

# John's Tech Blog

Had all I can take, I can't take no more. ~Popeye

## OP25 Raspberry Pi Streaming Part 2

Like

Be the first of your friends to like this.

So I finally figured out OP25 and I have this brand new Raspberry Pi 3 B + laying around doing nothing. In this segment we'll install OP25 on Raspberry Pi and then take our police scanner feed and send it to the internet on Broadcastify.

As someone pointed out to me yesterday, "There's an app for that" they are indeed correct. You can get Police Scanner Apps for IOS and Android. Guess where the feeds in those apps comes from? If you said Broadcastify you'd be correct. So if no one is feeding your municipality then there will be no feed in the app. We will be that feed.

So the assumption is that you have a Pi with Raspbian installed and you kind of know how to use it.

The first thing we will do is to install GQRX. GQRX is not a requirement but I like GQRX, it pulls down all the rtl-sdr drivers and GNURadio which IS a requirement and I use GQRX to hone in on the signals from the Control Channels, so it is a useful tool to me anyway. Boot your pi, open a terminal and type the following commands.

```
sudo apt update
```

```
sudo apt install gnuradio libvolk1-bin libusb-1.0-0 gr-iqbal
```

```
sudo apt install qt5-default libqt5svg5 libportaudio2
```

```
sudo cp udev/*.rules /etc/udev/rules.d/
```

Now make a directory and download gqrX

```
cd
```

```
mkdir gqrX
```

```
cd gqrx
wget https://github.com/csete/gqrx/releases/download/v2.11.5/gqrx-sdr-2.11.5-linux-rpi3.tar.xz
```

Unzip the file (I just did a VNC session and did it in the GUI) otherwise you need a program called xz-utils

```
sudo apt-get install xz-utils
```

Once unzipped the program is installed. We're not going to open or use it now. We got what we want. All you do is navigate to the folder that has the gqrx executable in it and double click or type `./gqrx` to launch the program in a terminal.

Now install OP25. Go back to Part 1 of this blog to learn how to do that. Installing and configuring OP25 is the same on Raspberry Pi as it is on Ubuntu Linux.

Ok the assumption here is that you are up and running and decoding a stream.

You have to have a RadioReference or Broadcastify account now and you have to apply for a stream. I'm going to use RR as an example. Under RadioReference go to [this link](#) to apply to stream a channel. Fill out the requested information EXACTLY as they request or they will kick your request back. Mine got kicked back once.

Once approved they'll give you a link which contains a server address, a mount point, and a password.

These feed technical details are provided for users who want to configure their own broadcasting clients.

Server	
Port	80
Mount	
Password	

Now go to [this page](#) to get the directions to stream to Broadcastify

All the directions come from his page. I DID NONE OF THIS.

```
sudo nano /etc/modprobe.d/bcm2835.conf
```

Insert the following:

```
softdep snd-bcm2835 post: snd-aloop
options snd-aloop enable=1 index=1 pcm_substreams=2
```

Hit CTL key + X key then Y to save the file

Now make an asound.conf file:

```
sudo nano /etc/asound.conf
```

```
# output device
pcm.loopout0 {
    type plug
    slave.pcm "hw:Loopback,0,0"
}

# input device
pcm.loopin0 {
    type dsnoop
    ipc_key 686592
    slave.pcm "hw:Loopback,1,0"
}

# duplex plug device
pcm.loop0 {
    type plug
    slave {
        pcm {
            type asym
            playback.pcm "loopout0"
            capture.pcm "loopin0"
        }
    }
}

# output device
pcm.loopout1 {
    type plug
    slave.pcm "hw:Loopback,0,1"
}

# input device
pcm.loopin1 {
    type dsnoop
    ipc_key 686593
    slave.pcm "hw:Loopback,1,1"
}

# duplex plug device
pcm.loop1 {
```

```

type plug
slave {
    pcm {
        type asym
        playback.pcm "loopout1"
        capture.pcm "loopin1"
    }
}
}

```

Hit CTL key + X key then Y to save the file

You just made a loopback device. On Windows this is kind of similar to Virtual Cable where you take your audio and pipe it into another program. Because you changed the audio source you need to change a couple things in the way you invoke OP25. I went back to my script file and changed the stuff in bold:

```

./rx.py --args 'rtl' -N 'LNA:47' -S 250000 -f 857.2625e6 -o 17e3 -q -2 -O loop0 -T trunk.tsv -V -2 -U

```

So your sound device is now the loopback device and you're passing the stream.

