This howto for a repeater controller was designed for a specific repeater but is presented here to show how a very capable controller can be assembled using Allstar at minimal cost.

Allstar Repeater Controller Design and Documentation

by Doug Crompton, WA3DSP

The custom repeater controller consists of a Raspberry Pi 4 running hamvoip Allstar software (hamvoip.org). This controls the Kenwood NXR-710 repeater and provides additional control and sensing capabilities. It has a 16 bit four channel A/D which allows the measurement of two temperature and two voltage channels, two single pole double throw relays, and four GPIO inputs. This will allow the measurement of temperatures both inside and outside the repeater shack, measurement of power voltages, relay control of power or other signals and monitoring of levels such as door security at the shack.

Software is programmed on a 16GB SD card with a backup card supplied. In addition there is a 32G USB stick for additional storage of programs and messages.

Front Panel LED's

Red – Power, Yellow – COS, Blue – PTT, Green - Heartbeat

Several scripts are included to control the external IO. This includes:

relay 1|2 on|off|cycle

Allowing control of two relays to either on or off or cycle which turns it on for 5 seconds and then off. The relays are double throw so the NC (normally closed) connection is enabled when off and by default. Turning the relay on or cycling the relay would open this connection either permanently or cycle it open and then closed.

read_voltage 1|2

This reads the voltage applied to the V1 or V2 terminals to ground. The absolute maximum voltage that can be applied to V1 or V2 is 50 volts. Voltage is read with a 16 bit A/D with .1 volt resolution.

read_temperature 1|2

This reads the temperature for either sensor 1 or 2 if they are connected. The temperature sensors require three connections to the rear panel, SV, S1|2, and ground. Temperature is read with a 16 bit A/D with .1 degree fahrenheit resolution.

Both temperature and voltage are read as a single value that can be used to annunciate the reading using the Allstar controller.

read_temperature inside|outside

This reads the inside (sensor 1) or outside (sensor 2) temperature and says the result on the primary node. This can also be initiated by the DTMF command *831 for inside and *832 for outside.

readbit 1|2|3|4

This reads the status of GPIO bits 1-4. Either a 1 or 0 is returned depending on whether the bit is high or low. These bits are pulled to +12V in the high or open state.

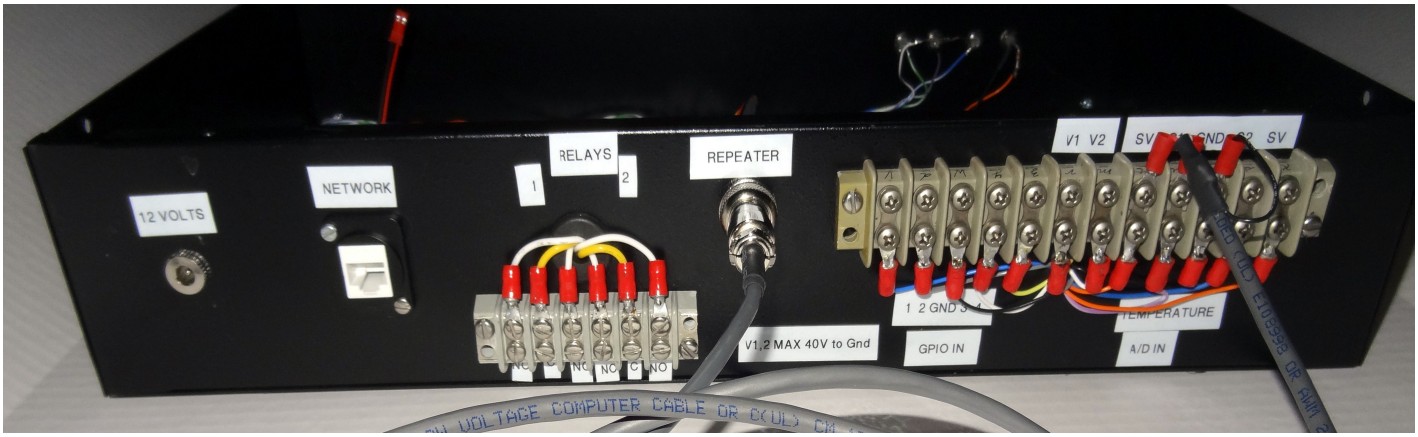
readbit_int 1|2|3|4

This script reads GPIO bits 1-4 using an interrupt. Nothing is returned unless the bit falls low for 5 seconds. This is the preferred way to read a bit continuously as it takes little CPU resources.

door_check

This script reads the repeater door status on GPIO bit 1. When this bit is low the door is locked. If it becomes high the door is unlocked. An announcement is made on the repeater when there is a status change of unlocked or secure. If the door remains in the unlocked condition and announcement is made every 15 minutes until it is locked. This script is run at boot in /etc/rc.local with the nohup command.

