Creating a DMR to P25 Bridge

Table Of Contents:

Reference Sheet

- # P25Clients
- # DVReflector
- # P25Hosts.txt
- # DVSwitch System Builder
- # BrandMeister
- # Free DNS Service

Configuring Operating System

# You can set the DMR to P25 Bridge up on a Raspberry Pi or Linux other systems.

# Using a Raspberry Pi with Debian 11 image or any other Debian system:
# For this tutorial we will be using Debian 11.

# Default login for Debian images on the Raspberry Pi and creating "p25reflector" user:

Login: root
Password: <none>

# Activate WiFi:

```
nano /etc/network/interfaces.d/wlan0
```

# Now Reboot

```
reboot -h now
```

# Once it has rebooted we will update with apt:

```
apt update
```

# Once the update has completed we will upgrade with apt:

```
apt upgrade
```

# Now we will add a user —> replace <username> with your own.

```
adduser <username>
```

# Change the root password:

```
passwd
```

```
# Change to root:

su - root

# Install sudo with apt:

apt install sudo

# Now give privileges to “sudo”:

sudo usermod -aG sudo <username>

# Update again with apt:

sudo apt update

# Upgrade again with apt:

sudo apt upgrade -y

# Now gather and build essential packages:

sudo apt-get install build-essential -y

# Install git

sudo apt install git -y

# Install your network tools (Optional)

sudo apt-get install net-tools -y

# Install wget

sudo apt-get install wget
sudo apt update

sudo adduser p25reflector

# Type new password for the p25reflector user.

# Switch to the p25reflector user.

sudo su - p25reflector

# Give sudo permissions to p25reflector.

sudo usermod -aG sudo p25reflector

# Create new directory named Downloads.

mkdir Downloads

# Change into the new “Downloads” Directory with the cd command.

cd Downloads

# Now we will download P25Clients: This is where the P25Gateway and
P25Parrot files reside.

sudo git clone https://github.com/g4klx/P25Clients.git

# Cloning into 'P25Clients'...
remote: Enumerating objects: 3307, done.
remote: Counting objects: 100% (1362/1362), done.
remote: Compressing objects: 100% (501/501), done.
remote: Total 3307 (delta 983), reused 1134 (delta 861), pack-reused
1945
Receiving objects: 100% (3307/3307), 3.00 MiB | 281.00 KiB/s, done.
Resolving deltas: 100% (2214/2214), done.

# Change into the P25Clients Directory with the cd command.

cd P25Clients

# Install DVReflectors: This includes NXDNReflector, P25Reflector and
YSFReflector.
```

```
sudo git clone https://github.com/nostar/DVReflectors.git

# Cloning into 'DVReflectors'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 91 (delta 6), reused 5 (delta 5), pack-reused 80
Receiving objects: 100% (91/91), 1.02 MiB | 2.39 MiB/s, done.
Resolving deltas: 100% (44/44), done.

# List the files with the ls command.

ls

# DVReflectors  Makefile      P25Gateway  README.md
LICENCE     P25Clients.sln  P25Parrot


# Change in the DVReflectors with the cd command.

cd DVReflectors

# List the files with the ls command.

ls

# LICENCE  NXDNReflector  P25Reflector  README.md  YSFReflector

# Now change into the P25Reflector file with cd command.

cd P25Reflector

# List the files with the ls command.

ls
#  You should see a file called "P25Reflector.cpp".

Conf.cpp     Mutex.cpp      P25Reflector.vcxproj        UDPSocket.cpp
Conf.h        Mutex.h   P25Reflector.vcxproj.filters  UDPSocket.h
DMRIds.dat    Network.cpp     StopWatch.cpp
Utils.cpp
DMRLookup.cpp  Network.h       StopWatch.h Utils.h
DMRLookup.h    P25Reflector.cpp  Thread.cpp
Version.h
Log.cpp        P25Reflector.h    Thread.h
Log.h          P25Reflector.ini  Timer.cpp
Makefile       P25Reflector.sh   Timer.h
```

# Now we need to copy the P25Reflector.ini to /etc with the cp command.

sudo cp P25Reflector.ini /etc

# Change to the /etc directory with the cd command.

cd /etc

# Edit the P25Reflector.ini with the nano command.

sudo nano P25Reflector.ini

# Now change the Daemon from a "1" to a "0".

```
[General]
Daemon=1 -> Daemon=0

[Id Lookup]
Name=DMRIds.dat
Time=24

[Log]
# Logging levels, 0=No logging
DisplayLevel=1
FileLevel=1
FilePath=.
FileRoot=P25Reflector
FileRotate=1

[Network]
Port=41000
Debug=0
```

