piHPSDR User's Manual

Christoph van Wüllen, DL1YCF
email: dl1ycf@darc.de

September 11, 2023

Copyright Notice:

Disclaimer. The manual has been written with the intention that it is useful. It is quite clear that it still contains errors, therefore it is stressed here that it comes without any warranty. The reader is hereby explicitly warned that through wrong use of an SDR program such as piHPSDR, it is possible to damage the radio hardware.

Trade marks. Registered trade marks are not marked with a sign in this manual. From the absence of a trademark sign, it cannot be concluded that a mark you find in this manual is not registered or not protected.

The author:

Christoph van Wüllen (DL1YCF) has contributed a lot to piHPSDR in the last few years, this manual refers to the code in his github account

`https://github.com/dl1ycf/pihpsdr`

where the LaTeX ,,source code" of this manual, together with all figures in .png format, can be found in the `release/LatexManual` directory. At this moment this code has significant developed compared to the piHPSDR code in John Melton's master repository, but there is still hope that both versions can be merged in the future, although this will be hard work.

If you think you can improve the manual, you are welcome. Simply fork the above repository and make a pull request, or (this is the recommended way) write an email to the author: `dl1ycf@darc.de`

Contents

1 Introduction **1**

2 Starting piHPSDR for the first time **5**

3 Main window layout **11**

3.1 One or two receivers 11

3.2 Spectrum scope options 13

3.3 Zoom and Pan 14

3.4 The `Hide` button 15

3.5 Window areas 16

3.6 Mouse clicks in the main window 18

3.7 VFO bar and status indicators 19

3.8 Meter section 23

4 The Main Menu: introduction **25**

4.1 The `Exit` Menu 27

4.2 The `About` Menu 28

5 The Main Menu: Radio-related menus **31**

5.1 The `Radio` Menu 31

5.2 The `Screen` Menu 37

5.3 The `Display` Menu 39
5.4 The `Meter` menu 42
5.5 The `XVTR` (Transverter) Menu 43

6 The Main Menu: VFO-related menus 45

6.1 The `VFO` menu 45
6.2 The `Band` menu 47
6.3 The `BndStack` (Bandstack) menu 48
6.4 The `Mode` menu 49
6.5 The `Memory` menu 50

7 The Main Menu: RX-related menus 53

7.1 The `RX` Menu 54
7.2 The `Filter` menu 54
7.3 The `Noise` Menu 57
7.4 The `AGC` Menu 59
7.5 The `Diversity` Menu 60

8 The Main Menu: TX-related menus 63

8.1 The `TX` Menu 63
8.2 The `PA` Menu 67
8.3 The `VOX` Menu 68
8.4 The `PS` (PureSignal) Menu 68
8.5 The `CW` Menu 70

9 The Main Menu: menus for RX and TX 71

9.1 The `FFT` (Signal Processing) Menu 71
9.2 The `Equalizer` Menu 72

9.3 The `Ant` (Antenna) Menu 72
9.4 The `OC` (OpenCollector) Menu 73

10 The Main Menu: controlling piHPSDR 75

10.1 The `Toolbar` Menu 75
10.2 The `RigCtl` (Rig control, or CAT) Menu 78
10.3 The `MIDI` Menu 78
10.4 The `Encoders` Menu 78
10.5 The `Switches` Menu 78

A List of piHPSDR „Actions” 85

B piHPSDR keyboard bindings 109

C piHPSDR CAT commands 111

D How to connect a Morse key/paddle 113

E Running piHPSDR alongside with DigiMode programs 115

F RaspPi: configure GPIO 117

G RaspPi: binary piHPDSR installation 119

H Linux: piHPDSR install from sources 121

I MacOS: piHPDSR install from sources 123

Chapter 1

Introduction

piHPSDR is a program that can operate with software defined radios (SDRs). As a graphical user interface, it uses the GTK-3 toolkit, while the actual signal processing is done by Warren Pratt's WDSP library. Thus, piHPSDR organizes the transfer of digitized radio frequency (RF) data between the radio hardware and the WDSP library, the transfer of audio data (either from a microphone or to a headphone), as well as the processing of user input (either by mouse/touch-screen, keyboard, or external "knobs and buttons"), and the graphical display of the RF data. piHPSDR is intended to run on different variants of Unix. It runs on all sorts of Linux systems, including a Raspberry Pi (hence the name piHPSDR), but equally well on Linux desktop or laptop computers, and on Apple Macintosh (Mac OSX) computers which have a Unix variant under the hood. The present author is not aware of piHPSDR running under the Windows operating system, although with environments such as MinGW, this should be possible.

Although piHPSDR can be operated entirely by using mouse and keyboard as input devices, many users prefer to have physical push-buttons and/or knobs or dials. To this end, piHPSDR can control push-buttons and rotary encoders connected to the GPIO (general purpose input/output) lines of a Raspberry Pi. At least two generations of such controllers have been put on the market by Apache labs, and I know of several projects where home-brewn controllers have successfully been made. As an alternative, MIDI devices can be used for user interaction. For desktop/laptop computers that do not have GPIO lines, MIDI offers an easy-to-use possibility of having push-bottons and dials that control piHPSDR. Apart from homebrew projects in which a micro-controller such as an Arduino Micro controls the actual buttons/knobs and acts as a MIDI device to the computer to which it is connected via USB, there are low-cost so-called "DJ controllers" (DJ stands for disk jockey) from various brands which have successfully been used with piHPSDR. A third possibility to control piHPSDR is via a serial interface through CAT (computer aided transceiver) commands. The CAT model used by piHPSDR is based on the Kenwood TS-2000 command set with lots of PowerSDR extensions.

Using a touch-screen instead of a mouse offers the possiblity to put the actual radio hardware together with a Raspberry Pi running piHPSDR and an assortment of buttons/knobs into a single enclosure. This way, one can build an SDR radio which can be operated like a conventional analog one.

The piHPSDR program has been written by John Melton G0ORX/N6LYT. It is free software that is licensed under the GNU (free software foundation) general public license. Many other radio amateurs have contributed to the code. A lot of extensions and improvements have been added by myself, therefore this document refers to the version of piHPSDR that can be found on my github account `https://github.com/dl1ycf/pihpsdr`.

Because piHPSDR can be used on many different types of computers, and because operating systems change rather quickly over time, I generally do not recommend to have a ,,binary release" with files that you can just copy to your computer and then it runs. Instead, my personal recommendation is to build piHPSDR and WDSP from the sources, only this procedure guarantees compatibility of the final program with your operating system. A manual of how to compile piHPSDR from the sources is available separately, see `https://github.com/dl1ycf/pihpsdr-compile-from-sources`, so this will not be covered in the present manual. This manual starts with the first invocation of a freshly compiled piHPSDR.