Back Panel Connections



12V power is supplied via a barrel connector shown at the left. Maximum current draw is about 2A

The Local LAN and Internet are via the next connector to the right. Two single pole double throw relay contacts via a six pin terminal strip are available. These are controlled by the relay script. Contacts are marked NC (normally closed), C (common), and NO (normally open). Relays are rated at 10A 250V.

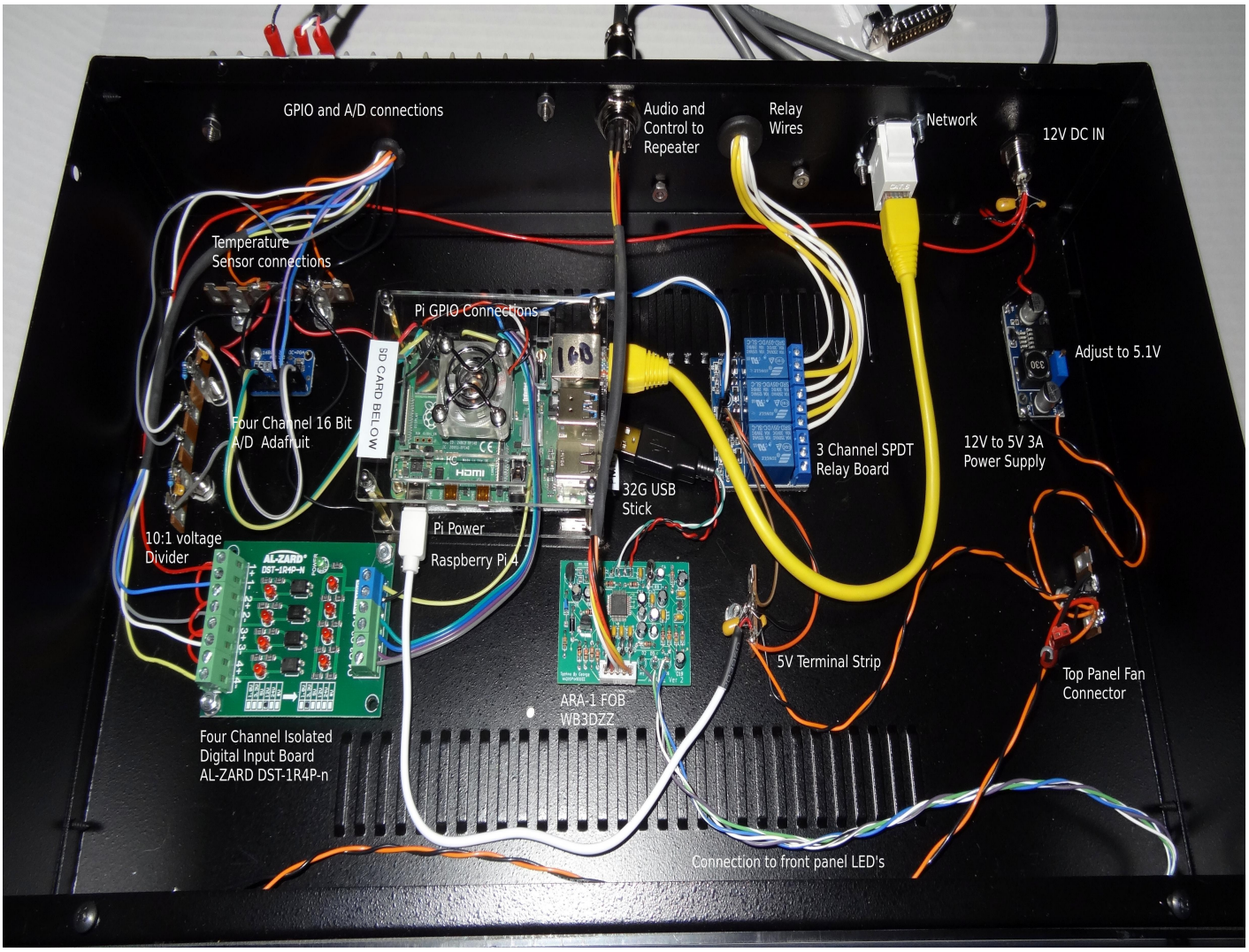
The repeater connects with an 8 pin microphone connector. See connection appendix for pinouts.

