CONSIDERATIONS ON CALIBRATION AND FILTERS IN uBITX

Thanks to Evan AC9TU, Jerry KE7ER, Ron and others.

Doubts periodically arise about the calibration procedure of the uBitx radio from https://www.hfsignals.com

As a result of this, a short time ago, I raised some questions on the forum about the operation of the radio, not because I had doubts about its design, but because I did not fully understand the principles on which it was based. It is obvious that the design is good enough for the simple reason that the radio works. And it works fine.

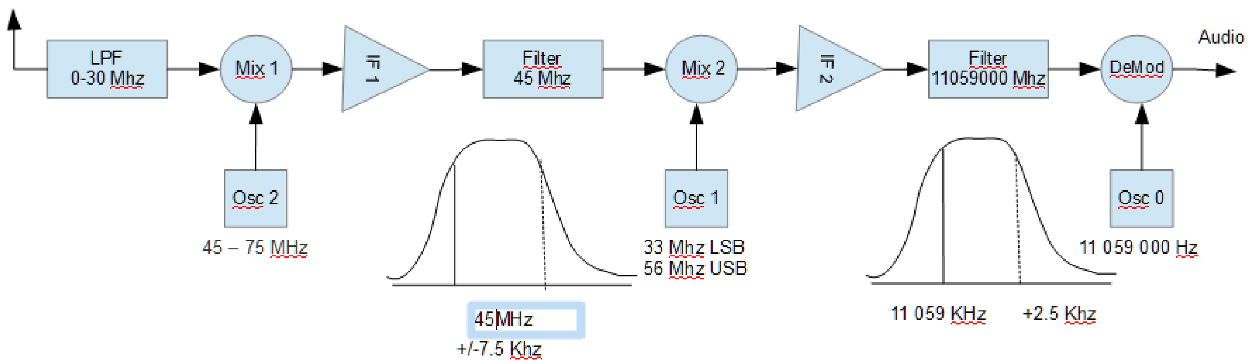
The original discussion can be found here:
https://groups.io/g/BITX20/topic/calibration_filters/89337641

The questions raised could be summarized as follows:

1. Is the spectrum of the sidebands well centered in the passband of the filters?
2. How to correctly adjust the frequencies of oscillators OSC0, OSC1 and OSC2?
3. Is there an instrument-based procedure that allows correct calibration of the radio or is the human ear required?

Antecedentes

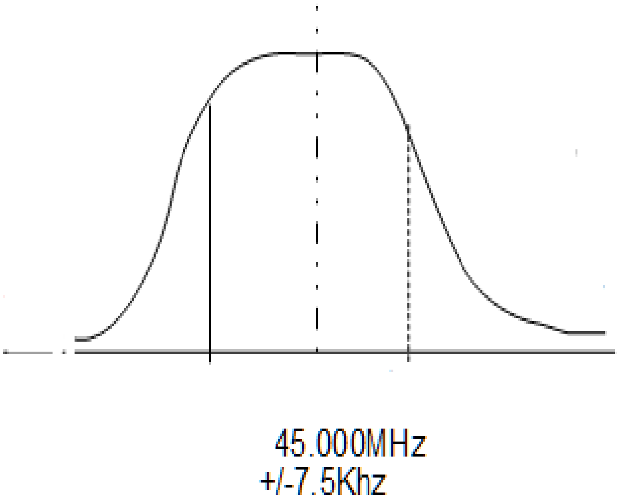
The diagram of the receiver in versions 5 and 6 is shown here. For previous versions the discussion is similar modifying the frequencies of the filters and oscillators.



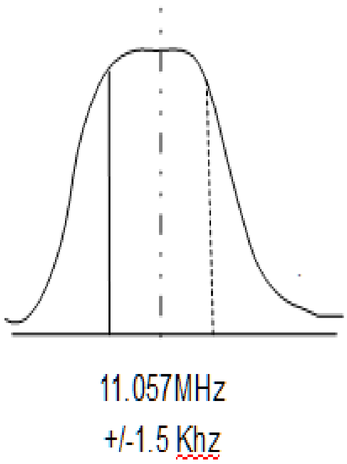
It seems accepted that

- The bandwidth for 3db of the 45MHz filter is about 15KHz.
- Typically the 45MHz filter appears to be centered at 44.995 Mhz.

