SoftRock v9.0 RX and VHF Converters - v15.10 & v15.11 firmware set-up

The following are the Yahoo Group postings in the order I originally made them, therefore some information will change in later ones, as I have learned a lot as I am going along. There are also a few minor corrections from what was posted. Additional notes, not posted are added towards the end.

Important:

27th July 2009, bug found in v15.10 affecting negative offsets needed for some programs, new v15.11 created and tested. See information later in this document regarding this.

Bob G8VOI

Issue 1.2 - 28th July 2009

Message 35151 - Posted 22nd July 2009

V15.10 Firmware set-up and general description of converter operation

Like a lot of you, I am keen to get my hands on these converters to start playing!

For those who were quick off the mark, I guess the kits will be dropping through your mail boxes shortly.

All of the following assumes you are using Fred PE0FKO's v15.10 firmware.

The one essential piece of information you need to use either the 6m or 2m converter with the v9.0 RX is the firmware LO multiplication factor:

50MHz converter = 4/3 entered as 1.333333333 144MHz converter = 4/5 entered as 0.8

To set up the firmware at present you will need to use Winrad and Fred PE0FKO's supporting dll file. Hopefully soon his stand-alone program should be available, which will simplify matters.

You need to select the 'LO' tab, and enter the multiplication factor in the 'LO Total' 'Multiply' box at the top of the page.

There is a slight anomaly due to the compiler used to create the code, that Fred is aware of and looking into, it produces an error if either a leading '0' or decimal point is entered first. This can be simply overcome by entering 8 and use the cursor keys to allow the zero and decimal point to be entered ahead of it.

For those interested in how the converter works, I will try to explain and elaborate on how the LO multiplication factors were arrived at. It did take me a long time to get it clear in my own head, particularly the concept of the variable frequency LO.

It is an extremely clever piece of design by Jan G0BBL, and certainly a case of 'thinking outside the box'.

To recap, for anyone not familiar with the conventional type VHF converters used to allow say the 2m band to be received on a HF receiver.

The incoming signal 144MHz is fed to a mixer, which is provided usually with a fixed frequency crystal local oscillator, typically 116MHz.

The mixer output provides both the difference (144 - 116 = 28MHz) and the sum (144 + 116MHz = 230MHz). Only the difference is required, so filtering is needed to remove the sum (image frequency).

Tuning the IF frequency between 28 - 30MHz allows you to listen to the 144 - 146MHz band segment (EU 2m allocation).

The first part of the v9.0 converter works in exactly the same way, the Si570 is used to generate the required LO frequency to the mixer U1. If you think of it in terms of the Si570 generating a 'fixed' frequency at any one instant in time, the explanation of the 'simple' converter above fits.

To use the converter with the v9.0, in addition to the LO frequency, it is essential also to produce a quadrature (sampling) clock at four times the I.F frequency.

This is where Jan's concept is brilliant! By a very careful choice of the LO frequency, the same frequency can be divided by four to produce the required quadrature clock.

For the 144MHz converter:

144MHz multiplied by 0.8 = 115.2MHz, that is the LO frequency required

When the LO frequency is subtracted from 144MHz:

144MHz - 115.20MHz = 28.8MHz, that is the IF frequency output

Dividing 115.20MHz by 4 = 28.8MHz, the required quadrature sampling clock.

If you do the sums for say 146 and 148MHz you will see the LO frequency is changed, and hence the IF output follows, but always fits the above calculations.

What I found hard to figure out at first was that the 4MHz segment 144 - 148MHz, produces an IF range covering 28.8 - 29.6MHz, but that is the effect created by the LO being variable.

For the 50MHz converter:

50MHz multiplied by 1.333333333 = 66.66666666MHz, that is the LO frequency required

When 50MHz is subtracted from 66.6666MHz:

66.66666MHz - 50MHz = 16.66666MHz, that is the IF frequency output

Dividing 66.6666MHz by 4 = 16.666666MHz, the required quadrature sampling clock.

One thing that might cause some confusion, you will see that the same LO multiplication factor (1.33333) is used as for the simple v9.0 BPF option for 6m. The difference being in that case, 1/3 sub-harmonic sampling is used, i.e. 50MHz divided by 3 and multiplied by 4.

