AllStarLink on Raspberry Pi

Session 11 – AllStarLink on a Raspberry Pi

This session explores how to install and run AllStar on a Raspberry Pi so you can have a hotspot to communicate with the AllStarLink network, connection to other nodes like W4AM on Lookout Mountain, or W4EDP.

Table of Contents

WHAT IS ALLSTARLINK?

"AllStarLink (ASL) is a world wide network of Amateur Radio repeaters, remote base stations, and hot spots accessible to each other via the Internet and/or private IP networks. AllStar software runs on a dedicated Linux computer (including the Raspberry Pi) that you host at your home, radio site, clubhouse, school, university, workplace, or computer center. AllStar is based on the open source Asterisk PBX and is released under the GNU GPL and is free for anyone to use. The core of AllStar and AllStarLink is the powerful app_rpt application and associated modules that load into the Asterisk PBX system."

--from https://wiki.allstarlink.org/wiki/Main_Page

CARC recently installed an AllStar node 602640 at its Lookout Mountain 144.390 MHz repeater. You can use it to connect with any other AllStar Node. You can also connect to it using your own AllStar node, as we will discover during this workshop session.

AllStarLink on Raspberry Pi

"A node, in simplest terms, is a computer that connects to the AllStarLink network. Nodes typically have an RF radio interface as well as an internet connection. Radios can range in size from a repeater radio to a low power radio integrated into a node. A node allows you to connect to other nodes in the AllStarLink network. Nodes take the form of an inexpensive Raspberry Pi computer or a PC running the Linux operating system."
-- https://wiki.allstarlink.org/wiki/Beginners_Guide#What_is_a_node.3F

AllStarLink is similar to other Internet supported ways of communicating with other hams. It joins other digital voice modes like DStar, DMR, Fusion, YSF, IRLP, Echolink, and more.

A Raspberry Pi provides an ideal, and often used, way to host a node.

BILL OF MATERIALS

- Raspberry Pi 3 or 4
- Power supply, 5v, 3-5 amps
- MicroSD card, at least 16 GB. Choose one size and type of card and stick to it, such as Sandisk Ultra 32GB, which is what I use.
- Optional: case, fan, heatsinks
- HDMI Monitor, keyboard, mouse with MicroHDMI to HDMI adapter cable (Pi 4) or HDMI to HDMI cable (Pi 3)
- Optional: Computer to use as an SSH client.
- Computer to load the system image on a MicroSD card
- USB to MicroSD card reader
- A second MicroSD card for backup
- Either a SHARI (SA818 Ham Allstar Radio Interface) or similar, or equivalent parts:
  - A TNC (Terminal Node Controller) or audio interface for your radio, or a radio with a built in sound card interface. Or you could purchase a PAUL.
  - A radio to use as the hotspot transceiver such as an HT or mobile radio.
  - SHARI , PAUL, and more are available at https://kits4hams.com/
- An HT radio that supports DTMF to communicate with your node.

EXPLORING A REPEATER NODE

Configurations for an AllStarLink network can be relatively simple, like W4AM, or complex, like N4LMC. N4LMC consists of about a dozen AllStar nodes. For more information, see https://lmarc.net/club/on-the-air/allstar/

W4AM is relatively simple, a single node which can be connected to other nodes.

In either case, you can interact with the node by sending commands to the repeater node. To send a command, use the DTMF function in your radio. Commands are:

(<NodeNumber> means enter a valid node number)

```
*1 <NodeNumber> - Disconnect Node
*2 <NodeNumber> - Connect Node in Monitor/Receive only mode
*3 <NodeNumber> - Connect node in full transceive mode
*70 - Get node status
```

These are suggested, but are typically configured by default on ASL nodes.

*80 Force System ID

*81 Say System Time

*980 Say app_rpt software version

*75 Link Connect (Local Monitor Only)

*72 Last active node (system-wide)

*73 System-wide connection status

*71 Disconnect all links (macro)

*74 Reconnect all links (macro)

Since each node is configured by its radio amateur owner, these will vary by node.