Thus, the typical response of the 45MHz filter would be



It also seems accepted that the 11059KHz filter would have a bandwidth of about 3KHz and the center frequency would be about 2KHz lower, at 11.057KHz, due to the series resonance of the circuit. The upper drop in response would be steeper than the lower.

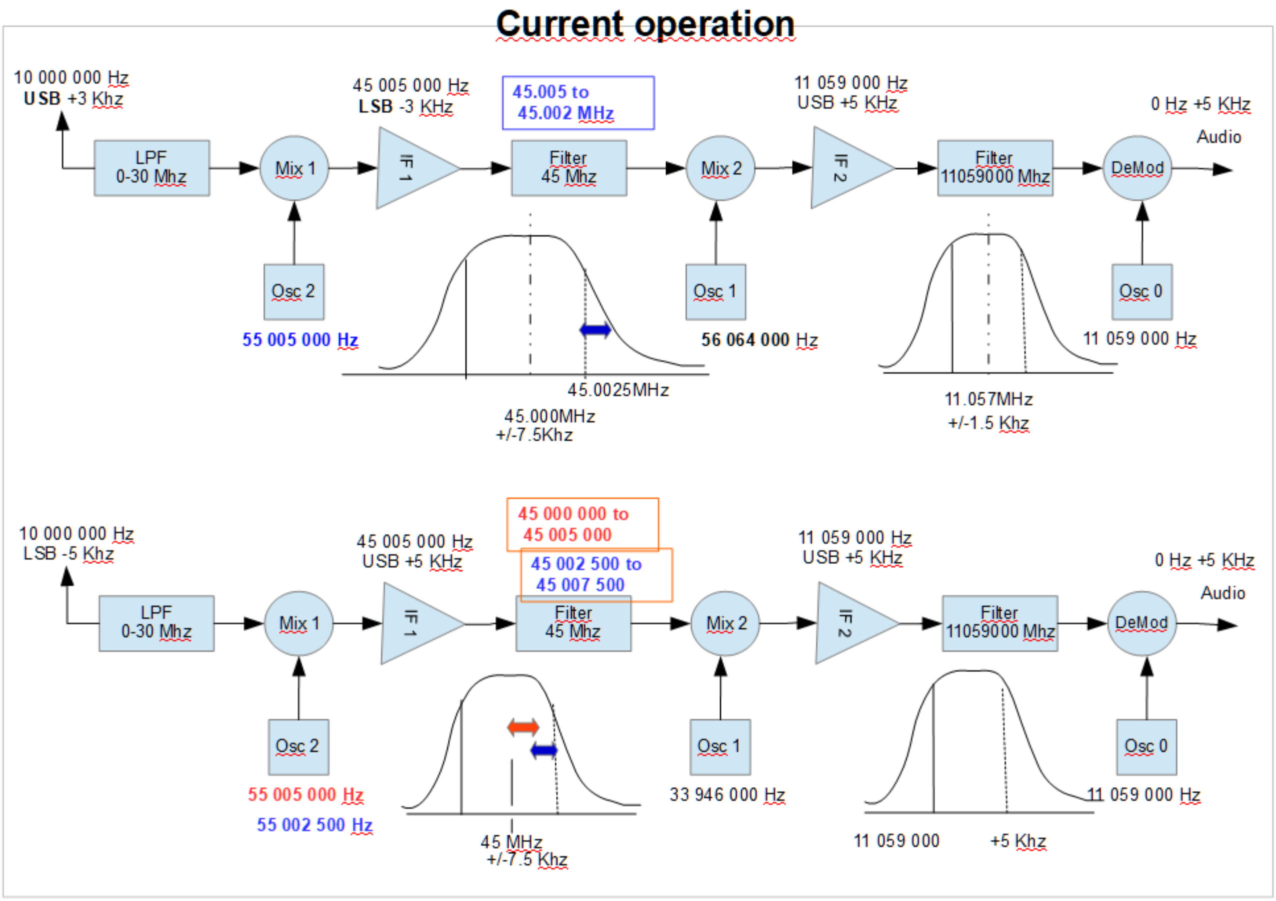


These would be the "theoretical" answers, without taking into account the tolerance and variation in each crystal unit.

We will first use these theoretical values to raise the terms of the discussion. Later it will be necessary to take into account the real variations for a correct calibration.

