ASL to DMR Bridge how to

We are going to take you through installing one of the most requested DVSwitch bridge types, an Analog to DMR bridge. Throughout this document, we will use real node numbers, real DMR IDs and real TG numbers. Where possible those values will be green. Please substitute your numbers. We will start with a already running ASL node and add the required parts to bridge it to BrandMeister DMR.

Throughout this document we will be testing along the way. If you do not get the expected results as shown, stop, check your work and if it all looks right, ask for help. It is a LOT easier to troubleshoot at the points along the way rather than at the end.

Everything we will do from this point on needs root privileges, so once logged in as user repeater:

Gain root privileges

```
sudo -s
```

Change to the asterisk configuration file directory:

```
cd /etc/asterisk
```

Check if asterisk is running:

```
systemctl status asterisk
```

It is much easier to make your changes and then run the program in the foreground to troubleshoot before continuing.

Stop asterisk and get its status:

```
systemctl stop asterisk
systemctl status asterisk
```
ASL will support multiple nodes on one server. For the bridge, this allows us to easily connect and disconnect ASL from DMR. Not all nodes on ASL need be in the public node list. These type of nodes are called private nodes. We are going to add a private node to ASL. This minimizes the changes required to ASL. The settings for node 1999 are selected to keep unwanted sounds from being sent to DMR from ASL.

Edit `/etc/asterisk/rpt.conf`
Copy the `[1999]` stanza from https://dvswitch.groups.io/g/allstarlink/wiki/home
Paste the stanza for the private node at the top of rpt.conf file.

Next we need to add the configuration for node 1999
In the `[nodes]` stanza:

```diff
[1999]
rxchannel = USRP/127.0.0.1:34001:32001  ; Use the USRP channel driver. Must be enabled in modules.conf
duplex = 0                              ; 0 = Half duplex with no telemetry tones or hang time. Ah, but Allison STILL talks!
hangtime = 0                            ; squelch tail hang time 0
althangtime = 0                         ; longer squelch tail hang time 0
holdofftelem = 1                        ; Hold off all telemetry when signal is present on receiver or from connected nodes except when an ID needs to be done and there is a signal coming from a connected node.
telemdefault = 0                        ; 0 = telemetry output off. Don’t send Allison to DMR !!!!!!!!!!!!!!!! Trust me.
telemdynamic = 0                        ; 0 = disallow users to change the local telemetry setting with a COP command,
linktolink = no                         ; disables forcing physical half-duplex operation of main repeater while still keeping half-duplex semantics (optional)
nounkeyct = 1                           ; Set to a 1 to eliminate courtesy tones and associated delays.
totime = 180000                         ; transmit time-out time (in ms) (optional, default 3 minutes 180000 ms)
idrecording = |ie                      ; id recording or morse string see http://ohnosec.org/drupal/node/87
idtalkover = |ie                        ; Talkover ID (optional) default is none see http://ohnosec.org/drupal/node/129
```
Add:

\[
1999 = \text{radio@127.0.0.1:4569/1999,NONE}
\]

The [nodes] stanza should look like this:

```
[nodes]
; Note, if you are using automatic update for allstar link nodes,
; no allstar link nodes should be defined here. Only place a definition
; for your local nodes, and private (off of allstar link) nodes here.

2100 = radio@127.0.0.1:4569/2100,NONE   ; This must be changed to your node number
1999 = radio@127.0.0.1:4569/1999,NONE
```

We are finished editing rpt.conf.

Next we must enable the USRP channel driver.
Edit /etc/asterisk/modules.conf
In the [modules] stanza
Change:
noload => chan_usrp.so
To:
load => chan_usrp.so

The line should look like this:

```
load => chan_usrp.so ; GNU Radio interface USRP Channel Driver
```

Now let's check our work. Run asterisk in the foreground

```
estarisk -c
```

```
root@repeater:/etc/asterisk# asterisk -c
```

You can see both the private node (1999) and the public node (2100) have started properly.
Normal Repeater Init  1999
Normal Repeater Init  2100
Exit asterisk with a Control-C

We will now add the private node to /etc/asterisk/extensions.conf
Edit /etc/asterisk/extensions.conf
in the [globals] stanza
add:
\texttt{NODE1 = 1999}
in the [radio-secure] stanza
add:
\texttt{exten => \{NODE1\},1,rpt,\{NODE1\}}

\begin{verbatim}
[globals]
HOMENPA = 999   ; change this to your Area Code
NODE = 2100     ; change this to your node number
NODE1 = 1999    ; This is the private node
[default]
exten => i,1,Hangup
[radio-secure]
exten => \{(NODE)\},1,rpt,\{(NODE)\}
exten => \{(NODE1)\},1,rpt,\{(NODE1)\}
\end{verbatim}

This completes the changes needed to be made to ASL Time to check our work.