For example, if I want to get the node status on W4AM's AllStar node, I can tune my HT to 145.390 MHz and send the command *70

On my FT1XD HT, I press and hold PTT and then key *70

The node responds with a spoken node status of what nodes, if any, are connected.

Note that the exact response varies by node and how it is programmed to respond.

I can get similar information by viewing the node dashboard. For W4AM, I access the dashboard using

AllStarLink on Raspberry Pi

which displays the current status of the node including what other nodes it is connected to,

AllStar Link   Stats Home   Map of Nodes   Keyed Nodes   Live Keyed Nodes (Beta)   Donate
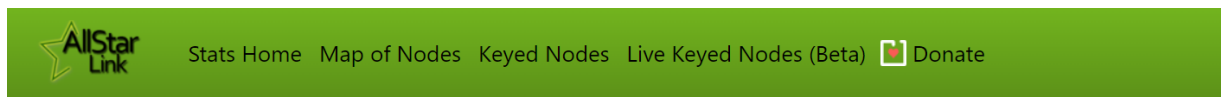
Node 602640

W4AM Chattanooga, TN

Click here for the Bubble Chart

Node	Callsign	Frequency	CTCSS	Location
45295	KA4ELN	147.375 + Analog/P25	123.0	Cleveland, TN
597110	KE4GBW	440.150 simplex		Maryville, Tennessee

Statistics

Actual Uptime	22hr 57m 4s
Keyups	85
Transmit Time	9m
Timeouts	0
Commands Executed	4

