# Script descriptions

**This document 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.

**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

```
@proglist = ("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 fisrt time you log into your newly installed image. It's purpose is setup your BBB 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!