Run asterisk in the foreground

\texttt{asterisk -c}

We can now test the ability to connect the public node (\texttt{2100}) to the private node (1999) that will later be the connection to DMR

At the CLI\textgreater{} prompt type:

\texttt{rpt cmd 2100 ilink 3 1999}

This tells ASL to connect node \texttt{2100} to node 1999 in tranceive mode.

Let's look at the statistics for node \texttt{2100}. At the CLI\textgreater{} prompt type

\texttt{rpt stats 2100}
Notice the line:

Nodes currently connected to us: 1999

Node 1999 is connected to node 2100. This tells us that our ASL configuration is correct. Press Control-C to exit asterisk.

It is best practice to build from the outside in. From shore to shore, so to speak. We can now move on to connecting to BrandMeister.

Let's install the DVSwitch programs needed to connect ASL to BrandMeister:
Make sure your repository list is up to date before trying to install more programs.

Run:

```bash
apt-get update
```
If you see no errors let's add the DVSwitch programs:

The ASL and DVSwitch Repository are both hosted on dvswitch.org so the DVSwitch programs are available.

Run:

```
apt-get install dvswitch -y
```
This installs Analog_Bridge, MMDVM_Bridge, and md380-emu. The scripts and datafiles needed to support these programs are also installed. Each of the programs is set to start at boot. You don’t have to do anything.

Before we start, Let’s take a look at how this all goes together. Usually in the message board you will see a simple ASCII diagram like this:

ASL <-> Analog_Bridge <-> MMDVM_Bridge <-> BrandMeister

The <-> denotes two way communications between the programs. This is done via UDP ports. Since the DVSwitch programs can be configured to build a large number of different bridges, the number of options can be staggering to a first time user. Focus on the project at hand. In reality there are not too many changes that need to be made. We have tried to be consistent in the naming between programs and to set the default value as someone would use in a simple bridge. Here is a port diagram of what you will be building:

From left to right, we start with The [1999] node stanza. Each node has a rxchannel. This defines how the node communicates. In this case, we will use USRP to communicate with Analog_Bridge. 127.0.0.1 is the address that ASL will send data to Analog_Bridge. 34001 is the UDP port that ASL will send data on. 32001 is the port that ASL will listen for data on.

In Analog_Bridge.ini the [USRP] stanza defines how Analog_Bridge will communicate with another USRP Partner. In this case ASL. 127.0.0.1 is the address that Analog_Bridge will send data to. Port 32001 is the port that Analog_Bridge will send data to ASL on. Port 34001 is the port that Analog_Bridge will listen for data from ASL on.
In Analog_Bridge.ini the [AMBE_AUDIO] stanza defines how Analog_Bridge will communicate with a digital Partner. In this case MMDVM_Bridge. 127.0.0.1 is the address that Analog_Bridge will send data to. Port 31103 is the port that Analog_Bridge will send data to MMDVM_Bridge on. Port 31100 is the port that Analog_Bridge will listen for data from MMDVM_Bridge on.

MMDVM_Bridge is the program that connects to BrandMeister. Let’s check it’s status:

```
root@repeater:/etc/asterisk# systemctl status mmdvm_bridge
● mmdvm_bridge.service - MMDVM_Bridge Service
 Loaded: loaded (/lib/systemd/system/mmdvm_bridge.service; enabled; vendor preset: enabled)
 Active: inactive (dead)
```

You will see the status of the mmdvm_bridge service

Good, MMDVM_Bridge is not running though it is enabled, meaning it will start at boot.

Change directory to /opt/MMDVM_Bridge

```
cd /opt/MMDVM_Bridge
```

Let’s see what files are in the directory /opt/MMDVM_Bridge

```
ls
```

You will see the program and it’s two configuration files.

We will now edit the callsign and Id

Edit /opt/MMDVM_Bridge/MMDVM_Bridge.ini
In the [General] stanza
Change:
Callsign=W1AW
To your callsign:
Callsign=N4IRS
Change:
Id=1234567
To your DMRID + 2 digit SSID
Id=311213805

The [General] stanza will look like this:

```
[General]
Callsign=N4IRS
```
In the [General] stanza, the callsign is your callsign. For the Id, It is STRONGLY recommended to use your DMRID and append a 2 digit number between 01 and 99. These 2 digits are referred to as SSID For this example that’s 3112138 plus 05. So the Id in the [General] stanza is 311213805. This is how MMDVM_Bridge logs into a BrandMeister Master. This will make it MUCH easier for you to control any static Talk Groups you want for your bridge.

