

## AllStar Handbook

Clicking on item listed in table of contents should take you straight to that part of the document.

Latest version of this document can be downloaded from:

<http://texomarepeatergroup.com/allstarhandbook.pdf>

I initially created this document for my own use, dumping in bits of information on using Allstar as I ran across them, so I could reference back to it. The information is based upon using a HAMVOIP Image. If anyone has suggestions for additional information, or see incorrect information, please pop me an email. QRZ has my email information. KF5LBT.

### Contents

Links For Getting Started .....	3
SD Card Formatter .....	3
Win32 Disk Imager .....	3
AllStar Node Setup and Configuration Guide .....	3
Allstar Link Site:.....	3
DMK Engineering.....	3
URI Cables.....	3
Hamvoip .....	4
Hamvoip Release Notes .....	4
How to use iaxrpt with Allstar on the hamvoip releases .....	4
WA2KJC Allstar and other Amateur Radio Tips .....	4
WiFi WPA2 and WPA3 Router Settings .....	5
SD Cards .....	5
Clone Pi SD Card .....	6
PC Software for Node Maintenance .....	7
PuTTY.....	7
WinSCP .....	7
Setting RX Level using PuTTY Session .....	7
Parrot Allstar Node 55553 .....	8
Extend a node number up to 10 nodes .....	9
Multiple Nodes Behind One Firewall.....	11
Port Forwarding A Node .....	14
CPU Stats .....	15
CPU Turbo Mode.....	16
Supermon .....	17

Configuration Editor - SimpleUSB-Tune-Control .....	17
RX Level Adjustment .....	18
Configuration Editor – Supermon-favorites.ini file .....	20
Using a Black list or White List .....	21
Connection Log .....	22
Load Average .....	22
CPU Temperature .....	22
Core Dump Message .....	22
Procedure to Change Refresh Time on Supermon .....	24
Procedure to Display Local Temperature and Weather on Supermon .....	24
Procedure to Add Internet Links to Supermon .....	25
Display Multiple Nodes on Supermon .....	25
Allmon2 .....	31
Set Pi Display Monitor Sleep Time .....	31
Local Node Announcements .....	32
Telemetry Audio Levels.....	32
Add Weather Conditions to the Hourly Time Local Announcement.....	33
Disable Time and Id .....	34
Unkey Courtesy Tone Enable/Disable .....	34
Node Auto Connect at Startup .....	35
ASL database - astdb.....	36
AutoSky Skywarn.....	36
Echolink Setup On Allstar Node.....	37
APRS Setup .....	39
IP Address .....	40
Hardware Identification .....	40
Node Registration Status .....	41
Halt Button (Hardware Means to Safely Shutdown) .....	41
ArchLinux .....	41
pacman commands .....	41
DTMF Tones.....	42
DTMF Function List.....	42
Confirming Tones Received .....	43
Sending Accurate Baofeng DTMF Tones from the UV-5R .....	46

Radioless Node.....	46
Radio Mobile - Online tool for predicting the propagation of radio waves.....	47
Distance and Bearing Calculator.....	48
Script descriptions .....	48
Scheduled Node Connect Disconnect crontab Example.....	52
Schedule Commands with crontab .....	53
About crontab .....	53
Command syntax .....	53
Overview .....	54
The crontab file.....	55
Environment settings .....	55
Configuration .....	56

## **Links For Getting Started**

### **[SD Card Formatter](#)**

This is a link to download software to your PC to be able to format SD Cards.

### **[Win32 Disk Imager](#)**

This is a link to download software to your PC to be able to copy Allstar Image to SD Card

### **[AllStar Node Setup and Configuration Guide](#)**

Has information on how to set up an AllStar node, hardware required, loading software etc.

### **[Allstar Link Site:](#)**

Where to go for obtaining a node number, along with much more information about the Allstar Link system. You will need to create a login to get a node number.

### **[DMK Engineering](#)**

This website is the home of DMK Engineering, the makers of the USB Radio Interface (URI).

### **[URI Cables](#)**

Radio interface cables made for various radios and Allstar Link configurations.

### **[Hamvoip](#)**

This is the link for information on the Hamvoip Allstar. Contains the information for creating an image to load onto an SD Card to create an Allstar node. Site contains many useful how to information links and information.

### **[Hamvoip Release Notes](#)**

This is a link to information on releases

### **[How to use iaxrpt with Allstar on the hamvoip releases](#)**

This is a link to a Hamvoip PDF How to document on setting up iaxrpt on windows, android.

### **[WA2KJC Allstar and other Amateur Radio Tips](#)**

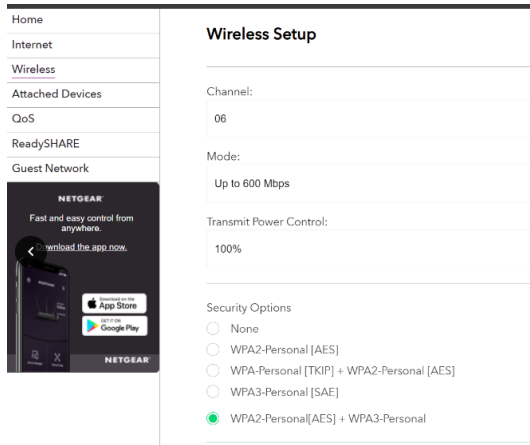
This is a link to WA2KJC (Dave) Dropbox shared files. Dave came across this handbook and contacted me with how to information he has created for Allstar and other Amateur Radio related topics. I recommend checking out what he has and check back to it occasionally as he continues to add content. Thanks Dave for suggesting this link to your information!

### **[Absolute Tech](#)**

This is a link that has great information on Raspberry Pi, Linux, Programming, Projects etc.

## WiFi WPA2 and WPA3 Router Settings

Tom, WOKFL, informed me of an issue that some folks might come across with newer home routers. Tom had upgraded to a new router and afterwards his Allstar node would no longer connect. He found out that the router was set to the latest more secure WPA3 protocol. He was able to change the setting in his router, (see screen shot below) to have it connect to WPA2 devices, and this fixed the problem. Thank you Tom for sharing this information !



## SD Cards

The information below is from Arm Allstar by Doug Crompton.

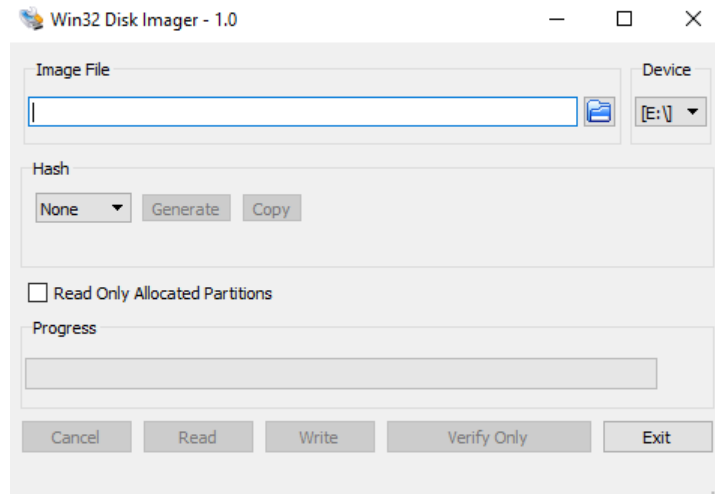
Just a reminder though that the read/write performance of the SD cards in a Linux application like Asterisk Allstar is immaterial after boot. So using a faster card is not going to make your system run faster. It would allow faster creation of the initial image and possibly faster boots (measured in a few seconds) but in most cases it would not be practical economically to purchase a super fast SD card.

The speed which a card writes is also heavily effected by the type of writer (usb2/3) and the SW used to write. So while the relative differences in these tests might be valid the absolute values of these tests could differ greatly based on the hardware and SW used to write.

Tried and true my choice is still the SanDisk Ultra but I am sure there are other cards in it's class that could be used. Also as I mentioned on the web page unless you have a compelling reason it is **NOT a good idea to expand the filesystem on these cards after installing the image**. Doing so makes total image backup a time consuming process especially on 16GB and larger cards and does nothing to improve performance or longevity.

## Clone Pi SD Card

1. Get everything set up just the way you want it on your Raspberry Pi, whatever you're using it for. Then shut down the Pi and remove the SD card. Insert the SD card into your computer.
2. Start up [Win32DiskImager](#), a program that you probably have from [when you first set up your Pi](#). (If you're on OS X or Linux, you'll have to [use the dd command as described here](#) instead of these steps).



3. In the "Image File" box, enter the path of your soon-to-be image file. For example, I put mine in C:\Users\hause\Documents\Ham Radio\Allstar\AllStarPi\Backup
4. Under the "Device" box, select your SD card.
5. Click the "Read" button to create the image file from your card.
6. When it's done creating the image file, you can eject your SD card and put it back in your Raspberry Pi. Keep that IMG file in a safe place.

Now, if anything ever goes wrong with your Pi, you can restore your fully-set-up image using the reverse instructions:

1. Insert the SD card back into your computer.
2. Head to the start menu or screen and type "disk management." Open the disk management program and find your SD card in the list.
3. Right-click and delete all the partitions on your SD card. When it's empty, right-click on it and format it (it doesn't matter what filesystem you format it to, your computer just needs to recognize it).
4. Open Win32DiskImager again and browse for your image file. Select your device from the Device dropdown just as you did before.
5. This time, click "Write" to write the image to the SD card.
6. When it finishes, eject the SD card and re-insert it into your Raspberry Pi. When you boot it up, it should be in the exact same state it was in when you first cloned the SD card.

# PC Software for Node Maintenance

## PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. Link to download: <https://www.putty.org/>

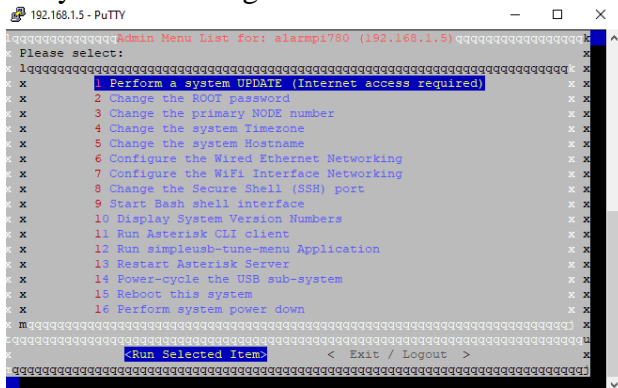
## WinSCP

Download this to PC, allows an easy way to transfer files between the Raspberry Pi and the PC. I had installed the PuTTY before hand, and saved a session for my Allstar Pi. The WinSCP detected the PuTTY on the PC, and allowed loading in of the Allstar Pi session into it. Link for WinSCP: <https://winscp.net/eng/index.php>

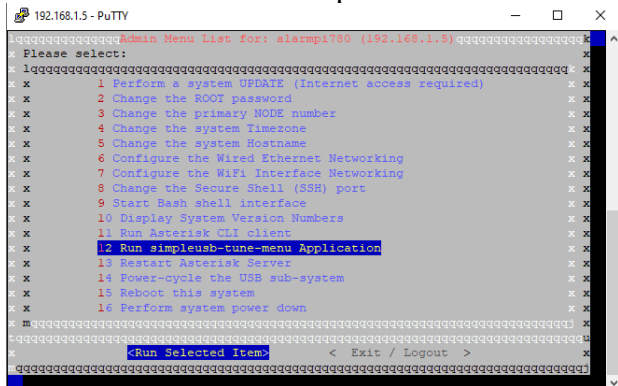
## Setting RX Level using PuTTY Session

There are 2 methods for setting the RX Level. One is using a PuTTY session described here, another is through Supermon also outlined in the Supermon section of this document.

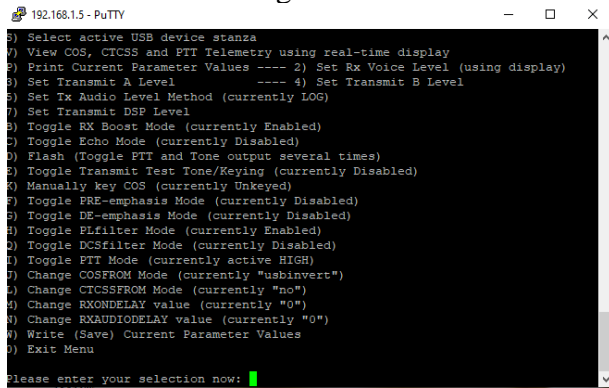
Putty into node to get to admin screen:



Select item 12 for the simpleusb



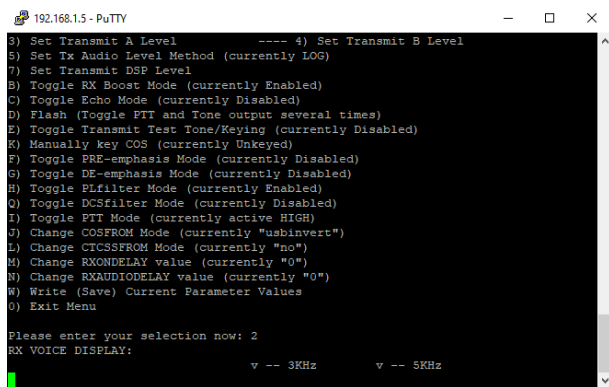
Will see the following



```
192.168.1.5 - PuTTY
5) 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)
S) Set Transmit A Level ---- 4) Set Transmit B Level
E) Set Tx Audio Level Method (currently LOG)
T) Set Transmit DSP Level
B) Toggle RX Boost Mode (currently Enabled)
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 Enabled)
Q) Toggle DCSfilter Mode (currently Disabled)
I) Toggle PTT Mode (currently active HIGH)
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
O) Exit Menu
Please enter your selection now: █
```

On SHARI nodes, PRE-emphasis (Item F) and DE-emphasis (Item G) need to be enabled and the RX Boost turned Disabled (Item B).