# The correct changes will look like this:

```
[General]
Daemon=0

[Id Lookup]
Name=DMRIds.dat
Time=24

[Log]
# Logging levels, 0=No logging
DisplayLevel=1
FileLevel=1
FilePath=.
FileRoot=P25Reflector
FileRotate=1

[Network]
Port=41000
Debug=0
```

```
# Now we will compile, change directory back to Downloads.

cd

# List the files with the ls command.

ls

# Change into the P25Reflector directory with the cd command.

cd Downloads/P25Clients/DVReflectors/P25Reflector

# List the files with the ls command.

ls
```

# You will see these files if you are in the correct spot:

```
Conf.cpp     Log.h      P25Reflector.cpp           StopWatch.cpp  UDPSocket.cpp
Conf.h          Makefile     P25Reflector.h                 StopWatch.h
UDPSocket.h
DMRIds.dat     Mutex.cpp    P25Reflector.ini               Thread.cpp
Utils.cpp
DMRLookup.cpp  Mutex.h      P25Reflector.sh                Thread.h
Utils.h
DMRLookup.h    Network.cpp  P25Reflector.vcxproj           Timer.cpp
Version.h
Log.cpp        Network.h    P25Reflector.vcxproj.filters  Timer.h
```

# Now we will use the make command to compile.

```
sudo make
```

# Now copy the files that are executable now.

```
sudo cp P25Reflector /usr/local/bin/
```

```
sudo cp P25Reflector.sh /usr/local/bin/
```

# We will now set file permissions for the P25Reflector.

```
sudo chmod +x /usr/local/bin/P25Reflector.sh
```

# Now it is time to test the reflector and start it up.

# Run this command as the root user.

```
sudo su - root
```

# Using the cd command we will change into —> /usr/local/bin

```
cd /usr/local/bin
```

# Once you are in /usr/local/bin we will start the reflector.

```
sudo ./P25Reflector.sh start /etc/P25Reflector.ini &
```

# When you enter this command you should see:

```
2024-04-12 04:06:04.795 Opening P25 network connection
I: 2024-04-12 04:06:04.797 Opening UDP port on 41004
W: 2024-04-12 04:06:04.797 Cannot open the Id lookup file -
DMRIds.dat
M: 2024-04-12 04:06:04.798 Starting P25Reflector-20210912
I: 2024-04-12 04:06:04.798 Started the DMR Id lookup reload
thread
```

# Lets open a new terminal, leave this one open. After a short amount of time you will receive a message saying, “No Repeaters Linked”…

Ex) `2024-04-12 04:08:05.022 No repeaters linked`

Build The P25Gateway

# In a new terminal with the other terminal still running.

# Switch to the p25reflector user:

```
su - p25reflector
```

# Change into the P25Gateway directory:

```
cd Downloads/P25Clients/P25Gateway
```

# List the files with the ls command.

```
ls
```

# You will see **P25Gateway.cpp** when you are in the correct directory:

```
Audio          Makefile                     P25Hosts.txt    StopWatch.h
Utils.h
Conf.cpp       Mutex.cpp                    P25Network.cpp  Thread.cpp
Version.h
Conf.h         Mutex.h                      P25Network.h    Thread.h
Voice.cpp
DMRIds.dat     P25Gateway.cpp               Reflectors.cpp  Timer.cpp
Voice.h
DMRLookup.cpp  P25Gateway.h                 Reflectors.h    Timer.h
DMRLookup.h    P25Gateway.ini               RptNetwork.cpp
UDPSocket.cpp
Log.cpp        P25Gateway.vcxproj           RptNetwork.h    UDPSocket.h
Log.h          P25Gateway.vcxproj.filters   StopWatch.cpp   Utils.cpp
```

# Copy the P25Hosts.txt to /etc with the cp command:

```
sudo cp P25Hosts.txt /etc
```

# Copy the P25Gateway.ini to /etc with the cp command:

```
sudo cp P25Gateway.ini /etc
```

# Compile the files with the make command:

```
sudo make
```

# Copy the P25 Gateway to /opt/P25Gateway:

```
sudo cp P25Gateway /opt/P25Gateway
```

Installing DVSwitch

# Change to the /tmp directory with the cd command:

```
cd /tmp
```

# Use the wget command:

```
sudo wget http://dvswitch.org/buster
```

# Once finished you will get:

```
--2024-04-11 22:25:21--  http://dvswitch.org/buster
Resolving dvswitch.org (dvswitch.org)... 44.103.34.4
Connecting to dvswitch.org (dvswitch.org)|44.103.34.4|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 2500 (2.4K)
Saving to: ‘buster’

buster                  100%[==============================>]   2.44K
--.-KB/s    in 0s

2024-04-11 22:25:22 (50.5 MB/s) - ‘buster’ saved [2500/2500]
```