We will now enable the DMR mode in MMDVM_Bridge.
In the [DMR] stanza
Change
Enable=0
To:
Enable=1

The [DMR] stanza will look like this:

```
[DMR]
Enable=1
ColorCode=1
EmbeddedOnly=1
DumpTAData=0
```

In this example, we will connect to the BrandMeister Master at 3108.repeater.net. You should use YOUR regional Master here.
In the [DMR_Network] stanza
Change:
Address=dvswitch.org
To:
Address=3108.repeater.net

The [DMR Network] stanza will look like this:

```
[DMR Network]
Enable=1
Address=3108.repeater.net
Port=62031
Jitter=360
Local=62032
Password=passw0rd
Slot1=0
Slot2=1
Debug=0
```

Time to test

Run MMDVM_Bridge in the foreground
I: 2018-06-04 14:12:48.435 MMDVM_Bridge: Portions Copyright (C) 2018 DVSwitch, INAD.
I: 2018-06-04 14:12:48.436 Hacks by Mike N4IRR and Steve N4IRS
I: 2018-06-04 14:12:48.436 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
I: 2018-06-04 14:12:48.436 This software is for use on amateur radio networks only, its use on commercial networks is strictly prohibited.
I: 2018-06-04 14:12:48.436 Copyright(C) 2015-2018 by Jonathan Naylor, G4KLX and others

M: 2018-06-04 14:12:48.437 MMDVM_Bridge-20180423 is starting

I: 2018-06-04 14:12:48.437 General Parameters
I: 2018-06-04 14:12:48.437     Id: 311213805

I: 2018-06-04 14:12:48.440     TX Delay: 100ms
I: 2018-06-04 14:12:48.440     RX Offset: 0Hz
I: 2018-06-04 14:12:48.440     TX Offset: 0Hz
I: 2018-06-04 14:12:48.440     RX DC Offset: 0
I: 2018-06-04 14:12:48.440     TX DC Offset: 0
I: 2018-06-04 14:12:48.440     RF Level: 100.0%
I: 2018-06-04 14:12:48.441     DMR Delay: 0 (0.0ms)
I: 2018-06-04 14:12:48.441     RX Frequency: 222340000Hz
I: 2018-06-04 14:12:48.441     TX Frequency: 224940000Hz
I: 2018-06-04 14:12:48.441     Height: 0m
I: 2018-06-04 14:12:48.441     Location: "Iceberg, North Atlantic"
I: 2018-06-04 14:12:48.441     Description: "MMDVM_Bridge"
I: 2018-06-04 14:12:48.441     URL: "https://groups.io/g/DVSwitch"

I: 2018-06-04 14:12:48.479 RSSI
W: 2018-06-04 14:12:48.479 Cannot open the RSSI data file - RSSI.dat

I: 2018-06-04 14:12:48.479 DMR Id Lookups
I: 2018-06-04 14:12:48.479     Reload: 24 hours

M: 2018-06-04 14:12:48.479 Loaded 100306 Ids to the DMR callsign lookup table
Here are the important lines for now:

This verifies the Callsign and the Id used to login to BrandMeister. The BrandMeister Master Address and Port you are connected to and that you have successfully logged in to the Master.

Let's take a look at BM. Login to your selfcare page at brandmeister.network. On the left hand side you should see “My hotspots”
Notice that **311213805** is showing the green “plug” icon. That means **311213805** is connected to BrandMeister. Next let's take a quick look at its edit page. Click on your hotspot number. For the example, that's **311213805**. This will take you to the edit page. In the center of the page you should see Sysops. You should be the only sysop for now. If you got this far you are most of the way there.

While we are here, we can setup a Static Talk Group to pass through the bridge. We will add the Talk Group number here:

Enter your Talk Group number in the box on the left and click on the right arrow. It will look like this, for the example I used Talkgroup **311222**:
At this point, you have both sides of bridge installed and working. Now it is time to connect them together. Since one network is analog and the other network is digital we need to convert the audio between the modes. That is the job of Analog_Bridge.

Press Control-C to stop MMDVM_Bridge.

Check the status of Analog_Bridge

```bash
systemctl status analog_bridge
```

Analog_Bridge is enabled and not running.

Change to the `/opt/Analog_Bridge` directory

```bash
cd /opt/Analog_Bridge
```

Let’s see what files are in the directory `/opt/Analog_Bridge`

```bash
ls
```

You should see the program Analog_Bridge and it’s configuration file Analog_Bridge.ini

Edit `/opt/Analog_Bridge/Analog_Bridge.ini` in the `[GENERAL]` stanza change:

decoderFallBack = true

The `[GENERAL]` stanza will look like this:

```ini
; General Section describes settings for Analog_Bridge itself.
```
The [AMBE_AUDIO] stanza sets up the connection to the Partner program. In this example, MMDVM_Bridge. RXPort is the UDP port that Analog_Bridge will listen to for data from MMDVM_Bridge. Address and TXPort are where Analog_Bridge will send data to MMDVM_Bridge. The default values for Address, RXPort and TXPort are correct for connecting to the [DMR] stanza in MMDVM_Bridge.ini gatewayDmrId is the DMRID that all analog traffic from ASL will appear to come from when someone is listening on DMR. repeaterID should match the ID you used to log into BM. txTg is the Talk Group that traffic from ASL will be sent to BrandMeister. txTs is the Time Slot the traffic from ASL will be on.