Select 2 Set RX Voice Level, will see following



```
192.168.1.5 - PuTTY
3) Set Transmit A Level ---- 4) Set Transmit B Level
5) Set Tx Audio Level Method (currently LOG)
T) Set Transmit DSP Level
B) Toggle RX Boost Mode (currently Enabled)
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 Enabled)
Q) Toggle DCSfilter Mode (currently Disabled)
I) Toggle PTT Mode (currently active HIGH)
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
O) Exit Menu
Please enter your selection now: 2
RX VOICE DISPLAY:
█ v -- 3KHz v -- 5KHz
```

Now transmit, and you will see a bar go across the bottom of the screen as you talk, press your enter, or Return key and it will display the current setting number, enter in a new number and repeat until your happy with the setting. Then **BE SURE to select W on the menu to write the setting, if you don't, your change won't be saved !**

## Parrot Allstar Node 55553

Tried this node out and it works well. Just connect to it, say a few words, it will come back and tell you if your audio is to low, high or about average.



## Extend a node number up to 10 nodes

Note: I have done this with one of my nodes, and it seems to work fine. After I did the node extension, I ran the script, `firsttime.sh` from the bash prompt on the node. The script does the updates to all the required files, just like when you first set up a node. I did notice that afterwards my node showed up as a private node, but that will take care of itself once the `asl` database node thing does its update which occurs in the early morning, so I just left my node powered up overnight, and by the next morning it was okay.

One can now extend a node number to 10 nodes. This eliminates having to request and wait for a new node numbers to be assigned. So for example lets say you have node number 12345. You now can have up to 10 nodes with the base number of 12345. The node numbers would go like this:

123450

123451

123452

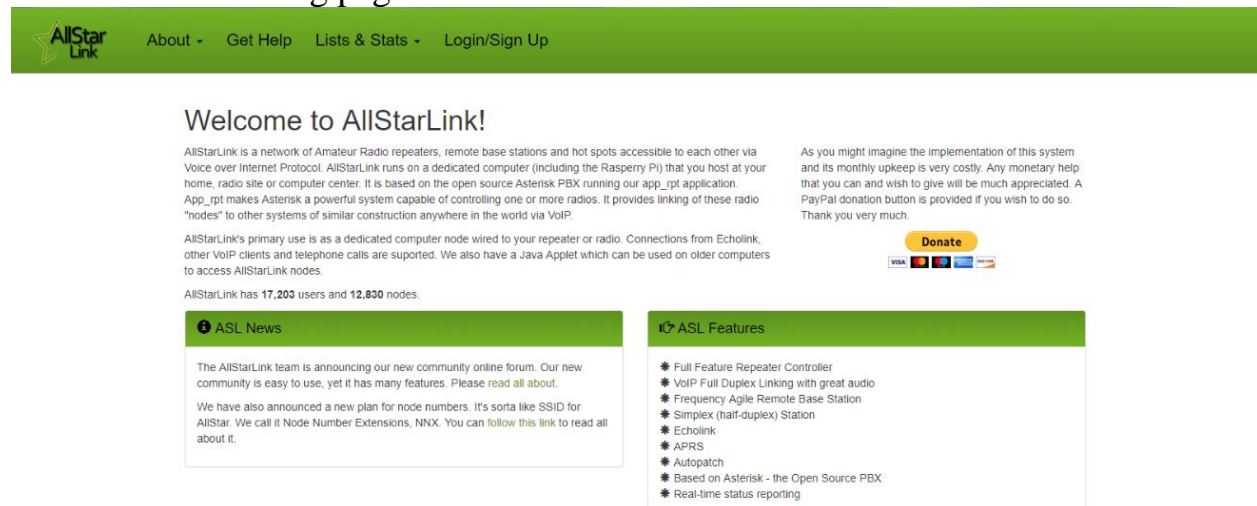
And so on down to

123459

To do this go to the Allstar Link web site.

Link to site: <https://web-tpa.allstarlink.org/>

Should see following page:



The screenshot shows the AllStarLink website homepage. At the top is a green navigation bar with the AllStarLink logo and links for 'About', 'Get Help', 'Lists & Stats', and 'Login/Sign Up'. Below the navigation bar is a 'Welcome to AllStarLink!' section. This section contains introductory text about the network, a 'Donate' button with payment icons, and two columns of content: 'ASL News' and 'ASL Features'. The 'ASL News' column contains two paragraphs of text. The 'ASL Features' column contains a bulleted list of features.

**AllStarLink** About - Get Help Lists & Stats - Login/Sign Up

### Welcome to AllStarLink!

AllStarLink is a network of Amateur Radio repeaters, remote base stations and hot spots accessible to each other via Voice over Internet Protocol. AllStarLink runs on a dedicated computer (including the Raspberry Pi) that you host at your home, radio site or computer center. It is based on the open source Asterisk PBX running our `app_rpt` application. `App_rpt` makes Asterisk a powerful system capable of controlling one or more radios. It provides linking of these radio "nodes" to other systems of similar construction anywhere in the world via VoIP.

AllStarLink's primary use is as a dedicated computer node wired to your repeater or radio. Connections from Echolink, other VoIP clients and telephone calls are supported. We also have a Java Applet which can be used on older computers to access AllStarLink nodes.

AllStarLink has **17,200** users and **12,830** nodes.

As you might imagine the implementation of this system and its monthly upkeep is very costly. Any monetary help that you can and wish to give will be much appreciated. A PayPal donation button is provided if you wish to do so. Thank you very much.

**Donate**

VISA M.C. AMEX

#### ASL News

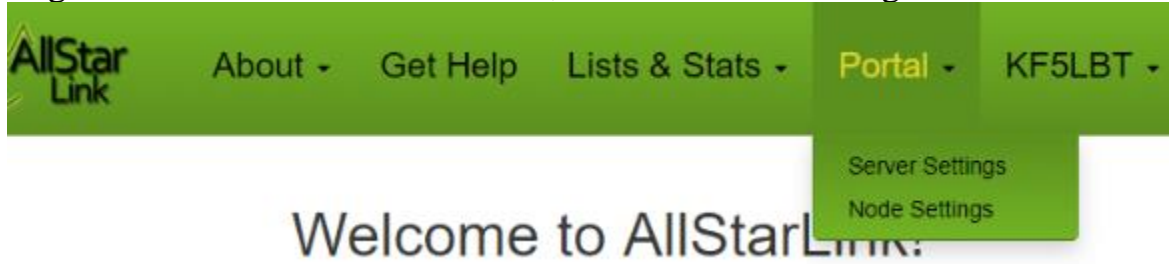
The AllStarLink team is announcing our new community online forum. Our new community is easy to use, yet it has many features. Please read [all about](#).

We have also announced a new plan for node numbers. It's sorta like SSID for AllStar. We call it Node Number Extensions, NNEX. You can follow [this link](#) to read all about it.

#### ASL Features

- Full Feature Repeater Controller
- VoIP Full Duplex Linking with great audio
- Frequency Agile Remote Base Station
- Simplex (half-duplex) Station
- Echolink
- APRS
- Autopatch
- Based on Asterisk - the Open Source PBX
- Real-time status reporting

Log in to the site and click on Portal, and then Node Settings.



Should then see information like what's below for your nodes

## KF5LBT Nodes

Press Continue to:

- Request a new node number from AllStarLink.
- Extend a node number up to 10 nodes.
- Delete a node number.

[Continue...](#)

Or click a node number below to edit it.

Node Number	Server	Callsign	Pa
40122	KF5LBT_40122	KF5LBT	
40174	KF5LBT_40174	KF5LBT	
45078	KF5LBT_45078	KF5LBT	
46738	KF5LBT_46738	KF5LBT	

Next to add an extension to a node click on the continue. Should see something like this:

## Request, Extend or Remove

**Extend** | Request | Remove

Extending a node allows you to assign yourself a Node Number Extension (an NNX). You do this yourself without the need to request more nodes from allstarlink.org. This saves you time and preserves valuable node number space.

The dropdown shows the node numbers that you've been issued and a count of extensions. It's possible to assign up to 9 extensions to any of your nodes.

If the dropdown is empty, you don't have any nodes yet. Please click request link (above) to request your first node number from AllStar.

Extend Node

Select a node and press Continue to begin.

Continue...

Next, clicking on continue, you will be asked to confirm changing the node number you had selected. So for this example 40122 would be changed to 401220. Note again by doing this to existing node, you will have to update all the associated image files on the pi to the new node number by running the `firsttime.sh` script.

### Multiple Nodes Behind One Firewall

If you are going to have multiple Nodes connected to you router, need to have unique bind ports for each one defined in each node `iax.conf` file, and also have the bind port information updated on the Allstar Link Server setting for your nodes. When you initially set up a node the bind port address is set to 4569 by default in the `iax.conf` file and the Allstar Link Server site settings. I have 5 nodes, so I set the bind ports like this:

Node 1 bind port 4569

Node 2 bind port 4570

Node 3 bind port 4571

Node 4 bind port 4572

Node 5 bind port 4573.

For each node I edited the iax.conf file to set the bind port. The iax.conf file is in the directory root/etc/asterisk/iax.conf. Around line 26 in the file is the setting for the bind post. Example:

```
; WARNING - THIS FILE WAS AUTOMATICALLY CONFIGURED FROM A
; TEMPLATE FILE IN /usr/local/etc/asterisk_tpl BY THE
; NODE-CONFIG.SH SCRIPT.
;
; EACH TIME THIS SCRIPT IS RUN, THIS FILE WILL BE OVERWRITTEN.
; IF YOU CHANGE ANYTHING IN THIS FILE AND RUN THE NODE-CONFIG.SH
; SCRIPT, IT WILL BE LOST.
;
; IF YOU INTEND TO USE THE NODE-CONFIG.SH SCRIPT, THEN YOU
SHOULD
; MAKE MODIFICATIONS TO THE ACTUAL TEMPLATE FILES LOCATED IN
; /usr/local/etc/asterisk_tpl directory.
;
; Inter-Asterisk eXchange driver definition
;
; For manual configuration edit the registration lines
; below and also passwords for various connections.

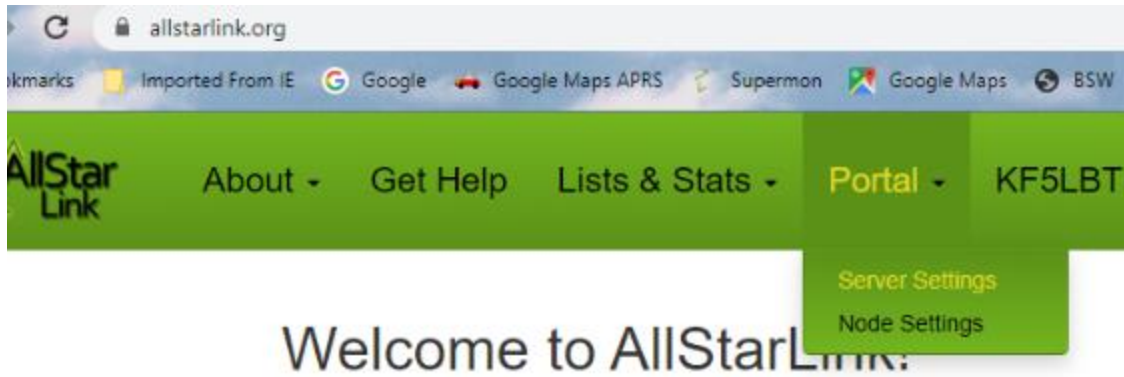
; Updated 1/2019 to include additional features
; Before attempting to add features ensure they
; are not already installed

; THIS IS AN IMPORTANT FILE! EDIT CAREFULLY!
; MAKE A BACKUP BEFORE EDITING

[general]
bindport=4570
bindaddr=0.0.0.0
disallow=all
```

Line highlighted above in the iax.conf file is the one to update for the bind port setting.

