SHARI and Dire Wolf

SHARI is a compact low power VHF or UHF transceiver packaged with a soundcard. It was originally intended for an Allstar Node but can be used with many other applications.

It has an antenna connector and two USB connectors, spaced so they can plug into a Raspberry Pi:

- CM119A USB audio adapter equivalent of the DINAH interface with provides transceiver audio and PTT control.
- USB to serial converter to configure the SA818 transceiver.

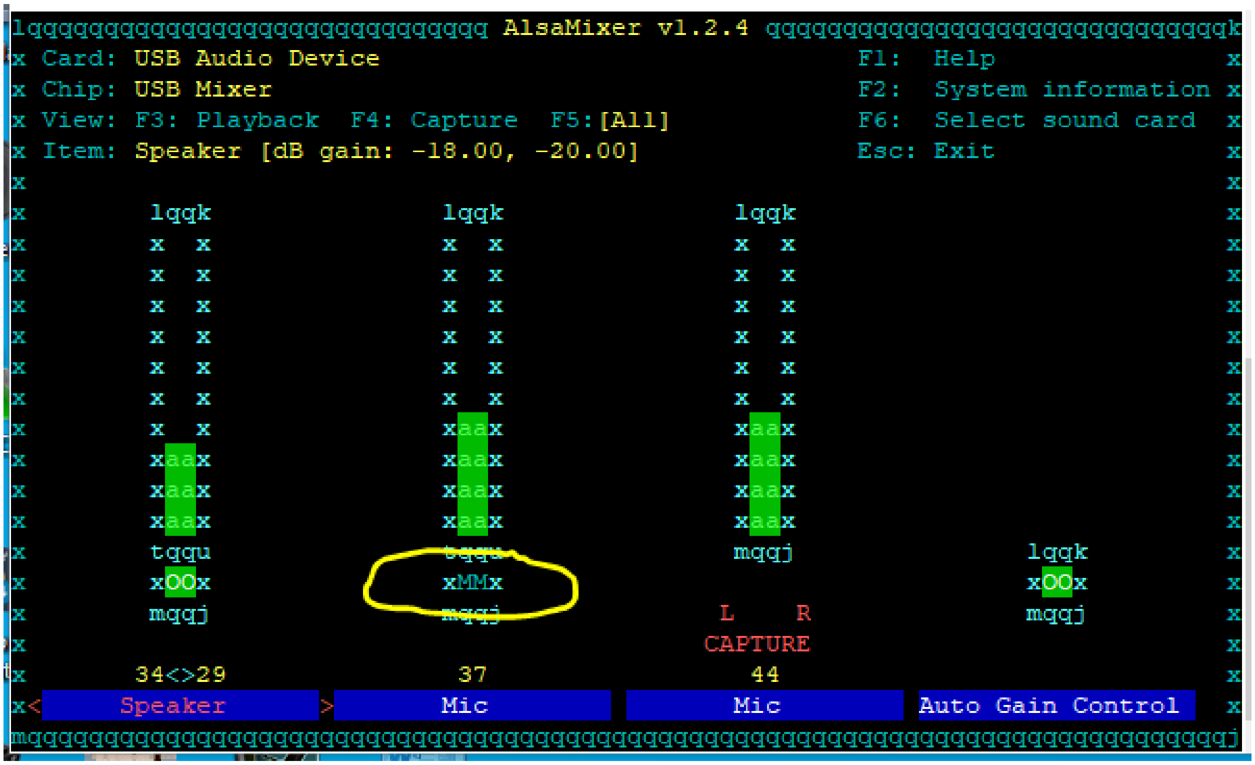
Identify audio device & unmute microphone