For all of the VHF converters, the 'normal' 4 times sampling clock is used at the IF (receive) frequency.

I have checked the firmware by using my SDR-kits USB Synth to verify everything above worked which also helped me to get my mind around it!

Message 35187 - Posted 23rd July 2009

SDR Program operation and compatibility

I have started to do some compatibility testing using my USB-synth and a frequency counter on the output to confirm that the correct LO frequencies can be generated for use with the 6m and 2m v9.0 converters.

50MHz LO = 66.66666MHz 144MHz LO = 115.2MHz

These LO frequencies are variable and only true at those exact frequencies.

At the present time, the following programs appear to working:

For 50MHz (firmware LO multiplication factor set to 1.33333333)

Winrad v1.32 Winrad v1.42 Rocky v3.6 PSDR v1.9.0 sr40 (Guido PE1NNZ) PSDR-IQ v1.12.20 (Christos SV1EIA) PSDR v1.16.2 PSDR v1.18

For 144MHz(firmware LO multiplication factor set to 0.8)

Winrad v1.42 works correctly Rocky v3.6 works correctly, edit Rocky.ini file to add additional centre frequencies as required PSDR v1.9.0 sr40 works correctly

The following cannot be used with just the simple firmware LO * 0.8 setting:

Winrad v1.32 - maximum LO can be set to is 99.9999MHz PSDR-IQ v1.12.20 - VFO maximum setting is 65MHz PSDR v1.16.2 - as above PSDR v1.18 - as above

I suspect the 65MHz is related to maximum CMOS Si570 frequency limit.

65MHz x 4 = 260MHz, original device CMOS maximum.

I am currently trying to get the above working using the LO offset functions, but have had no success yet.

To set Fred's v15.10 firmware to use either converter, you must use Winrad to access the firmware set-up pages, and press the 'Start' button.

Press 'H' on keyboard to access the set-up screens:

Select the 'LO' tab

In the top box, 'LO:Total' set the LO multiplication factor to:

1.3333333 for 50MHz 0.8 for 144MHz

Leave the 'Offset' set to '0'

Then press 'Save' followed by 'Read'

In the second box 'LO:DLL' the values should be:

Multiplication factor:

50MHz = 4.000000xx (error due to rounding) 144MHz = 4

Offset = 0

In the bottom greyed out box 'LO:Firmware'

50MHz = 0.33333333 (derived from 1.33333 / 4) 144MHz = 0.2 (derived from 0.8 / 4)

As I commented in my previous posting, there can be a little difficulty due to an anomaly in the compiler used for the dll file. Both a leading '0' and decimal point produce an error, so enter the 0.8 as 8 and use the cursor keys to shift to type in the '0' and decimal point. Fred is aware of this, and is going to see if it can be resolved.

As far as the programs that do not easily work the 144MHz converter, I will keep on playing and trying to find a solution, if any is available.

Hope this might help anyone who has received their kits, built them and now keen to try them out!

Message 35203 - Posted 24th July 2009

Winrad v1.32 and WinradHD

Just remembered to try WinradHD, essentially its the same as v1.32, so ok with the 50MHz converter, but same problem with the 144MHz, VFO maximum is 99.99MHz.

Not made any progress with set-up for 144MHz on those two versions of Winrad, or all versions of PSDR. Only Guido's PSDR v1.9.0 sr40 is currently working with the simple set-up.

Message 35xxx - Posted 26th July 2009

Christos SV1EIA PSDR-IQ v1.12.20 and 144MHz operation

I just had another look at using the v9.0 144MHz converter with the programs that I have not been able to get to work so far.

I have found a simple way to trick Christos SV1EIA's PSDR-IQ v1.12.20 into working with the 144MHz converter.

Once you have set the v15.10 firmware for the LO * 0.8 multiplier, then when you run that version of PSDR, in the set-up select:

'Hardware Configuration' tab, then 'check' the 'expert' box in the Si570 section. Currently the Si570 maximum frequency is set to 260MHz (the CMOS device limit). If you change this to say 600MHz, and apply it.