and if I click the "Bubble Chart" link, I can see what all the connected nodes are connected to, which can be a large number of nodes.
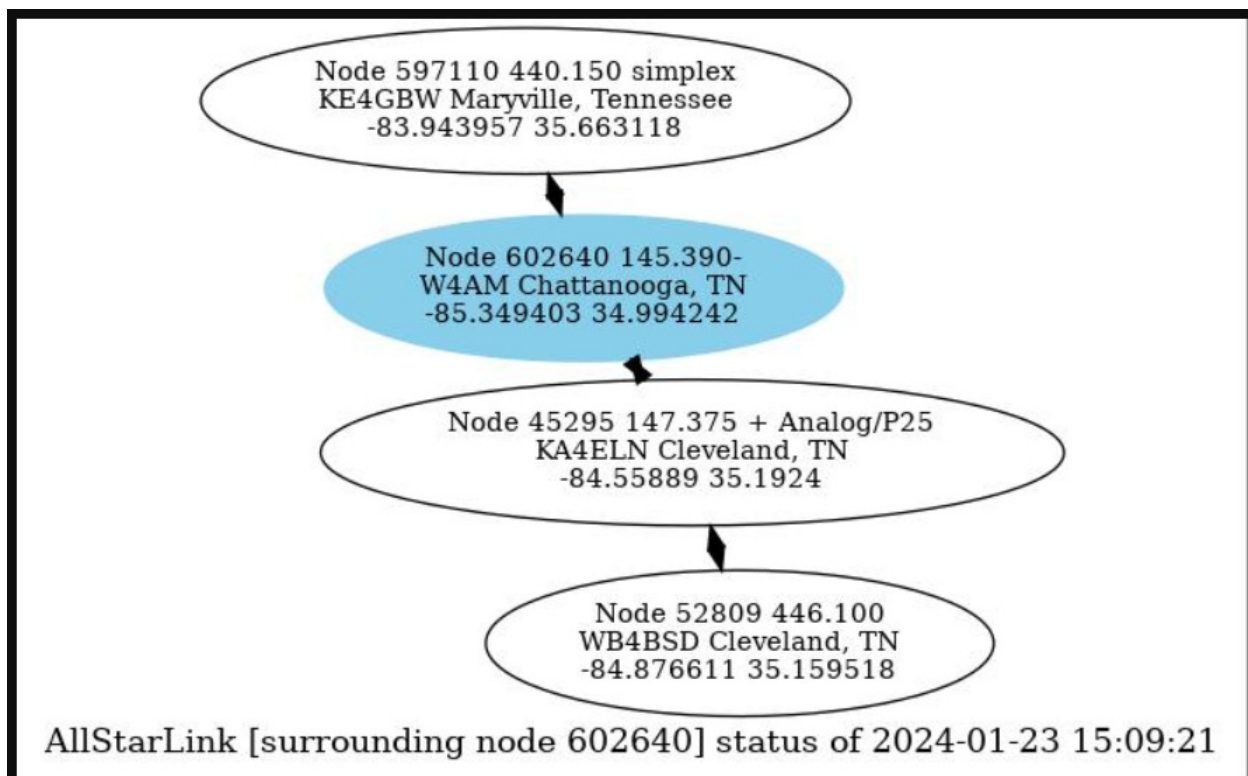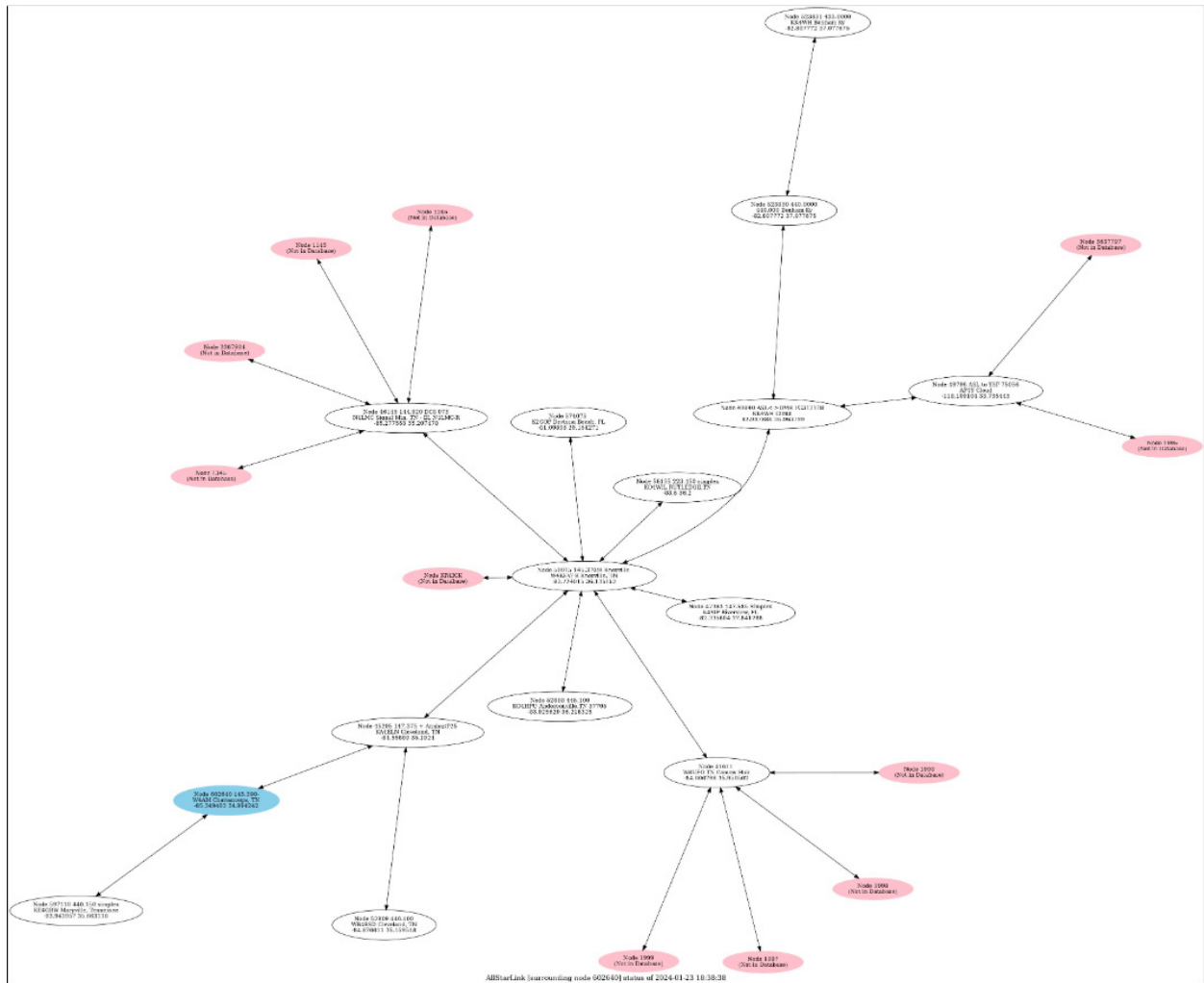
You can see from the bubble chart that node 45295 is connected to both 602640, the W4AM repeater node, and 52809, another node in Cleveland, TN.