A 12 pin terminal strip is for connection of GPIO, and A/D inputs. Starting from the left GPIO inputs pin 1 and 2, Pin 3 is ground then pins 4 and 5 are GPIO inputs 3 and 4. These are 12 V high and ground low inputs. Pin 1 is dedicated for door security – see door_check script above.

Pins 6 and 7 are voltage inputs to the A/D. These have a 50 volt absolute maximum voltage level and are typically used to measure power supply or other voltages. See the read_voltage script above.

Pins 8,9,10 and 10,11,12 are temperature probe inputs. These measure temperature using the read_temperature script above. Each probe uses an Analog Devices ADS1115 chip. Pin 8 (SV) supplies power to probe 1 and pin 9 is the sense voltage from the sensor. Pin 10 is a common ground. Pin 11 is sense voltage from probe 2 and pin 12 is probe 2 power. The probes wiring is color coded red for power, white for sense, and black for ground. The A/D is an Adafruit board – https://www.adafruit.com/product/1085. The 4 bit GPIO opto-isolated input board is a DST-1R4P-N available from Amazon or Ebay - https://www.amazon.com/Converter-Optocoupler-Isolator-Maluokasa-DST-1R4P-N/dp/B07NNLYWCB. The relay board is similar to this type - http://www.icstation.com/channel-relay-module-relay-expansion-board-with-optocoupler-insulation-arduino-p-9665.html

Controller Internals

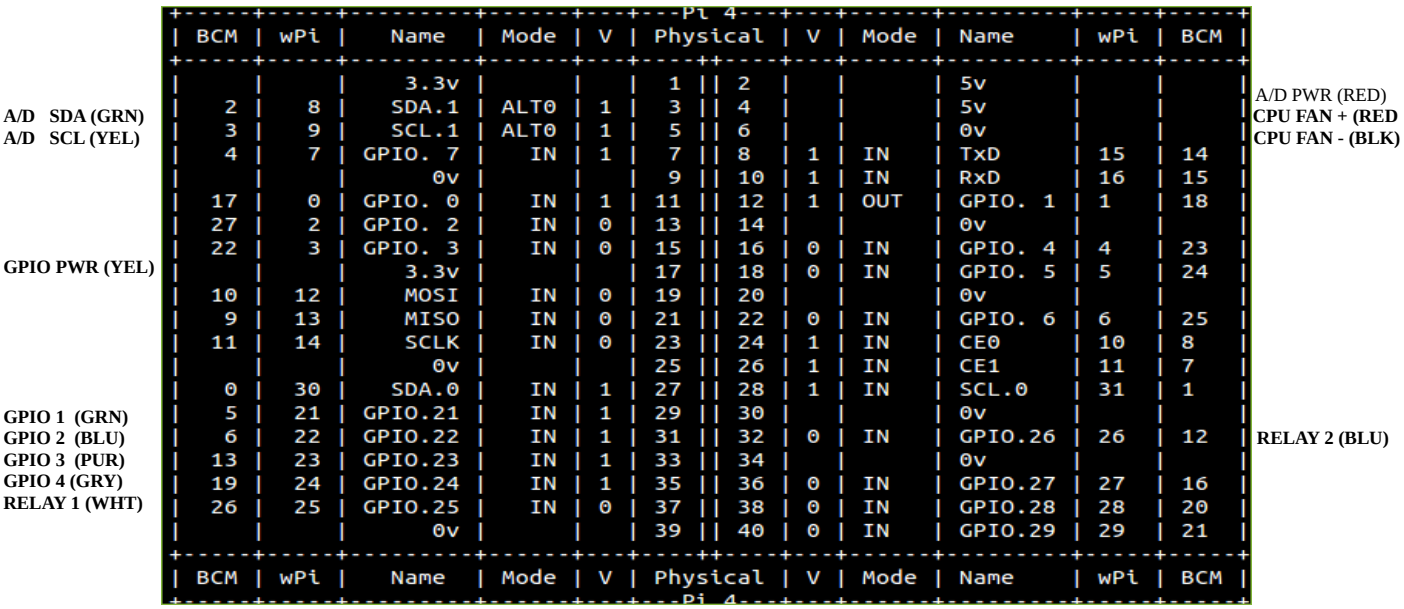


The controller is powered by 12v at the upper right through a barrel connector. This feeds to a 12v to 5V 3A power converter which supplies power for the majority of the components.

The main controller is the Raspberry Pi 4. It connects to the Internet via the RJ45 connecotr on the rear panel. It also connects via USB to the ARA-1 sound FOB. This interfaces the audio and control signals to the repeater via the 8 pin mic connector on the rear panel. Another Pi USB connection is used for a 32G USB stick. This is used to store messages, logs, and other data.