Within this manual, we shall use a typewriter font in red color if we refer to a text or a button within a menu of within the VFO bar. piHPSDR menus and commands are indicated through a typewriter font printed in blue. The author hopes that this improves readability. In some cases, if the name of a command or menu is written on the screen, you may find the same string both typed in red and blue in the description, depending on whether the text refers to the command in an abstract sense or to the string as it can be found on the screen. This may be confusing upon first reading, but I shall try to follow the coloring convention as laid out here.

Chapter 2

Starting piHPSDR for the first time

Let us assume you have an SDR (say, an ANAN-7000 or a HermesLite-II) powered up and connected to an antenna, and you have piHPSDR installed on a computer (say, a Raspberry Pi or an Apple Macintosh), the first thing to do is to establish a proper connection between the computer and the radio. Although advocated at many places, I do highly recommend against a WiFi connection. WiFi routers often use optimizations where they hold back data packets for a given client for a while, to be able to send a collection of them in a burst. While this certainly optimizes the through-put because it minimizes clear-channel arbitration events, such jitters are desastrous in SDR operation. The safest way of connecting the radio and the computer is to have a managed switch with a built-in DHCP server, and to connect both the computer and the radio with a suitable cable to the switch. If the computer has both a RJ45 jack for an ethernet cable, and a WiFi interface, my personal recommendation is to use WiFi to connect the computer to the internet, and use a single direct cable plugged into the RJ45 jacks of the computer and of the radio. This is a little bit tricky since both the computer and the radio have to be set to a fixed IP address (e.g. computer: 192.168.1.50, radio: 192.168.1.51) with the same netmask. However, once this has been done, this is the safest connection with no perturbations from elsewhere.

If the piHPSDR program is started for the first time, it opens a window that looks like Fig. 2.1. Besides stating a version number and when piHPSDR was

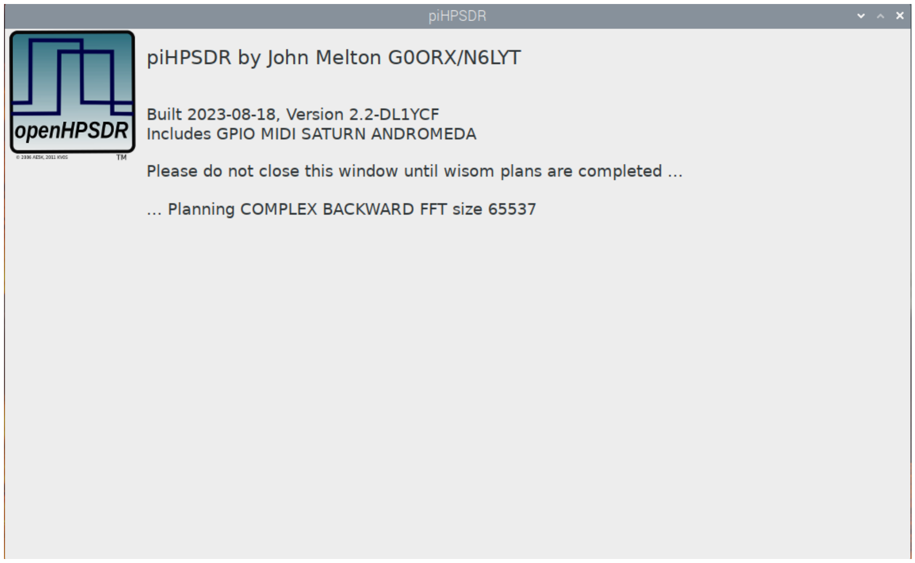


Fig. 2.1: piHPSDR screen while completing the *wisdom plans*.

built, a list of optional features (to be activated at compile time) is stated, in this case, GPIO, MIDI, SATURN, and ANDROMEDA. These options indicate that the program has GPIO support (this is only possible on Raspberry Pi or similar single board computers), that is has support for MIDI devices, that it can run natively on the compute module of the latest G2 (generation two) SDRs from Apache labs, and that is has support for Laurence Barker's ANDROMEDA controller. What is important here that you have to wait. This only applies to the very first time you start piHPSDR. On CPUs with a rather simple instruction set (like the ARM processor in the Raspberry Pi, or the Apple Silicon processor in recent Macintosh computers), this so-called *planning* step is quite fast. For example, on my Apple M2 Mac mini, this step only takes 6 seconds, and you have to wait for 34 seconds on a RaspberryPi 4. On the contrary, on CPUs with complex instruction sets, more planning is necessary: on my other Mac mini with a 3 GHz x86 processor, it takes 16 minutes! But note this has only to be done once, in subsequent starts of piHPSDR, the wisdom will simply be read from the file created during the *wisdom plans*. These plans contain, for a large number of dimensions, the fasted way how to perform FFTs (fast Fourier transformations) on the given CPU. When the wisdom is secured, piHPSDR tries to detect a radio on the network. If everything went well with the network connection, you then see

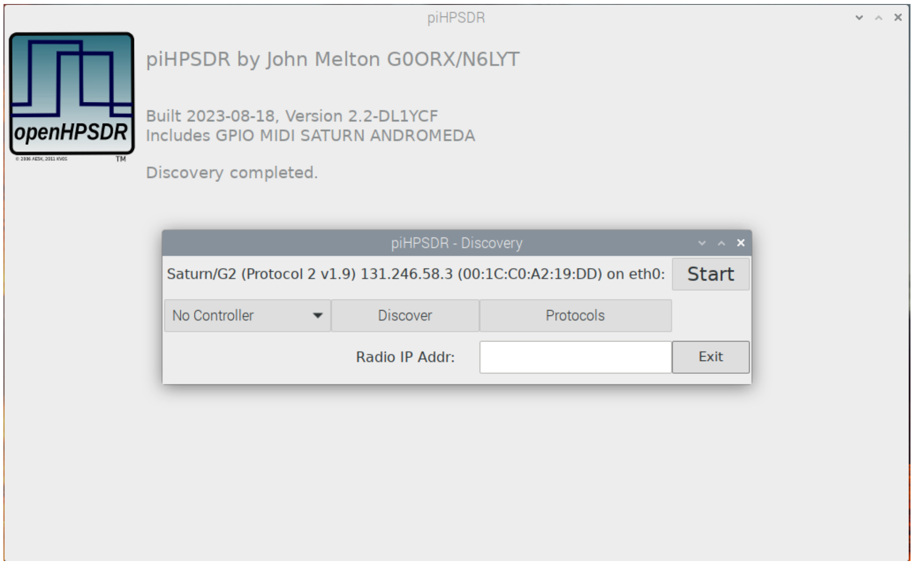


Fig. 2.2: A radio has been discoverd. You are ready to start it.

a screen with a discovery menu (Fig. 2.2).

