AllStar Link Network -- USB Radio

USB Device Configuration And Provisioning

Each USB device requires specification of operating mode parameters, which is done by setting various parameters in the /etc/asterisk/usbradio.conf file. This may either be done for you by this Portal, if you choose the Portal to create your configuration, or by manually editing the file. There are many different potential configurations of radio hardware and associated connection with the USB Radio device, and the various parameters are an attempt at addressing these differences.

Associating Physical USB Devices With Their Device Designations (USB Device Naming)

Every Physical USB Radio device, such as a URI or a USB fob is assigned a device name, either by the Portal (in the case of Portal-Based configuration, or by you when configured manually. When configured manually, it is not necessary to follow any particular standard, but calling it usb<Node-Name> (like for example, usb1999 would be associated with node 1999) might be a good idea. Click here for more details.

At this point, it is assumed that a configuration for all of the desired USB Radio devices has been generated (by whatever means), and has been placed in the configuration file /etc/asterisk/usbradio.conf, and the system is running with this configuration. If this is to be done manually, see below for details of how to do so. The USB configuration (and a number of others) must be done previous to starting the Asterisk system.

If your system has only one single USB Radio device, the association of the device and its name is quite easy and obvious (since there only is one of them). For systems with multiple USB Radio devices, it gets much more complicated. The problem is seriously compounded by the fact that all USB Radio devices (based on the CMedia CM108, CM119, etc) all look alike to the computer, and there is no way to tell them apart, other then by what actual USB connector (interface) that they are attached to.

There are several methods to associate and determine what USB devices go with what device names (designations). The most practical one (which we call the "flashing and swapping method") is described here:

1. Plug in all of your USB radio devices. This may be done with the system in operation, as both the system hardware and the software are perfectly capable of handling plugging and un-plugging of these USB devices while in operation, even while in use (the application software won't even know its happening).

2. The system will seemingly rather arbitrarily associate all your USB Radio devices with the names specified in the configuration file. There is a small but finite chance that it did it the way you wanted it. If not, it is possible to set things up in a desired manner.

3. Run the radio-tune-menu program.

4. Use menu item (1) to select the a USB device, and use the (F) option which will make the associated USB device cause its attached transmitter to transmit a 1000Hz tone on and off a couple of times (and make the red light on the interface flash on and off, if so equipped) in order to identify which USB

device is associated with the selected USB interface designator.

5. If the association is with the wrong one, use the (S) option to swap the USB device with another one and repeat step 4 and this step until you have found the correct one.

6. Repeat steps 4 and 5 until all the devices are associated with the correct designators.

7. Going through all the devices, one at a time, menu item (1) to select the device, then use item (W) to save the device parameters permanently. The USB device to designator association is one of the parameters that gets saved when you do this. If you happen to un-plug a USB device, be sure to plug it back into the same connector (USB interface) that it came out of, otherwise you may have to repeat this whole procedure again.

USB Radio Configuration File (/etc/asterisk/usbradio.conf)

