

Audio Recording on AllStar Link node

Audio Recording on AllStar Link nodes.

In `rpt.conf` there are stanzas that start with the node number, such as:

```
[28174] ; Change this to your assigned node number
rxchannel=Radio/usb
;rxchannel=Simpleusb/usb
duplex=0
~
archivedir=/media/USB
```

If it contains the “**archivedir**=<some recording location>” as shown above, then the audio passing through the radio will be recorded. In my case the recording location is a USB stick formatted as an **ext4** file system identified by the system as **/dev/sda1**. So edit the file **/etc/fstab** and add a line as shown in bold below:

```
#
# /etc/fstab: static file system information
#
# <file system> <dir> <type> <options> <dump> <pass>
/dev/mmcblk0p1 /boot vfat defaults 0 0
tmpfs /var/log tmpfs nodev,nosuid,size=50M 0 0
/dev/sda1 /media/USB ext4 errors=remount-ro 0 1
```

Reboot and the drive should be mounted. We can check that the USB is in fact mounted at the location **/media/USB** with the **df** command.

```
[root@AllStar-KA7U USB]# df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/root 3762048 1323764 2227464 38% /
devtmpfs 493436 0 493436 0% /dev
tmpfs 497712 0 497712 0% /dev/shm
tmpfs 497712 12860 484852 3% /run
tmpfs 497712 0 497712 0% /sys/fs/cgroup
tmpfs 497712 108 497604 1% /tmp
tmpfs 51200 272 50928 1% /var/log
/dev/mmcblk0p1 102182 14104 88078 14% /boot
/dev/sda1 7561896 19996 7134732 1% /media/USB
tmpfs 99544 0 99544 0% /run/user/0
```

```
[root@AllStar-KA7U USB]#
```

The AllStarLink system will add the node number to the directory structure, so when you navigate to the saved audio files, you will do something like this:

```
[root@AllStar-KA7U /]# cd /media/USB/28174
[root@AllStar-KA7U 28174]# ls -l
total 1188
-rw----- 1 root root 6154 Mar 15 21:22 20160315.txt
-rw----- 1 root root 12800 Mar 15 20:51 20160315205145.WAV
-rw----- 1 root root 5455 Mar 15 20:51 20160315205153.WAV
-rw----- 1 root root 5130 Mar 15 20:52 20160315205159.WAV
.... snip...
```

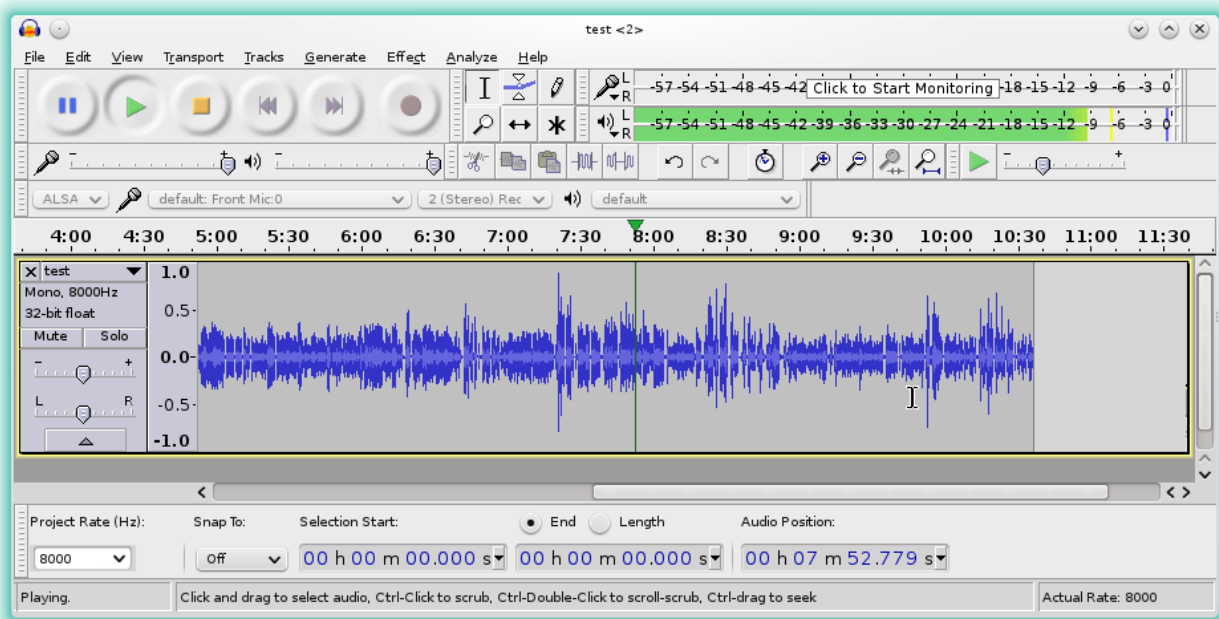
As you can see above the system has been busy recording audio files in the mounted USB stick. To use these files, we need to copy them to a computer capable of playing them. In this case the Raspberry PI is connected to another Linux computer by **SSH**, so we will copy the desired files to the connected computer with the **SCP** command issued from the connecting computer in the directory we wish the files to be copied to. Note: there is a space between *.WAV and ./ , don't forget the space. □

```
ron@linux-zdyj:~/wave> scp root@allstar-ka7u:/media/USB/28174/*.WAV ./
root@allstar-ka7u's password:
20160315205145.WAV                                100% 13KB
12.5KB/s 00:00
20160315205153.WAV                                100% 5455 5.3KB/s
00:00
20160315205159.WAV                                100% 5130 5.0KB/s
00:00
20160315205202.WAV                                100% 1685 1.7KB/s
00:00
20160315210008.WAV                                100% 4155 4.1KB/s
00:00
20160315210157.WAV                                100% 4480 4.4KB/s
00:00
... snip....
```

