Pacific Northwest Network

Introduction to the VA6EDN Network

Written by:
Martin Alcock, M. Sc, VE6VH

Revision: 0.3

Alberta AREDN Society,
43 Citadel Bay NW,
Calgary, AB T3G 3Y4
www.praebius.com

Table of Contents

List of Tables

List of Figures

Document Revision History

Date	Rev	Description
Oct 2021	0.1	Initial Draft
Oct 2021	0.2	Added detail about linking network
Nov 2021	0.3	Added topHAT SDR module

Table 1 Revision History

Distribution

Rev	Distribution
0.1	SARA Executive only
0.2/0.3	Any AREDN member

Table 2 Distribution

Reference Documents

[1] gnu.org, "General Public Licence," [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html. [Accessed 25th February 2018].

[2] National Oceanographic and Atmospheric Administration, "NOAA Weather Radio Alerts," [Online]. Available: https://www.weather.gov/grb/nwr_same. [Accessed 24 May 2020].

[3] Ettus Research, "USRP Hardware Driver and USRP Manual," [Online]. Available: https://files.ettus.com/manual/page_transport.html. [Accessed 21 October 2020].

[4] V. Martin C. Alcock, "A single chip repeater controller using an FPGA," *CQ Amateur Radio,* pp. 24-30, 2016.

[5] Network Working Group, "RFC 3350: RTP: A transport protocol for real-time applications," July 2003.

[6] Network Working Group, "RFC 2543: SIP: Session Initiation Protocol," 2002.

[7] Digital Voice Systems Inc, "AMBE+2 Vocoder," [Online]. Available: https://www.dvsinc.com/software/technology.shtml. [Accessed 7 October 2021].

[8] Wikipedia, "Goertzel Algorithm," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Goertzel_algorithm. [Accessed 1 11 2020].

Glossary of Terms

PBX	Private Branch Exchange. A node in a telephone network that provides connectivity for a series of local extensions to a set of trunks.
VOIP	Voice over Internet Protocol. A system where telephone calls are placed, and audio is exchanged using the Internet Protocol.
PCM	Pulse code modulation. An encoding scheme used in digital audio and PBX systems that varies in bit width depending on the compression scheme but always sampled at 8 KHz.
RTP	Real time protocol. A protocol used in VOIP systems for exchanging audio, employs 8 bit PCM coding using µ- or A-law compression.
USRP	Universal Software Radio Protocol. A protocol for exchanging audio packets between software radio devices employing 16 bit linear PCM coding.
FPGA	Field Programmable Gate Array. A general purpose logic device.
JNI	Java Native Interface. A method of calling native level methods from a Java program for more time-critical applications.
HAT	Hardware Added on Top. A standard for hardware add-on modules for the Raspberry Pi 2 or 3.
HAT^2	"Hat-on-Hat". A mezzanine card added on top of a HAT module.
JSON	JavaScript Object Notation. A lightweight data interchange format that is hierarchical in nature and is easy for humans to write and machines to parse.
JAM STAPL	A method of programming an FPGA using an embedded processor.
DMR	Digital Mobile Radio
AMBE	Advanced multi-band coding.
CODEC	An algorithm for translating one standard to another, from COder/DECoder.

Intellectual Property Notice

The hardware components and intellectual property described herein is the property of one or more corporate sponsors, all rights are reserved.

Each sponsor grants licence to any Amateur for personal or club use, on an as is and where is basis under the condition that its use is for non-commercial activities only, all other usages are strictly prohibited. Terms and conditions are governed by the GNU public licence [1].

No warranty, either express or implied or transfer of rights is granted in this licence and the owner is not liable for any outcome whatsoever arising from such usage.

MapleDSP, MaplePi and topHAT are registered trademarks of Praebius Communications Inc.

Disclaimer

This document is a preliminary release for hardware and software still in development and may be subject to change in future revisions.

Background

The Amateur Radio Emergency Data Network (AREDN™) is an always-on Internet Protocol network owned and operated by amateur radio operators, designed to provide high speed data communications and stay operational when local resources are not available due to an emergency or failure.

Beginnings

The network was born from earlier experimentation in repurposing LinkSys™ WRT54 routers with custom firmware, operating on the amateur portion of the 2.4 GHz band. The devices were able to interconnect with each other forming a 'mesh network', where each station, or 'node' in the network is capable of reaching any other by exchanging routing information. The net result was that the bar for on-air data exchanges was raised from less than 100Kb/s to in excess of 10 Mb/s, and the concept was found to be viable for on-air communications.

This project, known as broadband hamnet, was eventually abandoned as the manufacturer was purchased by another company, and due to a cost reduction exercise the firmware would no longer fit into in the memory on the revised hardware, limiting deployment to revisions that pre-dated the current and later hardware, also making it almost impossible to obtain except on the surplus market. In order for the network to be viable and grow, a stable source of new devices had to be found.

Creation of AREDN

The AREDN network was created not only to revive the mesh network concept and maintain the previously achieved on-air speeds, but also to extend the reach of the network. First, a new platform had to be found, and hardware built by Ubiquiti networks, particularly in their high power Air Router, known as the ARHP, was adopted. Figure 1 illustrates such a device.

Despite the higher power devices, distances between stations were still limited by consumer-grade low power transmitters and a lack of gain antennas. This led to a network topology where most stations were deployed in local 'clusters', which were separated from each other. Clusters tended to be localized to a particular rural or urban area and were not interconnected due to a lack of higher power equipment that could maintain the network speeds. Some clusters could be interconnected using many 'hops' between stations, however larger distances were unachievable.

The AREDN firmware added the capability of interconnecting nodes using the public internet and technology know as a 'virtual private network' to maintain security. This enabled clusters all over the world to be connected, producing the network that we have today. These are referred to as 'tunnels', any node hosting others is a tunnel hub, those connected in are tunnel clients.

Figure 1 Ubiquiti Networks Air Router HP

The network currently spans the globe with nodes in North and South America, the United Kingdom and European Union, Africa, Asia, Australia and New Zealand. A map of the network is shown in Figure 2.

Figure 2 AREDN Worldwide Network

AREDN in Alberta

As in all advancements serendipity has a role to play, and this is what brought the network to Alberta. At the 2017 Glacier hamfest Ken Oelke, VE6AFO, announced that QCWA would be hosting a cruise with Holland America Lines in the fall. We decided to join in and on this cruise a connection was made between Mark Herson, N2MH and myself, VE6VH, and my interest in the network was piqued.

Figure 3 Amateurs aboard the cruise in 2017

Following that, the first node in Alberta was deployed early in 2018, which was the ubiquiti node shown in Figure 1. Reasonably shortly after that, a second node was deployed at the QTH of Jerry, VE6TL, and the first RF connection on 5.7GHz was made between the two, and it is still in service today. Since then, the local network has grown in several directions to some 78 nodes, which include all provinces west of Manitoba, with the exception of Saskatchewan, to Vancouver Island, and also two States in the US.

To keep the number of nodes to a manageable amount, the network has been segmented into 8 different networks, all of which are accessible to each other. As of early 2021, what was the Alberta network has now been renamed to be the 'Pacific Northwest AREDN Network' and is a branch of its own. It is anticipated to grow even more in the ensuing years.

The deployment strategy to date has been to connect new nodes to the network using internet tunnels initially, and then replace those with an RF based connection as time goes on. The enables each amateur to enjoy using the network facilities right away, and alleviates a large capital investment, as the entry-level nodes can be purchased for less that $80.

Figure 4 Pacific Northwest AREDN network

The AREDN Society

As of 2021, it was decided that a society would be formed to preserve the interests, assets and operations of the network. This process is still ongoing, but a board of directors has been appointed, which consist of Martin Alcock, VE6VH, Wally Gardiner, VE6BGL, Garry Jacobs, VE6CIA, and Andrew Webb, VE6EN.

It was decided early on that aligning with ARES to provide for their needs would be beneficial, therefore they form part of the board.

Up to now, the network has been funded by participating amateurs, who have purchased their own equipment and donate the use of their internet connections. Capital equipment has been donated by private industry.

The society has little or no ongoing operating costs.

Getting Started

All that is needed to get started on the network is some node hardware, which can be purchased for under $100. The recommended starter hardware is the Microtik hAP (hybrid access point), which has a combination of ethernet and Wifi ports, and is the current replacement for the Ubiquiti Air Router previously shown and is shown in Figure 5.

Setting up a node

The node has to be programmed with the AREDN firmware, instructions on how to do it and more information can be found on the AREDN website, www.arednmesh.org.