At this point, you can start the radio by clicking the `Start` button, but let us first explain the purpose of the other buttons! Easiest to explain is the `Exit` button, this will simply terminate the program. Most likely, you may want to go into the `Protocols` menu sooner or later. By default, piHPSDR tries to discover the presence of a radio using all protocols known to piHPSDR. However, if you know that your radio, for example, uses P2 (Protocol 2), then trying to discover a P1 (Protocol 1) radio is just a waste of time. So if you know which types of radio you want to connect to, you can enable (only) these in the `Protocols` menu. The available protocols are

`Protocol 1` This is the "original" HPSDR protocol.

`Protocol 2` This is the "new" HPSDR protocol.

`Saturn XDMA` This is used to talk to a Saturn FPGA through the internal XDMA interface. Only available if piHPSDR is compiled with the `SATURN` option.

`USB OZY` This is used to talk to a radio using the legacy USB OZY interface. Only available if piHPSDR is compiled with the

USBOZY option.

`SoapySDR` This is used to talk to a radio through the SoapySDR library, for example to an AdalmPLUTO. Only available if piHPSDR is compiled with the `SOAPYSDR` option.

`STEMlab` This is used to connect to RedPitaya based SDRs through the WEB interface. Only available if piHPSDR is compiled with the `STEMLAB_DISCOVERY` option. Starting the radio using this protocol is a two-step process: first, the RedPitaya's WEB interface is located, and the `Start` button then starts the SDR app on the RedPitaya. Then, piHPSDR tries to connect to this SDR app and upon success offers a new `Start` button to start the radio. If the RedPitaya is exclusively used as a radio, it is recommended to auto-start the SDR app when the RedPitaya is powered up. In this case, the STEMlab protocol is not used, because the SDR app can be started through Protocol-2.

`Autostart` This is a very useful option. It indicates that if exactly one radio has been found, it is automatically started. So in normal operation, when starting piHPSDR subsequently, and all settings are still valid, the radio is started without user intervention. If this option is activated and one radio is present, you will not see this menu, so in order to make further changes here, you have to disconnect the radio from the ethernet cable, start piHPSDR until you see this menu, and reconnect the radio.

Sometimes piHPSDR needs to know the IP address of the radio. This is, for example, the case for the `STEMlab` discovery described above. In such a case the IP address in numerical form (xxx.xxx.xxx.xxx) can be entered in the box with the label `Radio IP Addr:`. If a legal IP address is contained in this box, protocol-1 and protocol-2 discoveries will also send to the IP address specified, in addition to the standard broadcast discovery packets which can only reach radios on the same network segment. With a known IP address, one can connect to radios which are not on the same subnet as the computer, in principle you can connect to any radio on the world provided it is on the internet. However, the original HPSDR standard states that a broadcast packet must be used, so several radios won't reply. On the other hand, there are some radios such as a RedPitaya or a HermesLite-II which allow being discovered by such a routed packet.

The `Discover` button re-starts the discovery process. This is useful if the radio has been powered up too late and was not yet ready when piHPSDR was started. Simply press `Discover` to give another try.

The combo-box (pop-down menu) to the left of the `Discover` button lets you choose which type of GPIO controller you have attached to the computer. This menu is only available if piHPSDR has been compiled with the `GPIO` option, which is not the case on desktop/laptop computers. The menu lets you choose between

`No Controller` Choose this if no GPIO controller is wired to your Raspberry Pi. This is the default when you first start piHPSDR.

`Contoller1` Choose this if you have a "version 1" piHPSDR controller.

`Controller2 V1` This option is valid for some early prototypes of the "version 2" controller with single encoders.

`Controller2 V2` Choose this if you have a "version 2" piHPSDR controller with double encocders.

`G2 Front Panel` Choose this if you have an ANAN G2 radio with a built-in controller.

Attention. Be sure to choose a controller only if such a controller is actually connected to your Raspberry Pi. If you choose, for example, a controller which uses an I2C expander for the switches, but no I2C interface is present on your Raspberry Pi, the program my hang when trying to open the I2C connection.

All settings (protocols, controller, IP address) made in this menu are stored in the global (radio-indepentend) settings and are restored when piHPSDR is started the next time.

If all went well, a radio could be discovered and you hit the `Start` button, the radio is started, and if this succeeds, you see something like shown in Fig. 2.3.

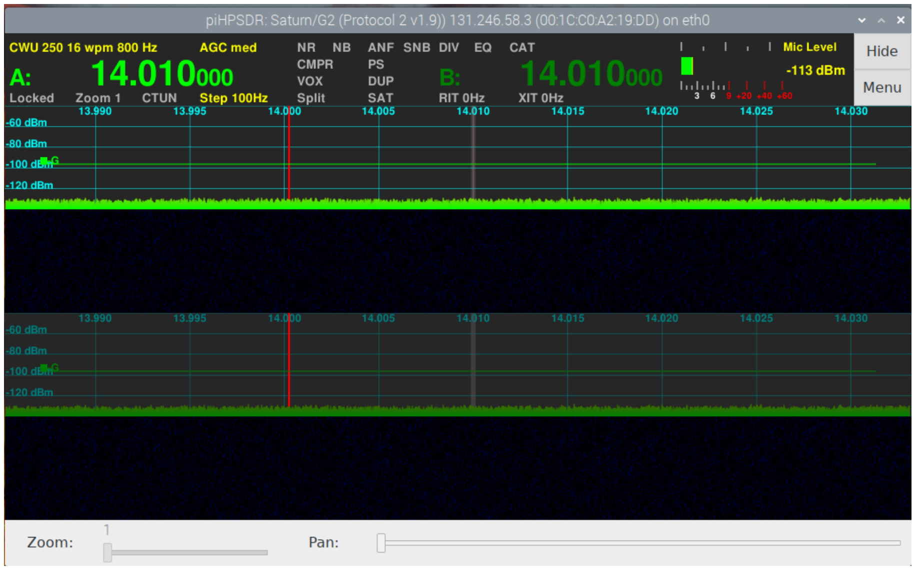


Fig. 2.3: The radio with two RX. Sliders and Toolbar are not on display by default when using a controller.

The bottom of the window looks different (more controls) if you have chosen `No Controller` in the preceeding menu. You see two receiver panels stacked vertically, both of them having a spectrum display and a waterfall area. At the top, just below the window title, you have the VFO bar which contains information on the frequencies of the two VFOs A and B, as well as lots of further information, to be explained later. At the top right, there are two buttons `Hide` and `Menu` which will be explained in the next chapter. To the left of these two buttons, there is the meter bar which by default is a digital S-meter. At this point, you have started piHPSDR successfully for the first time.