In the [AMBE_AUDIO] stanza
Change:

<table>
<thead>
<tr>
<th>gatewayDmrId = 3112138</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeaterID = 311213805</td>
</tr>
<tr>
<td>txTg = 311222</td>
</tr>
<tr>
<td>txPort = 31103</td>
</tr>
<tr>
<td>rxPort = 31100</td>
</tr>
</tbody>
</table>

Remember the values in green are for the example. You should use your values here.

The [AMBE_AUDIO] stanza will look like this:

| ; 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 = 2000      | Minimum time in MS for hang delay (0-10000)      |
| gatewayDmrId = 3112138  | ID to use when transmitting from Analog_Bridge   |
| repeaterID = 311213805  | ID of source repeater                           |
| txTg = 311222           | 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                  |

Now we will look at the Analog side of the connection:

The [USRP] stanza will look like this:
The [USRP] stanza is what is used to connect to ASL. This is also where you can tailor the audio in and out of ASL to DMR. For now the defaults will work as is. We are done editing Analog_Bridge. Let’s check our work.

Run Analog_Bridge in the foreground.

```
./Analog_Bridge Analog_Bridge.ini
```

I: 2018-06-04 15:09:34.105 Copyright (C) 2018 DVSwitch, INAD.
I: 2018-06-04 15:09:34.105 Created by Mike N4IRR and Steve N4IRS
I: 2018-06-04 15:09:34.105 Analog Bridge comes with ABSOLUTELY NO WARRANTY
I: 2018-06-04 15:09:34.105 This software is for use on amateur radio networks only, its use on commercial networks is strictly prohibited.
I: 2018-06-04 15:09:34.106
M: 2018-06-04 15:09:34.108 Setting [AMBE_AUDIO] address -> 127.0.0.1
M: 2018-06-04 15:09:34.109 Setting [AMBE_AUDIO] txTg -> 311222
M: 2018-06-04 15:09:34.109 Setting [USRP] address -> 127.0.0.1
M: 2018-06-04 15:09:34.110 Setting [USRP] txPort -> 32001
M: 2018-06-04 15:09:34.110 Setting [USRP] rxPort -> 34001
M: 2018-06-04 15:09:34.110 Setting [USRP] dmrGain -> 0.35
M: 2018-06-04 15:09:34.111 Setting [DV3000] address -> 127.0.0.1:2470
M: 2018-06-04 15:09:34.111 Setting [DV3000] rxPort -> 2460
W: 2018-06-04 15:09:34.111 ioctl reset error
W: 2018-06-04 15:09:34.111 ioctl speed error
W: 2018-06-04 15:09:34.111 ioctl stereo error
W: 2018-06-04 15:09:34.112 ioctl setfmt error
I: 2018-06-04 15:09:34.112 Project 25 IMBE Encoder/Decoder Fixed-Point implementation
Developed by Pavel Yazev E-mail: pyazev@gmail.com
Version 1.0 (c) Copyright 2009
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under certain conditions; see the file "LICENSE" for details
I: 2018-06-06 13:47:08.456 Default extended metadata <3112138>
Let's look at the important information:

Analog_Bridge is sending data to MMDVM_Bridge at Address 127.0.0.1 on port 31103
Analog_Bridge is listening for data from MMDVM_Bridge on port 31100.
Analog_Bridge is sending analog audio to ASL at Address 127.0.0.1 on port 32001
Analog_Bridge is listening for analog audio from ASL on port 34001.
Audio levels to and from ASL are unity. (no gain or reduction)

**Insert Vocoder discussion here:**

Analog_Bridge is capable of using three classes of vocoders

**Hardware**
- DV3000U, PiDV, DVMega AMBE
  - Support for DMR, YSFN, NXDN and D-Star
  - Direct serial and IP (ambe server) modes are supported

**Software**
- The md380-emulator
  - Support for DMR, YSFN, NXDN
- The OP25 vocoder
  - Support for DMR, YSFN, NXDN, D-Star, P25 and YSFW

**Quality vs cost**

Will your CPU support all vocoders?

As described in other sections, we support AMBE, IMBE and DSAMBE audio formats, but there are differing levels of support and quality for these formats.

AMBE is supported in hardware (USB or AMBEServer), md380-emulator on supported environments and OP25 vocoder for all platforms. The md380-emulator is available on x86 and ARM V7 (with div support) which includes the Raspberry Pi 2 and 3, Allwinner H3 and H5 processors.

IMBE is supported in software on all platforms using the OP25 vocoders
DSAMBE (D-Star) is supported in hardware (USB or AMBEServer) and (but you will not be happy) by using the OP25 vocoder.

If you got this far, you are doing well. It’s now time to test everything and see if we missed anything. For this portion you will run all of the bridge programs in the foreground. Each will be in a separate window so you can watch in real time. Again we will work from the outside in. Let’s start asterisk. We will also tell asterisk to use more verbose logging so that we can see the traffic coming from Analog_Bridge.

