

# Raspberry Pi APRS

January 2013

The Raspberry Pi makes a fine platform for running Dire Wolf and other ham radio applications.

Getting the audio device to work properly took some experimentation. It might be beneficial to follow these instructions closely before making your own modifications.

## 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+rpil' -
--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+rpil)
```

## System Setup & Configuration

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

The Raspbian “wheezy” distribution (2012-12-16) from <http://www.raspberrypi.org/downloads> had some glitches with the USB audio causing a low percentage of the received packets to be decoded. These problems went away after installing the latest updates on January 21, 2013. Be sure to perform the system wide upgrade in section 2.3.5 of the Debian Reference on the desktop.

## 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
```

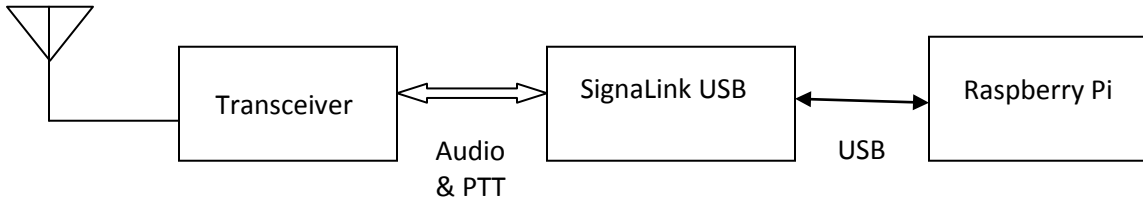
You can get a list of audio input devices with the “arecord -l” command:

```
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.



It should also be possible to use the built-in audio output and some sort of USB microphone adaptor.

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
  
```

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. The built-in serial interface does not have the RTS or DTR control signals. Possible alternatives include:

- Use a USB to RS-232 converter cable to supply a full RS-232 port.
- VOX operation where transmitter is activated by transmit audio signal. The SignalLink USB uses this technique.
- Use one of the General Purpose I/O (GPIO) pins. Future possibility – not implemented yet.

## Install Dire Wolf

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

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

Failure to perform this step will result in a compile error resembling “audio.c:....: fatal error: alsa/asoundlib.h: No such file or directory”

Download the source version (0.6 or later) from <http://home.comcast.net/~wb2osz/site/> and copy it to your Raspberry Pi.

The file should be named something like direwolf-src-0.6.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-src-0.6.zip
cd direwolf-0.6
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 <a href="http://aprs.fi">http://aprs.fi</a> or <a href="http://findu.com">http://findu.com</a>
/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/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: <a href="http://www.aprs.org/aprs11/tocalls.txt">http://www.aprs.org/aprs11/tocalls.txt</a>
/usr/share/direwolf/dw-icon.png	Icon for the desktop.
/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

Attempts to use the "hw:1,0" device directly were not very satisfying. Various experiments lead to the conclusion that best results were obtained when using **pulseaudio** between the application and hardware driver.

A script has been provided to perform the necessary configuration. Run this command:

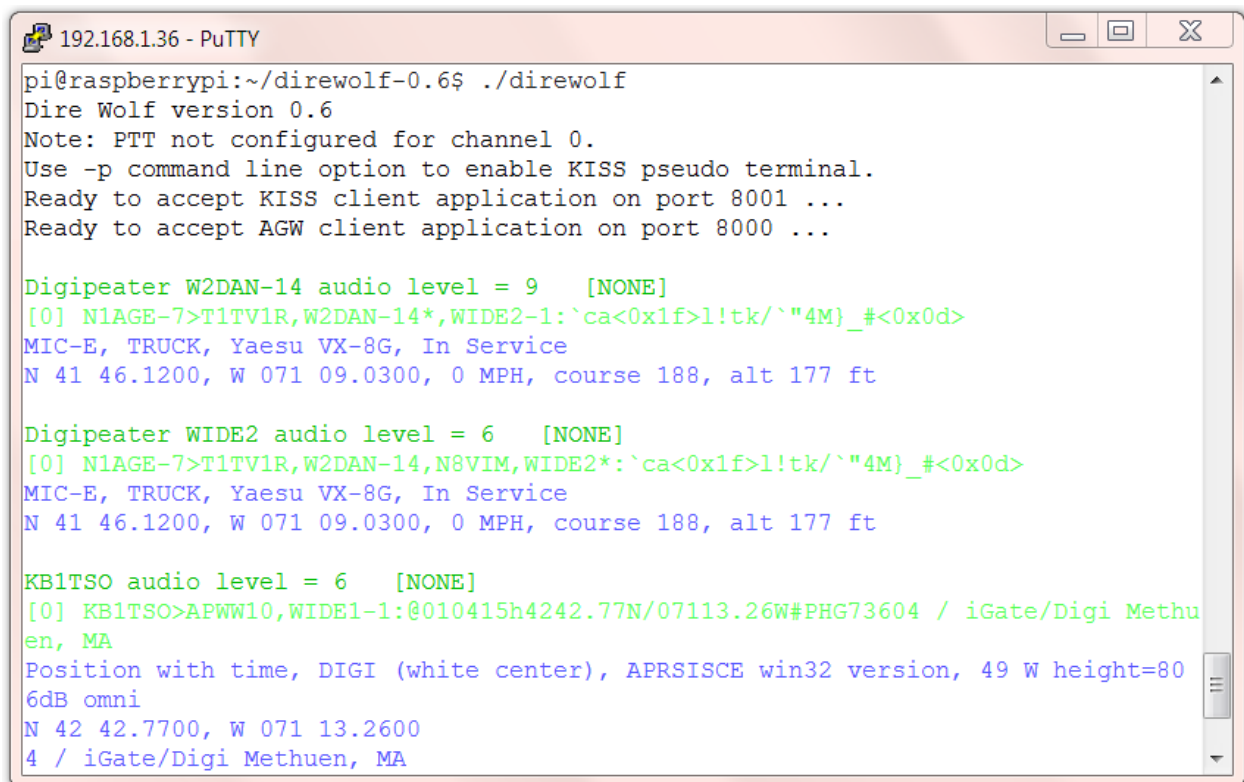
```
./rpi-audio.sh
```

## Run Dire Wolf application

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

```
direwolf
```

You should see something resembling this:



```
192.168.1.36 - 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.

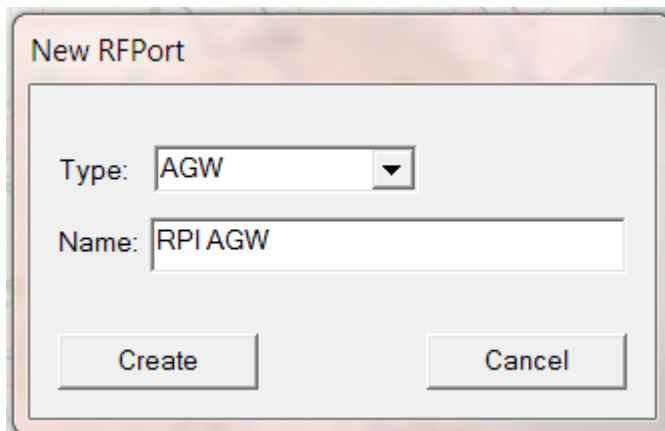
The final part of the greeting contains the network ports for the KISS and AGW protocols for use by client applications. You will need to know these numbers for the next step. If you don’t like these default values, they can be changed in a configuration file.

## 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 with Linux in your “shack” in a basement connected to your radio equipment. You might want to use a client application on a Windows laptop in a more comfortable location. Here is an example of how you could configure APRSIS32 in this situation.

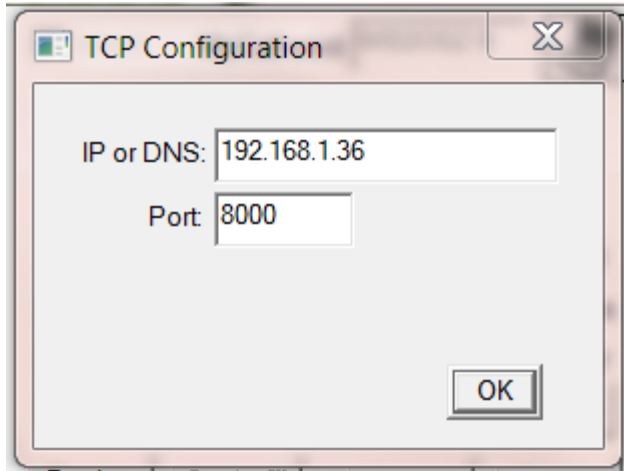
From the Configure menu, pick Ports → New Port...

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

Edit the `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 Beaconing

Look for a section like this in direwolf.conf:

```
# Example:
#
# BEACON 0 1 30 WB2OSZ-5>APDW06,WIDE1-1,WIDE2-1:!4237.14NS07120.83W#PHG7130Chelmsford, MA
# BEACON 0 11 30 WB2OSZ-5>APDW06:!4237.14NS07120.83W#PHG7130Chelmsford, MA
# BEACON 0 21 30 WB2OSZ-5>APDW06:!4237.14NS07120.83W#PHG7130Chelmsford, MA

# Change this for your particular situation before removing
# the # comment character from the beginning of the line!!!
```

Remove the “#” comment character from the beginning of the “BEACON” lines. Make necessary adjustments to the station call, latitude, longitude, and comment.

You can check the validity with the **decode\_aprs** utility mentioned earlier. This example has an intentional error:

```
echo 'WB2OSZ-5>APDW06:!4237.14NS07120.83x#PHG7130Chelmsford, MA' | decode_aprs
```

The response is:

```
WB2OSZ-5>APDW06:!4237.14NS07120.83x#PHG7130Chelmsford, MA
Error: 'x' found for longitude hemisphere. Specification requires upper case E or W.
Position, OVERLAY DIGI (green star) w/overlay S, DireWolf, WB2OSZ, 49 W height=20 3dB omni
N 42 37.1400, Invalid Longitude
Chelmsford, MA
```



## 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`.