When I went back later that day, the bubble chart had greatly expanded due to 45295 connecting to 50015.

This means that whatever was sent or received on any node in this chart will be heard in all of the other nodes!

Another local repeater node that you can use is 46331, for LMARC N4LMC W4GTA 145.350 on Lookout Mountain, GA.

Some node owners use an additional dashboard such as the N4LMC nodes in north Georgia which are found at https://lmarc.net/club/on-the-air/dashboards/ called Supermon. You can install this on your AllStar node, as described later.

BUILDING A LOCAL NODE

What is a local node? It is a node that is like a hotspot, that is, it is self-contained and dedicated to the function of being able to connect to other nodes and for you to transmit and receive either through another radio or directly within the node. It is usually portable.

Why would you want to have a local node? Let's say you are going to take a trip to Florida and while you are there, you want to participate in nets here in Chattanooga, such as the Tuesday Skywarn net. If you take a node with you, and you have Internet access where you are traveling, then you can use it to connect to the 146.390 MHz repeater vis its ASL node and participate just like you were in Chattanooga, assuming the AllStar node is functioning. You can also use it to connect to any ASL node and talk to whoever is connected to that node. Some of our CARC club members are getting their own AllStar nodes and if you learn their node numbers you can connect directly to them, assuming they have their nodes open for connection.

To check if the 146.390 MHz repeater's AllStar node is available, use this URL:
https://stats.allstarlink.org/stats/602640
If it is online, then you can connect to it using your local node.

To check if the 145.350 MHz repeater's AllStar node is available, use this URL:
https://stats.allstarlink.org/stats/46331
If it is online, then you can connect to it using your local node

You might want a local node to provide the ability to connect to any other node without having to go through a repeater. You can control what other nodes it is connected to and for how long.

The connection to the node can be by radio (using a VHF/UHF radio to connect to the node) or radio-less (using a microphone and speaker without a radio). We will explore a radio connection. With this type of connection and with the HT you brought along, you can participate in the net while you are poolside, or by the campfire,

(A possible radio-less node can be built using the Masters Communication RL-20
https://www.masterscommunications.com/products/radio-adapter/rl/rl20.html
Masters Communications provides a series of products in their RA series specifically meant for AllStarLink. I use the DRA-30 and DRA-50 for my digital USB sound cards which can also be used for AllStarLink.)

We will use software from https://hamvoip.org/   This source is based on a different Linux distribution, Arch Linux ARM. Kits4Hams uses this source. Although this version of Linux is less familiar to me, I was able to find my way through it.

Its guide to configuring a node for Kits4Hams hardware is
https://groups.io/g/SHARI/files/SHARI%20PiXX%20Documentation/SHARI%20Allstar%20Node%20Setup%20Procedure-Version%201.4.pdf

(If you are not familiar with a URL like this, the %20 indicates a space in the human readable URL. The human readable name is:

"SHARI Allstar Node Setup Procedure-Version 1.4")

This guide also worked for me to set up a node using my homegrown setup described below.

Setting up your node requires the use of the command line as well as the simple GUIs provided. Some have some pseudo-GUIs displaying screens where you can choose functions from a menu.