We also start from the values of OSC0, OSC1 and OSC2 used in the version 6 software that can be found here: https://github.com/afarhan/ubitxv6

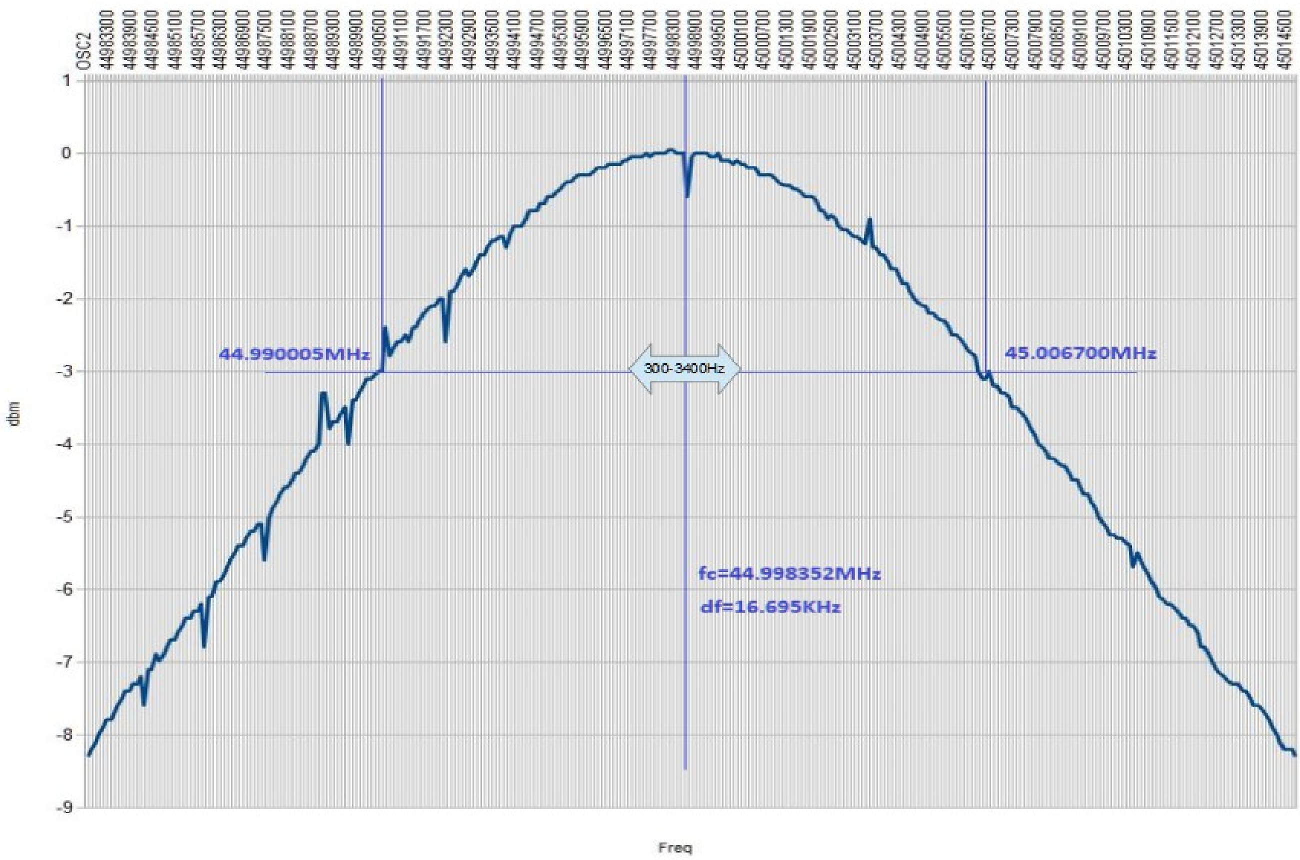
The diagram would therefore look like this