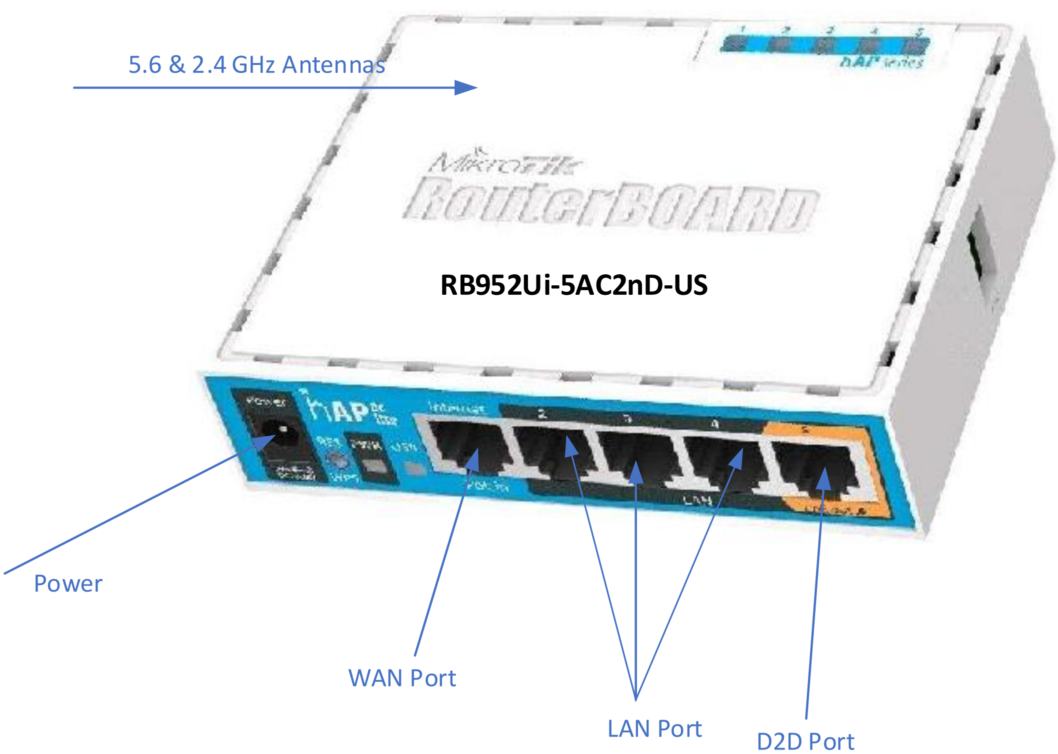


Figure 5 Mikrotik hAP Access point

Each node has three categories of ethernet ports:

1. A WAN (wide area network) port, which is only used for establishing a tunnel connection to another node.
2. A series of LAN (local area network) ports, which are available for local devices, such as a PC, phone, camera, etc.
3. A D2D (Device to Device) port, which is dedicated to connection to another node to implement an RF connection.

Connecting to the network

There are two methods for setting up a connection to the network:

1. Using an internet tunnel
2. Connecting with RF

Using a tunnel connection

An internet tunnel is the easiest and most inexpensive way to get started on the network, but it relies on an internet connection, but any consumer grade internet can be utilized, and static IP addresses are not required.

There are tunnel servers around the network that have been designated as 'tunnel hubs' and have fixed sub-domain addresses that are updated by specialized software in the server node. Figure 6 illustrates hubs that are currently active in Alberta and British Columbia.

Tunnel Hubs in Alberta

Address	Club Name	Location	Sysop	Callsign
meshnet.va6edn.org	AREDN Society Central Hub	North Calgary	Martin	VE6VH
ares.va6edn.org	Amateur Radio Emergency Service	North Calgary	Andrew	VE6EN
caarc.va6edn.org	Central Alberta Amateur Radio Club	Red Deer	Garry	VE6CIA
fars.va6edn.org	Foothills Amateur Radio Society	High River	Wally	VE6BGL
saarc.va6edn.org	Southern Alberta Amateur Radio Club	Lethbridge	Gary	VE6CV

Tunnel Hubs in British Columbia

Address	Club Name	Location	Sysop	Callsign
bcnet.va6edn.org	East Kootenay AREDN Network	Cranbrook	Dan	VE7MRP
norac.va7okn.org	North Okanagan Amateur Radio Club	Vernon	Wilf	VE7OHM
barc.va7okn.org	Burnaby Amateur Radio Club	Burnaby	Don	VA7DGP

Figure 6 AREDN network tunnel hubs

Tunnels have to be configured at both ends, the credentials for a connection are set up by the hub operator. These credentials are entered into the node to establish the connection.

Using an RF connection

An RF connection can be made if the local area has implemented it. Nodes for this type of connection are usually mounted on a roof or tower and are connected to the local node using the D2D port. These nodes also use POE (power over ethernet), so only one cable needs to be run.

Figure 7 Daisy Chained nodes on a tower

At the tower up to 2 units can be connected in a daisy-chain fashion on a single cable, as illustrated in Figure 7. The two nodes implement two legs of a 5.7GHz connection, the upper node is an Ubiquiti NSM-5, and the lower node is a MikroTik LHG. Both operate in the amateur portion of the 5.7GHz band and have sufficient separation so as not to desense each other.

Accessing the network

The first step to access services on the network is to connect a PC to the LAN port and obtain an IP address from the node. Then, using a web browser, type the following into the address bar:

localnode:8080

The landing page as shown in Figure 8 will appear.

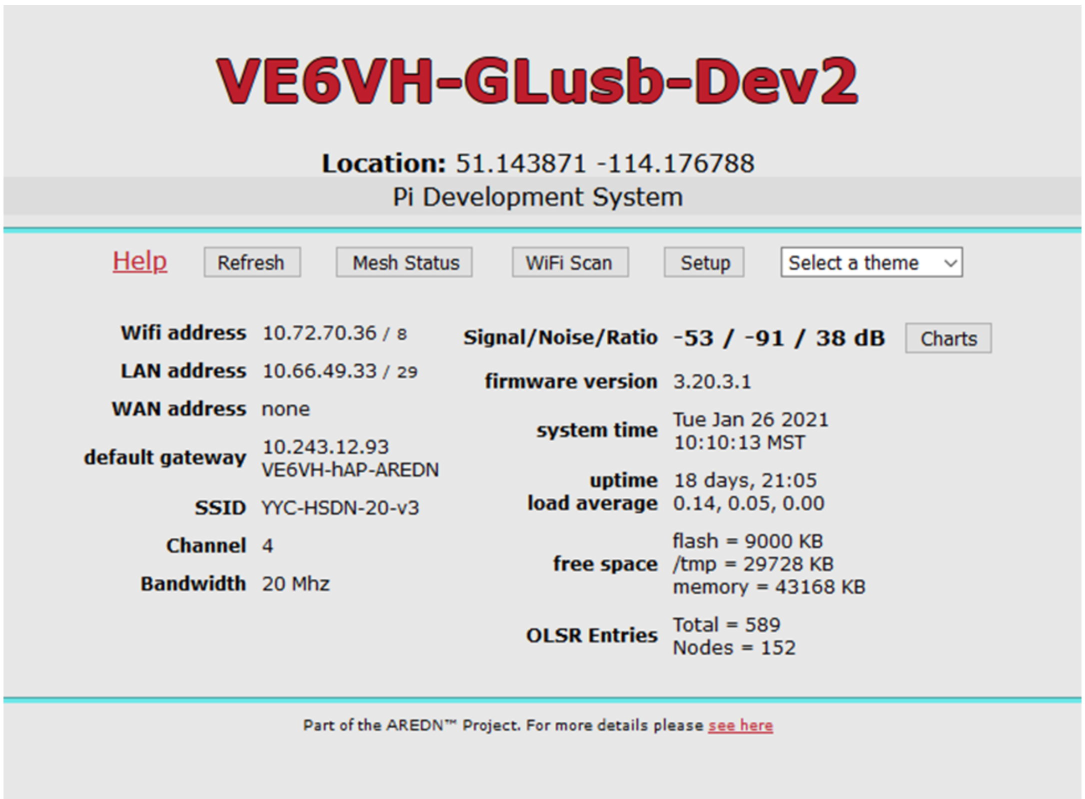


Figure 8 AREDN node landing page

Once at the landing page, options are presented to view the mesh status and access the node setup. Services are listed on the mesh status page, as shown in Figure 9.