# Set the file permission:

```
sudo chmod +x buster
```

# Now run the following commands one by one:

# Make sure there are two dashes before the “allow”: (-) (-) the format on these instructions merge them together, add two on your terminal to run the command correctly without an error: If you have any issues refer to the system builder.

```
sudo apt-get update —allow-releaseinfo-change
```

```
sudo ./buster
```

# When finished running the ./buster command you will see:

```
Starting DVSwitch repository install
Adding DVSwitch repositories to existing system
download package information from all configured sources
Installed repositories:
http://archive.raspberrypi.org/debianbullseye/main
http://dvswitch.org/DVSwitch_Repositorybuster/hamradio
http://raspbian.raspberrypi.org/raspbianbullseye/contrib
http://raspbian.raspberrypi.org/raspbianbullseye/main
http://raspbian.raspberrypi.org/raspbianbullseye/non-free
http://raspbian.raspberrypi.org/raspbianbullseye/rpi

Finished DVSwitch repository install
```

# Now we will update the system.

```
sudo apt-get update
```

# Install the DVSwitch Server (Make sure you have a space at the end of server before the -y or this will not work) - This command will take a few minutes to finishing the installation

```
sudo apt-get install dvswitch-server -y
```

# The following packages are installed with dvswitch-server if your system doesn't have them already:

```
analog-bridge dvswitch dvswitch-base dvswitch-dashboard dvswitch-menu
dvswitch-monit dvswitch-server ircddbgateway libc-ares2
libjs-highlight.js libnode72 libwxbase3.0-0v5 libwxbase3.0-dev
lighttpd lighttpd-mod-deflate lighttpd-mod-openssl md380-emu mmdvm-
bridgemonit node-agent-base node-commander node-debug node-lru-cache
node-ms node-mute-stream node-read node-semver node-tinycolor node-ws
node-yallist nodejs nodejs-doc nxdngateway nxdnparrot php-cgi php-
common php7.4-cgi php7.4-cli php7.4-common
php7.4-json php7.4-opcache php7.4-readline qemu-user-static quantar-
bridge spawn-fcgi wx3.0-headers
```

# **STOP!** You do not need to touch this part for this tutorial, if you want to explore it do so at your own risk as it will change how things are setup in this tutorial. Now that the DVSwitch Server has been installed you can access the configuration menu by using the cd command to change directories

—> /usr/local/dvs
—> ./dvs
cd /usr/local/dvs
sudo ./dvs

# Update the system again:

sudo apt-get update

# Change to the /opt directory with the cd command:

cd /opt

# Use the ls command to list the files:

ls

# Confirm that you have the following in the /opt directory:

```
Analog_Bridge  MMDVM_Bridge  NXDNParrot  P25Parrot  Quantar_Bridge
WidevineCdm  YSFGateway md380-emu      NXDNGateway   P25Gateway
pigpio     Web_Proxy       Wolfram      YSFParrot
```

# Change into the MMDVM_Bridge Directory with the cd command:

cd MMDVM_Bridge

# Use ls to list the directories.

```
ls
```

# Copy the MMDVM_Bridge.ini file to /etc

```
sudo cp MMDVM_Bridge.ini /etc
```

# For the Bridge to work correctly we will use two instances of Analog_Bridge
Run each command listed below one by one.

```
cd /opt/Analog_Bridge
```

```
sudo cp Analog_Bridge Analog_Bridge_P25
```

```
sudo cp Analog_Bridge.ini /etc/Analog_Bridge_P25.ini
```

```
sudo cp Analog_Bridge Analog_Bridge_DMR
```

```
sudo cp Analog_Bridge.ini /etc/Analog_Bridge_DMR.ini
```

# Now list all of the .ini files /etc

```
ls -la /etc|grep .ini
```

# You will see these files if done correctly:

```
-rw-r--r--   1 root root    5226 Apr 12 00:11 Analog_Bridge_DMR.ini
-rw-r--r--   1 root root    5226 Apr 12 00:11 Analog_Bridge_P25.ini
drwxr-xr-x   2 root root    4096 Dec  4 19:37 init
drwxr-xr-x   2 root root    4096 Apr 11 23:35 init.d
drwxr-xr-x   5 root root    4096 Dec  4 19:39 initramfs-tools
-rw-r--r--   1 root root    1463 Apr 12 00:07 MMDVM_Bridge.ini
-rw-r--r--   1 root root     578 Apr 11 22:19 P25Gateway.ini
-rw-r--r--   1 root root     197 Apr 11 21:52 P25Reflector.ini
-rw-r--r--   1 root root   12100 Aug 19  2015 RTIMULib.ini
```