Next thing to do is go to the Allstar Link Site, Log into your account, and then select the Portal, Server Settings.



Clicking on Server Settings will bring up something like this for your servers, below is mine:

## KF5LBT Servers

Add a new server or edit one below.

Name	Location	Site
<a href="#">KF5LBT_40122</a>	Grapevine, Tx	QTH
<a href="#">KF5LBT_40174</a>	Grapevine, Tx	QTH
<a href="#">KF5LBT_450780</a>	Grapevine, Tx	QTH
<a href="#">KF5LBT_450781</a>	Grapevine, Tx	QTH
<a href="#">KF5LBT_46738</a>	Grapevine, Tx	QTH

Under the Name column, click on the node you want to update the iax port on. Will see the following information, (shown below is one of mine).

## Edit Server Settings

### Server Name

25 characters max

### Server Location

City/State/Province of server location. 30 characters max.

### Site

Site Name, Bldg Number, etc. 30 characters max. Optional.

### Affiliation

Affiliated with a club, etc. 30 characters max. Optional.

### Hostname

Unix Hostname. 25 characters max. Optional.

### IAX Port

Normally 4569 when only one server behind a NAT router.

### Proxy IP

Normally blank. Only needed in rare cases. Optional.

Click the map to select server latitude & longitude or type them in.

### Latitude

### Server Longitude



Submit

Edit the IAX Port number shown circled above. Then click the Submit button at the bottom. Again be sure each node connected to your router has a unique IAX Port number.

## Port Forwarding A Node

Can also do port forwarding on a node to allow incoming connections. This would allow nodes behind a firewall to interconnect without having to do changes outlined in the Multiple Nodes Behind One Firewall section describe above and would also allow outside nodes to connect as well. To accomplish this, the changes are made on the router that the node connect to not the node itself. The router I have allows port forwarding to be done for an individual connection on the router, and I set up to port forward the iax port number for that particular node. The default iax port number for a node is 4569 unless you changed it. You only have to port forward that port number for UDP protocol.

## CPU Stats

If you have Supermon for your node, click on the CPU Status.



Another window will pop up with the information, like this:

```
Command: /bin/date
-----
Mon Jun 17 09:30:55 CDT 2019

Command: export TERM=vt100 && sudo /usr/local/sbin/supermon/ssinfo -
-----

alarmpi78: RPi-3B+Plus
CPU(s):          4@1.40 GHz - ARMv7 Processor rev 4 (v71)
Memory:          439MB
Swap:            0MB
Storage Devices: mmcblk0p2[7.42 GB]
OS Version:      Arch Linux ARM (Installed on 5Mar2015)
Domain:          Standalone(FILES)
Host ID:         a8c00501
Location:        Grapevine, Tx
System:          Radioless Node
Service:         Allstar/IRLP/Echolink System Manager
Uptime:          1 days
Primary User:    reboot Wed4.14.97-1-ARCH
Flags:           DST-OK

cpuinfo_cur_freq: 1400 MHz
cpuinfo_max_freq: 1400 MHz
cpuinfo_min_freq: 600 MHz
cpuinfo_transition_latency: 355000
CPU ondemand up threshold: 95
CPU scaling governor: ondemand

Current CPU Temperature: 113 F, 45 C / GREEN ;)

Core volt=1.3500V
sdram core volt=1.2500V
sdram I/O volt=1.2500V
sdram PHY volt=1.2250V
BogoMIPS : 38.40 X 4 CORES
```

To get the Status from a terminal connected, select item 9 from the Admin Menu





in more heavily loaded nodes. If you experience these effects and it is not due to an Internet issue then it might be a good idea to try the turbo mode but if you do decide to use it there are some cautions you must follow. Reference the full article: . [/cpu\\_turbo\\_howto.pdf](#)

The turbo mode can be turned on or off using the `cpu_turbo.sh` script which will install automatically in version 1.5 code if you have updated.

To turn on Turbo Mode - `cpu_turbo.sh on`

To turn off Turbo Mode - `cpu_turbo.sh off`

Changes take effect on the next reboot

## **Supermon**

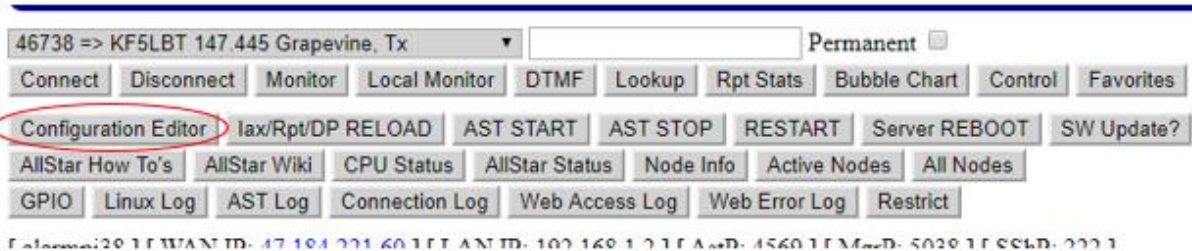
Below are notes collected on Supermon.

## **Configuration Editor - SimpleUSB-Tune-Control**

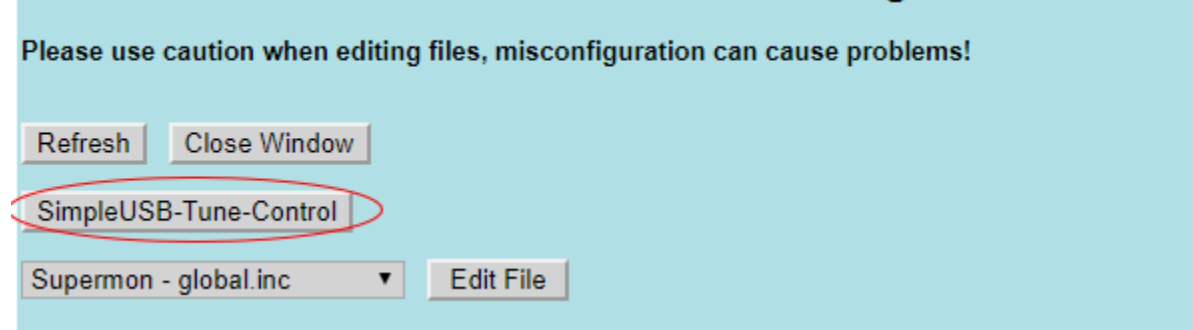
The Configuration Editor allows editing of certain files and adjustments in the SimpleUSB-Tune. Have added information here for setting the RX Level.

## RX Level Adjustment

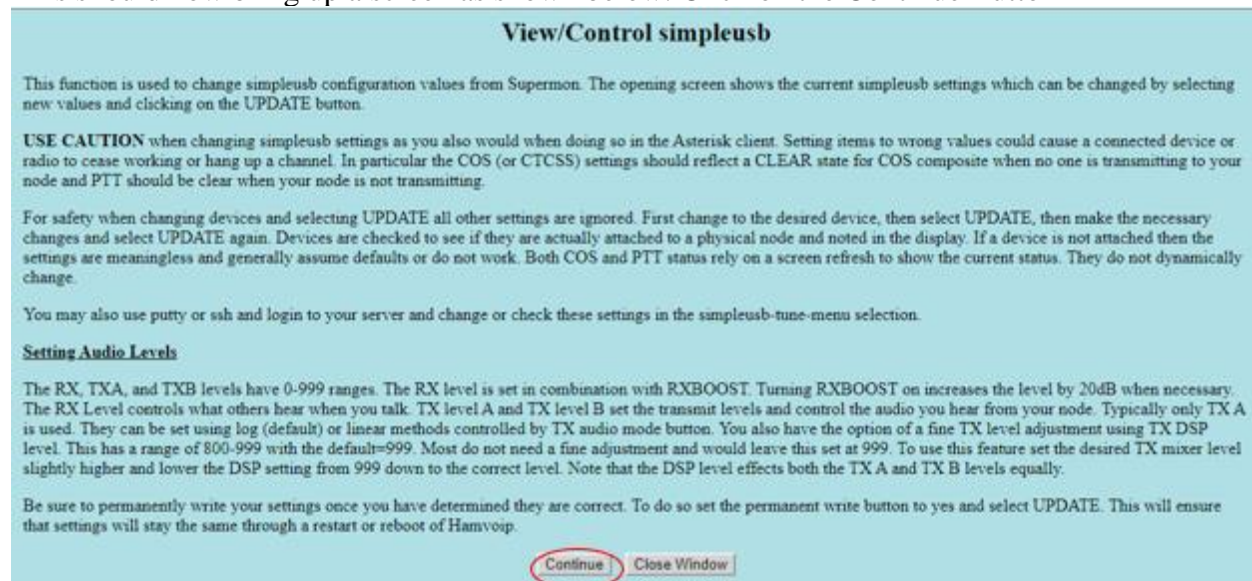
Outlined below is a method to adjust the RX level from Supermon. When doing the adjustment, it helps to be connected with someone so they can give you feedback as you adjust the level. On the Supermon page, you should see the controls below the top header as shown below. Click the Configuration Editor button. **NOTE:** There is a bug in image that if you make changes to other settings such as RX BOOST, PRE-EMPHASIS, etc, those changes will revert back to original settings when node is rebooted. Need to use procedure outlined in setting RX Levels using PuTTY in this document.



This should bring up a screen similar to the one below. Click on the SimpleUSB-Tune-Control



This should now bring up a screen as shown below. Click on the Continue Button



This will bring up the following Screen. You can now adjust the RX Level. On SHARI nodes, PRE-emphasis and DE-emphasis need to be ON and the RX Boost OFF. Enter in a new RX Level value, higher number to raise level, lower number to lower level. One you enter the number in the box, click on the Update button. Now you can talk to someone your connected to and repeat this process until your happy with the setting. Once done, select the YES at the bottom to select Permanently Write Settings, then click on the Update button on the top. Once done, click on the Close Window to exit out.

The screenshot shows a software interface titled "View/Control simpleusb". At the top, there are two buttons: "Update" and "Close Window", both circled in red. Below them, the text reads "Current Device is - usb" and "This device is Attached and Configurable".

The interface contains several settings with radio buttons:

- DE-EMPHASIS -  OFF  ON
- PL FILTER -  OFF  ON
- ECHO BACK -  OFF  ON
- RX BOOST -  OFF  ON
- PRE-EMPHASIS -  OFF  ON
- DCS FILTER -  OFF  ON
- KEY COS -  OFF  ON
- INVERT PTT -  OFF  ON
- COS -  None  USB  USB-invert
- CTCSS -  None  USB  USB-invert
- TX Audio Mode -  Log  Linear

At the bottom of the settings, there are three input fields:

- RX Level - 700 (circled in red)
- rxondelay - 0
- rxaudiodelay - 0
- TX Level A - 500
- TX Level B - 0
- TX DSP Level - 999

Below the input fields, there are two status indicators: "COS/CTCSS Composite Status - CLEAR" and "PTT Status - CLEAR". At the very bottom, there is a "Permanently Write Settings" label (circled in red) followed by radio buttons for "NO" and "YES" (the "YES" button is circled in red).

**Note: Transmit A and B Levels (From Doug Crompton)**

The A and B TX levels are the same audio just different outputs. The USB FOBS have stereo - 2 channel output. On the DMK URI this is pin 22 and 23.

We normally use 22 which is level A. I generally tell people without any measurement equipment to set the TX level so that the level you hear listening to your node is about the same level as listening to a local repeater on the same radio. This is only what you or someone listening to your node locally via RF hears.

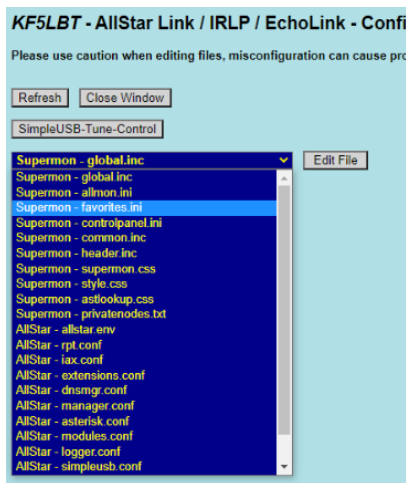
**Note: RXONDELAY and RXAUDIODELAY (From Doug Crompton)**

rxaudiodelay is primarily used for squelch tail elimination. It delays the audio in a buffer so that a drop of COS can cutoff a portion of the audio before the COS drops thus eliminating the tail of the transmission where a squelch crash would take place. Since all radios are different this value is adjustable. Valid values are from 0-24 in 20ms increments. Typical values in most cases would range from 5-10 or 100-200 ms. It is obviously not mandatory to use this, some people actually like squelch tails. The default is 0. If you have a way to monitor your node through another node or maybe iaxrpt you can listen when you drop TX on the node you want to adjust and move this value up until the tail is gone. Unless the tail is unusually long you should be able to eliminate it.

rxondelay delays the valid COS signal and is used to stop cyclic keying or ping-pong which can happen in rare cases. If you bring up simpleusb-tune-menu and select the "V" view mode you can see both the COS input and COS composite displays. To understand how rxondelay works set it to some high value like 20 or 400ms. When you transmit to your node you will see the COS input immediately go keyed and the 400ms delay of the composite. Setting this value too high would cut off the beginning of a transmission as it delays the actual COS beyond the real COS signal. The default is 0 and this would be appropriate in most all cases.