First, run “**arecord -l**” to get a list of audio capture devices. The result should look something like this:

```
**** List of CAPTURE Hardware Devices ****
card 2: Device [USB Audio Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

The important part is the card number, 2 in this case. We will need in a couple places later.

Run “**alsamixer -c 2 -V all**” to set the volume controls.

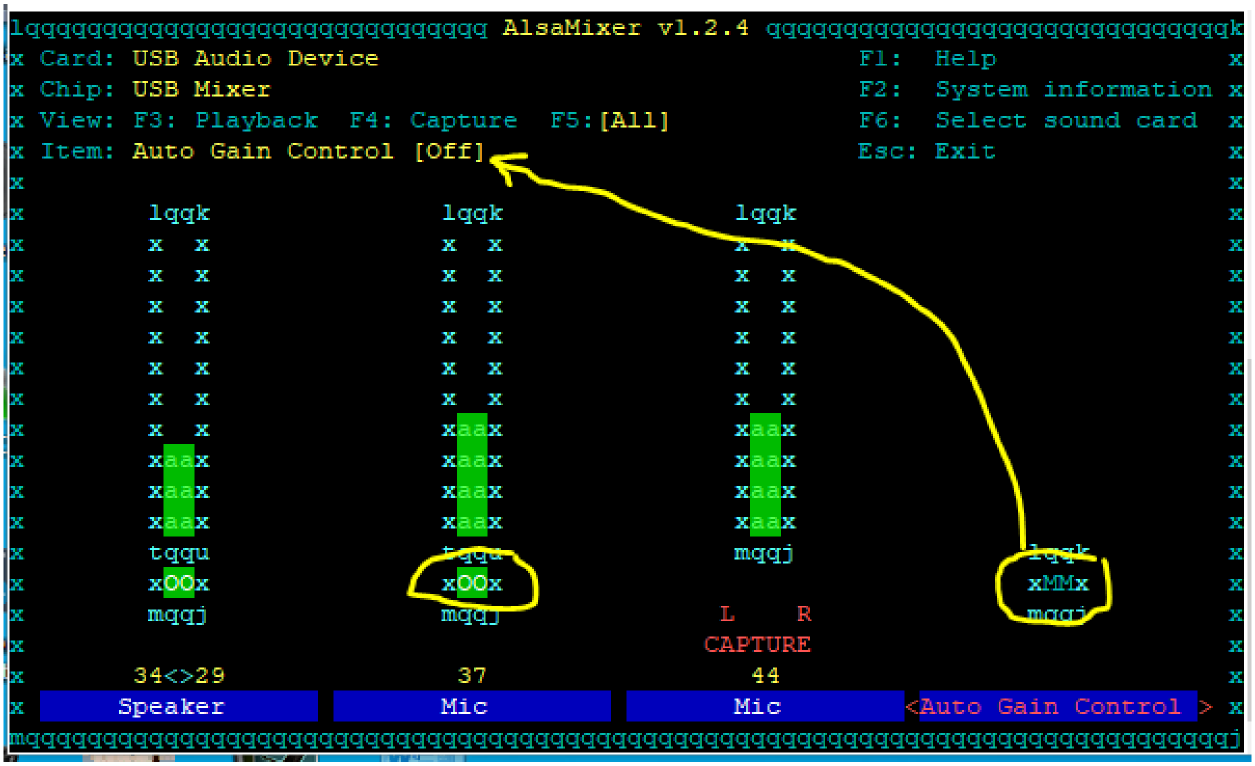


```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq AlsaMixer v1.2.4 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x Card: USB Audio Device                               F1:  Help               x
x Chip: USB Mixer                                      F2:  System information x
x View: F3: Playback  F4: Capture  F5:[All]            F6:  Select sound card  x
x Item: Speaker [dB gain: -18.00, -20.00]              Esc: Exit               x
x                                                                              x
x        lqqk               lqqk               lqqk                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        tqqu               tqqu               mqqj               lqqk         x
x        xOOx               xMMx                                  xOOx         x
x        mqqj               mqqj              L    R              mqqj         x
x                                            CAPTURE                           x
x       34<>29               37                 44                             x
x<     Speaker      >       Mic                Mic         Auto Gain Control   x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

The first time a USB audio device is seen, the microphone input is muted, as indicated by the "MM". Those not aware of this behavior can waste a lot of time wondering why there is no receive audio.