# Create the log directories:

sudo mkdir /var/log/P25Gateway

Editing P25 Gateway

# Edit the P25Gateway file in /etc with nano:

sudo nano /etc/P25Gateway.ini

```
  GNU nano 5.4
/etc/P25Gateway.ini

[General]
Callsign=G4KLX --> Put your callsign here
RptAddress=127.0.0.1
RptPort=32010
LocalPort=42020
Debug=0
Daemon=1 --> Change from “1” to “0”

[Id Lookup]
Name=DMRIds.dat
Time=24

[Voice]
Enabled=1
Language=en_GB --> Change to en_US if you’re in the US
Directory=./Audio

[Log]
# Logging levels, 0=No logging
DisplayLevel=1
FileLevel=1
FilePath=. --> Change to /var/log/P25Gateway/
FileRoot=P25Gateway
FileRotate=1
```

```
[Network]
Port=42010
HostsFile1=./P25Hosts.txt --> Change to /etc/P25Hosts.txt
HostsFile2=./private/P25Hosts.txt
ReloadTime=60
ParrotAddress=127.0.0.1
ParrotPort=42011
P252DMRAddress=127.0.0.1
P252DMRPort=42012
Static=10100,10200,10300,10400 --> Change to your P25 TG
RFHangTime=120
NetHangTime=60
Debug=0

[Remote Commands]
Enable=0
Port=6074


# Your /etc/P25Gateway.ini should look like this but with your info
```

```
  GNU nano 5.4
/etc/P25Gateway.ini

[General]
Callsign=KF7KGN
RptAddress=127.0.0.1
RptPort=32010
LocalPort=42020
Debug=0
Daemon=0

[Id Lookup]
Name=DMRIds.dat
Time=24

[Voice]
Enabled=1
Language=en_US
Directory=./Audio

[Log]
# Logging levels, 0=No logging
DisplayLevel=1
FileLevel=1 --> If you have issues running the Gateway to the file
FileLevel=1 to “0” or change the path
FilePath=/var/log/P25Gateway/ FileRoot=P25Gateway
FileRotate=1
```

```
[Network]
Port=42010
HostsFile1=/etc/P25Hosts.txt
HostsFile2=./private/P25Hosts.txt
ReloadTime=60
ParrotAddress=127.0.0.1
ParrotPort=42011
P252DMRAddress=127.0.0.1
P252DMRPort=42012
Static=65105
RFHangTime=120
NetHangTime=60
Debug=0

[Remote Commands]
Enable=0
Port=6074
```

Editing P25Hosts

# While in /etc we are also going to edit the P25Hosts.txt file with nano.

sudo nano /etc/P25Hosts.txt

# Decide which P25 TG you want to use, pick a number that is not on this list.

# IMPORTANT: The format you will use:

Your TG #(tab)IP Address or DNS(tab) Port #:41000

Testing

# Using cd, change to /opt/P26Gateway:

cd /opt/P25Gateway

# Run the command to activate the P25Gateway:

./P25Gateway /etc/P25Gateway.ini &

# If done correctly you will see:

```
2024-04-12 06:41:23.623 Opening P25 network connection
2024-04-12 06:41:23.623 Opening P25 network connection

I: 2024-04-12 06:41:44.744 Loaded 265 P25 reflectors
I: 2024-04-12 06:41:44.744 Loaded P25 parrot (TG10)
I: 2024-04-12 06:41:44.744 Loaded P252DMR (TG20)
W: 2024-04-12 06:41:44.744 Cannot open the Id lookup file - DMRIds.dat
I: 2024-04-12 06:41:44.744 Started the DMR Id lookup reload thread
I: 2024-04-12 06:41:44.745 Loaded the audio and index file for en_US
M: 2024-04-12 06:41:44.745 Starting P25Gateway-20200824
```