The USB Radio Configuration file /etc/asterisk/usbradio.conf consists of stanzas that each contain the configuration parameters for a USB Radio device, named with the devices name, such as [usb] or [usb1999]. Here is an example of such a file, containing one stanza (which happens to be the default configuration file):

```
;
; chan_usbradio.c configuration file
;
[general]               ; this has to be here and is RFU

[usb]                   ; USB Channel Name e.g. usb or usb1, usb2, ...
                        ; this starts the definition block for a device

; General Device Parameters


hwtype=0                ; USB Sound Adapter Hardware Type
                        ; Set to 0 for DMK Eng. URI, or USB sound adapters
                        ; modified using the instructions from usbfob.pdf.
                        ; Set to 1 for DingTel/W9SH modified usb adapters.

duplex=0                ; Connected Radio Device Duplex Capability
                        ; 0 = Simplex (PTT / Release to Listen)
                        ; 1 = Duplex (simultaneous Tx and Rx)

eeprom=0                ; 1 = Indicates that an EEPROM internal to the radio
                        ; adapter and cable is expected.
                        ; 0 = no warning message if no EEPROM found.

; Receiver-Related  Parameters


rxboost=0               ; Rx Audio Boost
                        ; 0 = 20db attenuator inserted,
```

```
                        ; 1= 20db attenuator removed
                        ; Set to 1 for additional gain if using a low-level
                        ; receiver output.


rxctcssrelax=1          ; Rx CTCSS Decoder Criteria Relax
                        ; reduce CTCSS talk-off from radios w/o CTCSS TX HPF
                        ; Set to 1 for Amateur and FRS radios.

carrierfrom=dsp         ; Carrier Detect Source
                        ; Options - no,usb,usbinvert,dsp,vox
                        ; no - no carrier detection at all
                        ; usb - via USB radio adapter COR connection
                        ; usbinvert - same as above but inverted polarity.
                        ; dsp - from RX noise using dsp methods
                        ; vox - voice activated via RX audio

ctcssfrom=dsp           ; CTCSS Decoder Source
                        ; Options = no,usb,dsp
                        ; no - CTCSS decoding, system will be carrier squelch
                        ; usb - CTCSS decoding using input from USB adapter
                        ; usbinvert - same as above but inverted polarity.
                        ; dsp - CTCSS decoding using RX audio in DSP.
                        ; rxdemod option must be set to flat for this to work.

rxdemod=flat            ; Rx Audio Source Type from Radio
                        ; Options = no, flat, speaker
                        ; no - RX audio input not used
                        ; flat - source is un-filtered discriminator signal
                        ; speaker - source is filtered, de-emphasized audio

; Transmitter-Related  Parameters


txboost=1               ;Set to 1 to fix the 6db down audio bug in "voice" mode


txprelim=yes            ; Tx Audio PreEmphasis and Limiting in software
                        ; no - Audio is not pre-emphasized and limited.
                        ; for application to radio microphone input
                        ; yes - Audio is pre-emphasized, limited and filtered
                        ; for direct connection to an FM modulator

txtoctype=notone        ; Transmit Tone Turn Off Code method:
                        ; Options = no,phase,notone
                        ; no     - CTCSS tone encodes up to tx carrier drop
                        ; phase  - encode reverse phase before tx carrier drop
                        ;         AKA ("reverse burst") before unkeying TX
                        ; notone - drop tone, wait and then drop carrier
```

```
                    ;       AKA ("chicken burst")

txmixa=composite    ; Tx Mix Output Channel A (Left) Output Type
                    ; Options = no,voice,tone,composite,auxvoice
                    ; no - Do not output anything
                    ; voice - output voice only
                    ; tone - CTCSS tone only
                    ; composite - voice and tone
                    ; auxvoice - voice output for monitoring

txmixb=no           ; Tx Mix Output Channel B (Right) Output Type
                    ; See txmixa above.

invertptt=0         ; Invert PTT
                    ; 0 = ground to transmit, 1 = open to transmit
                    ; This is the collector lead of the 2N4401 transistor
                    ; on the modified usb sound adapter.

; CTCSS-Related Parameters
; These will be ignored and my be omitted if CTCSS is not in use

txctcssdefault=88.5 ; Tx CTCSS Default Encode
                    ; Encodes this tone if no other tone in use.

rxctcssfreqs=88.5   ; rx ctcss frequency. this must be in the tone table.
txctcssfreqs=88.5   ; tx ctcss frequency. any frequency is permitted.

; end of file
```

Receiver with audio coming directly from the output of the discriminator:

```
carrierfrom=dsp
ctcssfrom=dsp
rxdemod=flat
```

Receiver with audio coming from de-empahsized low level audio, or the speaker output:

```
carrierfrom=usb     ; or usbinvert
ctcssfrom=usb       ; or usbinvert, or no if CTCSS not in use
rxdemod=speaker     ;Note, this requires connection to the receivers COR (carrier presence detector)
                    signal.
```