Open a window on the bridge
Gain root privileges

```
sudo -s
```

At the # prompt:

```
cd /etc/asterisk
```

Run asterisk in the foreground

```
asterisk -c
```

Make the asterisk messages more verbose

From the CLI> prompt

```
rpt debug level 6
```

```
root@repeater:/opt/Analog_Bridge# cd /etc/asterisk
root@repeater:/etc/asterisk# asterisk -c

AllStarLink Asterisk Version 1.01 2/13/2018 GIT Version 004b9dd
Copyright (C) 1999 - 2018 Digium, Inc. Jim Dixon, AllStarLink Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
[ Booting...
[ Reading Master Configuration ]
[ Initializing Custom Configuration Options ]
[Jun 4 11:26:40] NOTICE[1852]: dnsmgr.c:394 do_reload: Managed DNS entries will be refreshed every 300 seconds.
[Jun 4 11:26:40] NOTICE[1852]: loader.c:874 load_modules: 56 modules will be loaded.
........................................................ [ Booting...
[ Reading Master Configuration ]
[ Initializing Custom Configuration Options ]
[Jun 4 11:26:40] NOTICE[1852]: dnsmgr.c:394 do_reload: Managed DNS entries will be refreshed every 300 seconds.
[Jun 4 11:26:40] NOTICE[1852]: loader.c:874 load_modules: 56 modules will be loaded.
........................................................ ]
Asterisk Ready.

*CLI>rpt debug level 6
app_rpt Debugging enabled, previous level: 0, new level: 6

*CLI>
Open a another window on the bridge

Gain root privileges

```
sudo -s
```

At the # prompt:

```
cd /opt/MMDVM_Bridge
```

Start MMDVM_Bridge in the foreground

```
./MMDVM_Bridge MMDVM_Bridge.ini
```

```
root@repeater:# cd /opt/MMDVM_Bridge/
root@repeater:/opt/MMDVM_Bridge# ./MMDVM_Bridge MMDVM_Bridge.ini
I: 2018-06-04 15:28:54.546 Portions Copyright (C) 2018 DVSwitch, INAD.
I: 2018-06-04 15:28:54.546 Hacks by Mike N4IRR and Steve N4IRS
I: 2018-06-04 15:28:54.546 This software is for use on amateur radio networks only,
I: 2018-06-04 15:28:54.547 it is to be used for educational purposes only. Its use on
M: 2018-06-04 15:28:54.547 Copyright(C) 2015-2018 by Jonathan Naylor, G4KLX and others
M: 2018-06-04 15:28:54.547 MMDVM_Bridge-20180423 is starting
M: 2018-06-04 15:28:54.547 Built 15:47:01 May 30 2018 (GitID #9cdea41)
I: 2018-06-04 15:28:54.548     Timeout: 180s
I: 2018-06-04 15:28:54.549     Modem Parameters
I: 2018-06-04 15:28:54.550     RX Invert: no
I: 2018-06-04 15:28:54.550     TX Delay: 100ms
I: 2018-06-04 15:28:54.550     RX Offset: 0Hz
I: 2018-06-04 15:28:54.550     TX Offset: 0Hz
I: 2018-06-04 15:28:54.550     RX DC Offset: 0
I: 2018-06-04 15:28:54.550     TX DC Offset: 0
I: 2018-06-04 15:28:54.551     RF Level: 100.0%
I: 2018-06-04 15:28:54.551     DMR Delay: 0 (0.0ms)
I: 2018-06-04 15:28:54.551     RX Level: 50.0%
I: 2018-06-04 15:28:54.551     CW Id TX Level: 50.0%
I: 2018-06-04 15:28:54.551     D-Star TX Level: 50.0%
I: 2018-06-04 15:28:54.551     DMR TX Level: 50.0%
I: 2018-06-04 15:28:54.551     YSF TX Level: 50.0%
I: 2018-06-04 15:28:54.551     P25 TX Level: 50.0%
I: 2018-06-04 15:28:54.551     NXDN TX Level: 50.0%
I: 2018-06-04 15:28:54.551     RX Frequency: 222340000Hz (222340000Hz)
I: 2018-06-04 15:28:54.551     TX Frequency: 2224940000Hz (2224940000Hz)
I: 2018-06-04 15:28:54.552     RX Frequency: 222340000Hz (222340000Hz)
I: 2018-06-04 15:28:54.552     TX Frequency: 2224940000Hz (2224940000Hz)
Open another window on the bridge

Gain root privileges

```
sudo -s
```

At the # prompt:

```
cd /opt/Analog_Bridge
```

Run Analog_Bridge in the foreground

```
./Analog_Bridge Analog_Bridge.ini
```
I: 2018-06-04 15:36:45.328 Analog Bridge comes with ABSOLUTELY NO WARRANTY
I: 2018-06-04 15:36:45.328 This software is for use on amateur radio networks only, it is to be used for educational purposes only. Its use on commercial networks is strictly prohibited.