# You can close the P25Gateway connection by **Ctrl-C**

Ctrl + C

Editing Analog_Bridge_P25

# Using the nano command we will edit the Analog_Bridge_P25.ini in the /etc directory.

sudo nano /etc/Analog_Bridge_P25.ini

# Change everything in: **PURPLE** * * *The Analog_Bridge_P25.ini will look like this:

```
 GNU nano 5.4
/etc/Analog_Bridge_P25.ini

; Analog_Bridge configuration file.
;   This file should be placed in either /etc or it can be
;   supplied as a path on the Analog_Bridge command line.
; The best way to think of this file is that there are two main
sections, the USRP (analog audio) and
; AMBE_AUDIO (compressed audio).  Analog_Bridge will take everything
sent to the USRP port and
; encode it for transmission on the AMBE_AUDIO port.  The encoder will
use the format defined in ambeMode.
; As you might expect, the reverse direction is defined too.

; Analog_Bridge supports include files. More on this later...
include = dvsm.macro
; include = asl.macro


; General Section describes settings for Analog_Bridge itself.

[GENERAL]
logLevel = 2                            ; Show messages and above 0=No
logging, 1=Debug, 2=Message, 3=Info, 4=Warning, 5=Error, 6=Fatal

; Metadata management
exportMetadata = true                   ; Export metadata to USRP
partner (transcode setups require this)
transferRootDir = /tmp                  ; Export database files to
USRP partner
subscriberFile = /var/lib/dvswitch/subscriber_ids.csv   ; DMR ID to
callsign lookup data

; General vocoder setup information
decoderFallBack = true                  ; Allow software AMBE decoding
if a hardware decoder is not found
useEmulator = false                     ; Use the MD380 AMBE emulator
for AMBE72 (DMR/YSFN/NXDN)
emulatorAddress = 127.0.0.1:2470        ; IP address and port of the
md380 server

pcmPort = 2222                          ; UDP port to send to the
WebProxy

; Information for xx_Bridge (Where xx is MMDVM, Quantar, HB, IPSC)

[AMBE_AUDIO]
address = 127.0.0.1                     ; IP address of xx_Bridge
txPort = 31103 --> 34103                         ; Transmit TLV frames
to partner on this port
rxPort = 31100 --> 34100                         ; Listen for TLV
frames from partner on this port
ambeMode = DMR --> P25                         ; DMR, DMR_IPSC, DSTAR,
NXDN, P25, YSFN, YSFW (encode PCM to this format)
```

```
minTxTimeMS = 2500                    ; Analog -> Digital Minimum
time in MS for hang delay (0-10000)

; The metadata below is used when ASL is the source since it does not
have any concept of digital modes
gatewayDmrId = Your DMR ID
; ID to use when transmitting from Analog_Bridge 7 digit ID
repeaterID = Your DMR ID + 2 Digits                    ; ID of source
repeater 7 digit ID plus 2 digit SSID
txTg = 9 -> Change to your P25 TG you placed in P25Hosts.txt
; TG to use for all frames sent from Analog_Bridge -> xx_Bridge
txTs = 2                              ; Slot to use for frames sent
from Analog_Bridge -> xx_Bridge
colorCode = 1                         ; Color Code to assign DMR
frames

; Information for USRP channel driver.  This interface uses PCM to
transfer audio information
; There are two typical configurations, ASL and Transcode.  ASL
(AllstarLink) is for analog clients connected
; to a digital network.  Transcode is when Analog_Bridge actually
points its PCM interfaces back at itself,
; causing a TLV <-- (pcm <--> pcm) --> TLV type of architecture.
;   When using ASL, this matches the rpt.conf ASL file with a setting
like:
;   rxchannel = usrp/127.0.0.1:34001:32001
;   When Transcoding, make two ini files and set txPort equal to the
other instance rxPort (crossover).  Launch
;   each instance with its own ini file.

[USRP]
address = 127.0.0.1                   ; IP address of USRP partner
(Analog_Reflector, Allstar/Asterisk or another Analog_Bridge)
txPort = 31001 --> 34001                        ; Transmit USRP
frames on this port
rxPort = 31001 --> 32001                        ; Listen for USRP
frames on this port
usrpAudio = AUDIO_UNITY               ; Digital -> Analog

(AUDIO_UNITY, AUDIO_USE_GAIN, AUDIO_USE_AGC)
usrpGain = 1.10                       ; Gain factor when usrpAudio =
AUDIO_USE_GAIN (0.0 to 5.0) (1.0 = AUDIO_UNITY)
usrpAGC = -20,10,100                  ; Set the agc threshold (db),
slope (db) and decay (ms)
tlvAudio = AUDIO_UNITY                ; Analog -> Digital
(AUDIO_UNITY, AUDIO_USE_GAIN, AUDIO_BPF)
```