Chapter 3

Main window layout

3.1 One or two receivers

At the end of the previous chapter (Fig. 2.3), there were two receiver panels in the piHPSDR window, stacked vertically, and both including a spectrum scope (the green-coloured noise floor) and a waterfall. The waterfall area is completely black in the above picture since there was no RF signal. piHPSDR can be switched between having one or two receivers in the `Radio` menu. If there are two receivers (called RX1 and RX2), one of the two is the *active receiver*. If you look closely at the above picture, you will note that the spectrum scope of the lower (RX2) panel is shaded, while it is in bright colour for RX1. This indicates that RX1 is currently the active receiver. By simply clicking into the panel of the other (inactive) receiver, either with a mouse or on a touch screen, the formely inactive receiver becomes active.

Many conventional rigs with two independent receivers discriminate between the *main* and the *sub* receiver. It is important that this is **not** the case for piHPSDR. In piHPDSR, both receivers are largely equivalent. For example, if you start transmitting in normal (non-split) mode, the TX frequency matches the frequency of the active receiver, no matter whether this is RX1 or RX2. Likewise, in split mode, the TX frequency matches the frequency of the non-active receiver. Most of the receiver-specific controls, for example adjusting the AF volume or the AGC gain, refer to the current active receiver. If piHPSDR runs with two receivers, RX1 is always controlled by VFO-A while RX2 is controlled by VFO-B. The VFO settings not only include the frequency but also the current mode (e.g. LSB or CWU), the filter setting, the band and bandstack setting, whether RIT is enabled or not, and the RIT offset. So changing the RIT value only changes it for the active receiver. If you want to change the RIT value for RX2 while RX1 is the active receiver, you have to make RX2 active, change the RIT value and then make RX1 active again.

RX1 and RX2 are largely independent. They can receive on different bands. They can receive from different antennas provided the radio has two RF frontend with two analog-to-digital converter4s (ADC, as most modern radios do. In this case, one usually assigns the first ADC (ADC0) to RX1 and the second ADC (ADC1) to RX2. This can be done in the `RX` menu.

By default, if there are two receivers, they are vertically stacked, with RX1 in the upper part and RX2 in the lower part of the display. This can be changed in the `Screen` menu to horizontal stacking, where RX1 is in the left half and RX2 in the right half of the display. Changing the stacking trades vertical against horizontal resolution, of course.

Fig. 3.1: piHPSDR with a single RX and all controls (Zoom/Pan, Sliders, Toolbar) at the bottom.

Fig. 3.1 picture shows, for demonstration purpose, a piHPSDR window with a single receiver. The RX panel only contains a spectrum scope with a white line and no waterfall (this can be changed in the `Display` menu. In addition, you see the toolbar with eight buttons at the lower edge of the window, and above it an area with sliders. Showing the sliders is the default (and necessary) if there is no GPIO or MIDI controller attached, since then these sliders are the only way to change, for example, the AF volume. If there is only one receiver, it is controlled by VFO-A. VFO-B then actually controls nothing (except the TX frequency in split mode), but the data stored in VFO-B can be quickly used, for example by copying VFO-B to VFO-A (the `A<B` command), or by swapping the two VFOs (the `A<>B` command).

3.2 Spectrum scope options

You have already seen two different spectrum scopes: in the first picture, the spectrum was a filled green area, while in the last picture, there only was a white line (this is similar to what you would see on a spectrum analyzer). This can be adjusted to your personal preference in the `Display` menu (see below). There are two options which you can enable or disable, such that there are four different outcomes. The first option is the ,,Filled” option which discriminates between a line spectrum and a spectrum which is filled below the line. In the picture below, the first and third example have no filling, while the second and fourth spectrum are filled:

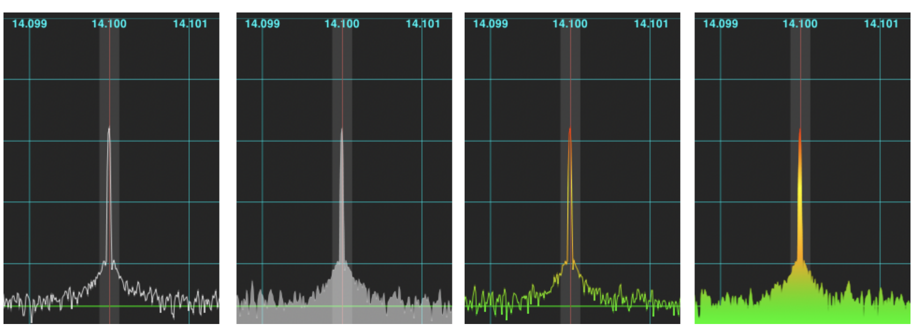


Fig. 3.2: Display options for the spectrum scope.

Then there is the ,,Gradient” option. Without this option, the spectrum is displayed in white colour. With the gradient option, the colour changes from green over yellow towards red depending on the signal strength (red colour is reached for S9). The above picture demonstrates the four possible combinations, and in the Display menu, you can make your choice. This setting refers to both receivers when there are two. Note that the TX spectrum can be a filled one or a line spectrum, but that the gradient option does not apply.

3.3 Zoom and Pan

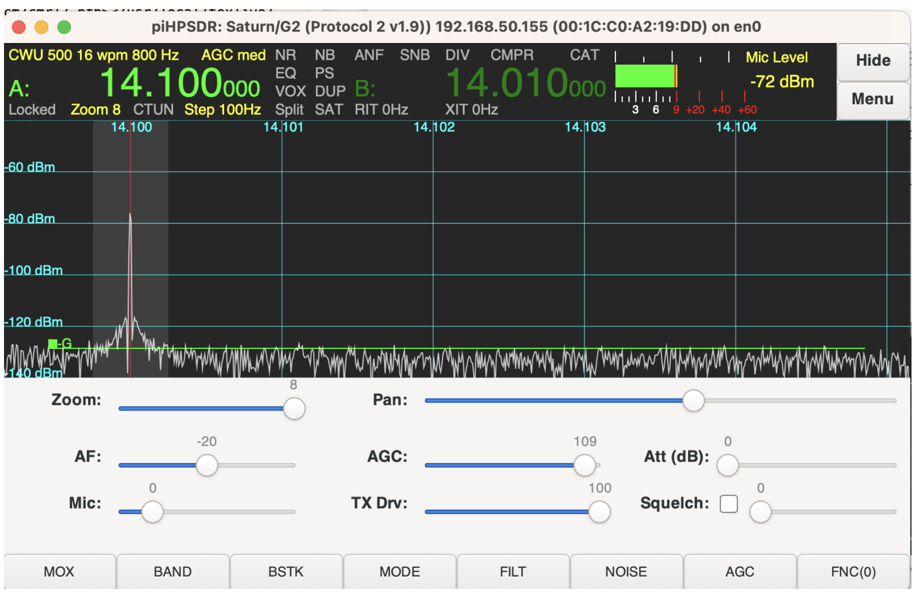


Fig. 3.3: The spectrum scope of Fig. 3.1 with a large Zoom value.