**Configuration Editor – Supermon-favorites.ini file**

The configuration editor can also be used to edit the favorites.ini file. Use this file to store your favorite nodes you connect to. Once you have edited this file with the nodes you want, then clicking on the Favorites button will bring up the list, and you can simply select the node you want and connect to it. To edit the favorites.ini file click on the Configuration Editor, then click on the drop down box showing Supermon-global.inc. A drop down menu of files will be displayed, click on the favorites.ini, then click on the Edit File button.

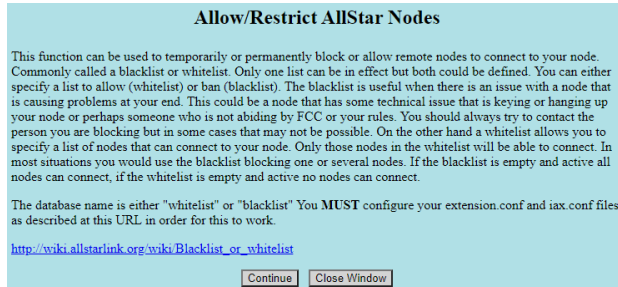


A window with the file will come up. The entries have pairs of 2 lines. A label line and a cmd line. The label line will be what shows up when you click on the Favorites Button. So just edit what is between the quotation marks on that line with whatever you want for the node description. Then the next line is the cmd line. This is what will be executed. Update the information between the quotes with the node number that you want to connect to.

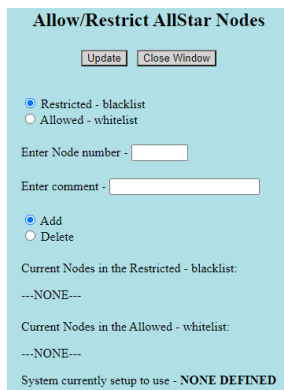
**Be sure to remove the semicolon at the beginning of the label and cmd lines.** This will make the label line show up when you click on the favorites button, and also make the cmd line executable.

## Using a Black list or White List

Supermon now provides a means to set up either a black or white list. The black list is used to block specified nodes, the white list specifies the nodes that are allowed to connect. Only one list can be active. To set up either list, click on the Restrict Button. A screen will pop up with a description of the white & black list function, click on the continue button.



The following window should come up



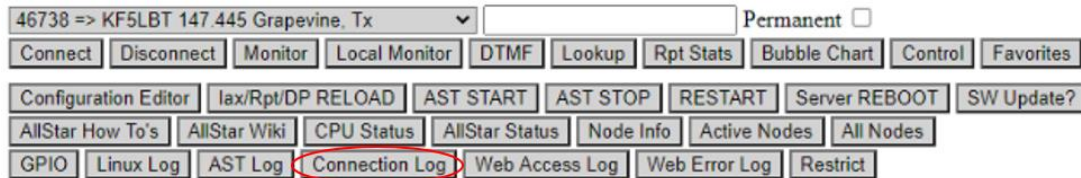
Make the selections you want for either Restricted or Allowed, the node number, and any comments you wish to have for that node. Then click on the Update button. Once you update it will show the nodes you have entered along with the comment for each node. Again you can set up nodes for both a black or white list, but only one list can be active at a time. LASTLY need to enable these functions in the root/etc/asterisk iax.conf file. Need to edit this file, and down around line 169 in the file should see the following:

```
context = radio-secure
;context = whitelist
;context = blacklist
;
```

Note the semicolon at the beginning of the lines for whitelist and blacklist. You must remove the semicolon's to enable the blacklist and whitelist to function.

## Connection Log

On the supermon page, you will see a button for Connection Log:



This log will show connections made over time. You must make a change in the rpt.conf file for this to work. There are two lines in the rpt.conf file that are commented so you need to remove the ; at the beginning of these lines to enable the logging.

Here is shot of my rpt.conf: Lines highlighted....

```
355 ; MISC. SETTINGS
356
357 parrot=0 ; One time parrot modr with DTMF A9
358
359 ;connpgm=yourconnectprogram
360 ; Program or script to run at connection
361
362 connpgm=/usr/local/sbin/supermon/smllogger 1 ; use this line for Supermon
363
364 ;discpgm=yourdisconnectprogram
365 ; Program or script to run at disconnection
366
367 discpgm=/usr/local/sbin/supermon/smllogger 0 ; use this line for Supermon
368
369 nounkeyct=0
```

Note, need to reboot node after making changes to the rpt.conf file.

## Load Average

On the Supermon, there is a line as follows:

[ Sunday, August 19, 2018 CDT 05:15:21 up 34 min, 0 users, load average: 1.06, 1.10, 1.02 ]

Load average doesn't necessarily indicate CPU usage, but rather how many processes are waiting to run. Link for more information: [Load Average](#)

## CPU Temperature

The CPU Temperature displayed on Supermon is highlighted in either Green, Yellow or Red. Temperature ranges are:

Green: Temperatures that are below 50° C (122° F)

Yellow: Temperatures between 50° C to 59° C (122° F to 138° F)

Red: Temperatures above 59 ° C (138° F)

## Core Dump Message

If you see Core Dump displayed on your Supermon, this can be caused by various reasons. The core dump file is located in the directory:

/var/lib/systemd/coredump

Deleting the core dump file will clear the error message on the Supermon. Might want to save the file someplace before deleting it in case there is a real issue you will still have the information.

## Procedure to Change Refresh Time on Supermon

The refresh time of the screen is now settable. This is necessary to update the CPU, weather, and AutoSky displays. The normal node updates which are immediate are not effected by this. Recommended refresh is 600 or 10 minutes. Add this to global.inc

```
$REFRESH_DELAY = "600";
```

## Procedure to Display Local Temperature and Weather on Supermon

You can now display the local temperature and weather condition using your zip code. If `$LOCALZIP = "zip";` is defined in global.inc the weather will appear below the display configuration and CPU temperature line. Add -

```
$LOCALZIP = "your-zip";
```

to your /srv/http/supermon/global.inc file. The line will not display if you omit or comment this variable. The line will look like this. Hopefully your temperature will be a little warmer!

```
[ Weather conditions for Richboro, PA 18954: 31°F, -1°C / Clear ]
```

For AutoSky users alerts now display in text in Supermon. This is displayed in red after the display configuration line or weather line if activated. You will see or hopefully won't see things like this - `*[ Tornado Warning ]*`



## Procedure to Add Internet Links to Supermon

Links to internet web pages can be added to the Supermon Screen on the line that appears below the top title blocks. Links are added in the almon.ini file, located in the supermon directory. Below is an example screen shot of a file with links added:

```
1  [46738]
2  host=127.0.0.1:5038
3  user=admin
4  passwd=
5  menu=yes
6  hideNodeURL=no
7  [1999]
8  host=127.0.0.1:5038
9  user=admin
10 passwd=wa3dsp
11 menu=yes
12 hideNodeURL=no
13 [All Nodes]
14 nodes=46738,1999
15 menu=yes
16 [lsNodes]
17 url="/cgi-bin/lsnodes_web?node=46738"
18 menu=yes
19 [IRLP Status]
20 url="http://status.irlp.net"
21 menu=yes
22 [HAMVOIP]
23 url="http://HamVOIP.org"
24 menu=yes
25 [VHF Prop]
26 url="http://aprs.mennolink.org/"
27 menu=yes
28 [Wind Speed]
29 url="https://wind.willyweather.com/tx/tarrant-county/grapevine.html"
30 menu=yes
```

## Display Multiple Nodes on Supermon

I have 5 nodes connected to my home router. I used Supermon to manage the nodes, using a separate tab on my web browser for each one. I found a method which allows control of all the nodes through one Supermon session. One can easily switch the information on Supermon to display another node connections, or one can also display the connection information for all nodes in Supermon. To implement this, the allmon.ini on the node you want to use Supermon on and the manager.conf files for the nodes that you want to control need to be edited. I always recommend making a backup copy of the files before you edit so just in case something goes wrong you can get back to your original set up.

The allmon.ini file is in the root/svr/http/supermon directory. The manager.conf file is in the root/etc/asterisk directory.

The allmon.ini file changes are shown below, note this is done only on the allmon.ini file for the node you will be running Supermon for. ( In this example that is node 46738) Added comments off to the side in red text.

```
[46738]
host=127.0.0.1:5038
user=admin
passwd=[REDACTED]
menu=yes
hideNodeURL=no
[450780]
host=192.168.1.5:5038
user=admin
passwd=[REDACTED]
menu=yes
hideNodeURL=no
lsnodes=http://192.168.1.5/cgi-bin/ltnodes_web?node=450780
[450781]
host=192.168.1.12:5038
user=admin
passwd=[REDACTED]
menu=yes
hideNodeURL=no
lsnodes=http://192.168.1.12/cgi-bin/ltnodes_web?node=450781
[40174]
host=192.168.1.9:5038
user=admin
passwd=[REDACTED]
menu=yes
hideNodeURL=no
lsnodes=http://192.168.1.9/cgi-bin/ltnodes_web?node=40174
[40122]
host=192.168.1.13:5038
user=admin
passwd=[REDACTED]
menu=yes
hideNodeURL=no
lsnodes=http://192.168.1.13/cgi-bin/ltnodes_web?node=40122
[All Nodes]
nodes=46738,450780,450781,40174,40122
menu=yes
```

Note, this should already be in the file, this is the node that will be used to run the Supermon session. Also note the host setting it needs to be set to 127.0.0.1:5038

Note: this should have originally been node 1999, you change this to the node number you want to add. Note here, the host setting set to the ip address for the node. Copy and paste this information for the other nodes you want to add. Be sure you update the node numbers and ip addresses after you paste.

Add to see ltnodes on Supermon

Be sure to have all the node numbers listed here

The manager.config file changes are shown below, note this is done only on the manager.config files for the nodes you want to add to Supermon ( In this example that was done on nodes 40122, 450780, 450781 and 40174). Added comments off to the side in red text. The lines edited are about 37 lines down into the file.

```
[general]
displayssystemname = yes
enabled = yes
;webenabled = yes
port = 5038
```

```
;httptimeout = 60
; a) httptimeout sets the Max-Age of the http cookie
; b) httptimeout is the amount of time the webserver waits
;   on a action=waitevent request (actually its httptimeout-10)
; c) httptimeout is also the amount of time the webserver keeps
;   a http session alive after completing a successful action
```

```
;bindaddr = 127.0.0.1 ; Local interface only!      This line needs to have the ; placed at the start of
                                                    the line to make it just a comment line
```

```
bindaddr = 0.0.0.0 ; Not secure                    This line will originally have a ; at the beginning of the
                                                    line, need to delete the ;
```

Once the above changes have been done, reboot the nodes.

After reboot, bring up the Superman for the node that allmon.ini was updated, in this example that was 46738. The following screen shot shows example of what you should see. Note the nodes I circled in red, these are the nodes that can now be controlled via the Superman for 46738.



Note the above Superman is now showing the connections for the 46738. Clicking on one of the nodes added, (shown in the red circle), will change the Superman to display the connections for that node.

Now for the interesting ability, by clicking on All Nodes, a new tab will open on your web browser, and show the connections for all the nodes.



See following screen shot...

**KF5LBT - Supermon 6.1+ AllStar Manager**  
 Logout  
 Grapevine, Tx  
 Simplex Node  
 Allstar/IRLP/Echolink System Manager

147.445 PL 74.4

46738 450780 450781 40174 40122 All Nodes W5MGM KD5HQF KF5LBT\_40122 KF5LBT\_4

46738 => KF5LBT 147.445 Grapevine, Tx  Permanent

Connect Disconnect Monitor Local Monitor DTMF Lookup Rpt Stats Bubble Chart Control Favorites

Configuration Editor laxiRpt/DP RELOAD AST START AST STOP RESTART Server REBOOT SW Update?