The GPIO of the Pi is connected to a four channel 16 Bit A/D converter board, A three channel relay board, and a four channel opto isolated GPIO input board.

GPIO CONNECTIONS



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1 || 2			5v		
2	8	SDA.1	ALT0	1	3 || 4			5v		
3	9	SCL.1	ALT0	1	5 || 6			0v		
4	7	GPIO. 7	IN	1	7 || 8	1	IN	TxD	15	14
		0v			9 || 10	1	IN	RxD	16	15
17	0	GPIO. 0	IN	1	11 || 12	1	OUT	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13 || 14			0v		
22	3	GPIO. 3	IN	0	15 || 16	0	IN	GPIO. 4	4	23
		3.3v			17 || 18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19 || 20			0v		
9	13	MISO	IN	0	21 || 22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23 || 24	1	IN	CE0	10	8
		0v			25 || 26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27 || 28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29 || 30			0v		
6	22	GPIO.22	IN	1	31 || 32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	1	33 || 34			0v		
19	24	GPIO.24	IN	1	35 || 36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37 || 38	0	IN	GPIO.28	28	20
		0v			39 || 40	0	IN	GPIO.29	29	21

Package or Setup Requirements

devmon is loaded for USB stick mounting. The 32G USB stick is formatted ext4 with a label of MS1 – thus creating a mounted /media/MS1 at boot. These commands are executed once at system setup to initialize auto-mounting -

systemctl enable devmon@root
systemctl start devmon@root

I2C Requirements

In directory /etc/modules-load.d create file called i2c.conf with the single line content
i2c-dev