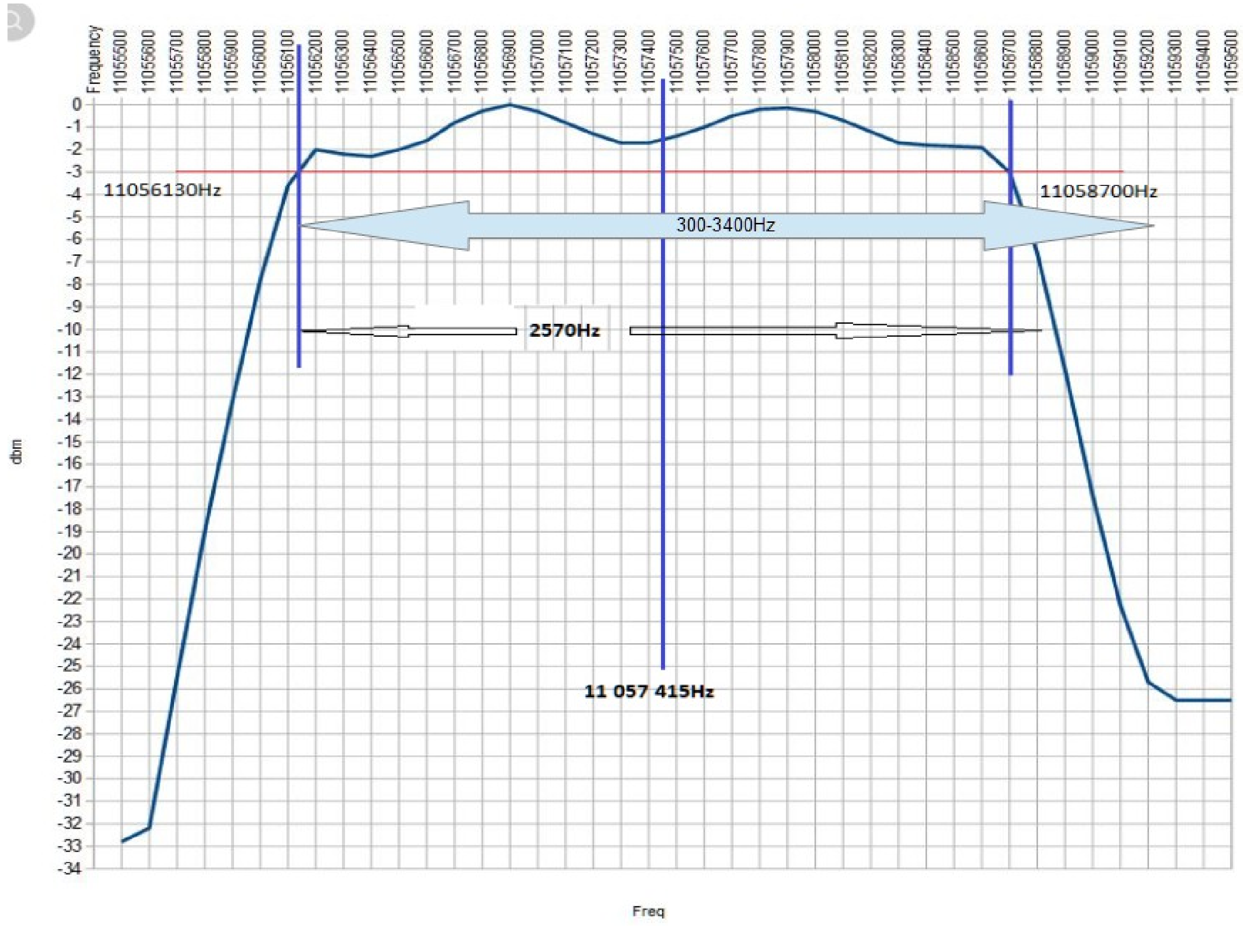
Next, I have plotted the actual response curves of both filters, 45MHz and 11.059 Mhz. We assume that the shape of the response curve is the same for all uBitx units and that it will only be shifted up or down depending on the quartz crystals of each unit.

45Mhz filter

45MHz filter response. Fc=44.998352MHz, dF=16.695KHz.

