Installing

Ubuntu 22.04 LTS

+

WSPRDaemon

+

GNURADIO

+

WSJTX

PA0SIM/PE0MJX

Augustus 2023

1 CONTENTS

2 Purpose of the document 3
3 Test setup 4
4 Installing Ubuntu 22.04 LTS 5
5 Installing GNURADIO 6
6 Installing sdrplay API 6
7 Installing SDRplay block for GNURADIO 7
8 GNURADIO model with RSP receiver blocks 8
9 Installing wsjt-x 9
10 Installing WSPRDAEMON 10
11 Virtual audio sources 12
12 Modifications in WSPRDAEMON for audio input 13
13 Spectrum Spread in WSJT-x 15
14 What to do for the 40m band 15
15 Starting up the wspr spots reporter 16

2 PURPOSE OF THE DOCUMENT

This document describes the procedure to set up a wsprdaemon spots reporting PC or laptop.
Be aware that it is for one band per SDRplay device.
For more bands with one receiver, you need other hardware (i.e. KIWI, RX 888 MKI, and others)
For this setup we need:

- A laptop let's say not older than 8 years. It must accept a bootable stick with Ubuntu on it. And preferable 2 cores, or more.
- An internet cable connection. Much faster than WIFI.
- An RSPx-device, we have positive results with RSP1a, and RSP2. The RSP2 is preferred because it has a hardware entry for frequency stabilization. During this document we use the RSP2.
- A Leo Bodnar mini-GPS referenced clock, or something similar.
- An additional screen at your laptop is nice because we need some screen space to do the experiments. For a final operational setup, it is not needed.
- The latest Ubuntu LTS distribution on a stick. We used Ubuntu 22.04 LTS.
- The latest Wsprdaemon software from the site of Rob Robinett
- The latest WSJT-x software from GitHub
- The GNURADIO package from GitHub, needed is V3.10
- Additional SDRplay block for GNURADIO
- The file for gnuradio to use:
    - rsp1a.grc
- The SDRplay driver V3, from the net.
- Changes in WSPRDAEMON modules
    - wsprdaemon.conf
    - decoding.sh
    - recording.sh
- Changes in the Ubuntu audio setup

In this document we will go through all the activities, so you should not be an expert to get this done.
However, a basic understanding of laptops, Ubuntu and SDR-technology is expected. Also, the concept of "Whisper" should be between the ears.

Have fun and success to get it done.
If you have serious troubles, you can contact us.

3 TEST SETUP

We will describe the setup for the 20m band. The differences for the 40m band are given in a separate chapter.. A few alternative settings are necessary then.

Most important of course is a good antenna. We used tuned dipoles, and inverted V's. But it doesn't matter as long as you have a good reception. Be aware that in domestic areas (as where I live) a lot of QRM and man-made noise is produced, so it will impact your reception. A rural environment or better is the best to receive a lot of spots!
Connect the antenna to the RSP-device. Connect also a Leo Bodnar mini-GPS referenced clock to the RSP device.
Do that the following way. Connect a SMA cable to the Leo Bodnar device output. The other end of the cable goes to a SMA T-connector. The second end of the T-connector is connected to the RSP2 device's clock input. The third end of the T-connector is stubbed with a 50 Ohm small dummy load.
I found this on the internet and the idea is that the RSP2 must see on power up a grounded voltage level to make sure it starts up using the external reference signal. Using 50 Ohm is perfect and is also helping to get the output level of the Leo Bodnar device to the correct voltage (+- 1 V).
You can check it on a scope or a Spectrum analyzer.
The RSP2 has a MCX female connector. You can apply a SMA to MCX adapter, or as I did, build a SMA connector on the plastic RSP2 case.
Connect then both devices to your laptop or PC, for power and data transmission.

On the PC/laptop we will run Ubuntu. A Gnuradio flowgraph will get all the RSP2 data, does a few calculations with it, and sends it to audio sinks. These will be read by WSPRDAEMON and WSJT-X. WSPRDAEMON reports the received spots, and on WSJT-X you can look also at the received spots. Just for fun and checks.

We will now describe the several steps to take.

4 INSTALLING UBUNTU 22.04 LTS

Download the Ubuntu 22.04 (or later) ISO file.
https://ubuntu.com/download/desktop
Burn that ISO file on a stick. We used the burn software of RUFUS.
https://rufus.ie/en/

Put the stick in your PC/Laptop and boot your device from the stick.
Make sure you make a note of the username and password you must enter. Basically, it is a standard install.