The SHARI guide provides all the information you need to:

- Create an account
- Download and install an image
- Login and create a server and node number
- Install and configure your node
- Set up your radio interface (for your access point hotspot)

I requested an AllStarLink account, server, and node. My server is AA6BD-1 and my node is 60068. You will see a node password. Record it as you will need it later.

AllStarLink can create additional nodes as needed. They encourage you to use NNX (Node Number Extension) which allows you to have up to 10 node numbers without needing to request more from AllStarLink and preserves node numbers. I requested NNX and my node number became 600680 for my DRA-50 based node. I have two nodes, and the second one will be 600681 for my SHARI node.

As I experimented with this, I found difficulty in setting up a node using my existing Pi3 with a DRA-30 sound interface and a selection of handheld and mobile radios.

In particular, I was unable to get DTMF decoding to work for me with my setup of RPi3 – DRA-30 – FTM-300D, and communicating using a Yaesu FT1-XDR. I consulted https://groups.io/g/SHARI/files/Setting%20the%20RX%20Level%20in%20HamVOIP%20Allstar.pdf but still could not get it working.

I posted my difficulty on the ARM-allstar list and got a response from the maker of the DRA-30. The difficulty seems to focus on COS.

COS is a function new to me. In AllStar, it is used to notify the AllStar node that its radio is receiving a signal. It is essential that this works properly.

As a result of what I learned, I purchased and built a DRA-50 kit and configured it and my FTM-300D to work on AllStar. I have a working node, 600680.

FTM-300D Configuration:

Setup > Function > TXPWR LO

Setup > Data >

- 1 Com port setting speed 9600, output packet
- 2 Data band select B-Band fix
- 3 Data speed Data 9600
- 4 Data squelch Data TX/RX Band, TX ON

ASL Configuration:

- PTT mode Low
- COSFROM mode USB
- CTCSSFROM mode no
- Preemphasis Enabled
- Deemphasis Enabled
- RX boost mode Enabled

The cost of someone else trying to duplicate what I did would be similar to the cost of purchasing a SHARI kit, so I elected to get a SHARI Pi3U (for Pi3 and UHF) for $70 to be used with my existing Pi3 and it provides everything needed except the Raspberry Pi.

There are a variety of SHARI units available including ones that include a Pi and one that uses a Pi Zero 2. One of them is the unit that Earl AK4ZX showed at the January CARC club meeting which included a Pi4 in a case with radio included.

You can find information on options and how to order at https://kits4hams.com/

I ordered a SHARI Pi3U with the SA818 radio module installed for $70 plus $9 shipping.

The construction and setup documents for the SHARI units are at https://groups.io/g/SHARI/files

To build my SHARI, I followed the guide https://groups.io/g/SHARI/files/SHARI%20PiXX%20Documentation/SHARI%20Construction%20Manual_v1.07.pdf
I had to solder 8 through-hole components to the circuit board, which was straightforward.

To configure my SHARI, I followed the guide https://groups.io/g/SHARI/files/SHARI%20PiXX%20Documentation/SHARI%20Allstar%20Node%20Setup%20Procedure-Version%201.4.pdf

In order to configure your node, you must start by connecting your Pi to the Internet using a wired connection. In my case, the wired connection is in another room, so I used a TP-Link TL-

WR802N router, a small travel router, to provide a wired connection from my wireless access in my house, called the "Client" operation mode.

In the Software chapter, I used BalenaEtcher to write the image to the SD card.

In the Configuring Allstar chapter, I used a monitor, keyboard, and mouse connected to the Pi so I didn't need MobaXterm. I did install MobaXterm and it works fine so you can have a node without needing a monitor, keyboard, and mouse.

I used "root" for the username and password.

(NOTE that the username is root. This means that you are using the root account which can change anything on the system, including deleting itself, so be careful!)

I downloaded updates and rebooted.

I logged in again and did not retrieve updates a second time.

I ran first setup.

I entered a new password. Do NOT forget what you entered or you will need to start over!

I answered No to private node.

I entered my public node number.