Now get darkice

```
sudo apt-get install darkice
```

Now make a config file:

```
sudo nano /etc/darkice-stream0.cfg
```

Paste in the following making sure to change all your data in red:

```

# sample DarkIce configuration file, edit for your needs before using
# see the darkice.cfg man page for details

# this section describes general aspects of the live streaming session
[general]
duration          = 0          # duration of encoding, in seconds. 0 means forever
bufferSecs        = 1          # size of internal slip buffer, in seconds
reconnect         = yes        # reconnect to the server(s) if disconnected

```

```
# this section describes the audio input that will be streamed
[input]
device          = loop0    # OSS DSP soundcard device for the audio input
sampleRate      = 22050    # sample rate in Hz. 22050 is the RR Standard
bitsPerSample    = 16      # bits per sample. 16 for mono feeds, 32 for stereo feeds
channel         = 1        # channels. 1 = mono, 2 = stereo

# this section describes a streaming connection to an IceCast server
# there may be up to 8 of these sections, named [icecast-0] ... [icecast-7]
# these can be mixed with [icecast2-x] and [shoutcast-x] sections

# If you want to send to your own icecast server, you basically copy
# this whole section and put in the values to point the feed to your own
# server instead of the RR information below

# Don't forget, the name [icecast2-0] needs to change to [icecast2-1]
# and 2 and 3 as you add more feed destinations.

[icecast2-0]
# The audio format to encode to, we're using mp3 here
format          = mp3

# The bit rate mode to use, the commented out lines are for a fixed bit
# rate "constant bit rate" at 16 bits. I prefer the variable bit rate
# for the sound quality it has and the throughput is very similar on avg

#bitrateMode     = cbr
#bitrate         = 16

sampleRate       = 22050    # sample rate in Hz. 22050 is the RR Standard
bitrateMode      = cbr
bitrate          = 16

# Quality at .1, max is 1.0, but as you increase, so does the bandwidth
# used. For mono scanner audio .1 is totally fine.
quality          = 0.1

# Takes the input and mixes it down to a mono output for the feed.
channel          = 1

# Cuts frequency response over 5kHz. For scanner feed this mostly
# eliminates hiss and harshness on paging tones. Saves a bunch of
# bandwidth as well.
lowpass          = 5000

# Server shown on your RR feed owner page that hosts your feed
# You need to log in to RR, go to Account -> My Live Audio Feeds ->
# click the feed owner page icon at the far right of your feed listing.
# Then you will see all this information.
```

```
# From Feed Technical Details, the "Server"
server          = SERVER.broadcastify.com

# From Feed Technical Details, the "Port"
port            = 80      # port of the IceCast server, usually 8000

# From Feed Technical Details, the Password exactly as shown
password        = PASSWORD

# From Feed Technical Details, the mount...remove the leading "/", it's
# not needed in this file. So if it is /999999 make it just 999999
mountPoint      = MOUNTPPOINT

# name of the stream
# On My Live Audio Feeds, it's the first part ending in a colon ":"
name            = Short name

# On My Live Audio Feeds, it's the part in parentheses in the Feed name
description     = Description
```

Hit CTL key + X key then Y to save the file

Now start your OP25 instance as you did from lesson 1. I used a script file. I tried and tried and tried to use the start at boot systemd script on the tutorial page but it just wouldn't work. My script file is called "op25.sh" and is located in the /home/pi/op25 folder. I finally figured out to start it at boot I call it from a crontab. OP25 needs to run in a terminal so I installed screen and did the following command from crontab -e

```
@reboot /usr/bin/screen -d -m /home/pi/op25/op25.sh
```

Or you can start it like this:

I finally figured out I could invoke that from the service file as well.

```
sudo nano /etc/systemd/system/op25-stream0.service
```

```
[Unit]
Description=op25
After=syslog.target network.target nss-lookup.target network-online.target
Requires=network-online.target

[Service]
User=pi
Group=pi
```

```
Type=forking
ExecStart=/usr/bin/screen -d -m /home/pi/op25/op25.sh
RestartSec=5
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Ctl + X then Y to save

```
sudo systemctl enable op25-stream0.service
sudo systemctl start op25-stream0.service
```

Now to start darkice is pretty easy. From a terminal type the following:

```
darkice -c /path/to/darkice.cfg
```

In my case that is

```
darkice -c /home/pi/darkice/darkice.cfg
```

If you followed the Tutorial from the other page it is probably in /etc. It doesn't matter where it is or what you call it as long as you point to it after running the darkice command.

To launch it automatically at boot type

```
sudo nano /etc/systemd/system/darkice-stream0.service
```

Paste in the following making sure to get the darkice.cfg path right:

```
[Unit]
Description=darkice-stream0
After=syslog.target network.target nss-lookup.target network-online.target
Requires=network-online.target

[Service]
ExecStart=/usr/bin/darkice -c /etc/darkice-stream0.cfg
RestartSec=5
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Now enable and start the service