Use the right arrow to highlight "Mic" then press "m" to unmute.

I have found better results when Auto Gain Control is off.   Use the right arrow to highlight it, the press "m" to disable it.



```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq AlsaMixer v1.2.4 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x Card: USB Audio Device                               F1:  Help               x
x Chip: USB Mixer                                      F2:  System information x
x View: F3: Playback  F4: Capture  F5:[All]            F6:  Select sound card  x
x Item: Auto Gain Control [Off]                        Esc: Exit               x
x                                                                              x
x        lqqk               lqqk               lqqk                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               x  x               x  x                            x
x        x  x               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        xaax               xaax               xaax                            x
x        tqqu               tqqu               mqqj               lqqk         x
x        xOOx               xOOx                                  xMMx         x
x        mqqj               mqqj              L    R              mqqj         x
x                                            CAPTURE                           x
x       34<>29               37                 44                             x
x      Speaker             Mic                Mic          <Auto Gain Control > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

We need to add these to the direwolf.conf file:

```
ADEVICE plughw:2,0
CHANNEL 0
PTT cm108
```

Set Radio Properties

If you don't have any other USB serial port devices plugged in, the SHARI control device will probably show up as /dev/ttyUSB0.

There are two ways to go about doing this:

1. Learn the command formats, and talk directly to the SA818 with a terminal emulator.
2. Use an application which asks you questions and sends the commands.

Let's start with the second option. Download the most recent stable version of **SA818-prog** from https://groups.io/g/shari/files and put it on your Raspberry Pi. Make it executable by typing "chmod +x SA818-prog"

We need to make a little change. The first line contains “#!/usr/bin/python2”. Python 2 is no longer supported after the beginning of 2020. It is not included in the initial installation of Raspberry Pi OS, bullseye. Rather than install Python 2, just change 2 to 3 in the first line.

Run it:

```
$ ./SA818-prog
No previous programming data available.

----------------------------------------------------

SA818-prog, Version 2.01
Programing SHARI PiXX / SHARI PiHat / SA818(U/V) Module
Programming Device name:
    /dev/ttyUSB0

----------------------------------------------------

Testing serial communications
Serial communications with the SA818 module are OK

What are you programming?
Enter 1,2 or 3 where SHARI PiXX=1, SHARI PiHat=2, SA818 Module=3: 1

      Programming a SHARI PiXX
      Wide channel spacing

Enter band (VHF=1, UHF=2): 1
      You chose VHF

Enter transmit frequency in MHz (xxx.xxxx): 144.3900
      The transmit frequency is 144.3900 MHz

Enter receive frequency in MHz (xxx.xxxx): 144.3900
      The receive frequency is 144.3900 MHz

Do you want to use a sub audible tone? (0 = No, 1 = CTCSS, 2= DCS): 0
      You chose No SubAudible Tone

Enter Squelch Value (0-8): 0
      Squelch is set to 0

Enter Volume (1-8): 5
      Volume is set to 5

Enable Pre/De-Emphasis (y/[n]): n
      Pre/De-Emphasis is not enabled
```

```
Enable High Pass Filter (y/[n]): n
        High Pass Filter is not enabled

Enable Low Pass Filter (y/[n]): n
        Low Pass Filter is not enabled


Verify:
-------------------------------------------------------
     Channel Spacing: 1
        Tx Frequency: 144.3900
        Rx Frequency: 144.3900
       Tx CTCSS code: 0000
       Rx CTCSS code: 0000
       Reverse Burst: n
       Squelch Value: 0
        Volume Value: 5
 PreEmphasis Enabled: n
   High Pass Enabled: n
    Low Pass Enabled: n
-------------------------------------------------------

Is this correct ([y]/n) ?y
Sending Frequency, Sub Audible Tone, and Squelch Information...
Frequency, Sub Audible Tone, and Squelch information correct
Setting Volume - 4
Setting Reverse Burst
Setting Filters

Programming Successful

Configuration log written to /home/pi/.SA818.log
```

The SA818 has a non-volatile configuration memory so the settings should be remembered when the power is shut off. NEED TO VERIFY.

Run Dire Wolf and see if we can receive

For anyone who might have skipped ahead, we previously mentioned that the direwolf.conf file must contain:

```
ADEVICE plughw:2,0
CHANNEL 0
PTT CM108
```

Run it.