I answered Yes to Set Node Configuration.

I did not change the time zone.

I changed my hostname to AA6BD-600681

I did not change the network interfacing setup.

I did not change the SSH port from 222.

I rebooted.

I logged in again.

I said Yes to continuing the configuration.

I left unchanged the node number and entered my callsign.

I said Yes to report status. This allows me to check the status of my node using https://stats.allstarlink.org/stats/600681

I selected Yes for voice ID.

I left Bind port unchanged at 4569.

I selected duplex mode of 1 which is the normal choice for a local node.

When asked for Node Password, I entered what I had recorded.

I did not configure the password for an iaxrpt connection.

I did not configure SimpleUSB settings now. We will do that later.

I restarted Asterisk.

Next, I see the Admin Menu List. We will be using it a lot. It appears whenever you restart the node.

The next chapter is "Configuring SimpleUSB-Tune-Menu for SHARI"

I ran the Simpleusb-tune-menu application and made the changes:

- Item B – Change RX Boost to Disabled
- Item F – Change PRE-emphasis mode to Disabled
- Item G – Change DE- emphasis mode to Disabled
- Enter W to save the changed settings then enter 0 to exit the SimpleUSB-Tune-Menu

I want to use WiFi rather than a wired Ethernet connection. To do this, I selected "Configure the Wi-Fi Interface Networking"

Select 1 Setup Wireless SSID and passphrase
Continue
Do you want to clear? Probably No
Do you want to manually enter the SSID? No
It will scan for WiFi networks. Select your SSID
Enter your passphrase
OK to ADD? Yes
Reboot the server and remove the wired connection.

(An older version described in https://hamvoip.org/howto/wireless_howto.pdf used older software and is not upward compatible. Use $ `wireless-setup.sh` to manage it.)

(This software is maintained as a hobby by radio amateurs. As such, you will sometimes find inconsistent documentation as things get revised and the documentation is not always revised, as happened with the Wi-Fi documentation.)

(Check out the `/etc/wpa_supplicant/wpa_supplicant.conf` file where the SSIDs and passwords are saved. You can save multiple SSIDs in this file and the Pi will connect to them in the order listed.   Use selection 4 to see the file.)

To view what your new IP address is, use:

```
$ ifconfig
```

Also, when you reboot your node, the node will say your node IP address. You will need your receiving radio on to hear this, of course.

During the WiFi connection process, some scripts were used. On this OS, where are these scripts stored? They are in /usr/local/sbin

The next chapter is "Connecting and Configuring SHARI."

I followed this chapter as documented. The only problem I had was that I needed to program the SA-818 twice to get it to transceive on the right frequency.

There was a document included with my kit that changed some of the setup which I followed. I changed:

- Squelch value: 1
- Volume value: 8
- Preemphasis enabled: y
- High pass enabled: y
- Low pass enabled: y

I then had to change the simpleusb-tune-menu options to

- B) rx boost disabled
- F) pre-emphasis mode disabled
- G) de-emphasis mode disabled
- I) PTT mode active Low
- J) COSFROM mode usbinvert

The simpleusb-tune-menu is:

```
Active simpleusb device stanza: [usb] -------------------------------------
S) Select active USB device stanza
V) View COS, CTCSS and PTT Telemetry using real-time display
P) Print Current Parameter Values ---- 2) Set Rx Voice Level (using display)
3) Set Transmit A Level            ---- 4) Set Transmit B Level
5) Set Tx Audio Level Method (currently LOG)
7) Set Transmit DSP Level
B) Toggle RX Boost Mode (currently Disabled)
C) Toggle Echo Mode (currently Disabled)
D) Flash (Toggle PTT and Tone output several times)
E) Toggle Transmit Test Tone/Keying (currently Disabled)
K) Manually key COS (currently Unkeyed)
F) Toggle PRE-emphasis Mode (currently Disabled)
G) Toggle DE-emphasis Mode (currently Disabled)
H) Toggle PLfilter Mode (currently Disabled)
Q) Toggle DCSfilter Mode (currently Disabled)
I) Toggle PTT Mode (currently active LOW)
J) Change COSFROM Mode (currently "usbinvert")
L) Change CTCSSFROM Mode (currently "no")
M) Change RXONDELAY value (currently "0")
N) Change RXAUDIODELAY value (currently "0")
W) Write (Save) Current Parameter Values
0) Exit Menu

Please enter your selection now: 
```

I set volume levels and DTMF as documented.