In /boot/config.txt add the line - dtparam=i2c_arm=on

Then reboot

Running 'gpio i2cd' should show the following – 16 bit A/D at i2c address 48

Multiple A/D's can be added at different addresses on the same bus.

```
# gpio i2cd

     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Repeater Connections

8 PIN Mic on Controller	**DB25 Male to NXR-710**
Pin 1 Audio in (RX) (RED)	Pin 11 (RA) (GRN)
Pin 2 Audio out (TX) (WHT)	Pin 9 (TA) (BRN)
Pin 3 COS (YEL)	Pin 21 TOR (WHT)
Pin 4 PTT (ORN)	Pin 16 (EPTT) (RED)
Pin 8 GND (BLK)	Pin 7, 12, 19 (DG,RXG,TXG) (Shield,BLK)

8 pin colors are internal from ARA1 FOB to chassis connector, DB25 colors are from 8 pin plug to DB25.

The repeater operates in duplex mode when using Allstar and can be switched between duplex (Allstar) and repeater (internal controller) with DTMF sent to the repeater receiver. This allows backup repeater internal control in the event of an Allstar failure.

Internet Routing

Allstar is statically addressed in Archlinux setup. Of course you could select any address compatible with your Internet connection an adjust the accordingly. Ports can be customized and can be configured to your liking remembering that when you change the web server and manager ports you need to use the changes in your browser and allmon.ini file.

The following standard ports are forwarded to the Allstar controller from the router -

AllstarIAX - 4569 UDP
AllstarSSH – 222 TCP
AllstarHTTP – 80 TCP
AllstarAMI – 5038 TCP/UDP
Echolink – 5198, 5199 UDP

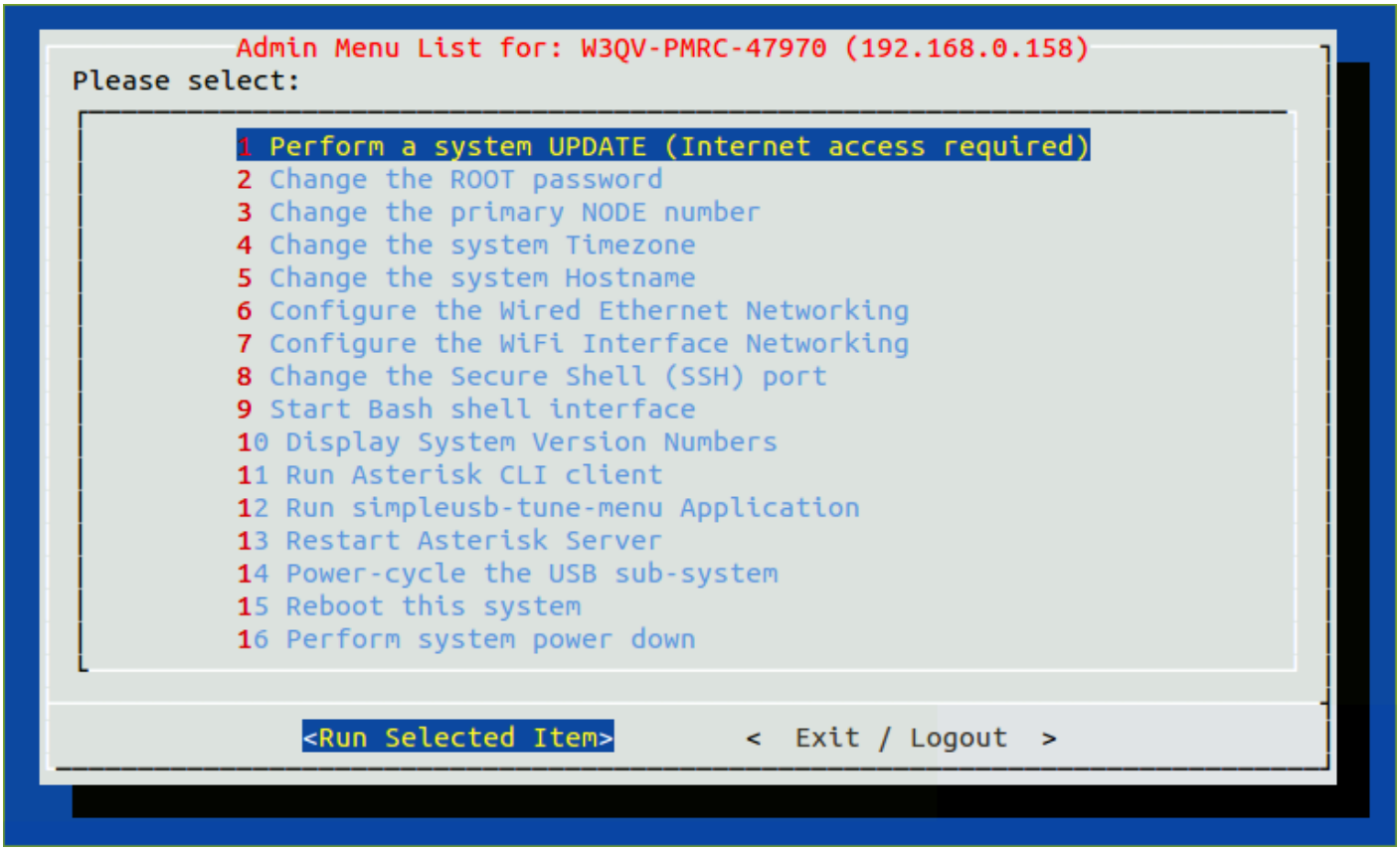
Accessing Allstar

Allstar can be accessed and controlled in several ways depending on whether you are at the repeater site using the local Internet or if you are at a remote Internet location.

The site can be accessed by administrators using the Cloudberry remote assistant. Cloudberry is preferred as Teamviewer tends to switch into non-free mode which is a real pain. If you are an administrator you will be given the login number and password to use. This program is a free download for any Windows computer. Once logged in you will see the full screen and have full control over the Windows computer at the repeater site.

There are screen icons for the programming the Kenwood repeater, and Putty for accessing the Allstar system via ssh login or using the firefox browser to locally run supermon.

Direct access to Allstar at the repeater site is via the Putty program which is an icon on the main Windows screen. Enter the IP address and port and save your selection. Select it and connect. It will ask for the username and password. The username is always root and the password will be given to administrators. After successfully logging in you will see this menu screen -



Most of the menu items are for experienced Allstar sysops but item 9 “Start Bash shell

interface” takes you to the Linux prompt where script commands and other system maintenance items can take place. At the Linux prompt you can directly run any of the scripts such as controlling the relays. To turn relay 1 on just type -

relay 1 on

To return to the main menu type 'exit' at the Linux prompt.

This same putty login can be directly accomplished from anywhere on the Internet using the sites public IP address and port 222 and login credentials just as you would locally.

Supermon can also be used locally at the repeater with the Firefox browser. The local URL would be (use your assigned IP address and port) -

http://your_PI_IP address:port/supermon2

To access Supermon remotely on the Internet the public IP address would be used -

http://Public IP:Port/supermon2

Anyone can use the public address to monitor status with Supermon but for control you must login to Supermon using the credentials supplied to administrators.

What if the public IP address at the repeater site changes?

Anyone running Allstar can determine the current IP address of any active node by using dns-query command. Enter the Bash shell item 9 from the main menu on your Allstar server and type -

dns-query <node-number>

Where <node-number> is the number of the remote node.

The current public IP address will be returned and can be used in the above examples.

Questions about this design or its operation can be directed to - wa3dsp@gmail.com

Example files for this project can be found at -

http://crompton.com/hamradio/repeater_controller/repeater_control_files.tar