I: 2018-06-04 15:36:45.329 Analog Bridge is starting
M: 2018-06-04 15:36:45.330 Setting [AMBE_AUDIO] address -> 127.0.0.1
M: 2018-06-04 15:36:45.331 Setting [AMBE_AUDIO] gatewayDmrId -> 3112138
M: 2018-06-04 15:36:45.332 Setting [USRP] address -> 127.0.0.1
M: 2018-06-04 15:36:45.332 Setting [USRP] txPort -> 32001
M: 2018-06-04 15:36:45.332 Setting [USRP] rxPort -> 34001
M: 2018-06-04 15:36:45.332 Setting [USRP] dmrGain -> 0.35
M: 2018-06-04 15:36:45.333 Setting [DV3000] address -> 127.0.0.1
M: 2018-06-04 15:36:45.333 Setting [DV3000] rxPort -> 2460
W: 2018-06-04 15:36:45.334 ioctl reset error
W: 2018-06-04 15:36:45.334 ioctl speed error
W: 2018-06-04 15:36:45.334 ioctl stereo error
W: 2018-06-04 15:36:45.334 ioctl setfmt error
M: 2018-06-04 15:36:45.335 Audio In/Out Device: /dev/null
I: 2018-06-04 15:36:45.335 Open UDP listener on 127.0.0.1:31100
I: 2018-06-04 15:36:45.335 Open USRP on 127.0.0.1:32001
M: 2018-06-04 15:36:45.335 Connecting to DV3000 hardware......
W: 2018-06-04 15:36:46.343 DV3000 not found at 127.0.0.1:2460

Project 25 IMBE Encoder/Decoder Fixed-Point implementation
Developed by Pavel Yazev E-mail: pyazev@gmail.com
Version 1.0 (c) Copyright 2009
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see the file 'LICENSE' for details.
I: 2018-06-04 15:36:46.344 Default extended metadata <3112138>
I: 2018-06-04 15:36:46.344 Subscriber IDs file not found.
I: 2018-06-04 15:36:46.344 Default extended metadata <3112138>
I: 2018-06-04 15:36:46.344 Connecting to emulator on host 127.0.0.1:2470
W: 2018-06-04 15:36:46.345 Using software MBE decoder version 1.2.3
I: 2018-06-04 15:36:46.345 Starting Analog_Bridge --> USRP thread
I: 2018-06-04 15:36:46.346 Open UDP listener on 127.0.0.1:34001
I: 2018-06-04 15:36:46.346 Starting USRP --> HB_Bridge thread

I will use my DMR radio and HotSpot to transmit on Time Slot 2 Talk Group 311222 to generate traffic

Go over to the MMDVM_Bridge window

M: 2018-06-04 15:51:24.087 MMDVM_Bridge-20180423 is running
I: 2018-06-04 15:51:24.087 Started the DMR Id lookup reload thread
D: 2018-06-04 15:51:34.153 DMR, Sending authorisation
D: 2018-06-04 15:51:34.218 DMR, Sending configuration
M: 2018-06-04 15:51:34.246 DMR, logged into the master successfully
M: 2018-06-04 15:52:04.825 DMR Slot 2, received network voice header from N4IRS to TG 311222
M: 2018-06-04 15:52:05.380 DMR Slot 2, received network end of voice transmission, 0.5 seconds, 0% packet loss, BER: 0.0%

In the above output from MMDVM_Bridge, Traffic from N4IRS on Slot 2, Talk Group 311222 has been received. Notice the time stamp: 2018-06-04 15:52:04.825
Go over to the Analog_Bridge window

In the above output from Analog_Bridge, traffic is received from MMDVM_Bridge. Notice the timestamp: 2018-06-04 15:52:04.825. BINGO! We have passed traffic from BrandMeister to Analog_Bridge through MMDVM_Bridge.

Look at the ASL window. You should see:

```
root@repeater:/opt/Analog_Bridge# cd /etc/asterisk
root@repeater:/etc/asterisk# asterisk -c
AllStarLink Asterisk Version 1.01 2/13/2018 GIT Version 004b9dd
Copyright (C) 1999 - 2018 Digium, Inc. Jim Dixon, AllStarlink Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
Asterisk comes with components licensed under the GNU General Public License version 2 and other licenses; you are welcome to redistribute it under certain conditions. Type 'core show license' for details.
=========================================================================   [ Booting...   [ Reading Master Configuration ]   [ Initializing Custom Configuration Options ]   [Jun 4 11:26:40] NOTICE[1852]: dnsmgr.c:394 do_reload: Managed DNS entries will be refreshed every 300 seconds.
-Jun 4 11:26:40] NOTICE[1852]: loader.c:874 load_modules: 56 modules will be loaded.
................................................................
Asterisk Ready.
*CLI> [Jun 4 11:52:04] NOTICE[2051]: app_rpt.c:20965 rpt: **** rx key
-Jun 4 11:52:04] NOTICE[2051]: app_rpt.c:21076 rpt: **** rx un-key
-Jun 4 11:52:04] NOTICE[2051]: app_rpt.c:20965 rpt: **** rx key
-Jun 4 11:52:04] NOTICE[2051]: app_rpt.c:21076 rpt: **** rx un-key
Traffic is flowing from DMR to ASL! Did you get the above results? If you did not, stop. Check your work before continuing. If you did, let’s test in the other direction from ASL to DMR. You could use a radio on the ASL node for this but there is a simply way for your first test to do this without an analog radio.
```