The width of the RX spectrum equals the sample rate of the receiver. This means that if you use, say, a sample rate of 96 kHz for a receiver, its spectrum will be 96 kHz wide, which may encompass a larger part of the spectrum than you are interested in. As a drawback, the part which is relevant to you may look a little bit compressed. This is where the Zoom command comes in. The zoom value can adopt integral values between 1 (no zoom) and 8. In the latter case, only 1/8 of the overall spectrum is displayed on the screen. In the picture below, you see that the RX scope is only 12 kHz wide (which is 1/8 of the RX sample rate, 96 kHz in our example). Note that what is displayed is in full resolution. Internally, a spectrum with 8 times the number of pixels of the screen width is created and only a part of it is displayed. The zoom value can be changed using the `Zoom` slider (at the left edge below the RX panel).

When using a zoom value larger than one, this means that a spectrum with more pixels than the actual screen width is produced. One can select which part of that area is displayed on the screen with the `Pan` slider (below the RX panel at the right side). Normally (Zoom=1), the VFO dial frequency is exactly in the middle of the RX scope, and marked with a thin red line. On the picture above, the dial frequency (14.100 MHz) is found in the RX panel close to the left edge, and this has been done by moving the `Pan` slider.

3.4 The `Hide` button

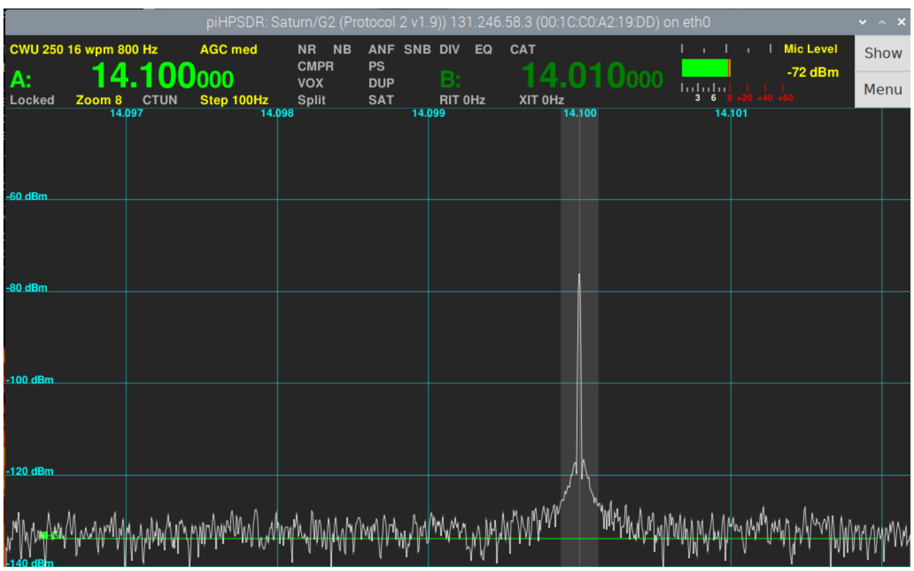


Fig. 3.4: piHPSDR window with the Toolbar/Sliders/Zoom area „hidden".

On small screens, space is scarce. This in in particular true for the vertical space if one used two RX panels and both with a spectrum scope and a waterfall. I this case, it may be hard to actually watch the signals if the screen is small. This is where the `Hide` button comes in. Clicking on this button „hides" the toolbar and slider area:

The text on the button then changes to `Show`, and clicking this button again will then return to the previous display.

3.5 Window areas

Look again at Fig. 3.1! Starting from the top, you see the title bar of the window. This bar is not visible in full screen mode, where the size of the piHPSDR window matches the display size. The title bar contains some basis information about the radio, e.g. its type, the protocol used, the IP and the hardware address of the radio. If you are really interested in this information, it is recommended to open the `About` menu.

Between the title bar and the RX spectrum scope, you see a small vertical area, most of which is taken by the VFO bar (containing the large frequency dials). At the rightmost end of this area, you see two buttons `Hide` (already discussed) and `Menu`. Clicking on the latter button opens the main menu, which will be discussed in detail in the following chapters. The `Menu` button is really important, since it enables access to one of the menus used for configuring piHPSDR. Between the VFO bar and the `Hide`/`Menu` buttons, you see the meter area where you find the S-meter (during RX) and information about output power, SWR, etc. during TX.

Below the RX spectrum scope, you see the Zoom/Pan area with the Zoom and Pan sliders, as already discussed. This area can be ,,hidden" with the `Display` menu to save some vertical space. Below the Zoom/Pan sliders you see a larger Sliders area containing several sliders for adjusting AF volume, TX drive level, RX AGC threshold, etc. Although the Sliders area can also be hidden via the `Display` menu, you should not do so unless you have a GPIO or MIDI controller which knobs that you can asssign to the slider functions. This is so since for normal operation, having access to the sliders is vital. Remember that for temporarily enlarging the space for the RX panel, there is the `Hide` button!

If you have a GPIO or MIDI console, and, say, assigned a knob there to control the AF volume, then turning the knob will auto-magically also move the AF slider if its on display (that is, if the sliders area is not hidden). If you turn a knob for which function there is slider on display, either because the slider area is hidden or because this function does not have a slider in

Fig. 3.5: A pop-up attenuation slider.

that area, then a graphical slider will temporarily pop up in the middle of the window to inform you about the changes you have made. To give one example, a knob at a MIDI console has been assigned to the RF attenuator (`Atten` function, see Appendix A), which controls the step attenuator in the RF front-end (if there is one). As long as the sliders are on display, the `Att` slider in the right part of the slider area moves when turning the knob. But when the sliders are not displayed, then a slider image pops up on the middle of the screen, and the bar contained therein moves when turning the knob, and the numerical value is displayed as well (Fig. 3.5). Such a pop-up slider always occurs if a knob on the GPIO or MIDI console is turned and no slider associated with the value changing is on display.

At the very bottom of the window, there it the toolbar. This can also be individually hidden/shown via the `Display` menu. The toolbar consists of eight ,,buttons" which you can click with a mouse or on a touchscreen. If you are using the original (V1) piHPSDR GPIO controller, is has eight push buttons below the screen and pressing those is equivalent to clicking the buttons on the screen. You might still want to keep the toolbar on display even if you are using the V1 controller since it shows you to which functions the buttons are actually assigned. This assignments consists of six ,,layers" (0 through 5). The rightmost button is hard-wired to the `Function` action which cycles through the layers. The button text includes the number of the currently active layer, and the button text of the other buttons reflect the functions assigned to the buttons in the current layer.