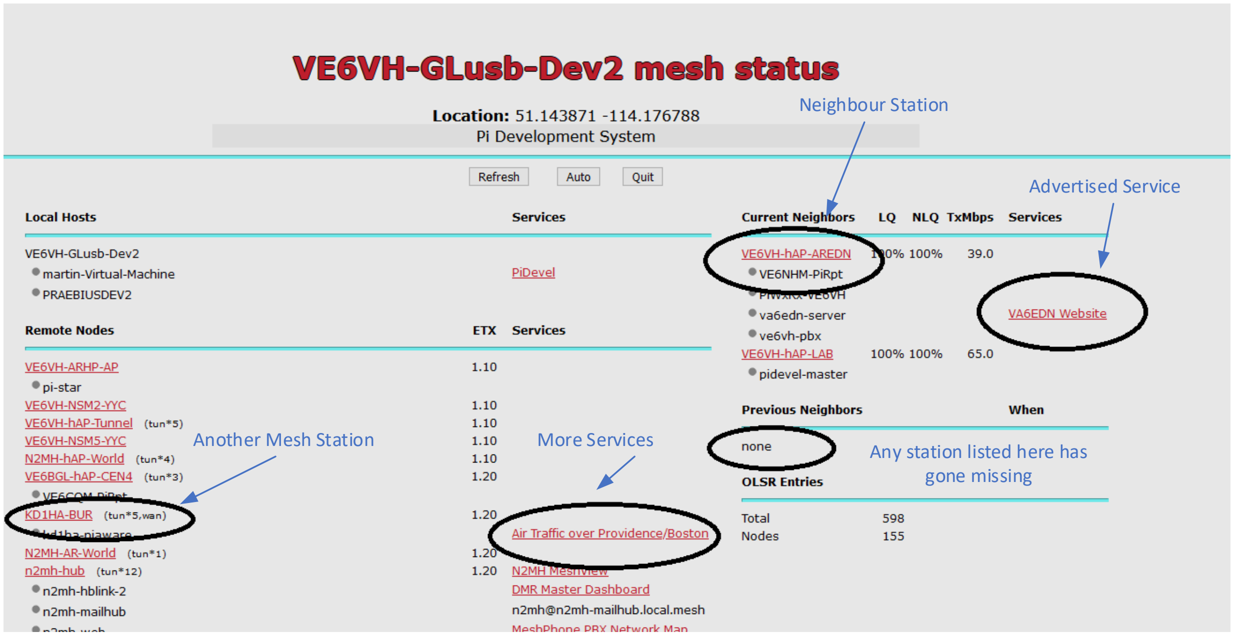


Figure 9 Mesh Status Page

Above is a snapshot of the mesh status on this node. There are two tables, each has two columns. The table on the left are all the stations in the local network, the table on the right is the immediate neighbours that are connected either connected by RF or a tunnel.

The left hand column has the name of the node, and a list of devices that can be reached by name. The right hand column are links to advertised services, which can be reached directly. Each is a hyperlink that can be ‘clicked’ to access it.

Local services on the network

There are several services on the local network, and many others around the network. Some of the local services are:

1. **Meshchat** A local chat room and messaging platform.
2. **Meshphone** An asterisk-based voice over IP network.
3. **Web Server** Local Linux server hosting site, email, FTP and databases.
4. **Weather Server** A raspberry Pi with an SDR dongle providing local weather alerts.
5. **DMR Hotspot and Repeater Linking** A service to connect DMR hotspots and local repeaters.

Meshchat

Meshchat is a node-based message exchange facility that can be local, regional or international. It is segregated into a series of 'chat rooms', or channels, for this purpose. It is freely available and can be installed in a couple of minutes. One of the common uses of Meshchat is for a local ARES group to check in, as can be seen in Figure 10.

Messages Search: Enter search Channel: Net Check In

Time	Message	Call Sign	Channel	Node
7/25/17 6:10 PM	N5HNE is checking in for the MCARES Net via the MESH	N5HNE	Net Check In	wa5eoc-ubm2-mont
7/25/17 5:47 PM	Checking in	N5MDT	Net Check In	wa5eoc-ubm2-mont
7/25/17 4:48 PM	Checking in via the Mesh for the net.	K5DLQ	Net Check In	k5dlq-pi1
7/18/17 6:58 PM	Checking in	W2MVR	Net Check In	wa5eoc-ubm2-mont
7/18/17 5:34 PM	Checking In 7-18-17	N5MDT	Net Check In	wa5eoc-ubm2-mont

Figure 10 Example Meshchat check-in

Meshphone

Probably the most used service on the network, Meshphone is a voice over IP network consisting of a series of 58 private branch exchanges (PBX), realized on the Asterisk and FreePBX platforms. Any VOIP phone that supports SIP can be utilized, however the recommended desk phone is the Polycom VVX411, and for a softphone the Grandstream Wave or Wave Lite.

Figure 11 illustrates the worldwide PBX network, which is overlaid on the AREDN network.

Figure 11 World wide Meshphone PBX network

Phone numbers consist of 7 digits, consisting of a 3-digit 'area' code, and 4 digit extension. There is a web-based phone directory available for all local numbers, and both phone types have the ability to look up a number from the local database.

Figure 12 An Example of a directory lookup

Another site has the entire worldwide data base which can also be accessed from any mesh node on the network. Demonstrations of Meshphone at club meetings usually involve dialling the date and time in Australia or New Zealand, or somewhere else around the world. This is a standard feature of all PBX's and part of the Asterisk software.

Figure 13 Polycom VVX411 SIP Phone

The polycom phones are managed by a server which contains current firmware, configuration and usage data. The configuration has been set up for single button access to PBX features such as: -

- On hook dialling, hands free operation.
- Voice mail
- Dial by Callsign
- Global dial
- Conference bridge
- Portal to Hamshack hotline

Additional features can be accessed from a menu which can be easily navigated, such as the local phone book. An example of such a lookup is shown in Figure 12.

Web Server

Table 3 lists the applications and software compliment on the Linux Web server.

Application	Name	Port	Components	Language
Email server	Meshmail	25/795	PostFIx/DoveCot	C/C++
Apache Server	Apache2	80	VA6EDN Website	HTML
Polycom phone provisioning	FTP Server	21	polycom config files	XML
Phone directory and white pages	LDAP Server	389	LDAP database	LDAP
Database Server	MySQL	3306	Node database	MySQL
Create/update node database	Meshcrawler		Meshcrawler	Java
Portal from email to SIP	SipPortal	none	PortalMain	Java
DMR Linking	HBLink	8085	hblink3	python 3
Bridge to HBLink	MMDVM_Bridge	31100/03	MMDVM_Bridge	C/C++

Table 3 Apache Server Software Compliment

Email Server

Known as ‘Meshmail’, it is a secure e-mail system implemented with readily-available servers on a Linux platform. It is secure not only in its delivery methodology, but also because it has no access in or out of the public internet, therefore external ‘spam’ has no ingress into the system.

Many different popular e-mail clients are supported including Outlook, Thunderbird and Mutt, and there also is a webmail portal available on the web server using Squirrel Mail.

Apache Server

The VA6EDN website is hosted on the Apache server, which contains general information about the mesh network, and is a portal to several other sites:

- Meshphone website: a ‘how to’ site listing instructions on how to dial on Meshphone
- Phone book: a listing of all local phone numbers
- Mesh map: a map of nodes on the network
- DMR hotspot linking network status
- Portal to webmail
- Head end UPS status
- Database administration for Meshphone site

FTP Server

A general purpose FTP server for provisioning polycom phones.

Database Server

A MySQL database server for holding current mesh network status information.

Weather Server

The weather server is realized on a Raspberry pi platform using an SDR dongle and GNU soft radio receiver. It serves two purposes:

1. To act as a Meshphone extension that can be called to hear local weather broadcasts.
2. To receive and decode local weather alert messages.

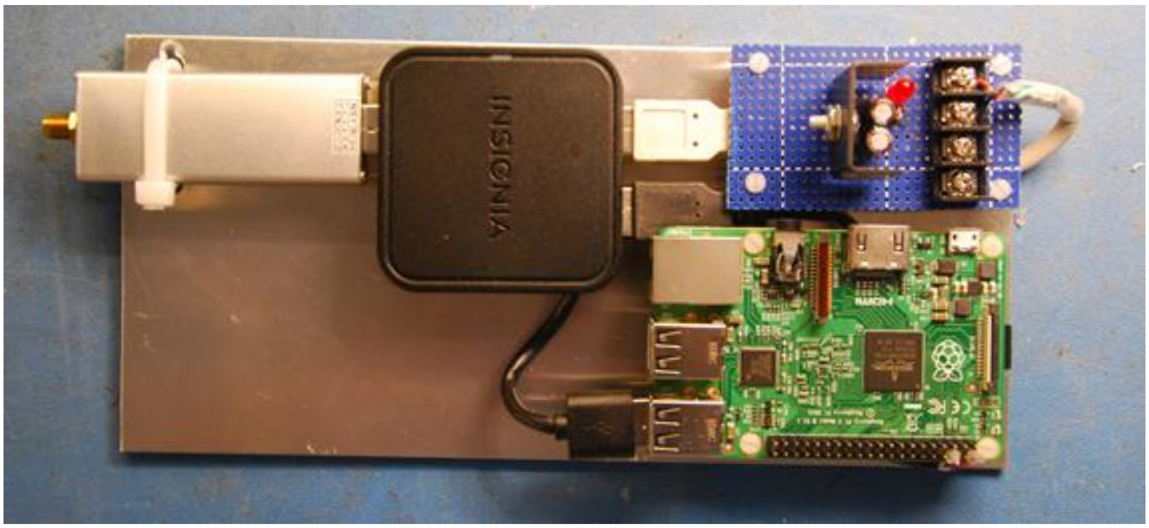


Figure 14 Pi based weather receiver hardware

As a Meshphone extension it can be called from any PBX on the network and will play back received audio from a local station. The receive frequency is programmed into a simple .xml file which can be managed as a text editor.