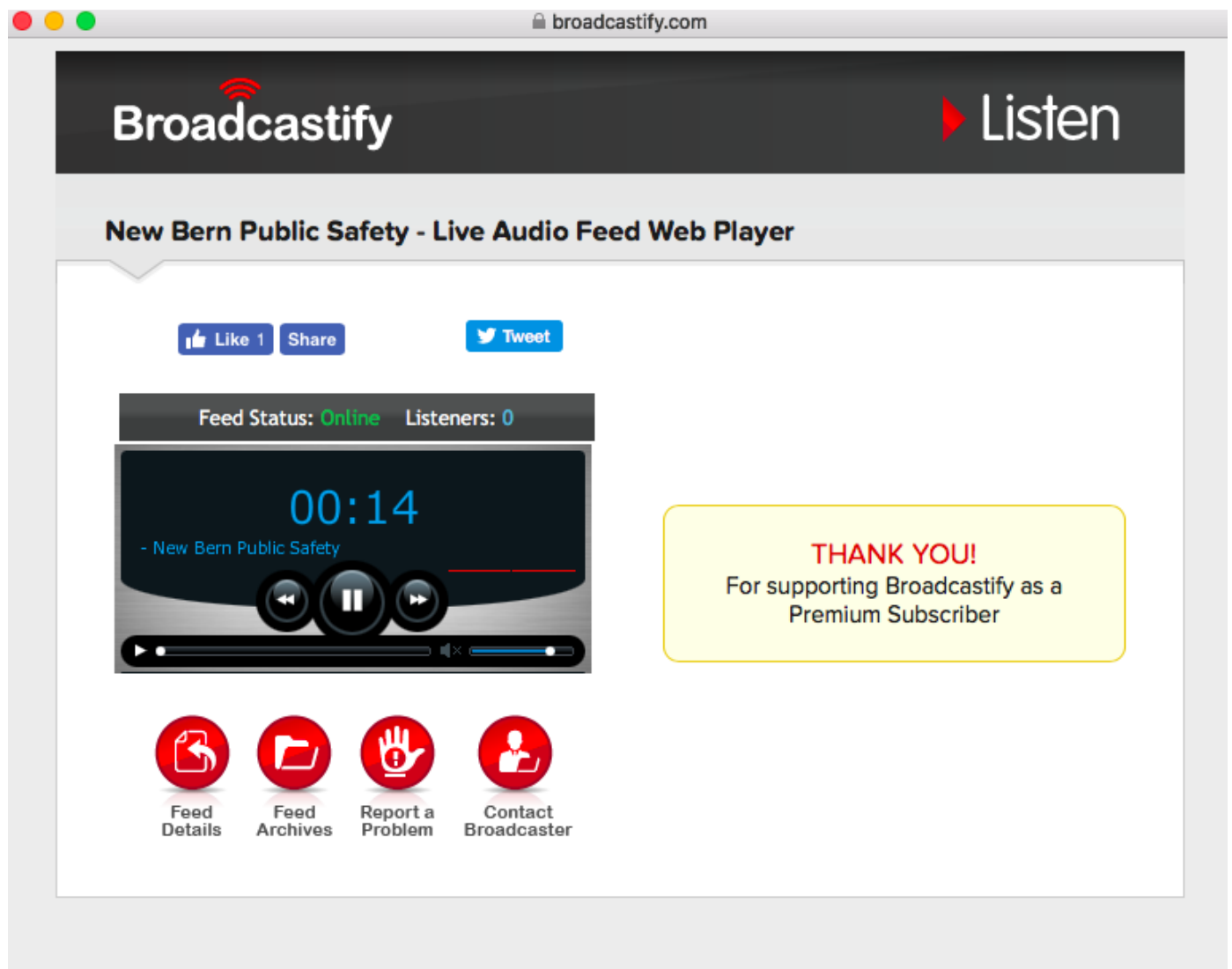
```
sudo systemctl enable darkice-stream0.service
```

```
sudo systemctl start darkice-stream0.service
```

Now when you reboot OP25 should start and run, and darkice should be streaming to your feed.

Here's my feed.

[New Bern Public Safety](#)



This entry was posted in Linux, Raspberry Pi, SDR, Software on July 20, 2018

[<https://www.hagensieker.com/wordpress/2018/07/20/op25-raspberry-pi-streaming-part-2/>] .



## 2 thoughts on “OP25 Raspberry Pi Streaming Part 2”



NJ

August 28, 2018 at 12:47 am

I'm getting:

E: Unable to locate package libvolk1-bin



**John Hagensieker**

Post author

August 28, 2018 at 7:42 am

Try:

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

```
sudo apt-get libvolk1-bin
```