AllStar How To's AllStar Wiki CPU Status AllStar Status Node Info Active Nodes All Nodes

GPIO Linux Log AST Log Connection Log Web Access Log Web Error Log Restrict

[ alarmpi38 ] [ WAN IP: 47.184.221.60 ] [ LAN IP: 192.168.1.2 ] [ AstP: 4569 ] [ MgrP: 5038 ] [ SShP: 222 ]  
 [ Tuesday, November 26, 2019 CST 09:56:03 up 5 days, 21:07, 0 users, load average: 0.13, 0.12, 0.09 ]

Display Configuration [ CPU: 96°F, 36°C @ 09:56 ]

[ Weather conditions for Grapevine, Tx 76051: 67°F, 19°C / Showers ]

**Node 46738 => KF5LBT 147.445 Grapevine, Tx** [Bubble Chart](#) [IsNodes](#)

Node	Node Information	Received	Link	Direction	Connected	Mode
46738	Idle					
45523	W5MGM ALLSTAR ECHOLINK HUB Addison, TX	000:01:06	ESTABLISHED	OUT	53:08:10	Transceive

**Node 450780 => KF5LBT Radioless Node Grapevine, Tx** [Bubble Chart](#) [IsNodes](#)

Node	Node Information	Received	Link	Direction	Connected	Mode
450780	Idle					
40122	KF5LBT NTx 220 HUB Grapevine, Tx	001:04:57	ESTABLISHED	OUT	101:53:45	Transceive

**Node 450781 => KF5LBT Radioless Node Grapevine, Tx** [Bubble Chart](#) [IsNodes](#)

Node	Node Information	Received	Link	Direction	Connected	Mode
450781	Idle					
40122	KF5LBT NTx 220 HUB Grapevine, Tx	001:04:57	ESTABLISHED	OUT	101:53:09	Transceive

**Node 40174 => KF5LBT 223.470 Grapevine, Tx** [Bubble Chart](#) [IsNodes](#)

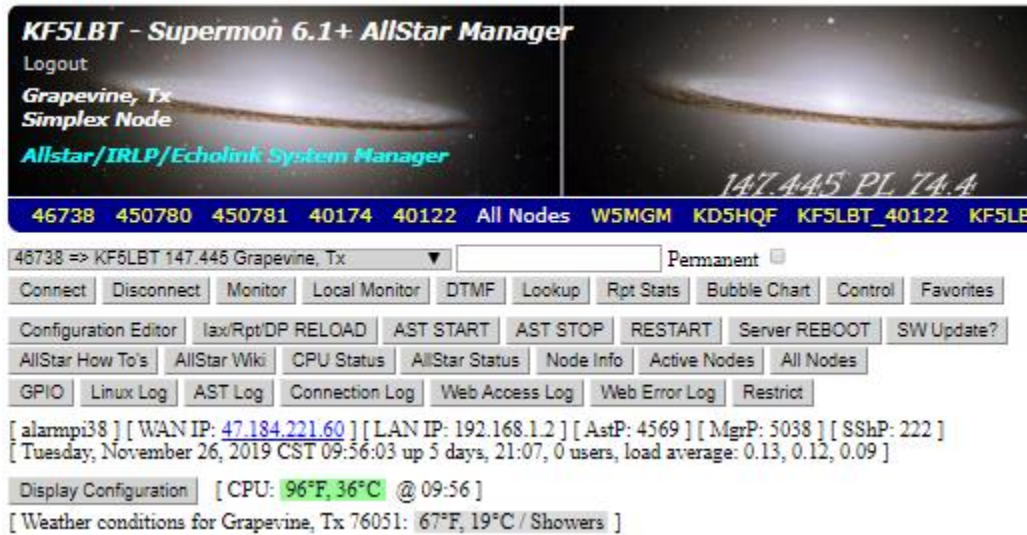
Node	Node Information	Received	Link	Direction	Connected	Mode
40174	Idle					
40122	KF5LBT NTx 220 HUB Grapevine, Tx	001:04:57	ESTABLISHED	OUT	101:54:25	Transceive

**Node 40122 => KF5LBT NTx 220 HUB Grapevine, Tx** [Bubble Chart](#) [IsNodes](#)

Node	Node Information	Received	Link	Direction	Connected	Mode
40122	Idle					
46273	KD5HQF 220 224.160- Howe, TX	001:04:57	ESTABLISHED	OUT	101:34:07	Transceive
450780	KF5LBT Radioless Node Grapevine, Tx	045:07:43	ESTABLISHED	IN	101:53:44	Transceive
40174	KF5LBT 223.470 Grapevine, Tx	Never	ESTABLISHED	IN	101:54:24	Transceive
450781	KF5LBT Radioless Node Grapevine, Tx	Never	ESTABLISHED	IN	101:53:08	Transceive

System maintained by: Jim, KF5LBT

Notice now that a drop-down menu is available to select what node you want to control connections, etc



Click on the little down arrow on the right of the box, and you will see the nodes you added. Selecting another node changes the Supermon controls to control that node.



One final note. The following screen shot shows two areas of buttons on the Supermon. The ones in Red Box will control the node that you have selected, however the buttons in blue box seem to only apply to the node which has the updated allmon.ini file no matter which node you have selected.



So that's it. If you have multiple nodes on your QTH router, you might find this useful. Have fun playing around !!!

## Allmon2

If you want to use Allmon2 for what ever reason, need to copy the password file created for Supermon over to the Allmon2 directory. From the allmon2 directory use the following command to copy the file:

```
htpasswd -c .htpasswd username
```

the username will be what want to login with.

## Set Pi Display Monitor Sleep Time

Background information. Last hamvoip update of 5/24/2018, which was a major system update, the monitor connected to the Pi no longer went into sleep mode after a few minutes. Talked with Doug who gave a suggestion, which by itself did not fix, then got another suggestion from Charles, and added that as well, and monitor now goes into sleep mode at time specified in command. Below are the two suggestions implemented.

Doug's change:

Edit the /boot/config.txt, added the line `hdmi_blanking=1` to end of file.

Charles change.

First thing done was to check if console blanking was on:

You can check console blanking with the command 'cat

/sys/module/kernel/parameters/consoleblank' to tell you whether blanking is on or off. It will return '0' if off. To enable in /boot/cmdline, the command is 'consoleblank=xxx' where xxx is

the number of seconds delay before the screen goes off. This is a kernel parameter.  
/boot/cmdline must remain a single line.

Executed the command and it returned 0, so edited the file /boot/cmdline.txt file.  
The file had the following line: (note in file this is one continuous line)

```
root=/dev/mmcblk0p2 rw rootwait console=ttyAMA0,115200 console=tty1 selinux=0  
plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 kgdboc=ttyAMA0,115200  
elevator=noop
```

Edited the line and added consoleblank=600 to the end to set the display to go into sleep mode  
after 10 minutes.

```
root=/dev/mmcblk0p2 rw rootwait console=ttyAMA0,115200 console=tty1 selinux=0  
plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 kgdboc=ttyAMA0,115200  
elevator=noop consoleblank=600
```

## Local Node Announcements

By default, a node is set up to say certain announcements local to the node, (Only you hear them). Following information on changing and controlling the announcements. Information for this came from arm-allstar site, by Doug Crompton

## Telemetry Audio Levels

The audio level for the local announcements you here on the node, such as time, connections etc. can be adjusted in the rpt.conf file. Below are the lines in the rpt.conf file to look at, they are about 194 lines down into the file. The default setting for the telemnomdb is 0, Changing this setting to -9 (Note the minus sign before the 9), will lower the audio level for the local announcements. You can play around with this setting to get it where you want the level to be.

; AUDIO LEVEL SETTINGS IN dB

telemnomdb=-9

; Overall reduction in telemetry level

telemduckdb=-9

; Telemetry level reduction with signal



## Add Weather Conditions to the Hourly Time Local Announcement

By default, the node announces the time on the hour. This is done through what is called a cron job. From the Admin screen (see below) select item 9 to bring up the terminal window.

```
lqqqqqqqqqqqqqqAdmin Menu List for: alarmpi38 (192.168.1.3)qqqqqqqqqqqqqqqqk
x Please select:
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x x      1 Perform a system UPDATE (Internet access required) x x
x x      2 Change the ROOT password x x
x x      3 Change the primary NODE number x x
x x      4 Change the system Timezone x x
x x      5 Change the system Hostname x x
x x      6 Configure the Wired Ethernet Networking x x
x x      7 Configure the WiFi Interface Networking x x
x x      8 Change the Secure Shell (SSH) port x x
x x      9 Start Bash shell interface x x
x x     10 Display System Version Numbers x x
x x     11 Run Asterisk CLI client x x
x x     12 Run simpleusb-tune-menu Application x x
x x     13 Restart Asterisk Server x x
x x     14 Power-cycle the USB sub-system x x
x x     15 Reboot this system x x
x x     16 Perform system power down x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq]
x qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq]
x <Run Selected Item> < Exit / Logout > x
x qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq]
```

At the prompt, be sure your in the root directory, and type the command:

```
crontab -e
```

Should see something like what's shown below:

```
# Do not remove the following line
# required for lsnodes and allmon
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/bin/nice -19 /usr/bin/perl
/usr/local/sbin/saytime.pl $NODE1 > /dev/null)
```

To have the weather conditions also announced with the time change the last line to include your zip code. See below where you would put your zip code in place of where YOUR ZIP CODE is.

```
# Do not remove the following line
# required for lsnodes and allmon
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/bin/nice -19 /usr/bin/perl
/usr/local/sbin/saytime.pl YOUR ZIP CODE $NODE1 > /dev/null)
```

For example mine looks like this:

```
# Do not remove the following line
# required for lsnodes and allmon
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
```

```
00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/bin/nice -19 /usr/bin/perl
/usr/local/sbin/saytime.pl 76051 $NODE1 > /dev/null)
```

## **Disable Time and Id**

Method to disable the time announcement

Start bash shell interface, and type command:

```
crontab -e
```

```
# Do not remove the following line
# required for lsnodes and allmon
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
#00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/bin/nice -19
/usr/bin/p.....
```

Add a # before the 00 0-23 line as shown above and save.

## **Unkey Courtesy Tone Enable/Disable**

By default, when you key up and transmit, and then unkey, the node will transmit a local tone to indicate unkeyed. This can be disabled in the rpt.conf file. Edit the line, (around line 369 in the file) in the rpt.conf file as shown below, setting the value = 1.

Default setting:

```
nounkeyct=0
```

```
; Disable courtest tones (=1)
```

Change this to 1 as shown below if you don't want to hear the unkey courtesy tone.

```
nounkeyct=1
```

## Node Auto Connect at Startup

To have your node automatically connect up to another node at start up can be useful. There is more information on start up macro's on the arm allstar site. Here is the link: [Auto Connect at Start Up](#). Below is information specific to setting up your node to auto-connect to another at start up. The changes are made in the rpt.conf file. The lines in the rpt.conf file that need to be changed are shown in **Red Text**.

```
;archiveaudio=0
```

```
; Disable saving audio files when  
; archiving. Use with caution on SDcards  
; This write a lot of data.
```

```
;archivetype=gsm
```

```
; Allows the selection of gsm (.wav) or  
; pcm (.ul) for archiving audio files
```

```
startup_macro = *57
```

```
Note, what changed here, is the ; at the beginning of line  
has been deleted. This line is about 433 lines down into  
the file.
```

```
; Macro to run at startup (optional)
```

```
Startup_macro_delay=30
```

```
Note change here, again ; at beginning of line deleted,  
And the delay time changed to 30 Note if your rpt.conf  
File does not have this line, then add it.
```

```
; Time in seconds to wait before executing  
; the startup macro
```

```
7=*73xxxxx
```

```
Need to add this line in the node macro stanza below the  
Following lines: This line is about 1007 lines down into  
file.
```

```
[macro450781]
```

```
1=*81 *80# ; play time and voice ID
```

```
7=*73xxxxx add this line. xxxxx is the node to connect to.
```

```
; TAIL MESSAGE
```

```
; The tailmessagetime,tailsquashedtime, and tailmessages need to be set  
; to support tail messages. They can be omitted otherwise.  
;
```

Last thing to change is down at the bottom of the file, around line 1006.

```
[macro40122] Note this should have your node number already in it.
```

```
;Macro number = command string (each command separated by space) -end with  
HASH
```

```
1=*81 *80# ; play time and voice ID
```

```
7=*7345523 Be sure this line is there without a ; at beginning.
```

; TELEMETRY COURTESY TONE DEFINITIONS

### **ASL database - astdb**

In the hamvoip code astdb is run from cron -

```
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
```

This runs astdb.php in the /usr/local/sbin directory once a day at 3 AM plus a 30 minute random offset. The random offset is so everyone does not hit at the same time.

The php script reads the database, appends the local private entries from

/etc/asterisk/local/privatenodes.txt, sorts the file and writes the contents out.

The output is stored in /var/log/asterisk/astdb.txt - ALL code wishing access to the database should either link or directly reference this file.

Note that changes in the database would only get updated this one time a 3AM + ~30 minutes daily. If you want to see the update immediately after a local change run the file manually.

To run the file manually do the following: (Not sure, but might need to reboot afterwards to take effect ?)