The message receiver is a purpose-built FSK demodulator for the NOAA message format [2], which is very different from any amateur packet service. At least once a week a test message is sent, but also other alerts for weather exceptions are also decoded.

The message protocol used for alerts is called 'Selective Area Message Encoding', or SAME. Currently these are posted to a Meshchat channel after being interpreted. The area that is affected by the alert is sent numerically in the message, which is translated by a local database.

Figure 15 illustrates an interpreted monthly test message that has been posted to Meshchat.

10/6/21 12:02 PM	Message from PiWxRx at VE6VH from XLF339 - Calgary AB Environment Canada has issued a Required monthly test bulletin at 11:00 local time on 2021/10/7 For the following areas: Kananaskis - Canmore M.D. of Bighorn near Canmore Exshaw and Ghost Lake Kananaskis Improvement District near Kananaskis Village Airdrie - Cochrane - Olds - Sundre Rocky View Co. near Airdrie and Crossfield Rocky View Co. near Bottrel and Madden Rocky View Co. near Cochrane Drumheller - Three Hills Rocky View Co. near Irricana Beiseker and Kathyrn City of Calgary Okotoks - High River - Claresholm Rocky View Co. near Bragg Creek and Tsuu T'ina Res. M.D. of Foothills near Priddis and Brown-Lowery Prov. Park	VE6VH

Figure 15 Interpreted Weather Alert Message

The software compliment of the weather server is shown in Figure 16. It consists of two 'layers', the upper, or application layer, contains the SIP client, an HTML server to review the configuration, and an e-mail server or SIP messaging to forward the message, the interpreter, message assembler and a JSON database for SAME interpretation. This layer is written in the Java language.

The lower layer contains the time-critical components, which includes an instance of GNU radio to demodulate the FM signal, the data modem and message receiver, and a G711 µLaw codec for encoding the voice audio, and an RTP streaming audio module which sends it to asterisk. The lower layers are all written in the C language for performance purposes.

All code can run on a Raspberry Pi 2B or 3, a dedicated extension must be allocated on the local PBX for it. A separate piece of software, written by Mark, N2MH, receives the interpreted e-mail message and posts it to the Meshchat forum.

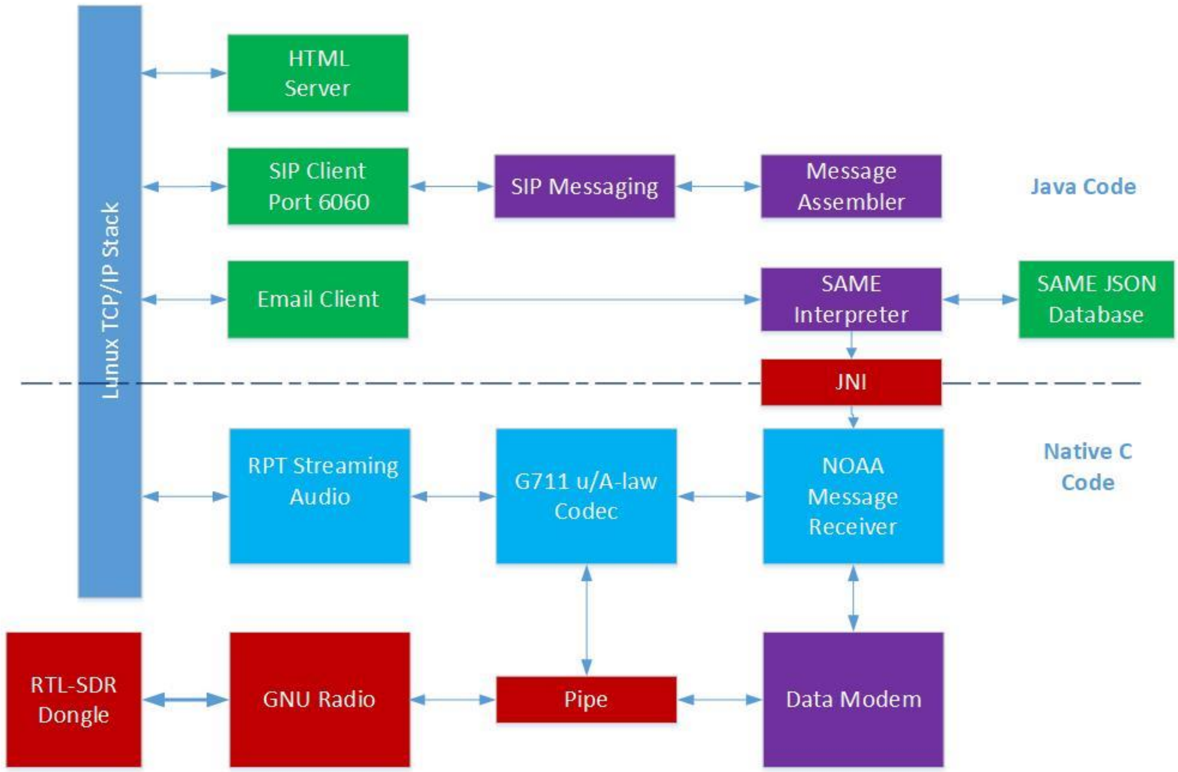


Figure 16 Pi Weather Receiver Software Compliment

1296 D-Star Portal

This project is currently in the planning phase. The intention is to provide remote access to the network from a 1296 MHz D-Star portal using the 128Kb/s mode. A potential user will connect to the portal using D-Star, the be redirected to a login page. Once logged in, they will be able to access the network facilities.

A site for a repeater is currently being sought, the network hardware will consist of an ICOM ID-1 radio.

Linking Network

The AREDN network is also used as an IP backbone for repeater linking, which comprises several software components as listed in Table 4.

Component Name	Function	Packet format	Source Language
HBLink	DMR linking	AMBE	Python
DVSwitch	Transcoding	PCM or AMBE	Object distribution
USRPLink	Repeater linking	PCM	Java/C++

Table 4 Linking system components

The system as it is currently deployed is illustrated in Figure 17.