From here we can play the files one by one in a player of choice, such as Audacity, or we can concatenate them into one or more files using **SOX**. Without our intervention, SOX will concatenate the files in the correct date and time order with the simplest command.

```
ron@linux-zdyj:~/wave> sox *.WAV test.wav
ron@linux-zdyj:~/wave> ls -l test.wav
-rw-r--r-- 1 ron users 1034926 Mar 15 21:42 test.wav
ron@linux-zdyj:~/wave>
```

Notice in the above that a new file “test.wav” now exists. Also note that the AllStarlink system uses a naming convention on the audio files using capital letters **WAV**. Audacity has the ability to further edit the file and change from wav to other codecs such as mpg. This screenshot shows the presentation of this audio file in Audacity.



[test.wav](#)

The file is linked above, if you would like to listen to it or work with it. This recording is quite useful in as much as one of the radios has an intermittent connection problem and one of the radios is over modulated. The picture in Audacity is worth a thousand words, and being able to listen to the audio more than once, makes it much easier to determine how things sound.

If the radio is breaking squelch to often, the recordings of that will be plentiful. So if we plan to record, the squelch should be sufficient to silence the radio until a signal appears.

Ron Morell
KA7U

Motorola R1200AX Service Monitor

Motorola R1200AX Service Monitor.



At present I do not have a manual but the unit does turn on and most functions seem to work. I did find limited instructions

here: <http://www.antiqueradios.com/forums/viewtopic.php?f=8&t=62528>

This is a copy of the relevant post. If you have relevant information, please leave a comment about it.

The R-1200AX is the basic 1200 with the option of a high stability timebase (the "X.")
The modules have been added later.

There is no internal load in this model. It is designed to monitor off-the-air signals or to be used with a pick-off enroute to an external dummy load or antenna.

I don't have the manual anymore, but this is how it works to measure deviation and frequency:

- 1) Dial up the known carrier frequency
- 2) Press the RECEIVE button
- 3) Press 1.5, 5 or 15 KHz to select range (FREQ ERROR)
- 4) Connect whip antenna or cable from pick-off to broadband mixer
- 5) Place wide/narrow switch in "wide" unless there's interference, in which case use "narrow."
- 6) Read frequency error on ERROR KHZ meter
- 7) Set deviation meter range as desired
- 8) Read deviation on meter when transmitter is being modulated

Unfortunately these are 30+ year old instruments which have usually seen a hard life.

They are worth little on today's market, as they have no value to commercial shops, thus intensive labor to get one running has to be considered a lost cause at some point.

I have had some where the electrolytics in the deviation meter module were totally dried out and open circuit. Other electrolytics in the unit, same story. Tantalum dip capacitors can be shorted, they are becoming notorious for this.

The manual, as I recall, is rather huge and would be a life's work for someone to scan and put on the internet. Operating the thing should be a pretty intuitive process.

For generating a signal, use the CW position generally as that is locked to the TCX0.

The red "Leveled" lamp should be lit normally. If it is out,

it means the output has either become unstable or the overload protect circuit has tripped (hence the reset button.)

The FM Cal knob sets the frequency when the generator is in the modulated FM mode.

It should be zeroed with the zero center meter. The stability while in the FM modulated mode is rather poor, it's just for testing receivers and not frequency setting. Leave it in CW normally.

posted by: Geoff Fors

OK, so after reading the above I setup to test the Service Monitor on the bench. I decided to use it in FM modulated mode to see how unstable that might be too. This video pretty much tells the story of how that worked out.