When you select the 'GEN' option, you can now tune the VFO right up to 150MHz, ($150 \times 4 = 600$), so now works correctly at 144 - 148MHz and PSDR recognises the 2m setting.

I have checked the LO frequencies are correct, but with the 9kHz 'psuedo IF' offset:

144MHz LO = 115.200MHz (ignoring +9kHz IF offset) 146MHz LO = 116.800MHz (as above) 148MHz LO = 118.400MHz (as above)

Unfortunately you cannot do the same for PSDR versions v1.16.2 or v1.18 as there are no set-up options for the Si570. I guess the limitation is in the program software based on the assumption that the Si570 can only go to 260MHz, based on the normal x4 clocking used, and not taking into account the possibility of externally modifying the LO multiplication factor in the firmware.

The same applies to Winrad v1.32 and Winrad HD, although you can set the Si570 maximum frequency much higher in the set-up, the VFO is limited to 99.999MHz on the display.

Some small progress, but I will keep on playing,

Message xxxxx - Posted 26th July 2009

Correction to PSDR-IQ information

Just a slight correction to my last posting concerning PSDR, the maximum VFO frequency you can set is 65MHz, which I guess is due to the program assuming 260MHz is the maximum Si570 frequency (260 / 4 = 65).

In Christos PSDR-IQ v1.12.20, if you check the 'expert' box for the Si570 set-up, you can alter this value, I set it to 600MHz, and then you can tune the VFO up to 150MHz (600 / 4) which allows you to use his version directly with the 144MHz converter.

As the LO frequency demanded is modified within the v9.0 firmware (LO * 0.8), this is well within the Si570's capability.

Unfortunately there is <u>NO</u> similar set-up facility for the Si570 maximum frequency in the FlexRadio's v1.16.2 or v1.18, so at the present time still its not possible to use those with the 144MHz converter.

Winrad v1.32 and HD, seem to be limited purely by the maximum you can set the VFO to on the screen.

Sorry for the numerous posts on this subject, this is a steep learning curve for me at the moment, and I am trying to keep everyone updated who might already have received their converter kits and wanting to start playing. I am still sitting here eagerly awaiting the arrival of mine!

Message xxxxx - Not posted yet 27th July 2009

After discussing the problem with Fred PE0FKO, he realised there was a 'bug' in the v15.10, this affected the 'offset' function, and was the cause of the difficulties I was experiencing. The problem only affected the use of negative offsets (needed for use with the 144MHz converter and PSDR).

The firmware had previously been tested using a positive offset, as needed to use the v9.0 RX with an external converter, i.e 144 to 28MHz with fixed 116MHz LO.

So it was not my understanding at fault (this time)!

A new corrected version (v15.11) was quickly produced, and I have carried out testing to confirm that the 'offset' function now works correctly. As a result, by using the offset function, it is now possible to set-up the firmware to use all of the outstanding SDR programs.

PE0FKO v15.11 Firmware

Winrad v1.32 and WinradHD with the 144MHz converter

As it is not possible to set the VFO to greater than 99.999MHz, the best option for the 144MHz converter using Winrad v1.32 or WinradHD is to add an offset of 100MHz to the VFO frequency.

Tuning 44 - 48MHz on the VFO, will then actually tune 144 - 148MHz with the converter.

In the LO:Total set-up

LO multiplier set to: 0.8 Offset set to: -100 (minus)

Then press 'Save' and 'Read' to update the firmware.

An alternative setting for those more used to a 'normal' fixed frequency converter would be to set the offset to -116, that would allow you to tune the VFO 28 - 32MHz to cover 144 - 148MHz.

Both options have been proven, but perhaps tuning 44 - 48MHz is the better option as it is, in effect a direct read out of the 2m frequency, less the leading digit.

FlexRadio's PowerSDR v1.16.2 and v1.18 with the 144MHz converter

In both of the above 'official' versions of PSDR, the VFO limit is 65MHz, therefore the options used with Winrad v1.32 / HD can also be used, i.e. tuning 44 - 48MHz with an offset of -100, or tuning 28 - 32MHz with an offset -116.