Go into directory `cd /usr/local/sbin`

Then enter `./astdb.php`

### **AutoSky Skywarn**

AutoSky is a Skywarn program designed to work with Allstar. It receives NOAA warnings for your county and announces them on your local node. It is easy to setup and has been adapted to work with hamvoip V1.5 software. AutoSky voices out simple voice alerts like "Tornado Watch" or "Flood Warning" at the drop of transmissions using the Allstar tailmessage command. It covers all the alert messages that would normally be encountered in a Skywarn system. AutoSky is an additional package in the hamvoip repository. Link for the "how to" for setting this up below.

**Note:** in the instructions, for the section shown below:

Add a start command

If you want AutoSky to run automatically at each boot freeing you of the manual starting steps add the following two lines to the end of the /etc/rc.local file.

```
/usr/local/bin/AUTOSKY/AutoSky
```

```
/usr/local/bin/AUTOSKY/AutoSky.ON
```

BE SURE to add the lines **above the exit 0** line in the file.

<https://hamvoip.org/AutoSky%20howto.pdf>

## Echolink Setup On Allstar Node

(Want to thank Fred, W5MGM for his help on putting this together)

First thing assumed here is that you currently have an Echolink sysop node number and password, ie your callsign with a -R. For example mine is KF5LBT-R.

Here is a link to the Hamvoip site how to document on setting up Echolink on a RPi.

[https://www.hamvoip.org/echolink\\_howto.pdf](https://www.hamvoip.org/echolink_howto.pdf)

There is one thing that's not stated in the document that one needs to do in order to get this to work.

Echolink requires that ports 5198 through 5200 are forwarded on your router. Set 5198, 5199 UDP, and 5200 TCP. I have put a screen shot below of my router settings, I realize this might not match what you will see on your router, but hopefully gives you an idea of what needs to be set up.

Applied rules:

Networked Computer / Device	Applications & Ports Forwarded	WAN Connection Type	Status	Delete
localhost 127.0.0.1	Frontier Service Tcp Any -> 4567	All Broadband Devices	Active	
192.168.1.101:63145	Application UDP Any -> 35003	All Broadband Devices	Active	
192.168.1.101:8082	Application TCP Any -> 8085	All Broadband Devices	Active	
192.168.1.100:8082	Application TCP Any -> 8086	All Broadband Devices	Active	
192.168.1.102:8082	Application TCP Any -> 8087	All Broadband Devices	Active	
192.168.1.100:63145	Application UDP Any -> 35002	All Broadband Devices	Active	
192.168.1.102:63145	Application UDP Any -> 35001	All Broadband Devices	Active	
<input type="checkbox"/> alarmpi192.168.1.2	Destination Ports 4569 UDP Any -> 4569	All Broadband Devices	Disabled	<input type="checkbox"/>
<input type="checkbox"/> alarmpi3192.168.1.6	Destination Ports 4571 TCP Any -> 4571	All Broadband Devices	Disabled	<input type="checkbox"/>
<input checked="" type="checkbox"/> alarmpi3192.168.1.6	Destination Ports 5198-5199 UDP Any -> 5198-5199	All Broadband Devices	Active	<input type="checkbox"/>
<input checked="" type="checkbox"/> alarmpi3192.168.1.6	Destination Ports 5200 TCP Any -> 5200	All Broadband Devices	Active	<input type="checkbox"/>

The how to outlines what to edit in the stock file named echolink.xxx located in the /etc/asterisk directory. Once you make the changes, you **save the changes to a new file name of echolink.conf**.

Below is what the echolink.xxx file contains. Note, have changed text font to red on the lines that need to be updated. Again once you made changes, **save file as echolink.conf**, then restart asterisks.

```
[e!0]
confmode=no ; no = only 1 connection, yes allows multiple connections.
call=your-call ; Change this to Echolink Call, mine for example is
                    KF5LBT-R
pwd=your-password ; Change this to the password for the Echolink node
name=your-name ; Change this to your real name!
qth=your-qth ; Change this to your actual QTH!
email=someone@somewhere.com ; Change this!
maxstns=20
rtcptimeout=10
node=your-Echolink-node-number ; Echolink node number - Change this!
recfile=/tmp/recorded.gsm
astnode=your-allstar-node-number ; Change this to your active Allstar node number!
context=radio-secure
server1=server1.echolink.org
server2=server5.echolink.org
server3=server3.echolink.org
;naeast.echolink.org
;nasouth.echolink.org
;servers.echolink.org
;backup.echolink.org
; Change following setting to your parameters
freq=446.1
tone=88.5
lat=40.223976
lon=-75.008529
power=20
height=50
gain=6
dir=0
;
; Message about your Echolink node. Up to 256 characters All in
; one line. Use \n for newline. Uncomment to use.
;
;message=This is my message\nand another line\n
;
; Deny-permit - Only use one, either whitelist using permit or
; blacklist using deny Echolink stations. Wildcards can be used
; Example - permit WA3DSP*,W3XYZ* would allow these calls
; by themselves or with -L or -R suffix. Uncomment to use.
;
;deny=
;permit=
```

## APRS Setup

Here is the link to a site I found that outlines how to setup the AllStar Raspberry to report position to Google APRS.

Link: <http://n6jer.blogspot.com/2017/07/updating-aprs-with-allstar-node.html>

Note the document instructions say to edit a `gps.conf` file located in the directory `/etc/asterisk`. My image load on the Raspberry did not have the file, so I created it, and all worked fine. Note be sure to do the edit of the `modules.conf` file outlined below.

Below is my `gps.conf` file. Did find out that If you have the opportunity to transmit your APRS packet out to RF either directly or indirectly, should update your comment to follow the APRS Spec for QSY format. Link to APR Spec: [APRS SPEC Addendum 1.2 Proposals](#).

```
comment = 147.445 C74.4 AllStar Node 46738  
If using CTCSS encode and decode
```

or

```
comment = 147.445 T74.4 AllStar Node 46738  
If you are using PL decode only
```

Passers by with mobile radios equipped to decode this can then push a TUNE button and QSY to your repeater instantly. If you are only pushing to APRSIS then it doesn't matter because mobile radios out in the wild will never hear it.

```
; Configuration for app_gps  
;  
[general]
```

```
call = KF5LBT-10  
password = 20412 ; Password for the APRS-IS server  
comment = 147.445 C74.4 Simplex AllStar Node, Node No. 46738  
server = rotate.aprs2.net  
port = 14580  
interval = 600 ; Beacon interval in seconds  
icon = r  
; comport = /dev/tty50  
; baudrate = 4800  
; debug = y  
freq = 147.465  
tone = 0.0  
lat = 32.91425  
lon = -97.11686  
elev = 4
```

```
power = 0
height = 1
gain = 3
dir = 0
```

In the same /etc/asterisk directory, type nano **modules.conf**

Scroll down 59 lines until you find:  
noload => app\_gps.so ; GPS interface module

Then simply delete the letters "no" so it looks like:  
load => app\_gps.so ; GPS interface module

Ctrl-x to save and exit.

## **IP Address**

To see the peer ip address information of nodes connected to you, type in the following command from the Asterisk CLI Client where Node# would be your node number.

```
CLI> rpt lstats Node#
```

If any server is online you can find its IP address and port from any other server by typing dns-query node

```
dns-query 27225
+OKradio@67.165.29.63:4569/27225,67.165.29.63
```

## **Hardware Identification**

At the Linux Prompt -

```
cat /proc/cpuinfo
```

At the bottom of the display there is a revision number.  
Go here and you can see the board type and who built it.

[https://elinux.org/RPi\\_HardwareHistory](https://elinux.org/RPi_HardwareHistory)

**Update 11/26/2018**, Can now just type the command:

```
whatsmyhardware.sh
```



## Node Registration Status

At the Bash Shell Prompt, execute the command: `check_reg.sh`  
It will return the registration status of the node.

## Halt Button (Hardware Means to Safely Shutdown)

It's always best to do a controlled shutdown of the Raspberry Pi. Just removing power can lead to having the SD Card image become corrupted.

For those with mobile nodes, transferring the node from one location can require removing power from the node, and if one can't ssh into the Pi to do a controlled shutdown, then you run the risk. Solution for this, add a Halt Button! By adding a momentary switch to the hardware, one can press and hold down a momentary switch, the Pi detects the switch and announces a goodbye, so you know it's safe to remove power. The link below has the details for adding the switch, and the required simple change to a file on the Pi image to enable the shutdown.

**The file is the `allstar.env` file. Set `SHUTDOWN_MONITOR="enabled"` in `/usr/local/etc/allstar.env` file**

Note the information states adding a halt button to a Raspberry Pi2, but it works on others as well. Remember after adding the hardware to edit the `/usr/local/etc/allstar.env` file and change `SHUTDOWN_MONITOR= "enabled"`

Link: [Halt Button](#)

## ArchLinux

### pacman commands

Below is a link to ArchLinux with more information:

[https://wiki.archlinux.org/index.php/pacman#Installing\\_packages](https://wiki.archlinux.org/index.php/pacman#Installing_packages)

## DTMF Tones

### DTMF Function List

*1<node> - Disconnect from a link	*77 - Reconnect previously disconnected links
*10 - Disconnect from the last link you connected to	*80 – System info
*2<node> - Connect to a link (monitor only)	*81 – Time (12 hr. format)
*3 - Connect to a link	*82 – Time (24 hr. format)
*4 – Remote Command	*83 – Full link status
*51 - Play macro (Time and ID)	*980 – Allstar version
*70 - System status	*99 Phone PTT
*71<node> - Disconnect permanently connected link	*A1 - Say local IP address
*72<node> - Permanent connect to a link (monitor only)	*A3 - Say public IP
*73 - Permanent connection to a link	*A5 - Say registration status
*74 Reconnect Links	*A9 - One time parrot (Say your message and release PTT)
*75 - Full system status	*B1 - Halt your Pi (Must cycle power to restart)
*76 - Disconnect all links	*B3 - Reboot your Pi
	*B6 - Restart Asterisk

NOTE: Some of these commands such as the B1, B3, B6 are commented in the rpt.conf file. To activate these commands uncomment the lines in the rpt.conf file.



```

== Manager 'admin' logged off from 192.168.1.2
== Manager 'admin' logged on from 192.168.1.2
[Oct 31 10:29:56] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char *
[Oct 31 10:29:57] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 7
[Oct 31 10:29:59] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 1
[Oct 31 10:30:00] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 4
[Oct 31 10:30:02] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 0
[Oct 31 10:30:03] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 1
[Oct 31 10:30:04] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 2
[Oct 31 10:30:05] NOTICE[368]: chan_simpleusb.c:2719 simpleusb_read: Got DTMF char 2
-- Hungup 'IAX2/172.245.118.100:4570-1572'
-- Hungup 'DAHDI/pseudo-983737040'
-- Hungup 'DAHDI/pseudo-251243618'
-- <DAHDI/pseudo-1453405903> Playing 'rpt/node' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'digits/4' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'digits/0' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'digits/1' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'digits/2' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'digits/2' (language 'en')
-- <DAHDI/pseudo-1453405903> Playing 'rpt/remote_disc' (language 'en')
-- Hungup 'DAHDI/pseudo-1453405903'
-- Hungup 'DAHDI/pseudo-374682405'
alarmpil74*CLI>

```

Type exit at the prompt when done to return to the main admin screen.

Main reasons for not being able to decode DTMF is improper level settings or an improper de-emphasis setting. You can change these items in the simpleusb-tune-menu which you can get to from the ADMIN menu or by just typing that at the Linux prompt.

In simpleusb-tune-menu item 2 is the level settings and item G toggles de-emphasis. Try turning de-emphasis on or off and check that the level is correct. Voice level should peak at 5khz on the simpleusb-tune-menu item 2 bar graph display.

Changes to the simpleusb-tune-menu in the V1.5 version are dynamic, they change instantly so there is no need to reload or restart.

You could open two putty sessions and have both of the screens open simultaneously to make it easier to see your results.

It is also possible Allstar is not receiving any audio from you radio. The COS line must be active to receive audio. In the simpleusb-tune-menu select item V for view. You will see a display at the bottom of the screen.

The COS input should show KEYED when you transmit TO your node radio, node radio receiving signal, and CLEAR otherwise. If this is not working your COS is not correctly connected. If it is reversed you need to toggle the COS mode - Item J - if this is working correctly and you are still not seeing RX audio (item 2) then check your audio connections from your radio.

Allstar has an often little known and used command \*4 mapped to ilink,4 which can be used to send DTMF commands to OTHER nodes just like you would to your own. The node you would control needs to be connected for the control to work. While this is in some cases a useful command it also has enormous security risks if used in a nefarious way. Users who need to use it can turn it on with this added command in the node stanza of each node requiring it in rpt.conf. Leaving the command out or setting it to 0 has the same effect of restricting its use. Few users will have a need

to turn this on -

```
remote_dtmf_allowed=1
```

This command is only needed at the node you want to control and on any nodes in between if it is not a direct connect. It is not required on the source node unless you also want it to be controlled from somewhere else. Keep in mind that it is a general security issue to turn this on and you should only do so if it is needed. This is the classic "back door" which probably made sense when Allstar was a small group with trust between users. We will be introducing a pin number security feature for this in a future update which will make it more secure to leave it on.