Traffic is flowing from DMR to ASL! Did you get the above results? If you did not, stop. Check your work before continuing. If you did, let’s test in the other direction from ASL to DMR. You could use a radio on the ASL node for this but there is a simply way for your first test to do this without an analog radio.

While still logged in to asterisk at the CLI> prompt
Tell asterisk to send a 1 KHz test tone.

```
*CLI> rpt cmd 1999 cop 4 1999
Traffic is flowing from DMR to ASL! Did you get the above results? If you did not, stop. Check your work before continuing. If you did, let’s test in the other direction from ASL to DMR. You could use a radio on the ASL node for this but there is a simply way for your first test to do this without an analog radio.

```

Traffic is flowing from DMR to ASL! Did you get the above results? If you did not, stop. Check your work before continuing. If you did, let’s test in the other direction from ASL to DMR. You could use a radio on the ASL node for this but there is a simply way for your first test to do this without an analog radio.

While still logged in to asterisk at the CLI> prompt
Tell asterisk to send a 1 KHz test tone.

```
*CLI> rpt cmd 1999 cop 4 1999
Look at the Analog_Bridge windows, you should see something like this:

M: 2018-06-05 14:03:09.787 PTT on

Look at the MMDVM_Bridge windows, you should see something like this:

M: 2018-06-05 14:05:13.971 DMR, TX state = ON
I: 2018-06-05 14:05:13.971 DMR, Begin TX: src=3112138 rpt=311213805 dst=311222 slot=2 cc=1 metadata=3112138

If you still have your DMR radio and HotSpot connected, you should hear the tone coming out of your radio. You do? We have liftoff!

Let’s stop the annoying tone.

While still logged in to asterisk at the CLI> prompt
Tell asterisk to STOP sending the 1 KHz test tone.

*CLI> rpt cmd 1999 cop 4 1999

Yes, the cop 4 command is a toggle, once on and once off. Please remember to turn it off.

Congratulations, You have built your very own ASL <-> DMR bridge. We hope that was not too bad and you learned a few things along the way. Just a few little cleanup tasks to do and you are ready to run.

The debug level in asterisk will quickly fill your log file with mostly unneeded information. Let’s set it back to a sane value.

Make the asterisk messages less verbose
From the CLI> prompt
You probably want your public node to connect to the DMR node when asterisk starts. We will use the startup macro in rpt.conf to do this for you.

Change to the /etc/asterisk directory

```
cd /etc/asterisk
```

Edit rpt.conf
Set the startup macro to connect your public node to your private node by setting startup_macro = *31999 in the [2100] node stanza.

Edit /etc/asterisk/rpt.conf
In the [2100] node stanza
Change:
startup_macro =
To:
startup_macro = *31999

The line should look like this:

```
startup_macro = *31999
```

Now it’s time to make sure everything will start at boot and do what you expect.

We will enable all the services we need just in case.
At the # prompt:

```
systemctl enable asterisk
systemctl enable analog_bridge
systemctl enable md380-emu
systemctl enable mmdvm_bridge
```

If the system was silent, that means those services were already enabled. No harm done.

OK, here we go:
At the # prompt:

```
reboot
```

Once the system is up:
Open a window on the bridge
Gain root privileges
sudo -s

At the # prompt:

systemctl status asterisk

root@repeater:/lib/systemd/system# systemctl status asterisk
● asterisk.service - Asterisk PBX and telephony daemon
Loaded: loaded (/lib/systemd/system/asterisk.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2018-06-06 14:24:18 EDT; 5min ago
Docs: man:asterisk(8)
Main PID: 402 (asterisk)
Tasks: 22 (limit: 4915)
  CGroup: /system.slice/asterisk.service
    └─ 402 /usr/sbin/asterisk -g -f -C /etc/asterisk/asterisk.conf
Jun 06 14:24:18 repeater systemd[1]: Started Asterisk PBX and telephony daemon.

At the # prompt:

systemctl status analog_bridge

root@repeater:/lib/systemd/system# systemctl status analog_bridge
● analog_bridge.service - Analog_Bridge Service
Loaded: loaded (/lib/systemd/system/analog_bridge.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2018-06-06 14:24:18 EDT; 7min ago
Main PID: 426 (Analog_Bridge)
Tasks: 3 (limit: 4915)
  CGroup: /system.slice/analog_bridge.service
     └─ 426 /opt/Analog_Bridge/Analog_Bridge /opt/Analog_Bridge/Analog_Bridge.ini
Jun 06 14:24:18 repeater systemd[1]: Started Analog_Bridge Service.

At the # prompt:

systemctl status md380-emu

root@repeater:/lib/systemd/system# systemctl status md380-emu
● md380-emu.service - MD-380 Emulator Service
Loaded: loaded (/lib/systemd/system/md380-emu.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2018-06-06 14:24:18 EDT; 9min ago
Main PID: 427 (md380-emu)
Tasks: 2 (limit: 4915)
  CGroup: /system.slice/md380-emu.service
       └─ 427 /usr/bin/qemu-arm-static /opt/md380-emu/md380-emu -S 2470
Jun 06 14:24:18 repeater systemd[1]: Started MD-380 Emulator Service.

At the # prompt:

systemctl status mmdvm_bridge

root@repeater:/lib/systemd/system# systemctl status mmdvm_bridge
● mmdvm_bridge.service - MMDVM_Bridge Service
Loaded: loaded (/lib/systemd/system/mmdvm_bridge.service; enabled; vendor preset: enabled)
OK, you have been a very diligent person, time for a little fun
Connect to the running instance of asterisk

Asterisk -r

At the CLI> prompt

rpt cmd 1999 cop 21 1999

Hmmm, nothing happened? You expected fireworks?
Key your DMR radio and identify.

Did you hear yourself (parrot)? That’s right, ASL has a parrot function and it just proved your bridge is working in both directions. Your DMR audio was actually parroted by ASL. A full round trip. A pretty quick and easy way to verify everything is working as expected.

Before you start getting strange phone calls, we better turn off the parrot.

At the CLI> prompt

rpt cmd 1999 cop 22 1999

Get up, walk around, pat yourself on the back. It works! We will cover some basic info and then you can go play. School is not quite out yet.

Below is some basic information you will need to maintain your bridge:

File locations

Programs
In /opt/program_name
/opMMDVM_Bridge/MMDVM_Bridge

Configuration files
With the program
/opMMDVM_Bridge/MMDVM_Bridge.ini

Datafiles
In /var/lib/mmdvm or /var/lib/dvswitch
/var/lib/mmdvm/DMRIds.dat

Logs
In /var/log/mmdvm, /var/log/dvswitch, /var/log/asterisk
/var/log/mmdvm/MMDVM_Bridge-2018-05-22.log

For the log files you can watch what one of the programs in real time by “tailing” the log files. For example:
tail -f /var/log/dvswitch/Analog_Bridge.log will show you what is going on with Analog_Bridge. Key your DMR radio and ID. See the traffic?

Scripts
In /usr/local/sbin
/usr/local/sbin/DMRIDUpdate.sh

Daily cron tasks to update data files
/etc/cron.daily/DMRIDUpdate (this is a symbolic link to the script
In /usr/local/sbin Note: If you want to add a link, the link name should not have a extension (.sh)

Status, stopping, starting, restarting programs:

AllStarLink
systemctl status asterisk
systemctl stop asterisk
systemctl start asterisk
systemctl restart asterisk

Analog_Bridge
systemctl status analog_bridge
systemctl stop analog_bridge
systemctl start analog_bridge
systemctl restart analog_bridge

MMDVM_Bridge
systemctl status mmdvm_bridge
systemctl stop mmdvm_bridge
systemctl start mmdvm_bridge
systemctl restart mmdvm_bridge

MD380-Emulator
systemctl status md380-emu
systemctl stop md380-emu
systemctl start md380-emu
systemctl restart md380-emu
There are some final adjustments you will probably want to make to your bridge. You can control the audio level from ASL to DMR. You can also control the audio level from DMR to ASL. Both these values are in Analog_Bridge.ini. Let’s take a look.

[USRP]
address = 127.0.0.1
txPort = 32001
rxPort = 34001
aslAudio = AUDIO_UNITY
agcGain = -20 ; Gain (in db) of the AGC filter
dmrAudio = AUDIO_UNITY
dmrGain = 0.35 ; Gain factor of audio from ASL (0.0-1.0)

AUDIO_UNITY Means pass the audio as is. No gain or reduction. AUDIO_USE_AGC Normally not used. AUDIO_USE_GAIN The Gain value can be from 0.0 to 10.0 or anything in between. A value less then 1.0 reduces the audio level to a percentage. Example: .75 is 75 percent of full volume or unity. A value of 1.0 to 10.0 increases the by a power of the number. Example: 3.0 is three time the level.

These values a labels will be changing to something more specific to the usage. ( Due to me trying to document usage ) Grin Funny how you find things this way.