```
$ direwolf
Dire Wolf DEVELOPMENT version 1.7 E (Jul 11 2022)
Includes optional support for:  cm108-ptt

Reading config file direwolf.conf
Audio device for both receive and transmit: plughw:2,0  (channel 0)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, A+, 44100 sample rate / 3.
Using /dev/hidraw0 GPIO 3 for channel 0 PTT control.
Ready to accept AGW client application 0 on port 8000 ...
Ready to accept KISS TCP client application 0 on port 8001 ...

Digipeater WB2OSZ-5 audio level = 61(13/17)    0000_____
[0.1] K2VUD-1>APWW11,WA1PLE-13,WB2OSZ-5*:@173552h4202.57N/07126.83W-
PHG/060APRS-IS for Win32<0x0d>
Position with time, House, APRSISCE win32 version
N 42 02.5700, W 071 26.8300, 111 km/h (69 MPH)
APRS-IS for Win32

Digipeater WB2OSZ-5 audio level = 61(12/17)    00000____
[0.2] KC1DDH-9>TRQU4Q,W1MRA,WB2OSZ-5*:`c7`m*0j/`146.520MHz 144.390 PL100_%<0x0d>
"100" in comment looks like it might be a CTCSS tone in non-standard format.
For most systems to recognize it, use exactly this form "T100" at near beginning of comment,
after any frequency.
MIC-E, JEEP, Yaesu FTM-400DR, Off Duty
N 42 15.4100, W 071 27.6800, 20 km/h (13 MPH), course 20, 146.520 MHz, PL 100.0
 144.390 PL100

WB2OSZ-5 audio level = 61(12/17)    0000_____
[0.1] WB2OSZ-5>APDW17:!4237.14NS07120.83W#PHG7140 Did you know that APRS comments
and messages can contain UTF-8 characters?   アマチュア無線
Position, SSn-N digipeater (includes WIDEn-N), DireWolf, WB2OSZ, 49 W height=20 4dBi omni
N 42 37.1400, W 071 20.8300
 Did you know that APRS comments and messages can contain UTF-8 characters?   アマチュア
無線

Digipeater WB2OSZ-5 audio level = 61(12/17)    00000____
[0.2] K1FFK>APMI03,AB1OC-10,WB2OSZ-
5*:@071247z4238.23N/07310.10W_000/000g000t066r000p000P000h75b10217
Weather Report, WEATHER Station (blue), SQ3PLX http://microsat.com.pl/
```

```
N 42 38.2300, W 073 10.1000
wind 0.0 mph, direction 0, gust 0, temperature 66, rain 0.00 in last hour, rain 0.00 in last 24
hours, rain 0.00 since midnight, humidity 75, barometer 30.17, ""
```

It works! Without much of an antenna, I'm only hearing my digipeater and no other stations directly.

I should mention the audio level. Ideally you should try to keep the average somewhere in the range of 10 to 100. For simplicity, I sometimes say around 50 but it is not critical. If it is too high, there will be a warning before the audio A/D converter begins to hit its limit.

This is a combination of the SA818 receive volume setting and the soundcard capture settings.

Just remember that a new USB audio adapter will have the microphone muted initially.

Now try transmitting

First make sure that a properly matched antenna for the frequency used.

We already configured the PTT control, so we just need to add a callsign and a position beacon command:

```
MYCALL WB2OSZ-14

PBEACON   LAT=42^37.14  LONG=-71^20.83
```

(Obviously, use your own callsign and location instead of copying this.)

MORE HERE

We are transmitting. That wasn't so hard.   **We need to do one more thing.**

Setting Transmit Audio Level

If set to low, you won’t be heard. If set to high, the signal will get limited, causing distortion, adjacent channel splatter, and less reliable decodes. If you have a deviation meter, somewhere around 3 or 3.5 kHz peak deviation is good. I don’t have fancy test equipment, so I just listen with another Dire Wolf setup, and compare audio levels with other stations. The Kenwood TM-D710 has a built in TNC, factory adjusted for proper deviation. Matching that would probably be a good setting.

Other Rambling

When using digital data, it is best to leave squelch off. When squelch is active:

- You might miss weak signals that the modem is able to decode.
- There is a delay at start of packet causing the beginning to get lost. This would be a problem if the sender sets TXDELAY too short.
- The audio can have a huge voltage jump (sounding like a click) when turning on and off.

You might be wondering about the settings:

- PreEmphasis Enabled: n
- High Pass Enabled: n
- Low Pass Enabled: n

FM voice transmitters usually use pre-emphasis. This is basically a high pass filter that increases the amplitude of higher frequencies.

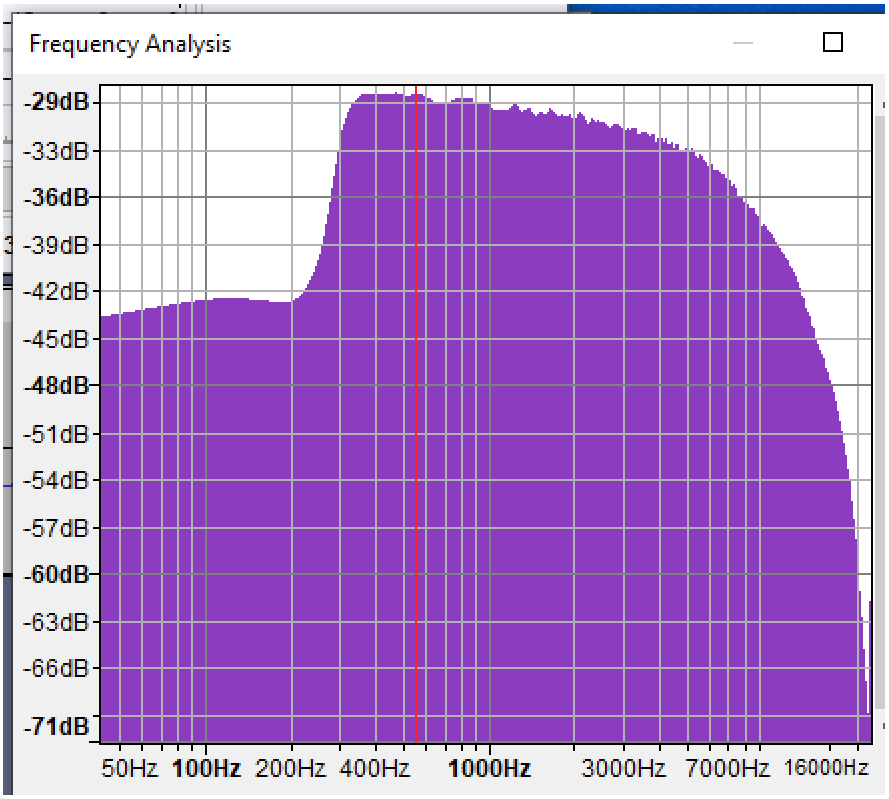
FM voice receivers usually use de-emphasis. This is basically a low pass filter which decreases the amplitude of higher frequencies.

In theory, they should cancel out and reduce the amount of noise. This might be beneficial for voice but distorting the signal - twice - is not good for digital data. The most commonly used 1200 bps AFSK can survive a lot of abuse but when data rates get higher, modems are not as forgiving about distortion.

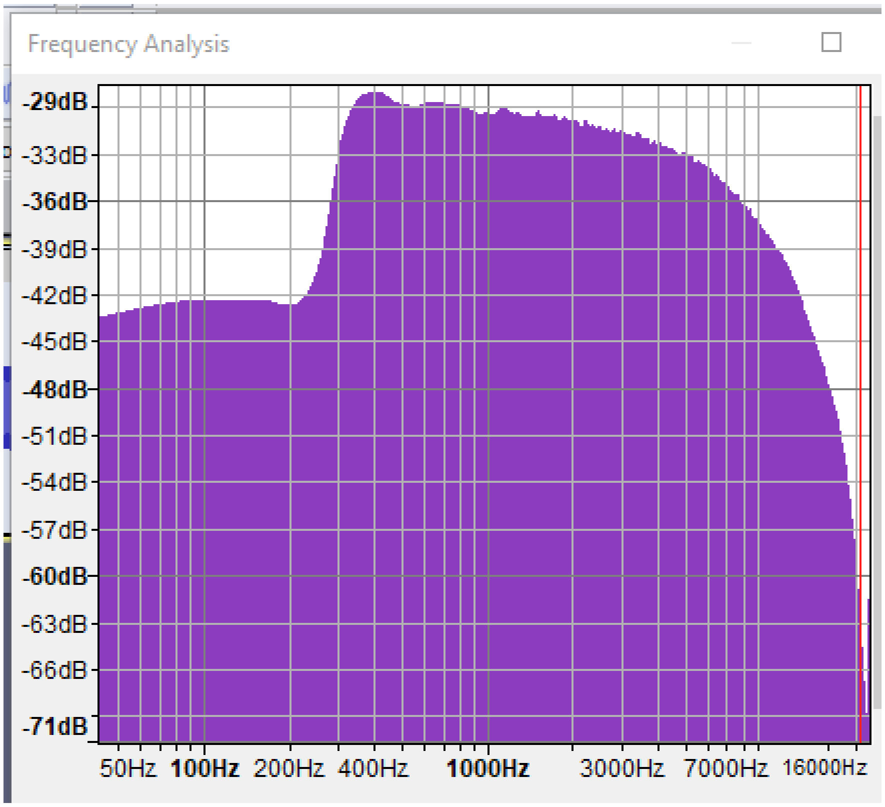
The NiceRF SA818 Programming Manual doesn't say anything about the characteristics of the high and low pass filters. Let's try a little test.   We will just record some random noise and take a look at the spectrum using Audacity.