## Sending Accurate Baofeng DTMF Tones from the UV-5R

While pressing the PTT + DTMF key(s), the mic does not mute. This can cause the mic to pick up any sidetones from the speaker and will overdrive and/or distort the transmitted tone(s).

### The Solution

The solution is in the Baofeng's menu settings. The menu #16 (DTMFST) setting needs to be set to "OFF". This is accomplished by following the steps below.

Press the following buttons on your Baofeng UV-5R:

Menu > 1 > 6 > Menu > 0 > Menu > Exit

To transmit the A,B C or D press the following buttons

Button	DTMF
Menu	A
▲	B
▼	C
EXIT	D

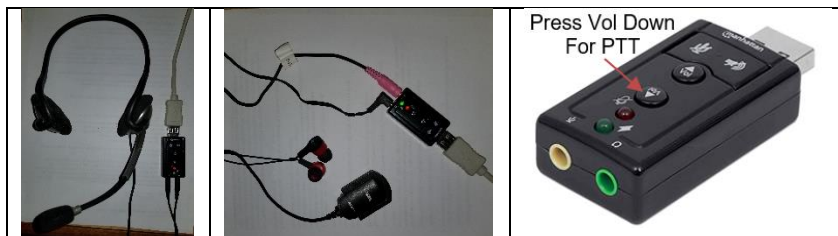
## Radioless Node

Below are two methods to operate on Allstar without a radio. Method 1 uses a sound fob that has volume control buttons on it. Method 2 also uses a sound fob that does NOT have volume control buttons on it.

Method 1 (Thanks to W5MGM gave me this information)

The sound fob with the volume control buttons on it that I used is the Manhattan USB 2.0 Sound Adapter. Below is a picture of it, and I bought it off Amazon. Here's a link to Amazon (note this link might not work if Amazon changes selling it) Link: [Amazon manhattan Sound fob](#)

What I did was use a USB extender cable to plug into the Pi so I didn't have the fob plugged directly into the back of the Pi. You can plug in ear buds and a separate mic, or a headset. Just press on the volume down button to activate the PTT.



## Method 2

Information below was found on Hamvoip site by Doug Crompton.

The simpleusb-tune-menu code in V1.5 (Item 12 on the ADMIN menu) has a manual key option. The is the (K) for key in the menu. This manually keys the COS line just like an incoming RX signal would from your node radio.

All you have to do is connect a headset and microphone to your USB FOB, open the simple-usb-tune-menu application and use the (K) option as PTT.

This toggles back and forth by typing (K) followed by a (CR) carriage return. You also must setup the RX and TX levels (menu item 2 and 3) for the headphone/mic.

Make sure you don't leave the(K) option keyed by accident. It will however clear (unkey) when you leave the simpleusb-tune-menu application.

## Radio Mobile - Online tool for predicting the propagation of radio waves

Radio Mobile Online is a tool for predicting the propagation of radio waves dedicated to the amateur radio community.

A mathematical model and digitized terrain information are used to simulate the transmission of waves between two fixed sites (radio link) or between a fixed site and a mobile (radio coverage).

Here is a link to the site:

[Radio Mobile](#)

Below is a screen shot of the site, you will need to create an account first. Click on the Create A New account.

This tool is sponsored by  commandite cet outil

[TowerCoverage.com](#)

  
**Radio Mobile Online / En ligne**

[Try the new Windows Desktop version - RmWeb 2.1.1.0 - Essayez la nouvelle version pour bureau Windows](#)

Utilisateur	<input type="text"/>	User
Mot de passe	<input type="password"/>	Password
<input type="button" value="Soumettre - Submit"/>		

[Create a New account](#) [English](#) [Lost your user name or password...](#)

[Créer un Nouveau compte](#) [Français](#) [Perdu votre nom d'utilisateur ou le mot de passe...](#)

[Crea una cuenta nueva](#) [Español](#) [Perdiste tu nombre de usuario o contraseña...](#)

[Crea un nuovo Account](#) [Italiano](#) [Dimenticato username o password...](#)

A free tool for amateur radio [ve2dbe@yahoo.ca](mailto:ve2dbe@yahoo.ca) Un outil gratuit pour la radio amateur

## Distance and Bearing Calculator

The link below is for a site that you can enter in the addresses, or Lat, Long for two points, and it will calculate the distance between them, and give the True Bearing locations. Could be helpful for pointing beam antenna's.

[Mobilefish.com](http://Mobilefish.com)

## Script descriptions

This applies to BBB Allstar versions starting at BBB v1.2 and RPi2 v1.0

Scripts located in /usr/local/sbin unless otherwise noted.

Note that example DTMF functions shown below as well as others have been entered in rpt.conf

–

these are commented so they do not work until properly configured and un-commented.

### **check\_reg.sh**

It will return the registration status of the node.

### **convert\_audio.sh**

Convert wav file to Asterisk/Allstar format

```
convert_audio.sh <input file.wav> [output_filename]
```

Input file must be .wav, output file is input filename.ul unless different second parameter filename is given.

### **halt.sh <node>**

See special note for RPi2 below

The BBB power button if held down momentarily executes the power-button-press script (see description below)

The halt.sh script sends a wav file to a local node announcing the shutdown and then ten seconds later properly shuts the system down.

This command can be executed from the command line or with a DTMF code. To use the radio keypad to shutdown your system enter the following in the rpt.conf functions stanza.

```
A1=cmd,/etc/asterisk/local/halt.sh <node>
```

A1 is the DTMF sequence. This can be any character combination you desire that does not conflict with other commands. <node> is the node where the shutdown wav file will be played.

If the node is not given then the shutdown is immediately executed. Note that A1 is only an example. On a public node a more complex code should probably be used.

The DTMF shutdown is a convenient way to shut a system down properly when you cannot ssh in to do it. Examples of these commands are in the functions stanza of rpt.conf

USE THIS REMOTELY WITH CAUTION! Once the system is shutdown you must remove and replace the power to reboot. You could not do this remotely. If your intention is to reboot use the reboot.sh instead.

The RPi2 does not have a power button. If you would like to have a halt button on the RPi2 you must fabricate one yourself. See the RPi2 halt how-to on the web page for information on how to do this.



**reboot.sh <node>**

Same as halt.sh but reboots the system instead of shutting it down.

**myip.sh**

Display the servers local (LAN or WAN) IP address

**mypublicip.sh**

Display the servers public (Internet) IP address

**sayip.sh <node>**

This script will voice the local (LAN or WAN) ip address of your server. This can be used from the command line or with a DTMF script. The IP address is said at boot by default. This can be controlled by enabling or disabling in /usr/local/etc/allstar.env A2=cmd,/etc/asterisk/local/sayip.sh <node>

**saypublicip.sh <node> <interface>**

Same as sayip.sh but say the servers public (internet) IP address. Interface is optional, defaults to eth0

**speaktext.sh “abc123” <node>**

Speak the letters or numbers to the specified node. This script is used by sayip and saypublicip. This can be used to say number or letters to the specified node.

**change\_vol.sh <inputfile> <outputfile> <volume\_change>**

Often it becomes necessary to change the volume of a sound file like voice ID's or messages you want to play over you node at a lower volume. This script will change the volume of the entire file.

The volume is expressed in decimal values above or below 1 which is no change. So .5 would represent a 50% reduction. You can experiment with different levels to suit your needs. Ex: change\_vol.sh myfile.wav mynewfile.wav .35

**saytime.pl <node>**

This Perl script when configured says the time on the specified node. It was created primarily to allow the volume to be changed on the fly but it also allows customization of what is being said. This script is typically called by a cron job often to announce the time on the hour.

An example of a cron entry to call this file is:

```
00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/local/sbin/saytime.pl $NODE1 > /dev/null)
```

This would announce the time each hour. This cron entry is installed in the stock BBB and RPi2 code. To turn it off comment the line with a # at the beginning using crontab -e

**say24time.pl <node>**

This Perl script works and is configured like saytime.pl except it outputs in 24 hour format saying the time like: 23 hours 45 minutes Eastern Standard Time to the specified node.

Whether to say the zone and the zone location are configurable in the file.

Both saytime.pl and say24time.pl can be added as a function in rpt.conf. Here is an example showing \*81 which normally says the time through the Asterisk saytime being replaced

with the external Perl script. This is in the functions context of rpt.conf. This example shows node 1998.

```
81=cmd,/usr/local/sbin/saytime.pl 1998
```

OR

```
81=cmd,/usr/local/sbin/say24time.pl 1998
```

To adjust the volume of either saytime.pl or say24time.pl change the volume value in the file. Here it is shown as reduced to .35

```
@proglis = ("sox --temp /tmp " . $outdir . "/temp.gsm " . $outdir . "/current-time.gsm vol -0.35");
```

### **power-button-press (Beagle Bone Black only)**

This script is called when the BBB power button (button closest to the RJ45 Ethernet connector) is pushed. Currently it gives verbal feedback on the node number you configure in this file and then shuts the system down in an orderly fashion. This can be used to shut the BBB down properly when at a location where you cannot ssh in. Note you can also use DTMF if you have it setup as shown for the halt and restart scripts. This script can be modified to perform other tasks when the power button is pressed if desired.

### **/usr/local/bin/tts\_audio.sh <-s> some.txt**

Creates an Asterisk playable audio file from a text file. This uses the google online TTS engine. If the text file is greater than 100 characters the -s option must be specified. See the Text to Speech how-to on the hamvoip.org web page for more details about tts on the BBB.

Note starting with BBB v1.3 and RPi2 v1.0 the -s option is not required.

### **backup.sh, restore.sh, usb-backup.sh, usb-restore.sh, image-backup.sh**

These scripts are used to backup and restore your Allstar configs to a file or USB stick and also make a complete copy of your SD card image to a USB stick. See the separate how-to on backup for more information.

### **pcsensor**

Reads temperature sensor devices over USB. Type 'pcsensor -h' for help

Inexpensive sensors can be purchased over the Internet. URL's will probably change but here are two examples. Also Google 'pcsensor' and/or 'temper1'

<http://pcsensor.com/index.php>

<http://www.ebay.com/itm/181265813495>

### **watchdog**

Monitors and resets the processor should a failure occur. This is a fail safe mechanism to restart a system especially in an inaccessible or remote location. To activate set the watchdog option to enable in /usr/local/etc/allstar.env save the file and reboot. To turn it off set the line to disable, save and and reboot. By default it is not activated. See the environment how-to.

### **firsttime.sh**

This is the script that is run the first time you log into your newly installed image. It's purpose is setup your login password, your system name, and your network settings. It also will ask if you want to configure your node.

While usually only run once at first configuration this script can be run again to change any of these settings. See the configuration how-to for more information.

### **node-config.sh**

This script is called by the firsttime.sh script if requested by the user. It configures all your node settings and updates the configuration files. It can be run at any time to change node settings but be sure to read the warnings in the configuration how-to.

The following scripts are only in the BBB starting at v1.3 and RPi2 starting at v1.0

### **818-prog, 818cli-prog**

These scripts are used to program the SA818 transceiver modules via a USB to serial adapter. See the SA818 how-to on the web page for more information.

### **allstar\_ssh.sh**

This script quickly connects you to a nodes ssh port. Syntax is -

allstar\_ssh.sh <node> <port>

The port is optional and defaults to 222. This only works for public IP addresses not LAN nodes. It is a quick way to connect without knowing or looking up the nodes IP address.

### **benchmark.sh**

Does a quick benchmark of the processor speed. It is by no means a comprehensive test but good for comparisons. See the web page listed in the run for more information.

### **cpu\_stats.sh (RPi2 only)**

List the CPU parameters and settings.

### **shutdown\_monitor.sh (RPi2 only)**

Monitors an I/O port and halts the processor based on a 6 second key press. See the RPi2 halt button how-to on the web page for more information. This script is NOT generally run by a user. It is started automatically at boot when enabled in /usr/local/etc/allstar.env

### **simpleusb-config.sh**

This script is normally run at initial startup as part of the setup routine but could be run manually to change the simpleusb.conf setting via a script rather than manually editing.

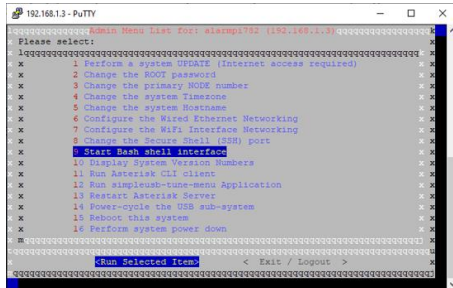
### **sshportsetup.sh**

Script to set the ssh port. This script is normally run at initial startup as part of the setup routine but could be run manually to change or check the ssh port. The default port is 222. If you change the port you need to logout and back in on the new port. Don't forget the port or you will not be able to log back in!