Hints:

- Don't do the : "try" but do the "install" option at the start of the boot.
- I did it in English but other languages should of course also work
- Keyboard layout ENG, ENG.
- Normal installation and downloading updates while installing options.
- Erase the disk and install Ubuntu.
- Choose your region, we did Amsterdam.
- Choose your name, username = " wsprdaemon" and password = UbuntuWD, but of course you can pick what you want.
- Install starts,…….copying files…….restart.
- When asked, remove the Ubuntu boot stick and "ENTER"
- Log in with user = wsprdaemon and password UbuntuWD.
- Set the options which are offered the way you like.
- Go to show applications (left down) and pick the terminal app. Make it a FAVORITE because we will need it a lot.
- Change the screen power settings to your feelings. I put it the screen to "Never" because I get really frustrated by typing in my password every time.
- Open the terminal and do a "sudo apt-get upgrade" and a "sudo apt-get update". Install the found updates.

It is possible that SOFTWARE UPDATER is reporting that new updates are available. Run it.

To be sure everything we need is installed, do:
sudo apt-get install libboost-all-dev
sudo apt search boost
sudo apt install python-is-python3

Open the terminal and proceed with Gnuradio.

5 INSTALLING GNURADIO

(Install it as user wsprdaemon).
Look at this site:
https://wiki.gnuradio.org/index.php/InstallingGR
Ubuntu 22.04 should install GNUradio V3.10, because we need that version.
On the command line of Ubuntu, do:
sudo apt-get install gnuradio

That takes a while dependent on your PC/Laptop speed.
In /etc/gnuradio/conf.d edit grc.conf the following line should be there:
xterm_executable = gnome-terminal
Do:
sudo nano /etc/gnuradio/conf.d/grc.conf and check it. If not, add it or change the current xterm executable.
Do then:
sudo ldconfig

To be sure that the config is taken into account, do a restart.

Do: gnuradio-companion (don't run gnuradio-companion as root)
You get two warnings about the vocoder codec. Ignore both warnings.
Than you are welcome!
And a link is given to the Block paths. Good.
The UI (user interface) is coming up. Later on, we will work on that one, but for now close it.

6 INSTALLING SDRPLAY API

(Install it as user wsprdaemon).

Get the driverAPI from the site:
https://www.sdrplay.com/dlfinishs/

And put it in the downloads-directory of Ubuntu. I downloaded the API on a stick and put it then in Ubuntu from the stick to the downloads-directory.
Go to ~/Downloads and check whether the API is there.
Make the file executable by:
Chmod +x SDRplay_RSP_API-Linux-3.07.1.run
Then execute it:
./SDRplay_RSP_API-Linux-3.07.1.run
Read the license agreement,…..and enter,….
Install it all, also the USB id part.

7 INSTALLING SDRPLAY BLOCK FOR GNURADIO

(install as initial user wsprdaemon).

Installed are SDRplay blocks that don't use Soapy. They only need the SDRplay API 3.07 .
Look for more info here:
https://github.com/fventuri/gr-sdrplay3

We first have to install "GIT".
cd ~
sudo apt install git

and CMAKE
sudo apt install cmake

Install the blocks by:
git clone https://github.com/fventuri/gr-sdrplay3.git
cd gr-sdrplay3/
sudo mkdir build
cd build
sudo cmake ..
sudo make
sudo make install

Some files appear to be in the wrong place, they need to be copied from /usr/local to usr/lib
Copy 3 files/links, see:
https://sdrplayusers.net/forums/topic/problem-sdrplay-with-gnu-radio-3-10-1-installed-via-ubuntu-22-04/
copy the files/links by:
cd /usr/lib/x86_64-linux-gnu/
sudo cp /usr/local/lib/x86_64-linux-gnu/* . Don't forget the * the space and the . at the end of the command !!

Check whether the files/dirs are copied. Look for
libgnuradio.sdrplay3.so
libgnuradio.sdrplay3.so..3.11.0git
libgnuradio.sdrplay3.so.v3.11.0.1git-1-ge292818c
in
/usr/lib/x86_64-linux-gnu/

After running all these commands, a number of files and directories are created.
One of these can be used to enable stereo (2 channels) audio sinks.
For stereo 2 channel audio sink:
Go to directory "~/ .gnuradio" and create a file conf.d (nano conf.d) with the following content:
[audio]
audio_module = portaudio

But it is not necessary. One channel works also.