```
arecord -D plughw:2,0 -t wav  -r 44100 -d 10 -f S16_LE recording1.wav
```

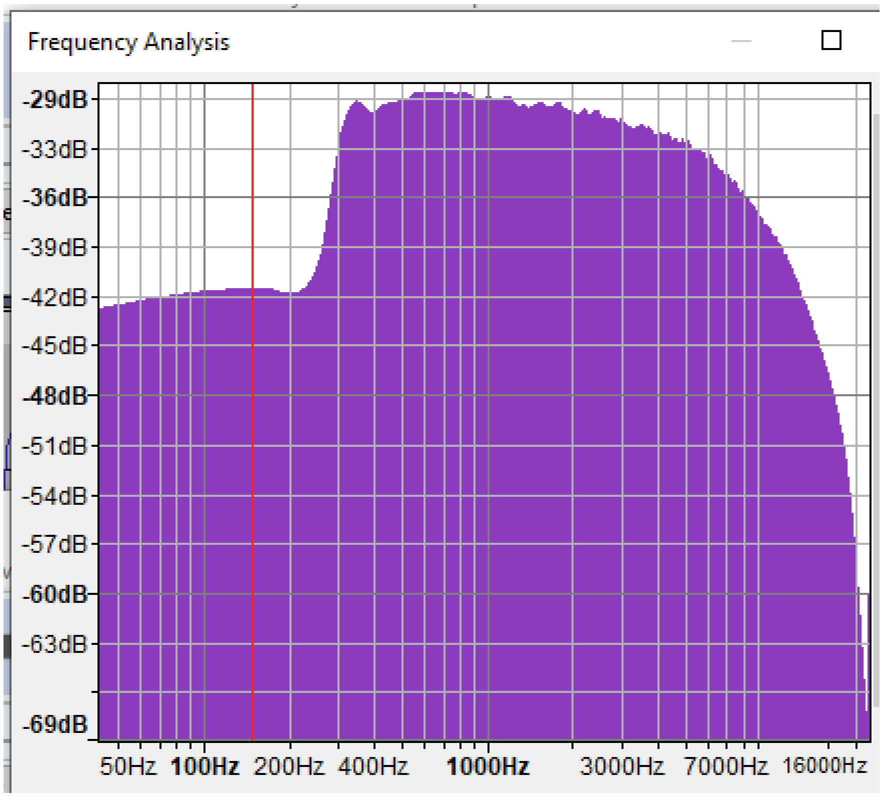
Here is the spectrum we see when all the filtering options are turned off. There is a very sharp cutoff below around 300 Hz. This would prevent us from hearing any CTCSS tones. I would have expected to see that only if the high pass filter was enabled.



Maybe the control value is backwards. Let's enable the high pass filter and see if that turns it off. It looks essentially the same. So the highpass filter setting seems to be always on. Is it a bug in SA818-prog or the SA818?



Now add the lowpass filter in addition to the highpass filter. I would expect this to attenuate frequencies over about 3 kHz which are not needed for voice. **It seems to make no difference.** . Is it a bug in SA818-prog, or the SA818, or my misunderstanding?



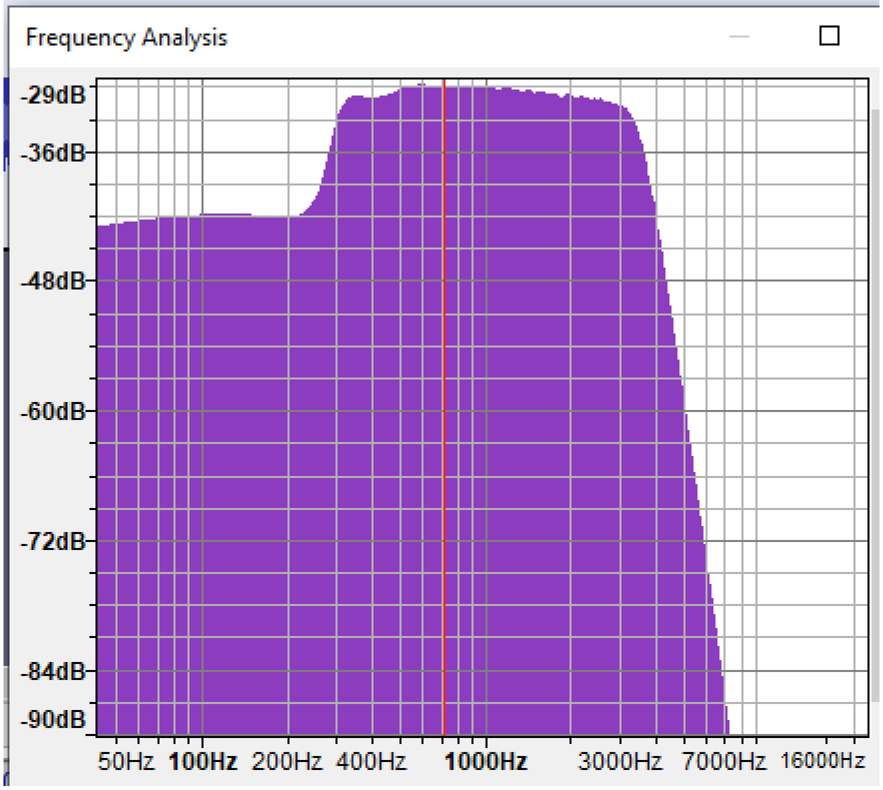
I added some debug code to see what is being sent to the SA818,