It might be best to use the 'transverter' option. This defaults to expecting a conventional converter with a 116MHz LO offset, and therefore tunes 28 - 32MHz.

If you set the firmware up for the 0.8 multiplier and -116 offset, all that is needed in PSDR is to select the 'XVTER' set-up tab, and set the following:

Enable - check box Button label 144 or 2m LO Offset 116 Frequency begin - 144.000 Frequency End - 148.000

Once done, when 'VHF+' is selected, the 144MHz 'transverter' can be selected.

The transverter option could in fact be used with all versions of PSDR and the 144MHz converter

For general interest, the following LO frequencies are generated by this process:

With LO * 0.8 multiplier and offset -116 (for tuning 28 - 32MHz = 144 - 148MHz)

28 + 116 = 144 * 0.8 = 115.200MHz LO 144.000 - 115.200 = 28.800MHz IF 115.200 / 4 = 28.800 30 + 116 = 146 * 0.8 = 116.800MHz LO 146.000 - 116.800 = 29.200MHz IF 116.800 / 4 = 29.200 32 + 116 = 148 * 0.8 = 118.400MHz LO 148.000 - 118.400 = 29.600MHz IF 118.400 / 4 = 29.600

With LO * 0.8 multiplier and offset = -100 (for tuning 44 - 48MHz = 144 - 148MHz)

 $\begin{array}{l} 44 + 100 = 144 \text{MHz} * 0.8 = 115.200 \text{MHz LO} \\ 46 + 100 = 146 \text{MHz} * 0.8 = 116.800 \text{MHz LO} \\ 48 + 100 = 148 \text{MHz} * 0.8 = 118.400 \text{MHz LO} \end{array}$ calculations are the same as above example

It is difficult to suggest which firmware set-up option is best to use, that will largely depend on which is your favourite SDR program, and if you need to swap between them. This only applies to the 144MHz converter.

If you are using the 50MHz converter, all SDR programs require the same set-up:

LO * 1.333333, with offset = 0

50MHz * 1.3333 = 66.6666MHz LO	66.6666 - 50 = 16.6666MHz IF	66.6666 / 4 = 16.6666
52MHz * 1.3333 = 69.3333MHz LO	69.3333 - 52 = 17.3333MHz IF	69.3333 / 4 = 17.3333
54MHz * 1.3333 = 71.9999MHz LO	71.9999 - 54 = 17.9999MHz IF	71.9999 / 4 = 17.9999

The 70MHz converter will work in a similar fashion to the 50MHz using just the LO * 1.333333 firmware setup option with all SDR programs except the two FlexRadio versions of PSDR, v1.16.2 and v1.18 as the VFO is limited to 65MHz maximum.

LO * 1.333333, with offset 0 in the firmware

70 * 1.3333 = 93.33333 LO	93.33333 - 70 = 23.3333MHz IF	93.33333 / 4 = 23.33333
71 * 1.3333 = 94.66666 LO	94.66666 - 71 = 23.6666MHz IF	94.666666 / 4 = 23.666666
72 * 1.3333 = 95.99999 LO	95.99999 - 72 = 23.9999MHz IF	95.99999 / 4 = 23.99999

For PSDR v1.16.2 and v1.18, I would suggest using the transverter option:

LO * 1.333333, with offset -42 in the firmware, as in a conventional 70 - 28MHz converter (42MHz LO).

With LO * 1.33333333 and offset -42 (for tuning 28 - 30MHz = 70 - 72MHz)

28 + 42 = 70 * 1.3333 = 93.33333 LO	93.33333 - 70 = 23.3333MHz IF	93.33333 / 4 = 23.33333
29 + 42 = 71 * 1.3333 = 94.66666 LO	94.66666 - 71 = 23.6666MHz IF	94.666666 / 4 = 23.666666
30 + 42 = 72 * 1.3333 = 95.99999 LO	95.99999 - 72 = 23.9999MHz IF	95.99999 / 4 = 23.99999

There are now solutions for using both the 2m and 6m VHF converters with all of the available SDR programs, and it would be advisable to update the firmware to v15.11, if you intend to use the FlexRadio versions of PSDR or earlier releases of Winrad.

73, Bob G8VOI