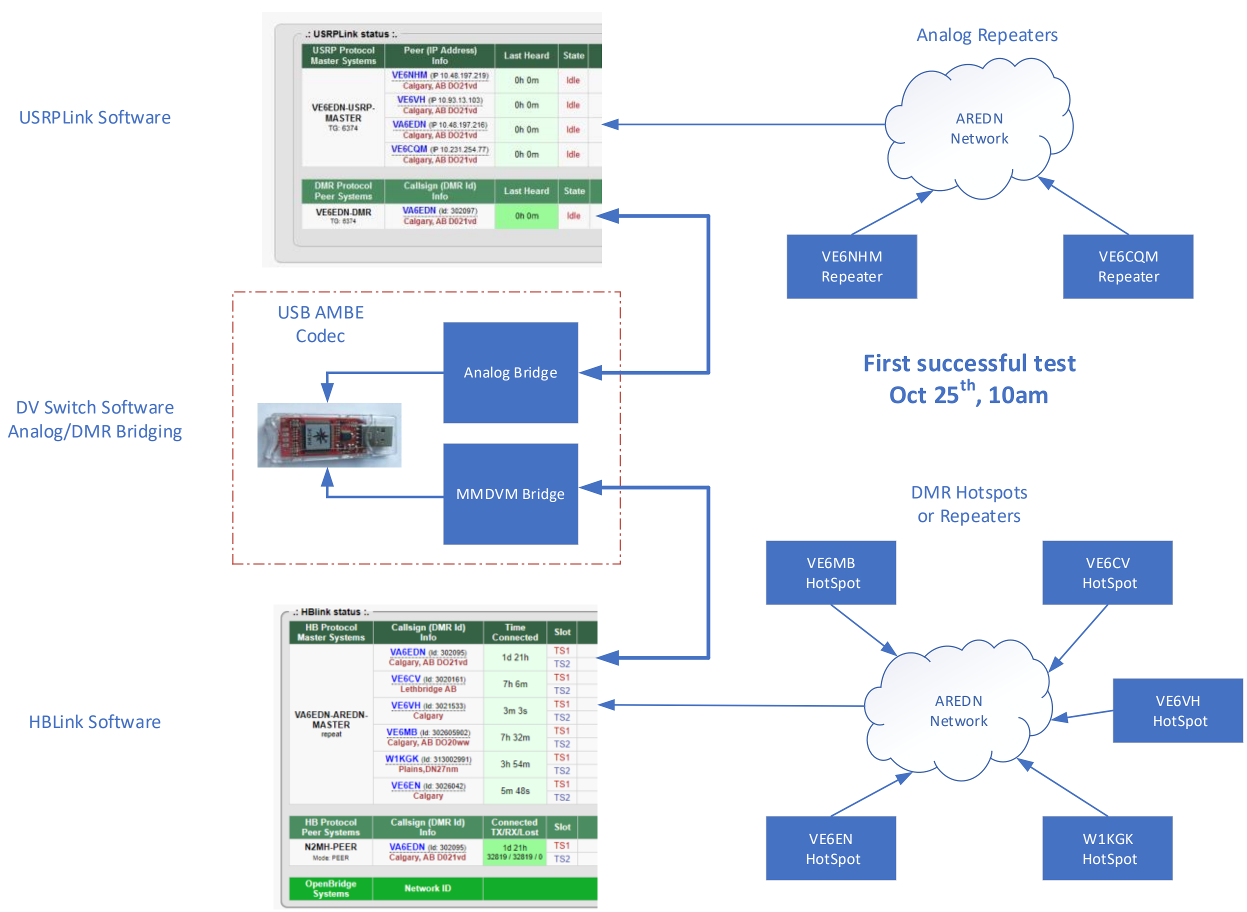


Figure 17 Digital/Analog Repeater Linking

The purpose of the network is not only to interconnect repeaters using digital networking, but also to be able to extend their reach to any location where an inexpensive hotspot can be located.

HBLink DMR Hotspot and Repeater Linking

The lower portion of Figure 17 illustrates the DMR hotspot and repeater linking network. Any hotspot that runs Pi-Star can be utilized, such as a Zumspot, LoneStar, SharkRF or other compatible device. The DMR protocol is used, there currently are no provisions for transcoding between D-Star, Fusion or other protocols. The linking in this domain is accomplished by software called HBLink, which is less complicated to implement than BrandMeister, and is also limited to the mesh network. Figure 18 shows the status screen which is available on the VA6EDN website.

.: HBlink status :.

HB Protocol Master Systems	Callsign (DMR Id) Info	Time Connected	Slot	Source Subscriber	Destination
VA6EDN-AREDN-MASTER repeat	VA6EDN_L (Id: 302095) Calgary, AB DO21vd	28d 17h	TS1		
			TS2		
	VE6MB (Id: 302605901) Calgary, AB DO20ww	6h 29m	TS1		
			TS2		
	VE6CV (Id: 3020161) Lethbridge AB	7h 46m	TS1		
			TS2		
	VE6EN (Id: 3026042) Calgary	8h 2m	TS1		
			TS2		
	VE6VH (Id: 3021533) Calgary	31m 59s	TS1		
			TS2		

HB Protocol Peer Systems	Callsign (DMR Id) Info	Connected TX/RX/Lost	Slot	Source Subscriber	Destination
N2MH-PEER Mode: PEER	VA6EDN (Id: 302095) Calgary, AB D021vd	156d 0h 1514 / 1513 / 1	TS1		
			TS2		

OpenBridge Systems	Network ID	Active Calls

Figure 18 HBLlink Status Screen

All hotpots connect to a master bridge, denoted VA6EDN-AREDN-MASTER on the figure. Routing of packets to each radio is determined by a talkgroup number, the one used for the mesh network is 6374 (M-E-S-H).

PiStar Setup

To set up a hotspot for the network, enter the 'Expert' mode of the Pi-Star setup and navigate down to the DMR network setup. Enter the credentials as shown below.

DMR Network	
Enable	1
Address	va6edn-server.local.mesh
Port	62031
Local	62032
Jitter	360
Password	s3cr37w0rd
Slot1	0
Slot2	1
Debug	0
ModeHang	20

Figure 19 DMR Hotspot Credentials

DVSwitch AMBE/PCM Transcoding

Transcoding of audio from the digital to analog domains is accomplished with DVSwitch software and hardware as illustrated in Table 5. This system is transparent to the network user.

Component	Embodiment	Function
MMDVM Bridge	Software/Python	Bridge to DMR radios
Analog Bridge	Software/Python	Bridge to USRP [3] network
AMBE Codec	Hardware/USB	Transcoding PCM to AMBE

Table 5 DVSwitch Software

USRPLink Bridging Software

Figure 20 illustrates the USRPLink software in more detail. In the diagram two repeaters and one radio that can source and sink USRP packets are connected by the IP network to a logical bridge construct implemented in the software. Several bridges can be implemented, routing between them is determined by either the talkgroup ID, or a default route. Each device connected to a bridge is called a 'Peer'.

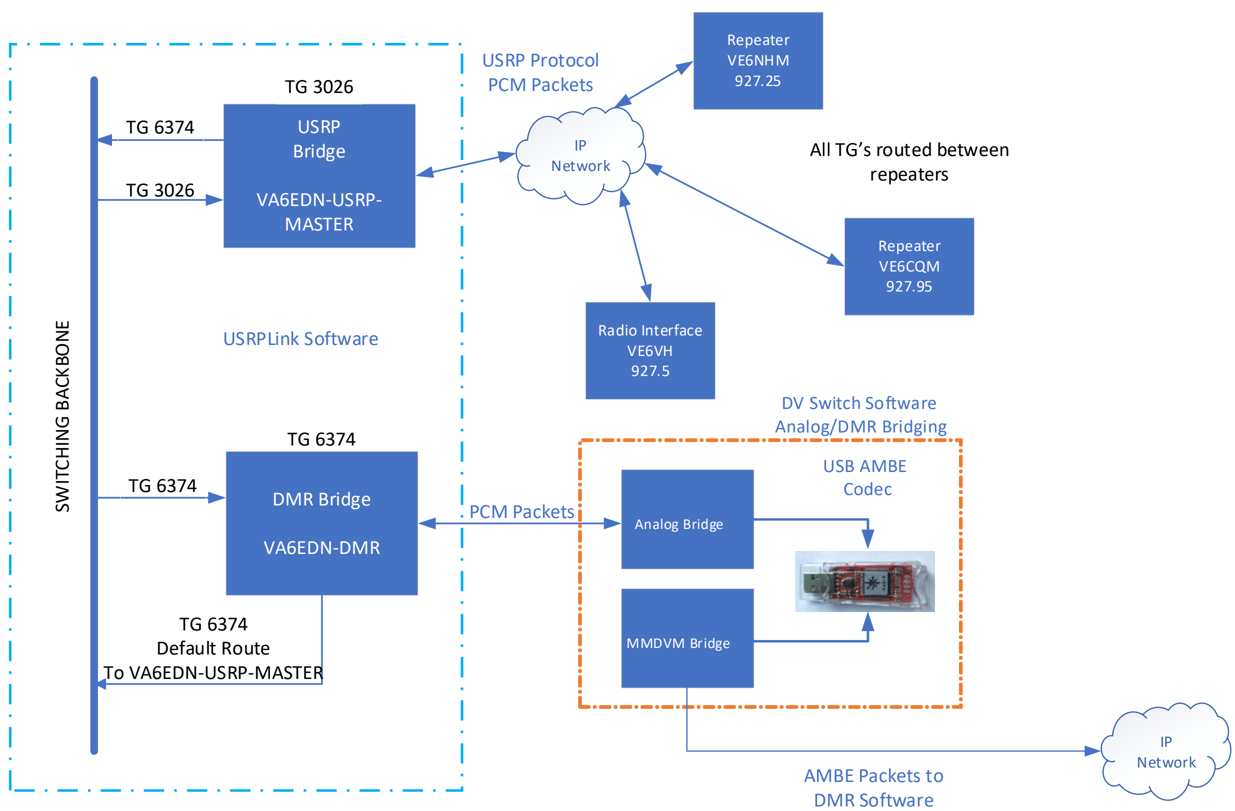


Figure 20 USRPLink Repeater and DMR Bridging software

In the example there are two bridge types shown, one implements the USRP protocol and the other DMR. The rules for the number of peers connected are as follows:

- A USRP bridge can have any number of peers.
- A DMR bridge can only have one peer.