Bonus for mouse users only. For the first seven toolbar buttons, there is no difference if you do a primary or secondary mouse click on that button (that is, it does not matter whether you use the left or right mouse button). But for the rightmost toolbar button, a normally mouse click cycles forward through the layers, while a secondary mouse click cycles backwards. If you use a V2 or G2-frontpanel GPIO controller or a MIDI console, then you can also map this function (`FuncRev`) to a spare button.

3.6 Mouse clicks in the main window

The main window ,,accepts" mouse or touchscreen click events. Some of them come from the standard handlers of the GUI. It is clear, for example, that clicking the `Hide` or `Menu` buttons, as well as clicking one of the toolbar buttons, will activate the function associated with these buttons. Furthermore, the sliders (and the squelch enable/disable checkbox) in the sliders and Zoom/Pan are are operated as usual. But there are additional functions coded into piHPSDR:

If there are two receivers, a mouse click (press and release) into the panel of the non-active receiver makes it active. On the other hand, a mouse click in the panel of the active receiver changes the VFO frequency of that receiver to the value clicked on. This means, if you see a signal in the spectrum scope, click on that signal and your VFO will move (*jump*) to that signal. Note the VFO frequency will be rounded to the next multiple of the VFO step size when jumping by a mouse click or touch screen press.

The second option to change the VFO frequency of the active receiver is to click (and hold) into its panel, then drag the mouse to the left or to the right, and then release the button. This will shift the VFO frequency by the amount dragged, it makes no difference where the first click actually occured, only the difference in horizontal position between click and release is used. You must drag at least three pixels so there is clear discrimination between a ,*VFO jump* (click then release) and a *VFO drag* (click, drag, and release) operation. Finally, the VFO frequency of the active receiver can be changed by the scroll wheel of the mouse, if there is any. Using the scroll wheel lets the VFO frequency move in multiples of the VFO step size, while mouse dragging can also be used for finer tuning.

Clicking into the VFO bar opens the `FREQ` (VFO) menu, for the VFO-A if clicked into the left half of the bar, and for VFO-B if clicked into the right half. This menu not only offers the possibility for direct frequency entry, but also lets you alter the RIT/XIT or VFO step size, or alter the Lock, Duplex, CTUN, or Split states. So a simple click in the VFO bar gets you quick access to often-used functions.

Clicking in the meter section (between the VFO bar and the Hide/Menu buttons) opens the `METER` menu, where you can change the meter properties (see below).

When operating with a mouse, there are usually two mouse buttons, the primary button (for right-handed mouses, this is usually the left button) and a secondary one. Secondary mouse clicks are difficult to apply with a touch-screen. Although there are touch-screen drivers which convert long presses to secondary clicks, they generate, for a long press, a primary click first and a secondary one later, so it is not possible to generate a single ,,secondary press" event. But for the benefit of mouse users, secondary mouse clicks are handled in a special way:

A secondary click into the VFO bar will open the `BAND` menu, so a band change can be made with really few mouse clicks. Likewise, a secondary click into the panel of a receiver (no matter if it the active or the non-active one) will open the `RX` menu for that receiver. This can be used to change the settings of a non-active receiver without making it temporarily active. In the same way, a secondary click in the TX panel will open the `TX` menu.

3.7 VFO bar and status indicators

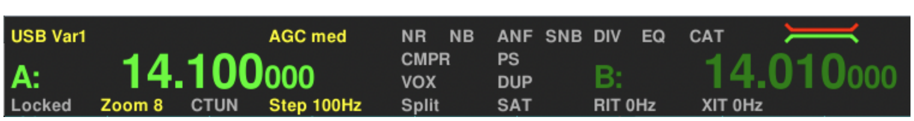


Fig. 3.6: The VFO bar

Fig. 3.6 shows the VFO bar layout in more detail. The example shown is a VFO bar whose width is 745 pixels and thus suitable for screens that are 1024 pixels wide (or more), since the meter area has a fixed width of 200 pixels, and the `Hide`/`Menu` buttons are 65 pixels wide. This layout is denoted `Large dials for 1024px windows`, as to the choice of VFO bar layouts, see the description of the `Screen` menu.

The large dials indicating the frequencies of VFO-A and VFO-B are easily recognized. The number to the left of the decimal point is the MHz part of the frequency, the three large digits to the right of the decimal point is the kHz part, and the last three (smaller) digits offer sub-kHz resolution. You may wonder why there is so much space to the left of the frequencies. This is so because with the advent of the QO-100 satellite, frequencies above 10 GHz can be used (with the transverter bands) and therefore eleven digits are needed!

Apart from the frequencies, you see a lot of text, most in light grey colour. As a general rule, a text in grey colour indicates a feature that is currently disabled, while features currently active are normally shown in yellow and sometimes in red.

At the top left corner of the VFO bar, the mode and filter of the currently active receiver is displayed. In Fig. 3.6, the text is `USB Var1` which indicates that the mode is USB using the Var1 filter with variable width (see the `Filter` menu). For the CW (CWU and CWL) modes, the CW speed (in wpm) and the side tone frequency (in Hz) is stated as well. For CW, the filter size may be appended by a ,,P”, which indicates whether the CW audio peak filter (see the `Filter` menu) is effective on top of the normal filter.

Now we continue line by line, from left to right and find the string `AGC med` printed in yellow. This means that automatic gain control (AGC) is effective in the active receiver, and that the AGC time constant is intermediate. Possible values for the time constant are Long, Slow, Medium and Fast which can be selected in the `AGC` menu. Here one can also disable AGC, in this case the VFO bar shows `AGC off` in grey colour.

Continuing to the left, we see the noise reduction settings, all printed in grey (that is, they are not effective). This can be changed in the `Noise` menu. We have two different noise reduction capabilities `NR1` and `NR2`, these strings are printed in yellow instead of the grey `NR` if they are effective. There are also two different noise blankers `NB1` and `NB2`, the automatic notch filter `ANF` and the spectral noise blanker `SNB`. Besides enabling/disabling these functions, there are further parameters you can tweak in the `Noise` menu.

The next strings whether Diversity reception is enabled or disabled (`DIV`), or whether an equalizer is effective `EQ`. Since there is a separate equalizer for the RX and TX audio chain, the equalizer indicator, if it is effective, not only turns yellow but reads `RXEQ` while receiving and `TXEQ` while transmitting. This means, if only the TX equalizer is enabled, the indicator will show a grey `EQ` while receiving and a yellow `TXEQ` while transmitting.