## Scheduled Node Connect Disconnect crontab Example

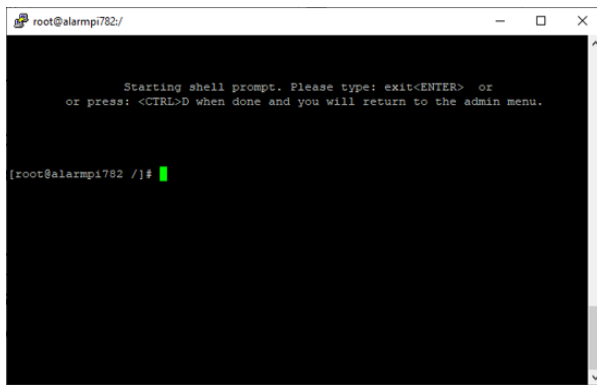
Procedure to edit crontab on Allstar node to schedule automatic connect and disconnect to another node. I used this to have one of my nodes automatically connect to a hub at 4:30 am and disconnect from it at 7:30pm.

PuTTY into the node to a bash shell prompt.



```
192.168.1.3 - PuTTY
Admin Menu List For: alarmpi782 (192.168.1.3)
Please select:
1 Perform a system UPDATE (Internet access required)
2 Change the ROOT password
3 Change the primary NOD number
4 Change the system Timezone
5 Change the system Hostname
6 Configure the Wired Ethernet Networking
7 Configure the WiFi Interface Networking
8 Change the Secure Shell (SSH) port
9 Start Bash shell interface
10 Start System Admin Menu
11 Run Asterisk CLI client
12 Run asteriskd-tune-menu Application
13 Restart Asterisk Server
14 Power-cycle the USB sub-system
15 Reboot this system
16 Perform system power down
Run Selected Item < Exit / Load >
```

I'm not positive if what follows has to be done from the root directory, but just in case, be sure once you're into the node you're at the root directory. There should be no directory listed after the prompt, see below:

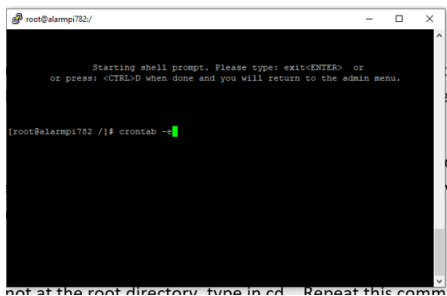


```
root@alarmpi782:/
Starting shell prompt. Please type: exit<ENTER> or
or press: <CTRL>D when done and you will return to the admin menu.

[root@alarmpi782 /]#
```

If your not at the root directory, type in `cd ..`. Repeat this command until you're at the root directory.

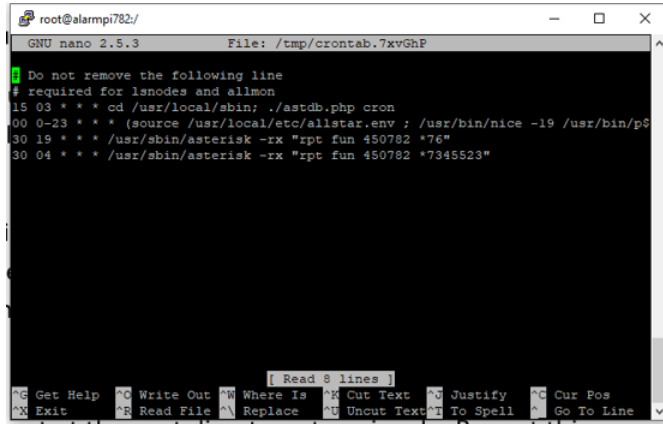
Next you edit the crontab. At the prompt, type `crontab -e`



```
root@alarmpi782:/
Starting shell prompt. Please type: exit<ENTER> or
or press: <CTRL>D when done and you will return to the admin menu.

[root@alarmpi782 /]# crontab -e
```

This will bring up the editor for making the changes. You should see something like what's shown below. The last 2 lines are the lines I added to connect and disconnect.



```
root@alarmpi782:/
GNU nano 2.5.3 File: /tmp/crontab.7xvGhP
# Do not remove the following line
# required for lsnodes and allmon
15 03 * * * cd /usr/local/sbin; ./astdb.php cron
00 0-23 * * * (source /usr/local/etc/allstar.env ; /usr/bin/nice -19 /usr/bin/p8
30 19 * * * /usr/sbin/asterisk -rx "rpt fun 450782 *76"
30 04 * * * /usr/sbin/asterisk -rx "rpt fun 450782 *7345523"
```

Note time is in 24 hour format. Also be sure you have a space between the minutes, hours, and all the \*'s. Also be sure to have the quotation marks shown in these command lines.

Line below used to set time to connect to another node:

```
mm hh * * * /usr/sbin/asterisk -rx "rpt fun YourNodeNumber *73NodeNumberToConnectTo"
```

Line below used to set time to disconnect from all nodes:

```
mm hh * * * /usr/sbin/asterisk -rx "rpt fun YourNodeNumber *76"
```

When done updating, do a Ctrl O to write out the file, and Ctrl X to exit. To then return to the admin menu, type exit at the prompt.

## Schedule Commands with crontab

Note, following information was obtained off internet and is general information, this is not specific for an Allstar node.

### About crontab

The crontab (short for "cron table") is a list of commands that are scheduled to run at regular time intervals on your computer system. The **crontab** command opens the crontab for editing, and lets you add, remove, or modify scheduled tasks.

The daemon which reads the crontab and executes the commands at the right time is called cron. It's named after Kronos, the Greek god of time.

## Command syntax

```
crontab [-u user] file
```

```
crontab [-u user] [-l | -r | -e | -i] [-s]
```

### Options

file	Load the crontab data from the specified file. If file is a dash ("-"), the crontab data is read from <a href="#">standard input</a> .
------	--

<b>-u</b> <i>user</i>	Specifies the user whose <b>crontab</b> is to be viewed or modified. If this option is not given, <b>crontab</b> opens the crontab of the user who ran <b>crontab</b> . Note: using <b>su</b> to switch users can confuse <b>crontab</b> , so if you are running it inside of <b>su</b> , always use the <b>-u</b> option to avoid ambiguity.
<b>-l</b>	Display the current crontab.
<b>-r</b>	Remove the current crontab.
<b>-e</b>	Edit the current crontab, using the editor specified in the environment variable <b>VISUAL</b> or <b>EDITOR</b> .
<b>-i</b>	Same as <b>-r</b> , but gives the user a yes/no confirmation <u>prompt</u> before removing the crontab.
<b>-s</b>	<u>SELinux</u> only: appends the current SELinux security context <u>string</u> as an <b>MLS_LEVEL</b> setting to the crontab file before editing or replacement occurs. See your SELinux documentation for detailed information.

## Overview

The **crontab** command is used to view or edit the table of commands to be run by **cron**.

Each user on your system can have a personal crontab.

Crontab files are located in **/var/spool/** (or a subdirectory such as **/var/spool/cron/crontabs**), but they are not intended to be edited directly. Instead, they are edited by running **crontab**.

### Cron command entries

Each **cron** command entry in the crontab file has five time and date fields (followed by a user name, only if it is the system crontab file), followed by a command.

Commands are executed by **cron** when the minute, hour, and month fields match the current time, and at least one of the two day fields (day of month, or day of week) match the current day. The **cron** daemon checks the crontab once every minute.

**Note:** Nonexistent times, such as "missing hours" during daylight savings "Spring forward" days, will never match. This causes jobs scheduled during the "missing times" not to run during those times. For the same reason, times that occur more than once during daylight savings (in the Autumn) will cause matching jobs to run twice.

### Time and date fields

field	allowed values
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names; see example below)
day of week	0-7 (0 or 7 is Sunday, or use names; see below)

Any of these fields can be set to an asterisk (\*), which stands for "first through last". For instance, to run a job every month, put \* in the Month field.

Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive; for example, **8-11** for an "hours" entry specifies execution at hours 8, 9, 10 and 11.

Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "1,2,5,9", "0-4,8-12".

Step values can be used in conjunction with ranges. For example, "0-23/2" in the Hours field means "every other hour." Steps are also permitted after an asterisk, so if you want to say "every two hours", you can use "\*/2".

Names can also be used for the "month" and "day of week" fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.

The "sixth" field (the rest of the line) specifies the command to be run. The entire command portion of the line, up to a newline or % character, will be executed by `/bin/sh` or by the shell specified in the SHELL variable of the crontab. Percent-signs (%) in the command, unless escaped with backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input.

Note that the day of a command's execution can be specified by two fields: day of month, and day of week. If both fields are restricted (in other words, they aren't \*), the command will be run when either field matches the current time. For example, "30 4 1,15 \* 5" would cause a command to be run at 4:30 am on the 1st and 15th of each month, plus every Friday.

### The crontab file

Each line of a crontab file is either "active" or "inactive". An "active" line is an environment setting, or a cron command entry. An "inactive" line is anything ignored, including comments. Blank lines and leading spaces and tabs are ignored. Lines whose first non-space character is a pound-sign (#) are interpreted as comments, and are ignored. Comments are not allowed on the same line as **cron** commands, because they will be interpreted as part of the command. For the same reason, comments are not allowed on the same line as environment variable settings.

### Environment settings

An environment setting line in the crontab can set environment variables for whenever cron runs a job.

**Tip:** Not every system's crontab can include environment settings. On Ubuntu and Debian, and systems that use GNU **mcron**, environment settings can be made in the crontab. On other systems, such as Arch Linux and Fedora, environment settings in the crontab are not allowed. Check your distribution's cron documentation for more information.

An environment setting in the crontab is formatted as:

```
name = value
```

The spaces around the equal sign (=) are optional, and any subsequent non-leading spaces in *value* will be part of the value assigned to *name*. The value string may be placed in quotes (single or double, but matching) to preserve leading or trailing blanks.

Some environment variables are set automatically by **cron**:

- **SHELL** is set to `/bin/sh`.
- **LOGNAME** and **HOME** are set from the `/etc/passwd` line of the crontab's owner. **HOME** and **SHELL** may be overridden at runtime by settings in the crontab; **LOGNAME** may not.
- The **LOGNAME** variable is sometimes called **USER** on [BSD](#) systems. On these systems, **USER** will be set also.

## Configuration

### Permitting users to run cron jobs

Cron jobs can be allowed or disallowed for individual users, as defined in the files `/etc/cron.allow` and `/etc/cron.deny`. If `cron.allow` exists, a user must be listed there to be allowed to use a given command. If the `cron.allow` file does not exist but the `cron.deny` file does, then a user must *not* be listed there to use a given command. If neither of these files exists, only the [superuser](#) will be allowed to use a given command.

Cron permissions can also be defined using PAM (pluggable authentication module) [authentication](#) to set up users who may or may not use `crontab` and system `cron` jobs. PAM configuration is located in `/etc/cron.d/`.

### Configuring the temp directory

The temporary directory for cron jobs can be set in environment variables listed below. If these variables are not defined, the default temporary directory `/tmp` is used.

### Configuration files

File	Description
<code>/etc/cron.allow</code>	If this file exists, users must be listed in this file to be able to run cron jobs.
<code>/etc/cron.deny</code>	If this file exists, users must <i>not</i> be listed in this file to be able to run cron jobs.

If neither configuration file exists, only the superuser may run cron jobs.

### Examples

Edit your crontab.

```
crontab -e
```

Display ("list") the contents of your crontab.

```
crontab -l
```

Remove your crontab, effectively un-scheduling all crontab jobs.

```
crontab -r
```

Edit the crontab of the user named `charles`. The `-u` option requires administrator privileges, so the command is executed using [sudo](#).

```
sudo crontab -u charles -e
```



View the crontab of user **jeff**.

```
sudo crontab -l -u jeff
```

Remove the crontab of user **sandy**.

```
sudo crontab -r -u sandy
```

## Crontab entries

The following are examples of entries which could be included in a crontab.

Run the shell script **/home/melissa/backup.sh** on January 2 at 6:15 A.M.:

```
15 6 2 1 * /home/melissa/backup.sh
```

Days and months can be listed by name (**Monday**) or abbreviation (**Jan**). Zeroes at the beginning of a number are valid, which can help you make multiple entries line up visually.

For instance, the next example runs the same script as above, at 12:01 AM, every Monday in January:

```
01 00 * Jan Monday /home/melissa/backup.sh
```

Run **/home/carl/hourly-archive.sh** every hour, on the hour, from 9 A.M. (**09:00**) through 6 P.M. (**18:00**), every day:

```
00 09-18 * * * /home/carl/hourly-archive.sh
```

Run **/home/wendy/script.sh** every Monday, at 9 A.M. and 6 P.M.:

```
0 9,18 * * Mon /home/wendy/script.sh
```

Run **/usr/local/bin/backup** at 10:30 P.M., every weekday:

```
30 22 * * Mon,Tue,Wed,Thu,Fri /usr/local/bin/backup
```