The basic rules of linking are pretty straightforward:

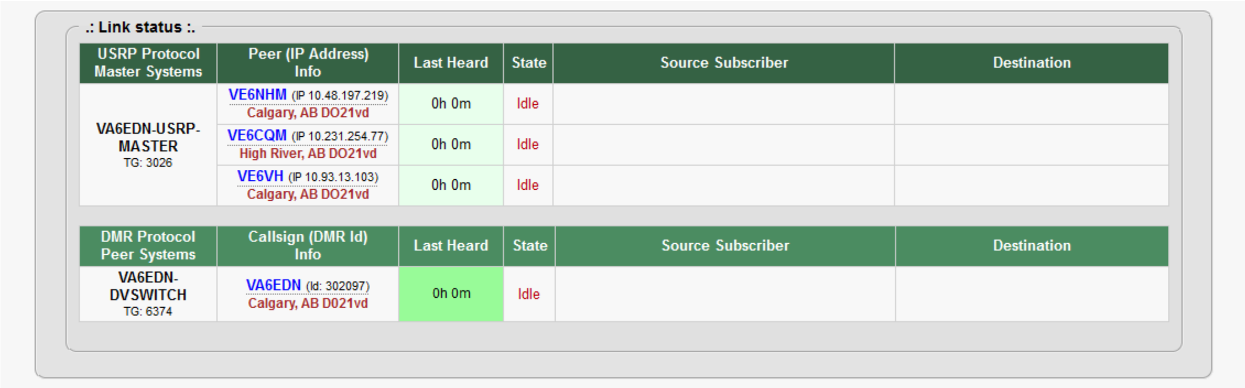
- All peers connected to the same bridge share traffic, regardless of the talkgroup ID (TG).
- Packets are routed to other bridges if they have the talkgroup ID of that bridge, and all peers also receive the same traffic.

This scheme enables repeaters to be grouped together onto a bridge, and packets to be routed between them based on talkgroup ID. In this example only one USRP bridge is implemented, however this is not a limitation of the software.

In the example, the USRP bridge is configured with TG 3026, and the DMR bridge with TG 6374. Traffic from any peer is repeated to all other peers on the bridge, regardless of the TG. The decision of whether to process an incoming TG is left to the repeater configuration. If a packet comes in with a TG of 6374, it will be repeated to all of the peers and also routed to the DMR bridge, using the switching backbone. This is only accessible to bridges. USRP peers can decide on which link to activate by specifying the TG, therefore steering the routing.

For the reverse route from DMR to the USRP, a default route is implemented, which states that any inbound traffic tagged with TG 6374 is to be routed to the USRP bridge. Any number of default routes can be added to a bridge.

Figure 21 Shows the USRPLink status page, which can be accessed from the VA6EDN website.


.: Link status :.

USRP Protocol Master Systems	Peer (IP Address) Info	Last Heard	State	Source Subscriber	Destination
VA6EDN-USRP-MASTER TG: 3026	VE6NHM (IP 10.48.197.219) Calgary, AB DO21vd	0h 0m	Idle		
	VE6CQM (IP 10.231.254.77) High River, AB DO21vd	0h 0m	Idle		
	VE6VH (IP 10.93.13.103) Calgary, AB DO21vd	0h 0m	Idle		

DMR Protocol Peer Systems	Callsign (DMR Id) Info	Last Heard	State	Source Subscriber	Destination
VA6EDN-DVSWITCH TG: 6374	VA6EDN (Id: 302097) Calgary, AB D021vd	0h 0m	Idle		

Figure 21 USRPLink software status

Experimental Repeater Project

The experimental repeater project was launched to develop a standard for a mesh-based repeater system and the hardware and software capabilities that would be utilized in an emergency situation. The 33 cm band was chosen as surplus commercial equipment was readily available, and it is currently being underutilized by the local amateur community.

The basic feature set is as follows:

1. Basic low power operation that can be run from a battery.
2. Fully remotely upgradeable.
3. Connection to the mesh PCM linking network.
4. Network routing using DTMF commands.

And an enhanced feature set to include the following:

1. Web browser based configuration.
2. Monitoring of local sensors, such as shack and PA temps and alarm reporting.
3. Access to Meshphone for VOIP calls.
4. Geolocation using an Android cell phone.
5. SMS messaging.

The repeaters on the network are low power experimental machines that are being used as to implement a local testbed. There are two, VE6NHM in NW Calgary, and VE6CQM in High River. The two are sufficiently spatially separated to ensure that they cannot hear each other and depend on the link for communication.

The chosen hardware platform for the repeater controller was a Raspberry Pi, which is a standalone single board computer that has the Linux operating system complete with all the necessary device drivers and has a large software compliment already available.

The strategy was to first develop the basic controller with remote upgrade capabilities, then move to a custom hardware platform known as a 'HAT' (Hardware Attached on Top), that would complete the functional requirements.

The first development was done on VE6NHM using mostly off the shelf components and breadboarding techniques. The second phase rolled out the HAT based controller, and VE6CQM was based on the production version of the board.

The prototypes satisfied the requirements of the basic feature set, these two repeaters are now in operation and connected to the linking network. They can be accessed either locally with a 900 MHz radio, or with a DMR radio and hotspot.

The enhanced feature set is the focus of current development.

Basic Feature Set

VE6NHM Phase One

The first phase of VE6NHM was designed to develop an FPGA based controller using a Finite state machine, that was first published in CQ Amateur Radio in 2016 [4]. This was a resurrection of an earlier project that dates back as far as 1986, realized in new flash-based FPGA technology, and the project also yielded an opportunity to experiment with CAD systems for PCB layout.

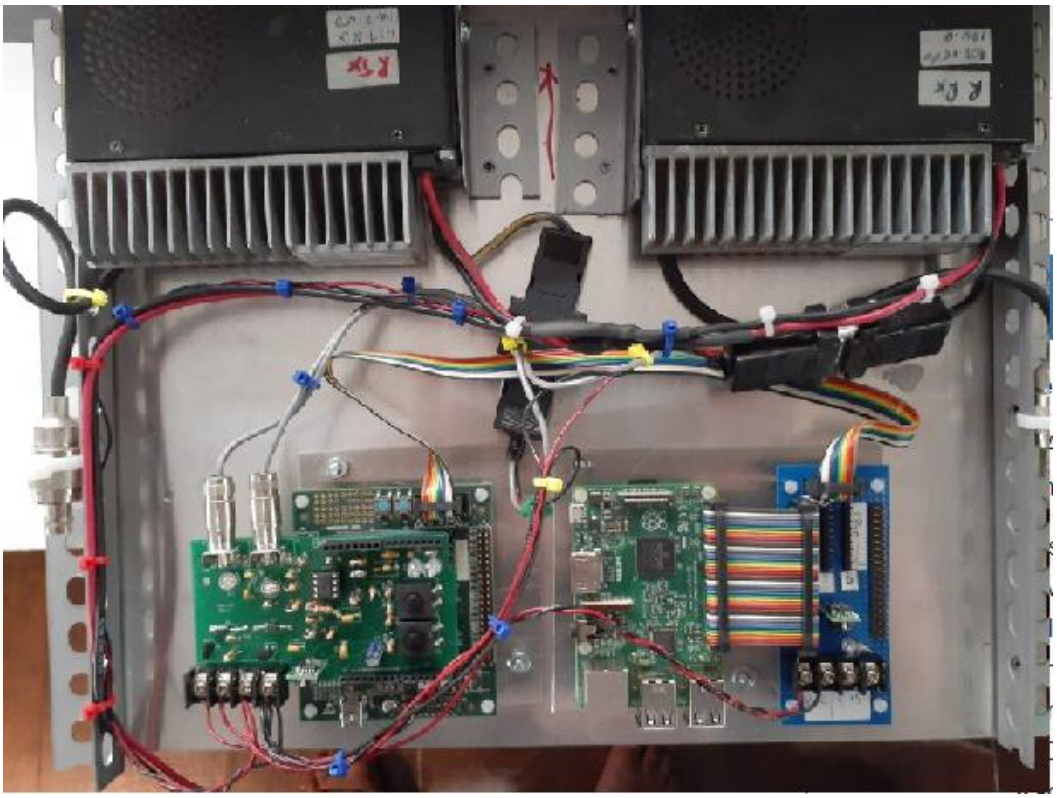


Figure 22 VE6NHM Phase 1

Figure 22 shows the phase 1 repeater, which was built using Kenwood TK941 transceivers for the receiver and transmitter. The receive radio had to be modified with new front end filters to accommodate a lower frequency, and the transmit radio had new firmware loaded into it to disable any timeouts.

The controller, on the lower left, was constructed from an off the shelf evaluation board for an Altera (now Intel) FPGA, that supports the Arduino style of connection. A second board was laid out to supply connectivity for the carrier sense, push to talk, and to provide an audio path and PWM filter for an identifier. The raspberry Pi, on the right, was set up to be able to remotely program the FPGA, a later requirement for deployment at a remote site.

This prototype implemented the basic feature set with the exception of the linking capability.

VE6NHM Phase Two

Figure 23 VE6NHM Phase 2

The second phase of VE6NHM was entered into to incorporate the previous control functions into a single HAT (Hardware Attached on Top) board for the raspberry Pi. The same FPGA was used, a better CODEC for audio digitizing was added, and a breakout board for connection to the 8-channel Analog to Digital Converter was also added. The previous analog board was refactored into an interface board that provided connectivity for audio, carrier and PTT, and is connected to the controller with an 8-pin modular cable.