The last indicator in the top row is `CAT` which indicates if the CAT module (see the `RIGCTL` menu) has accepted at least one connection. In total, piHPSDR can be CAT-controlled simultaneously by five different sources, two of them using a serial line and three of them a TCP connection.

The indicators in the middle, between the VFO dials, are related to transmitting. `CMPR` indicates if a speech processor (compressor) is enabled, if so, it prints in yellow, followed by a number between 1 and 20 indicating the compression value in dB. `PS` indicates whether adaptive pre-distortion („PureSignal") is enabled, PS settings can be made in the `PS` menu. `VOX` indicates whether VOX (voice control) is enabled. VOX means that if the microphone delivers an amplitude above a certain threshold, the radio is automatically put into TX mode. Enabling/Disabling VOX and setting the correct threshold can be done in the `VOX` menu. Finally, `DUP` indicates whether duplex mode is active. In duplex mode, the receiver(s) continue to work during transmit. Duplex mode when using the same antenna for RX and TX is no fun: you not only hear your own signal with a delay (from the cross-talk at the TRX relay), but this cross-talk signal is usually so strong that it leads to „AGC pumping", so your receiver is virtually deaf during the first second after TX/RX transition. For satellite operation, on the other hand, duplex mode is very convenient. Here you usually have two separate and well-decoupled antennas for RX and TX.

The bottom line of the VFO bar indicators are related to the VFO status. If the `Locked` string is red, it indicates that the VFO is locked and will not accept changes. There is a LOCK action which toggles the LOCK status and which can be assigned to a toolbar button or a push-button on a GPIO or MIDI console, but the Lock status can also be set/unset via the `FREQ` menu, accessibly by the main menu window, or just by clicking into the VFO bar.

The next indicator in the bottom line indicates the Zoom factor. If the Zoom factor is 1, the indicator is grey, otherwise it is yellow and also indicates the factor. Then there is a string `CTUN` which indicates whether the CTUN („click to tune") mode is off or on (the string is yellow in the latter case). The step size of the VFO controlling the active receiver is displayed next, this string is always yellow.

The split status is displayed by the next indicator, which is red in split mode. If split mode is off, transmitting is done on the frequency and the mode of the active receiver (if there are two receivers), or on the frequency/mode of VFO-A (if there is only one receiver). If split mode is on, transmitting occurs on the frequency/mode of the non-active receiver (if there are 2) or on VFO-B (if there is only one receiver).

The next indicator shows the SAT (satellite) mode, which can be off (then the indicator reads `SAT` in grey), or which can be SAT or RSAT (then the indicator displays this string). Once SAT mode is engaged, the two VFOs are tied together such that any frequency change of one of the two VFOs also applies to the other VFO. This is the best way to do cross-band operation with, e.g. the QO-100 satellite which is at a fixed position. In RSAT mode, a frequency change of one of the VFOs is applied to the other VFO with an opposite sign (so if you move up VFO-A by 2 kHZ, then VFO-B moves down by the same amount). This is what one needs for low-flying satellites which have inverting transponders which offer some sort of Doppler correction.

Finally there are the `RIT` (receiver incremental tuning) and `XIT` (transmitter incremental tuning) indicators. If RIT is off, receiving occurs on the VFO dial frequency. If RIT is on, the indicator becomes yellow and also indicates the RIT offset, that is, the frequency offset used while receiving. RIT is used, for example, if during your CW QSO the frequency of the transmitter of your QSO partner drifts and you want to follow without altering the frequency of your own transmitted signal. The RIT indicator corresponds to the active receiver. If XIT is active, the indicator becomes yellow and shows the offset of the true TX frequency from the VFO dial frequency.

Finally, in the top right corner you see a symbol with a green and a red line that only occurs if one of the variable filters (`Var1` or `Var2`) have been selected. The green caret indicates the default filter edges, while the red one above denotes the current filter edges.

3.8 Meter section

Fig. 3.7 shows the different designs that exist for the meter. To the left (right) there are the digital (analog) meters, while the top panels show the meter during RX and the lower panels during TX.

Fig. 3.7: Different designs for the meter.

The design can be switched between digital and analog in the `Meter` menu, which can be accessed quickly just by clicking into the meter area. During RX, an S-meter is shown together with the signal level in dBm. Note that -73 dBm corresponds to S9 for frequencies up to 30 MHz, while above 30 MHz, S9 corresponds to -93 dBm. Since the S meter is in steps of 6 dB, a signal level of S1 (below 30 MHz) corresponds to -121 dBm.

During TX, the output power is displayed, provided that the radio actually reports this power. The output power meter can be calibrated (see the `PA` menu). If the SWR exceeds a threshold for SWR warnings (the default is 1:3, but this can be changed in the `TX` menu), the SWR indicator turns red. If, in addition, SWR protection is enabled in the `TX` menu, the output drived will be reduced to zero if the SWR exceed that threshold. Furthermore, the ALC (automatic level control) value of the transmitter is shown. Negative ALC values (at least in peak mode) indicate that the volume of the TX input audio could be increased to get full output power.

Further info on the meters (e.g. switching between ,,peak” and ,,average” reporting) is described in the `Meter` menu.

Chapter 4

The Main Menu: introduction

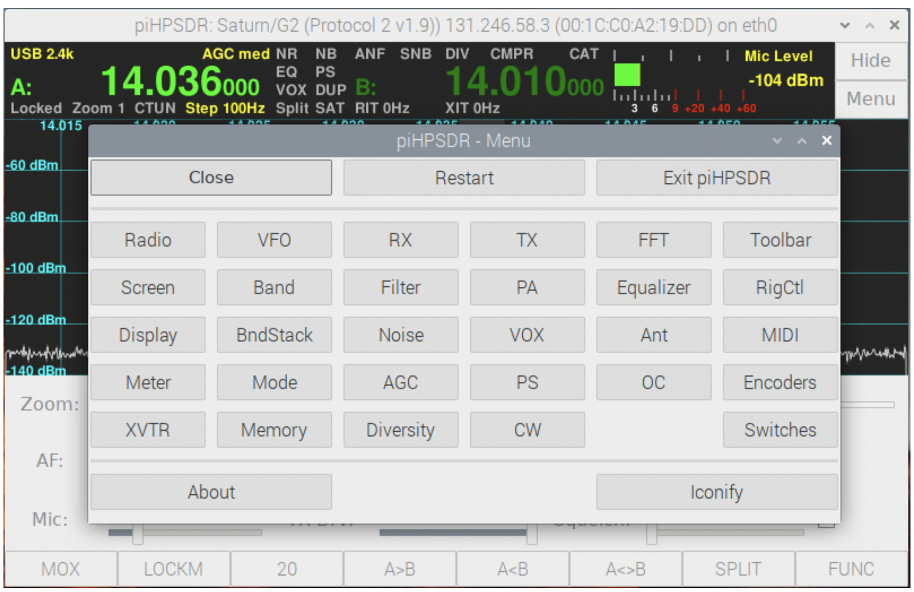


Fig. 4.1: The Main men, opened by the `Menu` button.