8 GNURADIO MODEL WITH RSP RECEIVER BLOCKS

(install it as user wsprdaemon)

We need a model for gnuradio to run. This is a file and it is called: rsp1a.grc
Get this file from the same place as where you got this installation guide.
Put the file in the Download directory of your initial user, in my case loek.
Now we can tell gnuradio where to get this file.
Open gnuradio with:
gnuradio-companion
And open the rsp1a.grc file from the Downloads directory.
We continue now for the 20m band, and the RSP1a.
In the RSP1a block we need:
- RSP selector: empty
- sample rate 512k
- Center frequency: 14095600 (Hz)
- Bandwidth (Hz): 200k

The other blocks are filled out with the proper parameters.

Do a restart of the PC/Laptop
Login in user wsprdaemon.
We can now check whether the script is running properly. But before doing that we need to activate the ALSA virtual audio sinks and sources by entering the command:
sudo modprobe snd-aloop
Restart gnuradio by:
gnuradio-companion

After getting a few warnings you should get a graph with the signal that is going to the BLOCK “QT GUI frequency sink. The signal shows the positive AND the negative frequencies. You can also see that the band is covering the 1400-1600 Hz part where the wspr spots are.
This means also that the signal is also going to the 2 sinks, meant for wsprdaemon and wsjt-x. But these packages are not yet installed.
My Laptop is pretty old (Acer Aspire 5738) and it is not supporting 48k sample rate. The max is 44.1k. So you have to change that in the sink blocks ! The good news is that gnuradio is telling you that the audio card is reporting that you give a signal with a to high sample rate. That is nice!
If we want to hear something out of the speakers of the laptop/pc, we have to add an audio sink block from the list at the right and put it besides the QT block. Sample rate 44.1k (in my case!), name default (in my case). I have put this block in the general script, if you don’t want is, just kick it out or disable it.

We proceed with the wsjt-x installation.

9 INSTALLING WSJT-X

This is the easy one,…..
Do as user wsprdaemon:
sudo apt install wsjtx

Intermezzo:
I started wsjt-x and I got some error reports while starting gnuradio after that. I found out that wsjt-x NEEDS 48kHz sample rate. No other possibilities. And I ran into a problem because my Acer laptop is not going higher than 44.1kHz. This causes the error reports from gnuradio, and it makes it impossible to receive/decode 48kHz data rates.
How can we solve that problem? Start all over again with a laptop which can do 48kHz or do something else. Well, I found in my junk box an USB sound stick. I put that into my laptop and in alsamixer you can make that stick the default sound"card".
I changed all the sample rates back to 48kHz in the gnuradio script and configured wsjt-x (settings/audio) with sysdefault:CARD=Loopback.
And now wsjt-x accepts the 48kHz data stream and receives and decodes the spots.
If you have a newer Laptop/PC you will most certainly have not this problem, because they support the 48kHz sample rate.

So, now we have gnuradio running, receiving the spots and wsjt-x is reading the spots. Be aware that if the tick box "Upload spots" in wsjt-x is ticked, you are reporting spots. In this phase not what you want, I guess.
Time to go to the wsprdaemon install.

10 INSTALLING WSPRDAEMON

Do :
cd ~
sudo su –
enter the wsprdaemon password (in my case UbuntuWD)
you are now root !
check with whoami
do:
sudo passwd root
Enter the new root password twice. And don't forget it!

Open a browser and go to:
https://github.com/rrobinett/wsprdaemon
scroll down to the "Greenfield" install instructions. And do exactly what is described there.
But you don't have to install the user wspdaemon, because we did that at installation of Ubuntu.
Do:
su wsprdaemon
cd ~
sudo usermod -aG sudo wsprdaemon
Enter your password (UbuntuWD for me)
exit
su wsprdaemon
cd ~
git clone https:/github.com/rrobinett/wsprdaemon.git
cd wsprdaemon
./wsprdaemon.sh -V

Some hints from our side:

- The first time you run wsprdaemon.sh -V, in our installation we got a warning that the "locale" was not properly set. In our PC/Laptop we wanted the "en_US.UTF-8 for the "LC-Numeric". Change it in the locale file, by:
  sudo nano /etc/default/locale (I use nano as editor)
  Restart the PC/Laptop, then the new locale will be active
  Do:
- cd wsprdaemon
- nano wsprdaemon.conf. Put the following in the wsprdaemon.conf file:
  # SIGNAL_LEVEL_ADJUSTMENTS = 12dB for the 20m band
  declare RECEIVER_LIST=("AUDIO_0 localhost:0,0 PE0MJX JO31cl NULL DEFAULT:12:20")
  declare WSPR_SCHEDULE_simple=("00:00 AUDIO_0,20,W2:F2")
  declare WSPR_SCHEDULE=("${WSPR_SCHEDULE_simple[@]}")
- Other changes in wsprdaemon.conf see further chapters.
- After restarting wsprdaemon, a lot of files are downloaded and installed.
  ./wsprdaemon.sh -V

After everything is installed we can make life easier by:

source ~/wsprdaemon/.wd_bash_aliases
wd-rci

After everything is done you should be able to use the following commands

- wda, start wsprdaemon
- wdz, stop wsprdaemon
- wds, print the status of wsprdaemon
- wdln, see the upload log of wsprdaemon
- wd-help, list all these commands

do:
wda
and see wsprdaemon starting up.
Wdz
to stop it again.
The sox-process is giving some warnings:
Rate clipped
Dither clipped
The reason for this is that the audio signal is to strong.
We will later tune the strength of the signal in the whole signal train.

11 VIRTUAL AUDIO SOURCES

We need internal loop back services from gnuradio to wsprdaemon and wsjt-x.
For that we do:
sudo modprobe snd-aloop

Then look for the allocated cards with the command:
alsamixer
In my case I have three sound cards:
default:0 HDA Intel
default:1 USB PnP sound device
default:2 Loopback

So we need to connect to Loopback, which is 2.

Start gnuradion-companion
Do:
gnuradion-companion

Two audio sinks in GNUradio: **hw:2,0** and **hw:2,1**

As you can see in the alsamixer, it depends on which sound hardware you have.

Startup wsjtx by giving the command:
wsjtx

Go to File/Settings/General and
Input your call and your grid in the first two fields. Then go to the audio tab and in the sound card INPUT
Pull the list down and you could choose (in my case, yours can be different)

sysdefault:CARD=Loopback

but we have two listeners (wsjt-x and wspdaemon) so we pick
plughw:CARD=Loopback,DEV=0 for wsprdaemon
and
plughw:CARD=Loopback,DEV=1 for wsjt-x

WD and WSJT-X now run in parallel using exactly the same signal.
Only wsprdaemon reports the spots!
WSJT-X is only used to compare spots. So, untick the report spots box.

See also for the difference between *hw* and *plughw*:
https://raspberrypi.stackexchange.com/questions/69058/difference-between-hwplug-and-hw

12 MODIFICATIONS IN WSPRDAEMON FOR AUDIO INPUT

In wsprdaemon.conf

SPOT_FREQ_ADJ_HZ="" (un-commented, (no 0.1Hz correction)
SIGNAL_LEVEL_UPLOAD="yes"
SIGNAL_LEVEL_UPLOAD_ID="PE0MJX"

In decoding.sh

All the changes are done in the file decoding.sh
Get the file from the same place as you got this document and put it in the Downloads directory.
Copy it from there to the ~/wsprdaemon directory
(Make a copy of the existing file first)
cd ~/Downloads
cp decoding.sh ~/wsprdaemon ,the old file is overwritten.

Some explanation for whom is interested.

Line 1363: **function decoding_daemon()**
Decoding FTS4W using JT9:
-F 220 is needed to cover the whole 200Hz WSPR band when centered at 1500Hz:

Change:
timeout ${WSPRD_TIMEOUT_SECS-110} nice ${JT9_CMD} -a ${decode_dir_path} -p ${returned_seconds} --fst4w -p ${returned_seconds} -f 1500 -F 100 ${decoder_input_wav_filename} >& jt9_output.txt
Into:
timeout ${WSPRD_TIMEOUT_SECS-110} nice ${JT9_CMD} -a ${decode_dir_path} -p ${returned_seconds} --st4w -f 1500 -F **220** ${decoder_input_wav_filename} >& jt9_output.txt

In recording.sh

All the changes are done in the file recording.sh
Get the file from the same place as you got this document and put it in the Downloads directory.
Copy it from there to the ~/wsprdaemon directory
(Make a copy of the existing file first)
cd ~/Downloads
cp recording.sh ~/wsprdaemon ,the old file is overwritten.

Some explanation for whom is interested.