Editing Analog_Bridge_DMR

# Edit the /etc/Analog_Bridge_DMR.ini with nano.

sudo nano /etc/Analog_Bridge_DMR.ini

# Change everything in: **PURPLE** * * * The Analog_Bridge_DMR will look like this:

```
 GNU nano 5.4
/etc/Analog_Bridge_DMR.ini

; Analog_Bridge configuration file.
;   This file should be placed in either /etc or it can be
;   supplied as a path on the Analog_Bridge command line.

; The best way to think of this file is that there are two main
sections, the USRP (analog audio) and
; AMBE_AUDIO (compressed audio).  Analog_Bridge will take everything
sent to the USRP port and
; encode it for transmission on the AMBE_AUDIO port.  The encoder will
use the format defined in ambeMode.
; As you might expect, the reverse direction is defined too.

; Analog_Bridge supports include files. More on this later...
include = dvsm.macro
; include = asl.macro

; General Section describes settings for Analog_Bridge itself.

[GENERAL]
logLevel = 2                            ; Show messages and above 0=No
logging, 1=Debug, 2=Message, 3=Info, 4=Warning, 5=Error, 6=Fatal

; Metadata management
exportMetadata = true                   ; Export metadata to USRP
partner (transcode setups require this)
```

```
transferRootDir = /tmp                  ; Export database files to
USRP partner
subscriberFile = /var/lib/dvswitch/subscriber_ids.csv   ; DMR ID to
callsign lookup data

; General vocoder setup information
decoderFallBack = true                  ; Allow software AMBE decoding
if a hardware decoder is not found
useEmulator = false                     ; Use the MD380 AMBE emulator
for AMBE72 (DMR/YSFN/NXDN)
emulatorAddress = 127.0.0.1:2470        ; IP address and port of the
md380 server

pcmPort = 2222                          ; UDP port to send to the
WebProxy

; Information for xx_Bridge (Where xx is MMDVM, Quantar, HB, IPSC)

[AMBE_AUDIO]
address = 127.0.0.1                     ; IP address of xx_Bridge
txPort = 31103                          ; Transmit TLV frames to
partner on this port
rxPort = 31100                          ; Listen for TLV frames from
partner on this port
ambeMode = DMR                          ; DMR, DMR_IPSC, DSTAR, NXDN,
P25, YSFN, YSFW (encode PCM to this format)
minTxTimeMS = 2500                      ; Analog -> Digital Minimum
time in MS for hang delay (0-10000)

; The metadata below is used when ASL is the source since it does not
have any concept of digital modes
gatewayDmrId = Your DMR ID (7 digits only)                  ; ID to
use when transmitting from Analog_Bridge 7 digit ID
repeaterID = Your DMR ID + 2 digit ID                 ; ID of source
repeater 7 digit ID plus 2 digit SSID
txTg = 9 --> Your DMR TG Here                                ; TG to
use for all frames sent from Analog_Bridge -> xx_Bridge
txTs = 2                                ; Slot to use for frames sent
from Analog_Bridge -> xx_Bridge
colorCode = 1                           ; Color Code to assign DMR
frames

; Information for USRP channel driver.  This interface uses PCM to
transfer audio information
; There are two typical configurations, ASL and Transcode.  ASL
(AllstarLink) is for analog clients connected
; to a digital network.  Transcode is when Analog_Bridge actually
points its PCM interfaces back at itself,
; causing a TLV <-- (pcm <--> pcm) --> TLV type of architecture.
;   When using ASL, this matches the rpt.conf ASL file with a setting
like:
;   rxchannel = usrp/127.0.0.1:34001:32001
;   When Transcoding, make two ini files and set txPort equal to the
other instance rxPort (crossover).  Launch
;   each instance with its own ini file.

[USRP]
address = 127.0.0.1                     ; IP address of USRP partner
(Analog_Reflector, Allstar/Asterisk or another Analog_Bridge)
txPort = 31001 --> 32001                          ; Transmit USRP
frames on this port
rxPort = 31001 --> 34001                          ; Listen for USRP
frames on this port
usrpAudio = AUDIO_UNITY                 ; Digital -> Analog
(AUDIO_UNITY, AUDIO_USE_GAIN, AUDIO_USE_AGC)
usrpGain = 1.10                         ; Gain factor when usrpAudio =
AUDIO_USE_GAIN (0.0 to 5.0) (1.0 = AUDIO_UNITY)
usrpAGC = -20,10,100                    ; Set the agc threshold (db),
slope (db) and decay (ms)
tlvAudio = AUDIO_UNITY                  ; Analog -> Digital
(AUDIO_UNITY, AUDIO_USE_GAIN, AUDIO_BPF)
```