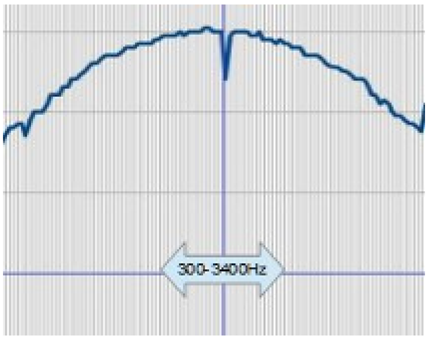


11.059 Mhz filter

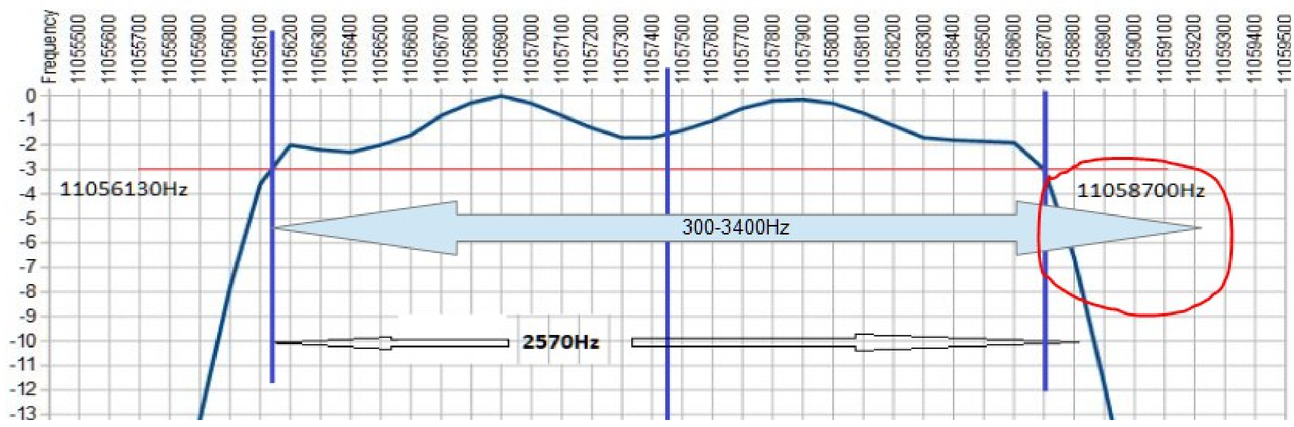


Suppose that the band that we want to transmit in a sideband goes from 300Hz to 3400Hz (voice band usually used in telephony), with a width of 3100Hz.

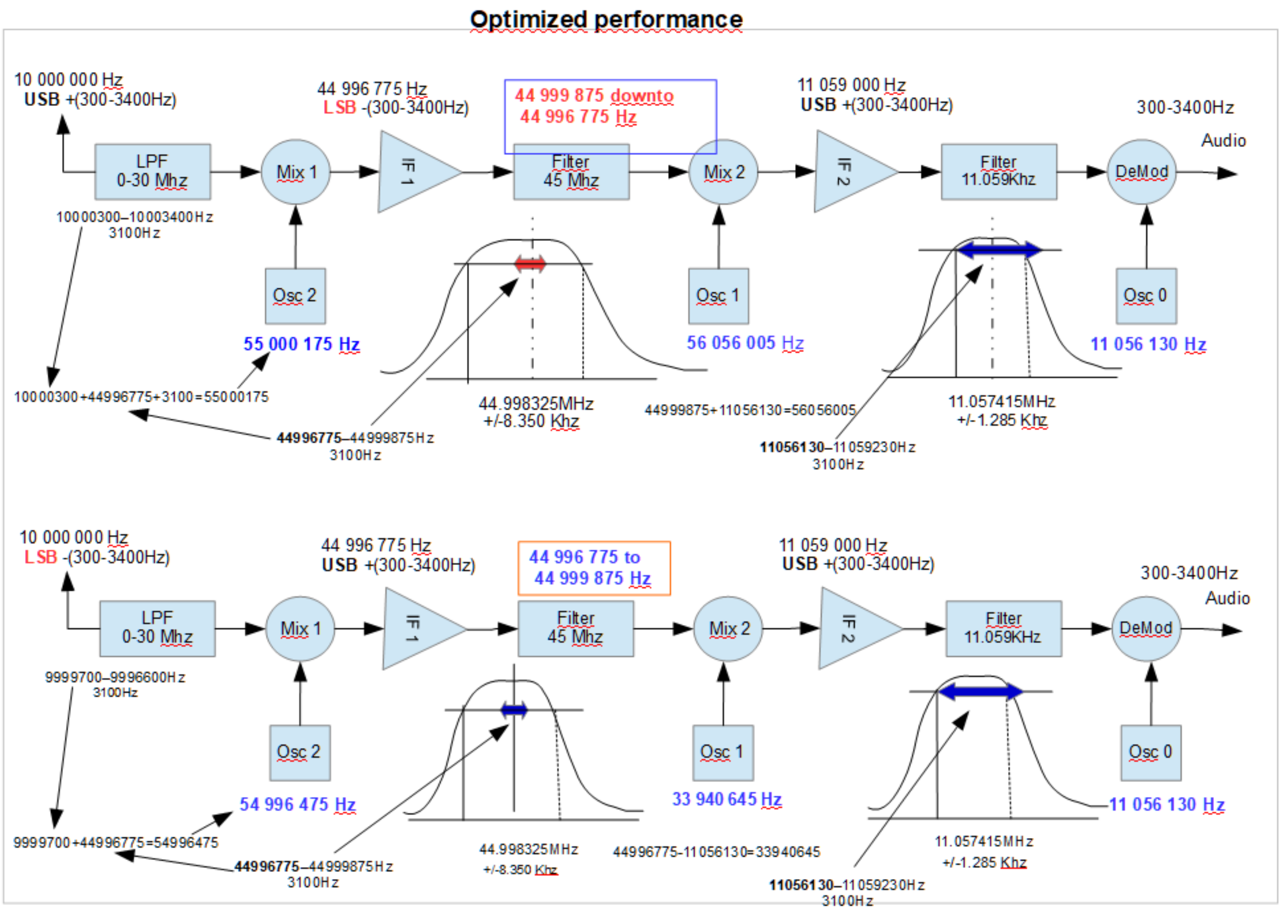
In the first filter, of 45 Mhz, we should place this spectrum in the center of the pass band of the filter