Trasnsmitter with audio going directly to the frequency modulator:

```
txprelim=yes
txtoctype=notone    ; This may be whatever is appropriate
txmixa=composite    ; Tx Mix Output Channel A (Left) Output of Composite Audio
txmixb=no           ; No output on B
invertptt=0         ; Invert PTT , set a appropriate (almost always 0)
```

Transmitter with audio going low level input or microphone input:

```
txprelim=no
txmixa=voice        ; Tx Mix Output Channel A (Left) Output of Audio
txmixb=no           ; No output on B
invertptt=0         ; Invert PTT , set a appropriate (almost always 0)
```

If you are using CTCSS (processed by the USB device, as opposed to the radio itself), be sure to set the CTCSS parameters appropriately.

Be sure also, to set 'duplex' and 'eeprom' to appropriate values.

All of these configuration options apply to both manual configuration and configuration through use of the Portal.

USB Device Setup And Tuning

Once your system is configured and running, your USB Radio devices still need to be 'tuned' (which means setting various audio levels, etc). The easiest way to perform this is to log in into your Linux system, either on the physical console, or via SSH from a remote system. There is a program called

radio-tune-menu

Which provides a menu of items appropriate for performing these 'tuning' functions. NOTE, this requires all of the Asterisk system to be running from the SVN sources as of 6/7/2010. To get instructions on upgrading your Asterisk system, Click Here.

This program generates a menu which looks something like this:

```
Active (command) USB Radio device is [usb1999]
1) Select USB device
2) Auto-Detect Rx Noise Level Value (with no carrier)
3) Set Rx Voice Level (using display)
4) Auto-Detect Rx CTCSS Level Value (with carrier + CTCSS)
5) Set Rx Squelch Level
6) Set Transmit Voice Level
7) Set Transmit Aux Voice Level
8) Set Transmit CTCSS Level
9) Auto-Detect Rx Voice Level Value (with carrier + 1KHz @ 3KHz Dev)
E) Toggle Echo Mode (currently Disabled)
F) Flash (Toggle PTT and Tone output several times)
P) Print Current Parameter Values
S) Swap Current USB device with another USB device
T) Toggle Transmit Test Tone/Keying (currently Disabled)
W) Write (Save) Current Parameter Values
0) Exit Menu

Please enter your selection now:
```

Note: If your USB device is on a node configured as a remote base, then you must first make a node connection to the remote base from an external node, because the USB device is not actually opened until an external node is connected. This does not apply to all other node configurations.

Scenarios And Environments

The best (optimum) method

The optimum case, and the only way to truly perform precision calibration of your audio levels is with the use of either a Service Monitor, or some sort of precision FM signal generator and Deviation meter.

If you are fortunate enough to have access to such a device, please perform the following 8 steps:

1. If the desired USB device is not currently selected, use menu item (1) to select the desired USB device.

2. Without any signal on the input of the receiver, use menu item (2) to automatically calibrate the base received signal level. If the device requires adjust of a squelch control in order to make it output noise (standard lack of signal static noise), please do so. Please return the squelch control to its normal position after performing this step.

3. Apply a hard-limiting full-quieting signal on the input of the receiver, with modulation of 1000Hz at 3Khz deviation. It should not be necessary to encode CTCSS at this time. If you do, please make sure that there is 3 KHz of deviation at 1000Hz, plus the deviation of the CTCSS (nominally 650Hz). Then, use menu item (9) to automatically set the receive voice level.

4. If your receive source for this device is flat (discriminator) audio, remove the 1000Hz modulation from the signal, and encode appropriate CTCSS tone at 650Hz Deviation. Then, use menu item (4) to automatically set the receive CTCSS level. If your receive source is not flat audio (or you do not use CTCSS), skip this step.

5. If menu item (T) does not show that Transmit Tone/Keying is enabled, use menu item (T) to enable it.

6. Remove the signal from the Receiver. Make sure the transmitter is connected to a proper load. Use menu item (6) to manually adjust to audio output to 3KHz deviation with the 1000Hz test tone generated and the transmitter is keyed by the USB driver.