Editing MMDVM_Bridge.ini

# Edit the MMDVM_Bridge.ini in /etc with nano:

sudo nano /etc/MMDVM_Bridge.ini

# Change everything in: **PURPLE** * * * The MMDVM_Bridge.ini will look like this:

# BM Servers to choose from in the **[DMR Network]** section:

Master	Domain	IP Address	Location
3102	3102.repeater.net	74.91.114.19	Dallas, Texas
3103	3103.repeater.net	74.91.118.251	San Jose, California
3104	3104.repeater.net	162.248.88.117	Chicago, Illinois

```
General]
Callsign=N0CALL --> Change to your callsign
Id=1234567 --> Change to your DMR ID + 2 Digits
Timeout=180
Duplex=0

[Info] --> Change the info section if you want BM to reflect this
information
RXFrequency=222340000
TXFrequency=224940000
Power=1
Latitude=41.7333
Longitude=-50.3999
Height=0
Location=Iceberg, North Atlantic
Description=MMDVM_Bridge
URL=https://groups.io/g/DVSwitch

[Log]
# Logging levels, 0=No logging, 1=Debug, 2=Message, 3=Info, 4=Warning,
5=Error, 6=Fatal
DisplayLevel=1
FileLevel=2
FilePath=/var/log/mmdvm
FileRoot=MMDVM_Bridge

[DMR Id Lookup]
File=/var/lib/mmdvm/DMRIds.dat
Time=24
```

```
[NXDN Id Lookup]
File=/var/lib/mmdvm/NXDN.csv
Time=24

[Modem]
Port=/dev/null
RSSIMappingFile=/dev/null
Trace=0
Debug=0

[D-Star]
Enable=0
Module=B

[DMR]
Enable=0 --> Change from “0” to “1”
ColorCode=1
EmbeddedLCOnly=1
DumpTAData=0

[System Fusion]
Enable=0

[P25]
Enable=0 --> Change from “0” to “1”
NAC=293 --> When programming your radio make sure the NAC matches your
HotSpot or repeater.

[NXDN]
Enable=0
RAN=1
Id=12345

[D-Star Network]
Enable=0
GatewayAddress=127.0.0.1
GatewayPort=20010
LocalPort=20011
Debug=0

[DMR Network]
Enable=0
Address=hblink.dvswitch.org --> Change to Brandmeister server in your
area.
Port=62031
Jitter=360
Local=62032
Password=passw0rd --> Change to your BM password.
```

```
# for DMR+ see https://github.com/DVSwitch/MMDVM_Bridge/blob/master/
DOC/DMRplus_startup_options.md
# for XLX the syntax is: Options=XLX:4009
# Options=
Slot1=0
Slot2=1
Debug=0

[System Fusion Network]
Enable=0
LocalAddress=0
LocalPort=3200
GatewayAddress=127.0.0.1
GatewayPort=4200
Debug=0

[P25 Network]
Enable=0 --> Change from “0” to “1”
GatewayAddress=127.0.0.1
GatewayPort=42020
LocalPort=32010
Debug=0

[NXDN Network]
Enable=0
#LocalAddress=127.0.0.1
Debug=0
LocalPort=14021
GatewayAddress=127.0.0.1
GatewayPort=14020
```

