateway (IGate)

Dire Wolf can serve as a gateway between the radio network and servers on the Internet. This allows information to be retrieved from locations such as <http://aprs.fi> or <http://findu.com>. Information can optionally be relayed from the servers, through your station, and on to the radio.

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

Each name has multiple addresses to achieve load balancing and resiliency. Visit <http://aprs2.net/> for the most recent information. You also need to specify your login name and passcode. For example:

```
IGSERVER noam.aprs2.net
IGLOGIN WB2OSZ-5 123456
```

Contact the author if you can't figure out how to generate a passcode for your ham radio cal. If you want to transmit information from the servers, you need to specify two additional pieces of information. First, you need to specify the radio channel and the via path for the packet header. Examples:

```
IGTXVIA 0 WIDE1-1,WIDE2-1
IGTXVIA 1 WZ9ZZZ
IGTXVIA 0
```

In the first case packets will be transmitted on the first radio channel with a path of WIDE1-1,WIDE2-1. In the second case, packets are transmitted on the second radio channel and directed to a known nearby digipeater with wide coverage. In the third case, there will be no digipeating.

You might want to apply a filter for what packets will be obtained from the server. Read about filters here: <http://www.aprs2.net/wiki/pmwiki.php/Main/FilterGuide> Example:

IGFILTER m/50

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 warning is displayed in red.

IGTXLIMIT 6 10

Receive Decoding Performance and CPU Requirements

As explained in the User Guide, multiple decoders are available and it is possible to run them at the same time. They were tuned in different ways and differ by effectiveness and CPU power used.

Decoder	Packets decoded from WA8LMF test CD	Percent of CPU time required.	Comment
A	965	23 - 30	
B	968	34 - 39	
C	971	45 - 50	
A & B & C	976	87 - 92	Not a good idea because there is no idle time left over. Data will be lost.
F	965	17 - 25	Only for 1200 baud, 1200/2200 Hz, 44100 sample rate.

You can pick one depending on the requirements of your particular situation with configuration options like this:

```
MODEM 1200 1200 2200 C
MODEM 1200 1200 2200 F
```

When building for the ARM processor, decoder F is the default to minimize CPU load.

Troubleshooting

First check the audio gain on your input device. I've always found it necessary set it at the maximum or fairly high. Different systems may have different applications for configuring the sound system. If using "alsamixer" follow this procedure:

- Press F6 to select the "sound card." Use up and down arrows and press Enter.
- Press F4 to set the Capture level. Use up arrow to set level to the max or fairly high.

Did you run apt-get and rpi-update to get the latest software and firmware? After many months of aggravation, it finally worked right after an update mid October 2013.

Before that time...

The Signalink USB worked OK but I couldn't get a cheap USB audio adapter to operate properly.

The same USB audio adapter worked fine on Microsoft Windows. It worked fine with Ubuntu Linux on a regular PC. It was a miserable failure on the Raspberry Pi.

Every 100 seconds, there was a debugging message similar to these in green:

Past 100 seconds, 3924900 audio samples, 0 errors.

Past 100 seconds, 3977820 audio samples, 0 errors.

Past 100 seconds, 3949155 audio samples, 0 errors.

It reveals that the audio input system was providing less than 4000000 samples during a period of 100 seconds. We expect this to be close to 4410000 for the 44.1 kHz sample rate. Instead we find it is about 10% lower than expected. With the sample rate that far off, or 10% of the audio samples getting lost somewhere, nothing gets decoded. If no signals are decoded, look for these messages for a clue.

The USB audio device was fine on Windows and Ubuntu Linux on a regular PC, so it was probably a software driver issue. After the most recent software/firmware update, it's now fine. The messages every 100 seconds show the expected number of audio samples for that time period.

If you can't copy any signals, make sure you have software/firmware no earlier than mid October 2013, crank up the audio input level, and then look for the debug messages every 100 seconds.