This is the first to fully implement all functions in the digital domain. Generation of CW and ID tones, linking, DTMF detection and basic operation are all done digitally, and Analog signals are converted as they come onto and leave the board.

The controller will support up to 4 different state machines that are independent, addition analog ports can be added using a unique 'HAT on HAT' architecture.

This prototype implemented the full basic feature set.

VE6CQM

VE6CQM was the second repeater to be built on the Pi HAT platform, also using Kenwood TK941 radios. The controller is a production version that was built in Calgary by a local contract manufacturer.

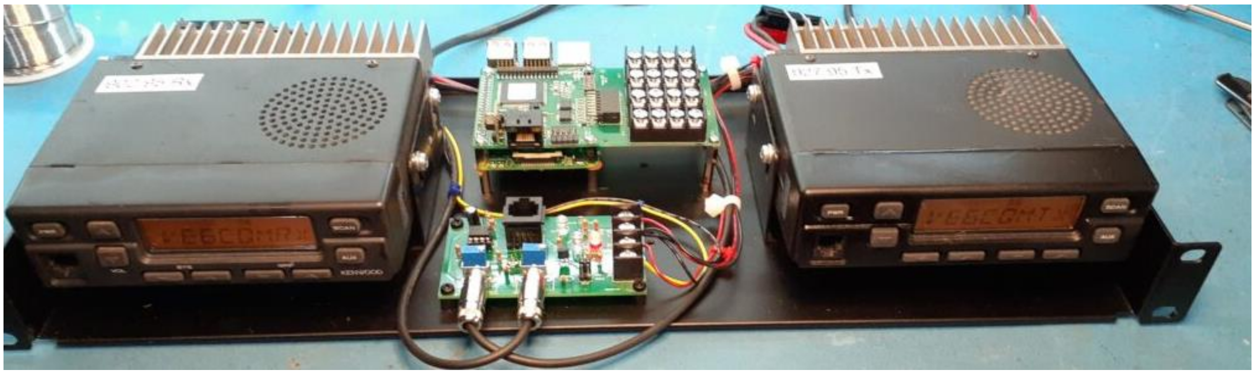


Figure 24 VE6CQM Production repeater

The raspberry Pi's are connected to an AREDN mesh node, which provides connectivity to the linking server. Digital audio is sent and received using the USRP [3] protocol.

Enhanced Feature Set

Implementation of the full feature set involved the development of a software 'stack' that runs mostly on the raspberry Pi. It consists of up to three layers, the first two implement the basic functionality and the third adds the enhanced functionality.

1. Hardware Layer. This provides connection to the radio, implements the state machine controller, a DSP based DTMF decoder. A data modem can also be included in this layer. Connection to the raspberry Pi is accomplished with two SPI (Serial Peripheral Interface) ports. This layer is coded in the Verilog language. A separate module can also be added using the 'topHAT' connector for additional functionality.
2. Interface layer. This layer sits between the hardware and application layer to handle time-critical functions such as audio streaming using the RTP [5] and USRP [3] protocols for VOIP calls and linking purposes. This layer is coded in the C language.
3. Application Layer. This layer implements the Web based (HTML) configuration, a SIP [6] client for managing VOIP calls and messaging, A/D alarm reporting, and a data manager for sending alarm messages and managing on-air data formats used by the chosen modem. This layer is coded in the Java language.

The software can be packaged in two different forms: one with only the lower 2 layers for basic functionality, and all three for the enhanced.

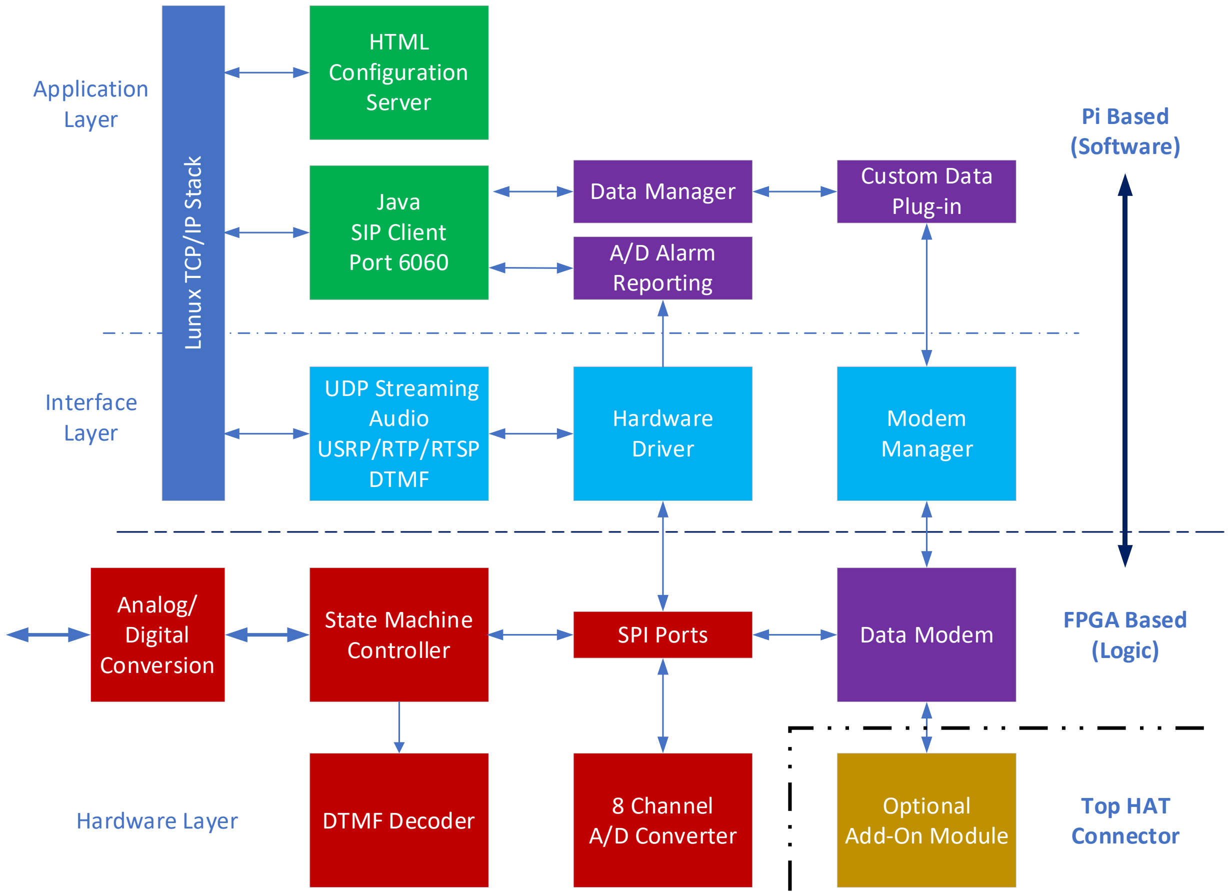


Figure 25 Controller Software Stack

Geolocation

The Geolocation project is a separate project that utilizes the data modem or RS232 interface, and a custom data plug-in to support the protocol. Currently two different avenues are being pursued:

1. Built-in modems. The Kenwood TK981 radio already has a built-in data modem using MSK that can support speeds up to 2400 bps, called ‘Fleetsync’. The first phase of the geolocation project was to explore the use of this mode. Restricted to radios that have this capability only.
2. DSP based modems. A DSP based 202C modem has been developed, that can support the AX.25 physical frame format. This requires external TNC devices at the radio as this is not a built-in mode but enables any radio to be utilized.

The first goal was to enable a cell phone to be used as the mobile terminal device for both geolocation and SMS messaging by providing a local area network using WiFi to which the phone can connect, and APRSDroid software to send the position information, as well as process SMS messages.

The goal was accomplished using an ESP8266 based IOT processor to provide an access point for the cell phone, and software was developed to convert APRS-IS messages into standard NMEA sentences, which are then sent to a TK981 radio using the serial port.

A diagram of the mobile installation is shown in Figure 26.