7. If your transmit source for this device is configured to generate CTCSS, use menu item (8) to manually adjust the transmitted CTCSS level to output 650Hz of deviation, while the USB driver generates the CTCSS tone and keys the transmitter.

8. Be sure to save the parameters that you just changed, by using the (W) (Write Parameters) menu item. This writes the parameters to a file and optionally the EEPROM connected to the USB device (if so configured).

The "kind of acceptable" method

This method relies on generated DTMF tones from a user's transmitter, and a "good" ear. It involves mostly manual adjustment of audio levels.

To accomplish this, please perform the following 8 steps:

1. If the desired USB device is not currently selected, use menu item (1) to select the desired USB device.

2. If your receive source for this device is flat (discriminator) audio, then without any signal on the input of the receiver, use menu item (2) to automatically calibrate the base received signal level. If your receive source is not flat audio, skip this step.

3. Using a users transmitter (preferably within feet of the receiver being adjusted, or at least having a hard-limiting full-quieting signal on the input of the receiver), transmit DTMF tones to the receiver while using menu item (3) to display the perceived audio level, and make adjustments to it so that the DTMF tones are roughly at a perceived 3KHz deviation. It is probably a good idea to try different DTMF tones, and perhaps try several different users radios to get the best consensus of the desired setting.

4. If your receive source for this device is flat (discriminator) audio, transmit an unmodulated signal from the users radio to the receiver (encoding the appropriate CTCSS tone). Then, use menu item (4) to automatically set the receive CTCSS level. If your receive source is not flat audio (or you do not use CTCSS), skip this step.

5. If menu item (T) shows that Transmit Tone/Keying is enabled, use menu item (T) to disable it.

6. Here's the tricky part. You need to have a source of "known correct" level audio to adjust the transmitter audio level. If this is a full duplex repeater, then you can use a signal on the input of the system (the receiver), since we just calibrated it and we know the level is correct. If it is not a full-duplex system, then you need some other source of audio. The use of "echo mode" (menu item (E)) is a good way of doing this. It records a received transmission (20 secs max), then after the transmission is over, it re-transmits it back out exactly as received. You can also bring up a link to a system with users talking at a known-good level. If you enable "echo mode", be sure to disable it when done testing. Regardless of how you accomplish this, the bottom line is to use another user's

receiver to an A/B comparison of the audio levels of a live, direct signal (received directly from the user's transmitter that is providing the signal to the receiver connected to the USB device being adjusted) and the signal being transmitted from the transmitter connected to the USB device that is being adjusted. The object is to get the two compared signals to be at the same audio level.

7. If your transmit source for this device is configured to generate CTCSS, you can use menu item (8) to manually adjust the transmitted CTCSS level if necessary. If it seems to work, leave it alone (it defaults to a set value of 200). If its not enough to reliably be decoded by user's radios, raise it. If its so loud it can be heard, lower it considerably. Without proper equipment, this is a difficult one to even come close to truly getting right.

8. Be sure to save the parameters that you just changed, by using the (W) (Write Parameters) menu item. This writes the parameters to a file and optionally the EEPROM connected to the USB device (if so configured).

The "if this is the best you can do" method

This method relies on speech levels from a user's transmitter, and a "good" ear. It should be avoided if at all possible. It's basically identical to the "kind of acceptable" method (see above) with step 3 different, as follows:

3. Using a users transmitter (preferably within feet of the receiver being adjusted, or at least having a hard-limiting full-quieting signal on the input of the receiver), Speak in a normal voice into the transmitter, and transmit to the receiver while using menu item (3) to display the perceived audio level, and make adjustments to it so that the speech level peaks roughly just below a perceived 5KHz deviation. It is probably a good idea to try different people speaking, and perhaps try several different users radios to get the best consensus of the desired setting.

Axillary Audio Output

If your USB device has one of its outputs configured as an "auxiliary" output, you can use menu item (7) to manually set this level.