However, in the second filter, since the sideband spectrum is larger than the passband, it seems more correct to match the lower limit of the spectrum to the lower limit of the passband. In this case we would lose the upper part of the spectrum, above 2870 Hz.



To achieve this, it is necessary to modify the software so that the frequencies of the OSC2, OSC1 and OSC0 clocks are those shown in the following diagram, in the event that the received frequency is 10.0 MHZ in USB and LSB.



Generalizing the values for any received frequency and sideband, being

Fr : received frequency
Fc1 : Filter 1 center frequency (45 MHZ)
Fc2 : Filter 2 center frequency (11.059 Mhz)

Thus, in the **case of USB,** the calculation of the frequencies of the oscillators would be:

OSC2 = Fr + 300Hz + Fc1 + 3100Hz/2

Ejemplo: *Osc2 = 10000000 + 300 + 44998325 + 1550 = 55 000 175 Hz*

OSC1 = Fc1 + 3100Hz/2 + Fc2 – 1285Hz

Ejemplo: *Osc1 = 44998325 + 1550 + 11057415 – 1285 = 56 056 005 Hz*

OSC0 = Fc2 – 1285

Ejemplo: *Osc0 = 11057415 – 1285 = 11 056 130 Hz*

For the lower sideband LSB

OSC2 = Fr – 300 + Fc1 – 3100Hz/2

Ejemplo: *Osc2 = 10000000 -300 + 44998325 - 1550Hz = 54 996 475 Hz*

Osc1 = Fc1 – 3100Hz/2 – Fc2 + 1285

Ejemplo: *Osc2 = 44998325 – 1550 – 11057415 + 1285 = 33 940 645 Hz*

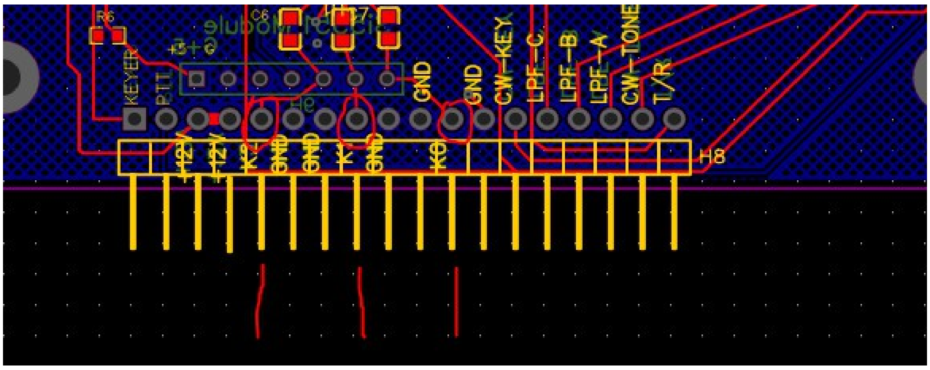
Osc0 = FC2 – 1285

Ejemplo: *Osc0 = 11057415 – 1285 = 11 056 130 Hz*

Si5351 main oscillator calibration

Before carrying out the entire filter measurement process, it is necessary to calibrate the Si5351 generator. We will use a reliable frequency counter and measure any one of the three generated clock frequencies CLK0, CLK1 and CLK2.