<https://www.youtube.com/watch?v=a5lSjb3t4nc>

Ron Morell
KA7U

CISCO SPA941 SPA942 configure for Asterisk

Configuring the SPA941/942 Telephones for use with ASTERISK.

[SPA941_Manual.](#)

The link above should load the manual for these phones.

For a quick Start: plug your telephone into the network jack

and connect 5vdc power to the telephone. After the telephone



starts up, press the setup button located on the front of the telephone. Use the arrow keys to scroll through the options until you find #14 "Factory Reset". Use the select button and accept to reset the telephone to the factory defaults. The telephone should reboot and when it does use the setup button and scroll to item #9 "Network". Select Network and you should find the IP address of the telephone. Enter this IP address into a Web Browser address bar (running on a computer served by the same network), and the telephone HTML setup screen should be displayed in the Web Browser. In the upper right menu area, select Admin and then Advanced. You should see a setup menu like the example found at this link:

[SPA941/942 Web Setup Menu](#)

The configuration example found at the above site is for the radio/telephone system described in the posts about [AllStar and Asterisk ver.13](#) in use at KA7U at the time of this writing. While you can use it and replicate what I have going on here, it is intended to be an example of how to do the configurations. By considering it and the manual, you might be able to configure the telephones to fit into other telephone networks as you wish for them to operate in that environment.

The most currently available firmware for the SPA941or SPA942 Telephones can be downloaded here:

[SPA941-Firmware](#)

[SPA942-Firmware](#)

Updating firmware has risks and you should know what you are about and what those risks are before upgrading firmware on your telephone. The risks are all yours...

Ron Morell

KA7U

Raspberry Pi2 RTL_TCP Server

The Raspberry Pi2 makes a nice RTL_TCP server. You can place it anywhere on the network close to a good antenna. The sample rate requires good throughput on the Network and generally Ethernet has more throughput than WiFi and Ethernet is preferred for connectivity on the LAN. Fast WiFi will work however. So to setup the RPi2 for this service, you will first install Raspbian Jessie on the SanDisk card and boot the Pi with it. Once you have logged in to the Pi, open a terminal and do the following:

1. `Passwd`
2. `sudo raspi-config`
3. `sudo apt-get update`
4. `sudo apt-get upgrade`
5. `sudo rpi-update`
6. `reboot`

log back into the Raspberry Pi

7. `cd /etc/modprobe.d`
8. `sudo nano rasp-blacklist.conf` #and add the following, then save the file:

```
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
```

9. `sudo update-initramfs -u`
10. `cd`
11. `sudo apt-get install git`
12. `sudo apt-get install cmake`
13. `sudo apt-get install libusb-1.0-0.dev`
14. `sudo apt-get install build-essential`

15. sudo apt-get autoremove
16. git clone git://git.osmocom.org/rtl-sdr.git
17. cd rtl-sdr
18. mkdir build
19. cd build
20. cmake ../
21. make
22. sudo make install
23. sudo ldconfig
24. cd /etc/udev
25. sudo cp ~/rtl-sdr/rtl-sdr.rules ./
26. sudo reboot

Log back into the Pi and start the RTL_TCP Service with:
rtl_tcp -a <the Pi's IP address>

Then using SDR# or HDSDR or whatever receiver you use on your local computer, configure the TCP interface for the server's IP address and set the sample rate and other parameters as appropriate to start receiving from the RTL-DVB USB receiver served by the Raspberry Pi.

<https://www.youtube.com/watch?v=pSq39ydj1QU>

<https://www.youtube.com/watch?v=M02N2wZe03A>

Ron – KA7U
March 4, 2016

Asterisk ver.13 on the

Raspberry Pi2

When I needed to install Asterisk on the RPi2 I searched google and found this "how to":

http://www.klaverstyn.com.au/david/wiki/index.php?title=Asterisk_Install_on_Debian

Never wanting to reinvent the wheel and noting that this install provides many useful options, I followed the recipe there. For my archive I have copied the instruction set below.

The configurations that bridge this PBX with my AllStar Link system will follow in the next post.

Ron – KA7U

March 3, 2016

To install Asterisk on a Raspberry Pi with a Debian based system.