Line 175: **function sleep_until_next_even_minute()** {
……..
sleep_seconds=$((${sleep_seconds}-2)) # PA0SIM: add this line
sleep ${sleep_seconds}
}
Audio files showed a too late start of the recording.

Only WSPR signals were decoded, not a single FST4W spots found.
WSJT-X however, running on the same computer, did decode all the FST4W signals!
Time on the computer was accurate. Compared with another Windows computer after a manual forced synchronization on that computer.
FST4W seems more critical for timing!
JT9 didn't decode any FST4W signals in the recorded files also.
Even WSJT-X didn't find FST4W signals, only WSPR signals were found in the files.
Starting 2 sec earlier resulted in FST4W spots!
Note: maybe 1 second is also enough? Problem is I don't know why the recording starts too late.
So maybe this is not the right solution. It affects also the Kiwi receivers.

Line 259: **function audio_recording_daemon()**

Three "return 1" lines are commented out !!!!!
Audio device name is manually set in this function to plughw:CARD=Loopback,DEV=0
The file to be generated has to be single channel and 16 bits: -c 1 --bits 16

WD expects two 1 minute audio wav files!
The generated single 115 seconds total audio wav file had to be split into two separate files with the correct name. The 2e file name has to be 1 minute later.
A 5 seconds dummy audio file is concatenated to the 55 seconds long 2e audio file.
The total audio time should be 120 seconds for JT9 (not sure why?)

```
while true; do
[[ $verbosity -ge 1 ]] && echo "$(date): waiting for the next even two minute" && [[ -f ${TEST_CONFIGS} ]] &&
source ${TEST_CONFIGS}
local start_time=$(sleep_until_next_even_minute)
local wav_file_name="${start_time}_${arg_rx_freq_hz}_usb.wav"
# PA0SIM next lines
local tmp_wav_file_name="tmp_${wav_file_name}_usb.old"
local dummy_wav_file_name="dummy_${wav_file_name}_usb.wav"
local copy_wav_file_name="copy_${wav_file_name}_usb.wav"
[[ $verbosity -ge 1 ]] && echo "$(date): starting a ${capture_secs} second capture from AUDIO device
${audio_device},${audio_subdevice} to '${wav_file_name}'"
##sox -q -t --type --channels 1 alsa hw:${audio_device},${audio_subdevice} --rate 12k ${wav_file_name} trim 0
${capture_secs} ${SOX_MIX_OPTIONS-}
sox -t alsa plughw:CARD=Loopback,DEV=1 --rate 12k -c 1 --bits 16 ${wav_file_name} trim 0 ${capture_secs}
${SOX_MIX_OPTIONS-}
#sox -t alsa hw:1,1 --rate 12k -c 1 --bits 16 ${wav_file_name} trim 0 ${capture_secs} ${SOX_MIX_OPTIONS-}
local sox_stats=$(sox ${wav_file_name} -n stats 2>&1)
local raw1_wav_file_name="${start_time}_${arg_rx_freq_hz}_usb.wav"
local raw2_wav_file_name="${start_time}_${arg_rx_freq_hz}_usb.wav"
local file_start_minute=$((1+${wav_file_name:12:1}))
raw2_wav_file_name="${raw2_wav_file_name:0:12}${file_start_minute}${raw2_wav_file_name:13}"
mv ${wav_file_name} ${tmp_wav_file_name}
sox ${tmp_wav_file_name} ${raw2_wav_file_name} trim 60 55
sox ${tmp_wav_file_name} ${raw1_wav_file_name} trim 0 60
# adding 5 seconds so total time is 120 seconds (copy of last 5 seconds volume scaled with 0.01)
sox -v 0.01 ${raw2_wav_file_name} ${dummy_wav_file_name} trim 50 5
cp ${raw2_wav_file_name} ${copy_wav_file_name}
rm ${raw2_wav_file_name}
sox ${copy_wav_file_name} ${dummy_wav_file_name} ${raw2_wav_file_name}
rm ${dummy_wav_file_name}
rm ${copy_wav_file_name}
rm ${tmp_wav_file_name}
# end PA0SIM
if [[ $verbosity -ge 1 ]] ; then
printf "$(date): stats for '${wav_file_name}':\n${sox_stats}\n"
fi
```

done

13 SPECTRUM SPREAD IN WSJT-X

In the standard wsjt-x application, you don't see the spectrum spread column.
If you want that (of course you want it,….) do the following:
Make a dir in the home dir called runwsjtx.
cd ~
cd ..
sudo mkdir runwsjtx