Whatever the version of Raduino, the original or any of the Raduino32, the measurement points are these connector pins between the Raduino/32 and the uBitx main board.



K2, K1 and K0 from left to right, seen from the front.

We put the uBitx in VFO Calibration mod and adjust the value until the frequency measured at point K0 (OSC0) is equal to the value that the BFO is calibrated to (approx. 11.056000MHz). It does not matter if it is the correct value, this step will be done later.

With this adjustment we will have compensated for the deviations of the 25 MHZ crystal of the Si5351 oscillator and we will be able to continue the process.

Calculation of the central frequencies of the filters

Obviously, to apply this procedure it is necessary to find the central frequencies of the filters **Fc1** and **Fc2**. A frequency generator and a level meter are required.

Depending on the equipment available, the procedure may vary. I had a Juntek JDS2800 generator and an Anritsu ML424A level meter.

Filter 2, 11.059 Mhz

With the radio turned off, inject frequencies between 11,055 Mhz and 11,059 Mhz with the generator, sweeping every 100 Hz. Save the data and plot the curve on an Excel sheet or on a grid sheet. With the curve drawn, look for the lower and upper cutoff points with levels of -3db with respect to the maximum level that we will consider as 0db. With the lower and upper limit frequencies we calculate the arithmetic mean and we will have the central frequency **Fc2**.
This central frequency should be between 11.056 MHz and 11.058 MHz, according to what is known for this type of filter.

So, in this case, we have:

F lower = 11.056130 Mhz
F higher = 11.058700 Mhz

Fc2 = central F = (11.056130 + 11.058700) / 2 = **11.057415 MHz**

Filter 1, 45 Mhz

In my case, since the ML424A level meter does not measure well above 20 Mhz, I had to resort to a trick to measure the response curve of the 45MHz filter.

(Valid for the Raduino32 board, but in the original Raduino it would be done in a similar way)

With the uBitx radio on and the antenna disconnected, adjust the BFO value (OSC0) to the center frequency of filter 2 to achieve the maximum response, in this case 11.057415 Mhz.

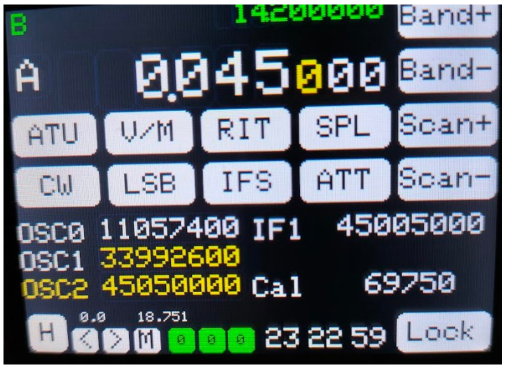
Make a temporary modification in the software, in the function ***void setFrequency(unsigned long f)*** *roughly on the line 340* del fichero *ubitx32_20.ino.*

```
341  void setFrequency(unsigned long f){
342    setTXFilters(f);
343    //alternative to reduce the intermod spur
344    IF1=conf.firstIF;
345    if (conf.isUSB==1)
346      {
347      //OSC1 = IF1 + conf.usbCarrier + f;
348      OSC1 = IF1 + conf.usbCarrier;
349      if (conf.cwMode) { OSC2 = IF1 + f + conf.sideTone; }
350      else             { OSC2 = IF1 + f; }
351      }
352    else
353      {        // LSB
354      //OSC1 = IF1 - conf.usbCarrier + f;
355      OSC1 = IF1 - conf.usbCarrier;
356      if (conf.cwMode) { OSC2 = IF1 + f + conf.sideTone;  }
357      else             { OSC2 = IF1 + f;     }
358      }
359    si5351bx_setfreq(2, OSC2);
360    si5351bx_setfreq(1, OSC1);
```

Comment lines 348 and 355 and uncomment lines 347 and 354 so that OSC1 =IF1 + usbCarrier + f.

Compile and load in Raduino32.

By pressing the “H” key go to screen mode 3, where the frequencies of the 3 clocks OSC0, OSC1 and OSC2 are shown.



We will sweep with the dial control the frequency that makes OSC2 go from 44.950000MHz to 45.050000 Mhz. We will first do a sweep every 1000 Hz to draw a rough curve and then every 100 Hz to get a more precise curve. Ignore what the main frequency indicates, only the values of OSC0 and OSC2.