Raspberry Pi SDR IGate

Last update ~~11/9/2015~~

7/3/2016 – added note about /etc/modprobe.d/raspi-blacklist.conf.

It's easy to build a receive-only APRS Internet Gateway (IGate) with only a Raspberry Pi and a software defined radio (RTL-SDR) dongle. Here's how.

Hardware Required

- **Raspberry Pi**

  I happened to use the model 2 so I can't say, with certainty that the earlier models would be fast enough to keep up. "top" shows about 93% cpu idle time so the older models are probably more than adequate.

  The procedure here is known to work with the Raspbian operating system. Some adjustments might be required for other operating systems.

- **SDR Dongle**

  This connects to the USB port and an antenna. This is the one I used.

  http://www.amazon.com/NooElec-RTL-SDR-RTL2832U-Software-Packages/dp/B008S7AVTC

  There are many others that appear to be equivalent such as
  https://www.adafruit.com/products/1497

Software Required

- **Dire Wolf**

  Install following the instructions in **Raspberry-Pi-APRS.pdf.**

  You can stop at the section called **Interface for Radio**. Here we are using the SDR dongle rather than a USB audio adapter.

  Don't worry about the configuration part because we will build our own configuration file here.

- **RTL-SDR** Library from http://sdr.osmocom.org/trac/wiki/rtl-sdr

Install it like this:

```
sudo apt-get update
sudo apt-get install cmake build-essential libusb-1.0-0-dev
cd ~
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON
make
sudo make install
sudo ldconfig
```

I didn't find it to be necessary, with the software version I was using, but community feedback suggests adding the following to **/etc/modprobe.d/raspi-blacklist.conf** to prevent the use of undesired device drivers.

```
blacklist dvb_usb_rtl28xxu
blacklist dvb_usb_v2
blacklist rtl_2830
blacklist rtl_2832
blacklist r820t
```

Configuration

We need to construct a configuration file for Dire Wolf. Take the following text and put it into a text file, called **sdr.conf**, in our home directory. You might find a copy of this file in /usr/share/doc/direwolf/examples.

```
#
# Sample configuration for SDR read-only IGate.
#

# We might not have an audio output device so set to null.
# We will override the input half on the command line.
ADEVICE null null
CHANNEL 0
MYCALL xxx

# 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 noam.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.

IGLOGIN xxx 123456

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

Put your callsign and optional SSID in the two places that have xxx above. Put your actual passcode in place of 123456.

Run It

```
rtl_fm -f 144.39M - | direwolf -c sdr.conf -r 24000 -D 1 -
```

Note the "-" at the end of the line which means read audio from stdin. Alternatively this could have been done in the configuration file, using either of these forms:

```
ADEVICE stdin null
ADEVICE - null
```

You should now have a functioning receive only IGate. Consult the **User Guide** for more details.

Automatic Startup on Reboot

You might be tempted to put the command line above into /etc/rc.local. It's not that simple and there are some disadvantages. First, it runs as root. It's best to avoid running as root, when possible, because it's harder to accidentally trash your system. When /etc/rc.local is running, $PATH is set to /sbin:/usr/sbin:/bin:/usr/bin so it wouldn't find anything in /usr/local/bin. You would need to specify the full path. The current working directory and $HOME are both / so you would need to put the configuration file there or specify a different location. The text output, for troubleshooting, is not readily available. Finally, it there is a failure, it won't restart until the next reboot.

My suggestion would be to modify the included **dw_start.sh** script so that it also runs rtl_fm. Look for this line and remove the # at the beginning.

```
#DWCMD="bash -c 'rtl_fm -f 144.39M - | direwolf -c sdr.conf -r 24000 -D 1 -'"
```

Run "crontab –e" to edit your crontab file. Assuming that you are running as user "pi" and you have a copy of dw-start.sh in the home directory, 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`.

Calibration

The cheap RTL SDR dongles might be a few kHz off and drift even more with temperature variations.

In one test, I found that it worked best when I specified a frequency 8 kHz lower than the desired frequency.

For best results you will want to calibrate them using a nearby station of known frequency. Here is a script that will perform calibration periodically: https://github.com/khaytsus/direwolf-init