- Run simpleusb-tune-menu
- Select item 2 and speak into the microphone and verify that the level extends just slightly beyond "5kh."
- Save and exit
- 11 Start the Asterisk client
- Send DTMF commands and verify they are seen.
- Use *A1 to hear the IP address of your node.

I enabled the COS LED.

- $ cd /etc/asterisk
- $ nano rpt.conf
- Edit as shown in the setup procedure document
- Remove comments for
  cop,62,GPIO4:1 = c|t|RPT_RXKEYED
  cop,62,GPIO4:0 = c|f|RPT_RXKEYED
- Exit and save changes
- Reboot the node
- Check that COS light lights up

TEST YOUR NODE

To test your node, you can connect to the Parrot Audio test node by sending DTMF *355553

Parrot will report how good your audio level is and play back your transmission.

To disconnect, send DTMF *10

SUPERMON

A feature included with the HamVOIP image is Supermon, a web based program to monitor and control your node. It is documented at https://hamvoip.org/howto/supermon_howto.pdf

I was able to set up Supermon and connect to my node from my Windows laptop and see what its status is, connect to other nodes from the web site, send DTMF commands to the node, read log files, and more.

AllStarLink on Raspberry Pi

There is another vendor of AllStar hardware: https://allscan.info

This vendor provides an alternate to Supermon named AllScan. It allows you to connect to other nodes that you have saved as favorites, and has a little nicer interface. Both are useful.

I am finding that there are many alternatives in the AllStarLink galaxy.

TIME ANNOUNCEMENT

When you build your local node, it announces the time every hour. Can you change this? Yes if you know how it does it. There is a script included with your node called "saytime.pl" and this script will announce the time. Do you remember cron? Cron is the facility to schedule things in Linux. To check what is scheduled, type

$ cronjob –e

This will show you the cron jobs that are scheduled for your node. If I type it on my node, I see

Notice the last line. It executes the script "saytime.pl" every hour on the hour. You could modify or delete this line as desired to change how the time is announced on your node, or just comment it out by placing a # before the line.

You can view https://hamvoip.org/howto/bbb_scripts.pdf to see this and other scripts.

INTERNET CONNECTION TO YOUR NODE

So far, I have created a node and connected to other nodes by using their node numbers. Can I connect to my node from other nodes? You might want to do this if you want other hams to connect to your node so you can have a conversation without using another node to both connect to. Your conversation becomes more private than if you used a repeater connected node.

We configured a "Bind port" of 4569. This is the port on our router that we can use to allow remote access to our node. You must change the configuration of your router to enable this.

Your node reports its IP address to allstarink.org so that others can find your node.

On my ORBI router, I first reserved the address of my node, 192.168.8.44, so that it would always have that IP address. Under Advanced Setup, Port Forwarding, the notes for ASL says that the port should be number 4569 and type UDP. I did not see a type UDP, so I created a custom service UDP using external port 4569.

DVSWITCH

DVSwitch Mobile is an Android App that can connect to your node. You can use it to connect to your node

https://hamvoip.org/howto/iaxrpt-config.pdf

APPENDIX

Another source of information on AllStar is https://w2ymm.home.blog/allstar-howto/

An alternate site for building a node is https://allstarsetup.com/

An alternate guide to building your node is https://wiki.allstarlink.org/wiki/Beginners_Guide

It is based on Raspberry Pi OS Debian 10 Buster, which is familiar to us although two releases behind the current one.

At https://github.com/Allan-N/ASL-Install you can find instructions on how to install ASL on the latest release of Pi OS and run it on a Pi 5.