Dependencies

```
apt-get -y install libv4l-dev ffmpeg libx264-130
```

```
cd /usr/src
```

```
wget http://www.libsdl.org/release/SDL2-2.0.4.tar.gz
```

```
tar xvf SDL2-2.0.4.tar.gz
rm -f SDL2-2.0.4.tar.gz
cd SDL2-2.0.4
./configure
make
make install

cd /usr/src
wget
http://www.pjsip.org/release/2.4.5/pjproject-2.4.5.tar.bz2
tar xvf pjproject-2.4.5.tar.bz2
rm -f pjproject-2.4.5.tar.bz2
cd pjproject-2.4.5
./configure --enable-shared
make dep
make
make install

cd /usr/src/
wget http://www.digip.org/jansson/releases/jansson-2.7.tar.gz
tar xvf jansson-2.7.tar.gz
rm -f jansson-2.7.tar.gz
cd jansson-2.7
./configure
make
make install
make check

apt-get -y install libtiff5-dev
echo /usr/local/lib/ >> /etc/ld.so.conf

cd /usr/src
wget
http://www.soft-switch.org/downloads/spandsp/spandsp-0.0.6.tar
.gz
tar xvf spandsp-0.0.6.tar.gz
rm -f spandsp-0.0.6.tar.gz
cd spandsp-0.0.6
./configure
make
make install
ldconfig
```

Asterisk Prerequisite

```
apt-get -y install subversion libspeexdsp-dev libmyodbc  
unixodbc-dev libmysqlclient-dev libnewt-dev libncurses5-dev  
uuid-dev libxml2-dev libsqlite3-dev libssl-dev portaudio19-dev
```

Install Asterisk

```
cd /usr/src  
wget  
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-  
13-current.tar.gz  
tar xvf asterisk-13-current.tar.gz  
rm -f asterisk-13-current.tar.gz  
cd asterisk-13.7.0  
./configure  
./contrib/scripts/get_mp3_source.sh  
make menuselect  
make  
make install  
make config  
make samples
```

Debian Cleanup

```
apt-get -y autoclean  
apt-get -y autoremove
```

AllStar Link on the RPI2

After a few attempts to configure a Raspberry PI 2 to be an AllStar Link node, I finally settled on using the ARCH version that is discussed here:

<http://crompton.com/hamradio/BeagleBoneBlackAllstar/>

There is a group reflector that supports the users of this version of the AllStar Link system and it is found here:

<http://lists.hamvoip.org/pipermail/arm-allstar/>

KB4FXC and WA3DSP seem to be the active developers of this ARCH Linux AllStar Link system for the Raspberry PI 2. They have modified the app_rpt.so module to correct certain issues that seem common to other Raspberry PI and ARM processor implementations of the software.

If you decided to install the AllStar Link software on a Raspberry PI 2 or BBB, and having successfully installed it, then if you are interested in some of my configurations read on.

The ARCH image is minimal as it is delivered. Everything you need for the service is there and then some, but if you want something else, well then it might be a problem because if you update the repositories you will bring in dependencies that are not compatible and will break the software. For example I wanted the terminal shell called "Screen". If you use Screen you know why, if you don't well then it might not matter, but the example of how I installed it will be useful to you for other things you might want to add. There are archived repositories available for ARCH and you can find the appropriate versions of software to match with the ARCH image you are using.

<http://fraggod.net/static/mirror/packages/archlinuxarm/armv7h/>

This link should take you to the program archives that contain useful programs for the RPi2_VER1.0_ALLSTAR.zip image, if that is the one you have installed. If not, you will need to ask on the reflector for the correct archive... Next you need to try to determine which program version will work with the dependencies already installed on your system. I guessed by checking version and date info on gcc.

```
[root@AllStar-KA7U ~]# gcc -version  
gcc (GCC) 4.9.2 20141224 (prerelease)
```

Then a scan of the gcc versions available in the archive led

me to believe the install package dated around Nov-Dec of 2014. Then scanning the *Screen* versions I found one that was created in that time frame and took a chance that it was good in the system. I found this one: `screen-4.2.1-3-armv7h.pkg.tar.xz` 12-Nov-2014 03:37 , and so it would be installed using:

```
pacman -U  
http://fraggod.net/static/mirror/packages/archlinuxarm/armv7h/  
screen-4.2.1-3-armv7h.pkg.tar.xz  
and that worked just fine.
```

Once the AllStar Link system is installed and configured with the first boot and login, I wanted to configure a telephone system to work with it. The AllStar Link system is an Asterisk PBX and functions for that purpose, but newer versions of Asterisk (ver.13 at this time) offer better security and NAT traversal without port forwarding, as well as voice mail and all the other modern features of telephone systems, so I decided to setup another Raspberry Pi2 running Asterisk ver.13 and use a SIP bridge to connect them. This configuration is involved and will be covered in another post.

Ron – KA7U

March 3, 2016