```
AT+SETFILTER=1,0,0
```

That agrees with the programming guide and success status is being returned.

Does the de-emphasis control have any impact?

PreEmphasis Enabled: y
High Pass Enabled: n
Low Pass Enabled: n



That is not what I expected at all. De-emphasis should decrease the signal by 6 dB per octave. That means half the amplitude when the frequency is doubled. Instead, we appear to have the highpass and lowpass filters. . Is it a bug in SA818-prog or the SA818?

The command sent was

```
AT+SETFILTER=0,1,1
```

so that looks right for enabling de-emphasis but bypassing the high and low pass filters. “1” means bypass so the look is intuitively backwards.

Before moving along, I did have one other thought. The programming guide shows a space after each comma. Could that make a difference?

Now try turning them all off.

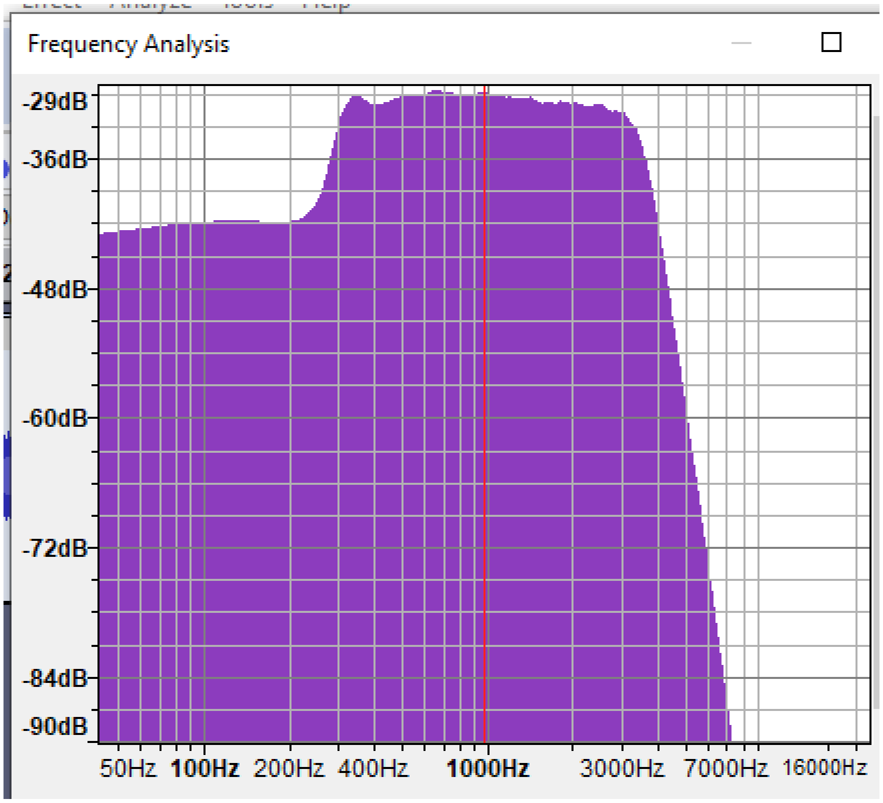
PreEmphasis Enabled: n
High Pass Enabled: n
Low Pass Enabled: n

No - that produces
JWL DEBUG command = AT+SETFILTER=1, 1, 1

```
   Error, invalid information (+DMOSETFILTER:1)...
 Command Sent:
   AT+SETFILTER=1, 1, 1
```

Spmeone with more patience can test all eight combinations. For our final test, let’s say yes to all.

PreEmphasis Enabled: y
High Pass Enabled: y
Low Pass Enabled: y



Experimentation reveals:

- The first parameter controls low pass filter, rather than the documented de-emphasis.
- The other two parameters don’t have any effect.

To be continued…

Resources

HB9BLA Wireless: We use Baofeng's "heart" for Our Projects (SA818, DRA818)

https://www.youtube.com/watch?v=CIeT4SWneeo