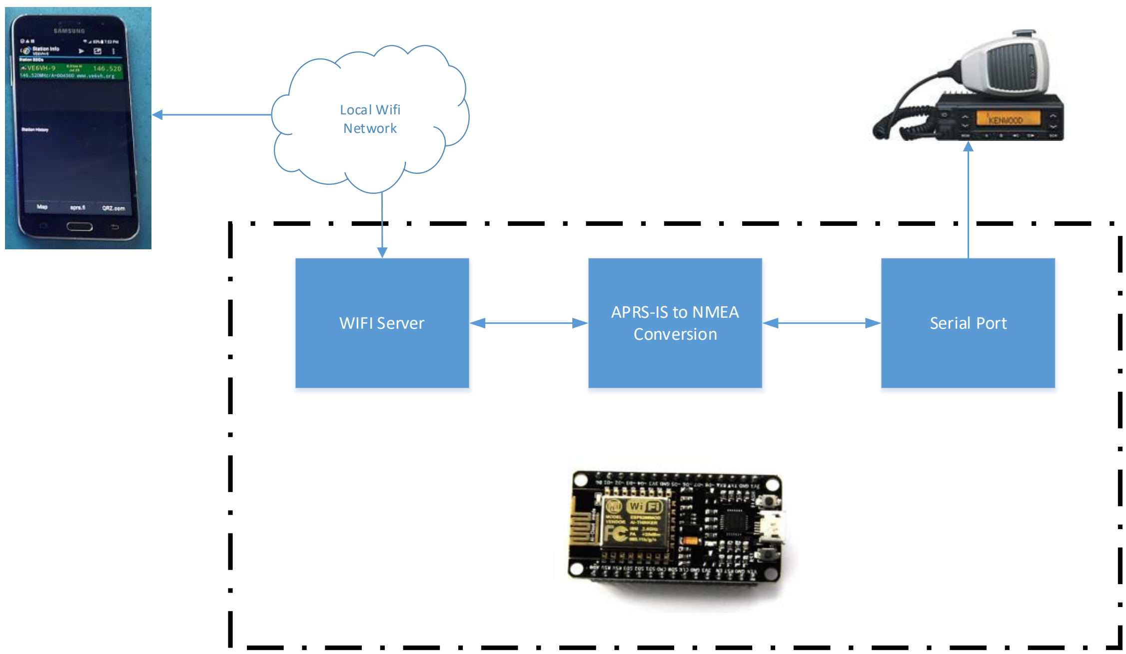


Figure 26 Cell phone to TK981 mobile hardware

A second TK981 radio was used as the receiver, which sent the received geolocation messages to the serial port of the controller. Figure 27 show the messages being source by APRSDroid and received at the controller using test software.

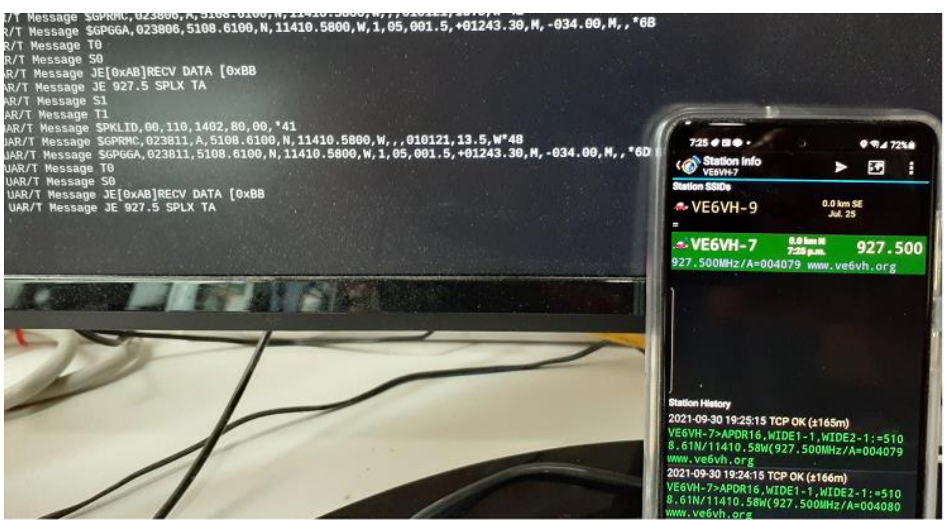


Figure 27 APRSDroid to Controller messaging

Messaging

This will be the final step in the development cycle and is still being defined. At present a software portal from Meshmail to a SIP phone has been developed, which runs on the local server and is illustrated in Figure 28. The end goal is to extend this capability to a mobile radio, using a data transport and by enhancing the software in the mobile hardware platform.

This software implements a transport between either an external or mesh based email address. The message body is reformatted and sent to the PBX as a SIP message, where it can be displayed on a mesh phone, as can be seen in Figure 29.

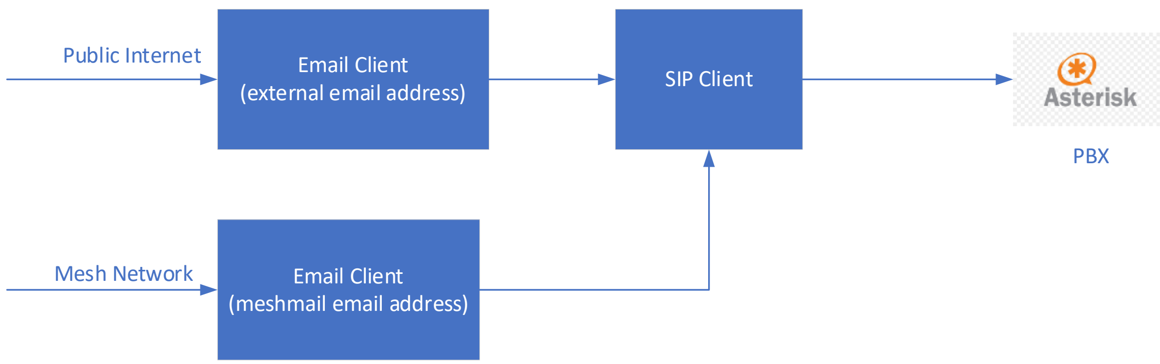


Figure 28 Email to SIP portal

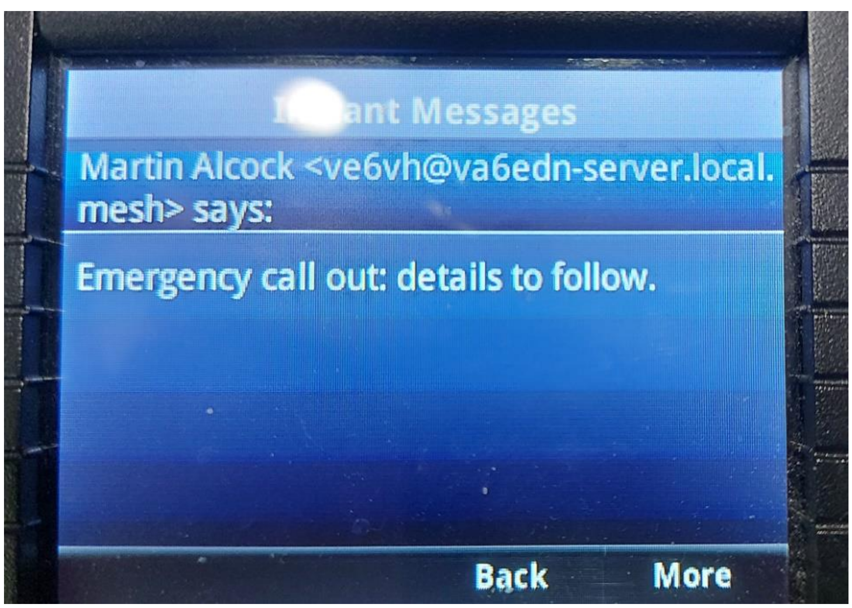


Figure 29 SIP message on a Polycom phone

SDR based Linking

Not to be confused with the baseband audio linking project, the SDR based link project is designed for providing an RF link between repeaters that is not normally accessible using the mesh network hardware.

The plan is to add an SDR transceiver to the controller that is based on an AT86RF215 transceiver chip, that is capable of operating in the 900MHz or 2.4 GHz frequency bands. An outboard power amplifier can be added to the transceiver that will provide up to 10W continuously, or 20W peak.

There are several built-in modulation schemes in the chip that comply with the 802.16.4 WPAN protocol, software will have to be developed for sending ethernet packets back and forth.

The transmitter calculations are shown in Table 6 and the receiver in Table 7. An assumption that a Yagi antenna will be used with a minimum of 6dB gain.

Parameter	dBm
Transmitter output power	14.5
Antenna Gain	6
Coax Loss	-1.5
PA gain	30
Effective Radiated Power	49

Table 6 Transmitter calculations

Parameter	dBm
Receiver sensitivity	-85
Antenna Gain	-6
Coax Loss	1.5
LNA gain	-20
Fade Margin	30
Minimum input signal to antenna	-79.5

Table 7 Receiver calculations

From the range equation, the distance achievable can be found from the formula:

$$20 \times \left[Log\left(\frac{4\pi}{0.3}\right) + Log(f(GHz)) + Log(d)\right]$$

Without the distance term, the constants are 31.66dB. Give the dynamic range 49-(-79.5) or 128 dB, less the constants leaves approx. 97 dB attributable to the distance term. This yields a maximum of approx. 70 Km, which will work in most link applications.