When starting in the /bin directory
WSJT-X crashes ones in a while and is not stable when decoding FST4W.
Seems to crash only when a FST4W signal is present/decoded.
Reported is:
Subprocess failed with exit code 2
Running: /usr/bin/jt9 -s WSJT-X -w 1 -m 3 -e /usr/bin -a /home/wsprdaemon/.local/share/WSJT-X -t /tmp/WSJT-X
At line 997 of file /build/wsjtx-92hGpk/wsjtx-2.5.4+repack/lib/fst4_decode.f90 (unit = 52)
Fortran runtime error: Cannot open file 'fort.52': Permission denied

14 WHAT TO DO FOR THE 40M BAND

Most important is that you change the frequency of the receiver, the RSP1a (or RSP2) to the 40m frequency. We do that in gnuradio in the block of the receiver. Change the center frequency in that block from 14095600 to 70386000 Hz. Kill the gnuradio flowgraph and execute it again. The effect of this is that gnuradio is now sending the wspr band of the 40m band to the audio sinks. Wsprdaemon and wsjtx don't know that this info is from the 40m band or where ever. So we have to inform wsprdaemon and wsjtx. In wsjtx it is easy, we go to the wsjtx screen and change the freq there to the 40m band. For wsprdaemon, we have to do the following:

- go to wsprdaemon.conf (cd ~/wsprdaemon)
- edit the file wsprdaemon.conf (nano wsprdaemon.conf)
- change in the "AUDIO_0…….line of the RECEIVERLIST the 20 to 40
- and the same in the Schedule line.
- Save it.

The 40m band spots are now received and reported to wsprnet.

Don't forget to change your antenna. It would be nice when it is also a 40m band antenna!

15 STARTING UP THE WSPR SPOTS REPORTER

To start up the wspr/ fts4w spot reporter, do the following:

- (re)start of the PC/Laptop
- Login with user wsprdaemon and password (for me UbuntuWD)
- we need the virtual channels, so:
  sudo modprobe snd-aloop
- startup gnuradio with
  gnuradio-companion (ignore the vocoder warning)
- you should see the flowgraph of gnuradio with the correct file name (rsp1a.grc)
- run the script: run-execute
- click it away from the desktop because we need the space. A second screen would be nice.
- Open a second terminal and do:
  wda to start wsprdaemon. The daemon is spawned, and the file wav-archive.d is searched for. But this file is not yet there. Ignore it.
- After a little while you see that the daemon is searching for spot files. They have to be created first ,…….by another daemon.
- Open a third terminal and do: wdln
- It reports whether spot files are found yet.
- Open a 4th terminal and do wds
- You see three activities posting, decoding and recording.
  And three daemons: a watchdog, an upload wsprnet, and an upload wsprdaemon
  All waiting for the action,……..getting the spots from the gnu script, which gets the spots from the RSP device.
- Go to the folder runwsjtx, cd ~, cd ../runwsjtx
- Do: /bin/wsjtx probably just wsjtx will also work.
- Wsjt-x will start up and you should see a lot of whispers coming into wsjt-x
- Now we want to see whether wspdaemon is reporting the whispers to wsprnet?
- Start up on a separate PC/Laptop a browser with the address wsprnet
  http://www.wsprnet.org/drupal/
  go to the menu DATABASE (right upper corner), and fill out the following parameters:
  - Band: 20m
  - Mode: ALL
  - Count: 50
  - Call: PE0MJX (this is my situation. You probably have put your call in the wsprdaemon.conf file)
  - Reporter: PE0MJX (you pick your call)
  - In last: hour
  - Scroll to the bottom, and update
  - Here you are, all the reported spots !!!!
  - In the last column you see W-2 or F-120. Guess what, W-2 stands for spots of wspr (2 minutes) and F-120 is FST4W (120 seconds)
  - Lat check: set wsjt-x to FST4 mode and erase the screen. After a while you could be seeing FST4W spots,….with in the last column the spectrum spread,…..

Yessss, we did it. Well done

It was quite a challenge but the result is nice,……and we learned a lot !!!

Please let us know if you got this working,…..use the forum:

https://groups.io/g/wsprdaemon/topics

pick the topic: wsprdaemon+GNU+SDRPlay devices.

PA0SIM/PE0MJX

Aug 2023