MMDVM_Bridge Testing

# Login as root:

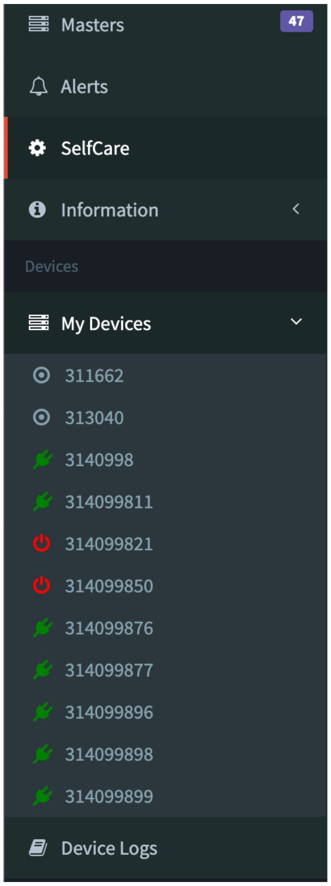
sudo su - root

# Using the cd command change directories to /opt/MMDVM_Bridge

cd /opt/MMDVM_Bridge

# Once in the /opt/MMDVM_Bridge directory use this command to test the connection to BM.

sudo ./MMDVM_Bridge /etc/MMDVM_Bridge.ini &

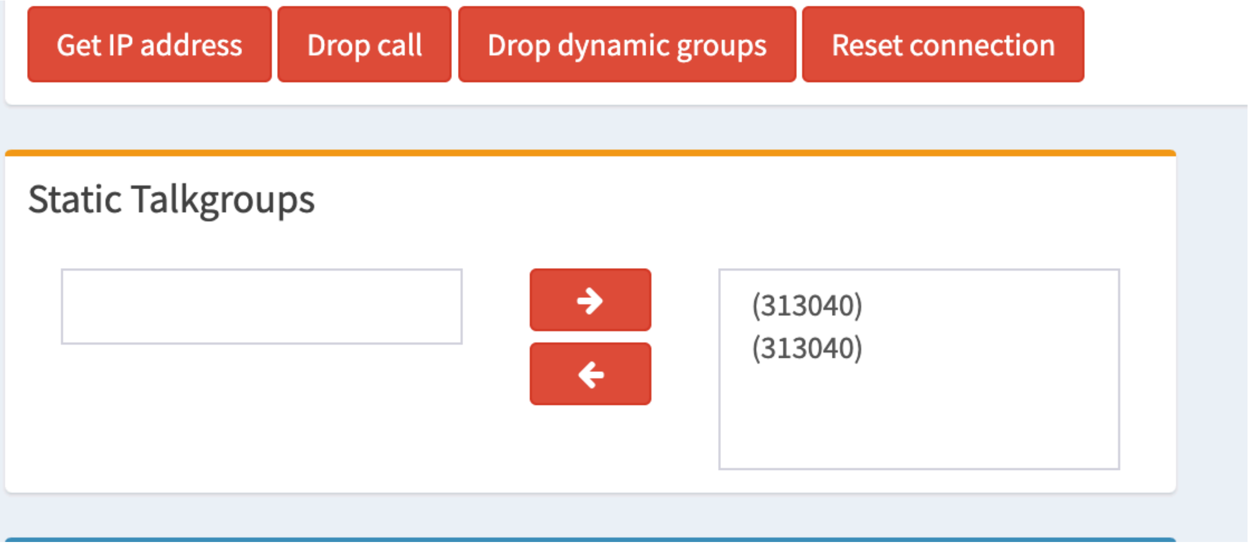


# In a moment, "MMDVM_Bridge is Running" should appear, signaling a successful connection to the BM master. Navigate to your Brandmeister dashboard to find your newly registered 'hotspot' under "My Devices". If this doesn't work, review the MMDVM_Bridge.ini file for any errors.

# You will see it connected under your devices, of course your DMR ID will be different.

Configuring Brandmeister

# After confirming the connection of MMDVM_Bridge to BrandMeister, proceed to set up a static talk group for your hotspot. This ensures the smooth flow of DMR traffic from BrandMeister to your hotspot.
Select the identical DMR talk group as specified in the ANALOG_BRIDGE_DMR.ini file.



(313040)
(313040)

Configuring Pi-Star

# Optional step for setting up a hotspot running Pi-Star to connect to your bridged Talk group. You'll need the IP address or FQDN that Hotspots will use to access your reflector.

# SSH into your Pi-Star via the method of your choosing, and set the filesystem to read/write:

```
rpi-rw
```

# Get permissions to modify files:

```
sudo su
```

# Add your reflector to the P25Hosts file:

```
sudo nano /root/P25Hosts.txt
```

# Enter your information in the format: {TG#}tab{ReflectorIP}tab(41000) - there are also instructions shown in the file.

Save and Exit.

# Force Pi-star to re-load the P25Host file:

```
sudo pistar-update
```

When the update completes, exit the pi-star SSH session, and pull up the Dashboard. You should see the TG and IP domain in the "P25 Startup Host" on the Pi-Star configuration screen. You can set this as your startup now or not...

Running the DMR to P25 Bridge

```
# Let's get started. Begin by rebooting your device for a fresh start.

# Then, open each of these commands in separate terminal windows initially. This will facilitate troubleshooting.

cd /usr/local/bin/ && ./P25Reflector.sh start /etc/P25Gateway.ini

cd /opt/P25Gateway && ./P25Gateway /etc/P25Gateway.ini

cd /opt/Analog_Bridge && ./Analog_Bridge_DMR /etc/Analog_Bridge_DMR.ini

cd /opt/Analog_Bridge && ./Analog_Bridge_P25 /etc/Analog_Bridge_P25.ini

sudo systemctl stop mmdvm_bridge.service && cd /opt/MMDVM_Bridge && ./MMDVM_Bridge /etc/MMDVM_Bridge.ini
```