Now we have a series of chapters that discuss all the piHPDSR menus. Many menus can be opened by a button click (or a push-botton on an external controller), e.g. hitting the `MODE`, `FILT`, or `NOISE` button on the toolbar you have seen in the last picture. You already know that the VFO and Meter menus can be opened by clicking into the VFO or meter section at the top of the window. When operating with a mouse, a secondary click in the RX or TX panadapter opens the RX or TX menu. But there is one place from which *all* piHPSDR menus are at hand, and this is the "Main Menu". It can be opened by clicking into the `Menu` button at the top right corne of the piHPSDR window, the outcome is shown in Fig. 4.1.

Some remarks have to be made about menus in general. Since piHPSDR is optimized for working with small screens, only one menu can be open at a time. If a menu is open and one tries to open another one, the first menu will be destroyed (closed) and the new one will be opened. For example, if you hit the `FILT` button in the toolbar when starting from Fig. 4.1, the main menu closes and the `Filter` menu opens. If you try to open a menu that is already open, then the menu will be closed. So, starting from Fig. 4.1 hitting the `Menu` button again will close the menu. Likewise, when the Filter menu has been opened, either via the Main Menu or with the `FILT` button, then hitting this button again will close the `Filter` menu.

While the menus are looking quite diverse, some effort has been invested to keep some things consistent throughout. For example, at the top left corner of the menu you usually find the "Close" button which closes the menu. The close button is somewht emphasised (slightly larger letters, and a thin border) so you will always quickly find it. Of course, it it possible to close a menu by deleting the menu window (on RaspberryPi, this is the small cross at the left of the title bar) but this is neither necessary nor recommended.

There are some commands available here that do not directly affect the radio operation, so these commands are found in the top and bottom line of the Main Menu. We first mention the `Restart` button in the middle of the top line. This restarts the radio protocol. While not needed under normal circumstances, it my happen (especially with beta releases of radio FPGA firmware) that the data exchange between piHPSDR and the radio gets out-of-sync. I observed such problems with early versions of the P2 firmware for Orion2 boards and that is the reason the `Restart` button is there, since this made a quick recovery possible without loosing the QSO. At the bottom right, there is the `Iconify` button which ,,minimizes" the piHPSDR window. Normally, if needed, one can do so by standard methods of the operating system in the title bar of the piHPSDR window. If piHPSDR, however, runs in full-screen mode (this is the case on very small touch screens), then the `Iconify` button to make the piHPSDR window temporarily disappear without breaking the connetion to the radio, do some work with the operating system, and get the piHPSDR window back. Note in earlier versions of piHPSDR this function was associated with the "Hide" button in the top right corner of the main window. Then, there are two menus (`Exit` and `About`) which are described in due course and which one can open by clicking either `Exit piHPDSR` or `About` in the main menu.

The other buttons, between the two horizontal separator lines, give access to piHPSDR control and fine tuning. They are organized in six columns, namely radio related menus (first column), VFO related menus (second column), RX and TX related menus (third and fourth column), menus affecting both RX and TX (fifth column) and, finally, menus for adjusting how you can control piHPSDR (sixth column), either via Toolbar, MIDI, or GPIO encoders or switches. „**Encoders**" are knobs which you can turn, and which can be used to change AF volume or TX output power. „**Switches**" are push-buttons which can be used to trigger a function such as transmitting a carrier for tuning, toggle between RX and TX, open a menu, and so forth.

4.1 The `Exit` Menu

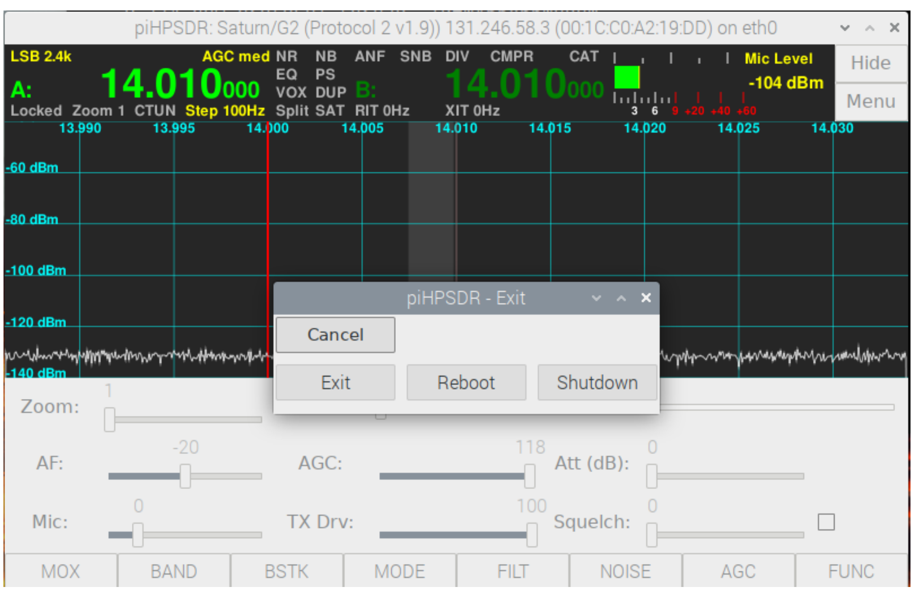


Fig. 4.2: The `Exit` menu.

Via the `Exit` menu, you can leave the piHPSDR program. When leaving the program, the radio protocol is stopped and all the settings are written to a preferences file. This file is located in the piHPSDR directory and takes the name xx-xx-xx-xx-xx-xx.progs, where the xx encode the MAC address for the radio. So the preferences for different radios (if you have more than one) are stored in different files. To leave the program, just click the "Exit" button in this menu. If you decide you want to continue, you can leave the `Exit` menu by clicking the "Cancel" button. This is the button which closes the menu and has the same position and look as the "Close" buttons in all the other menus.

If piHPSDR runs with administrator privileges, you can even leave the program and either re-boot or switch off the computer via the "Reboot" and "Shutdown" buttons. This makes sense for setups where a Raspberry Pi running piHPSDR, a small SDR radio, a touch-screen and several encoders and switches are built into a single common enclosure. On the other hand, when running piHPSDR on desktop or laptop computers, clicking "Reboot" or "Shutdown" both leave the piHPSDR program but no re-boot or shutdown takes place, due to missing administrator privileges.

4.2 The `About` Menu

The about menu gives you some information about piHPSDR, first the original author and an (incomplete) list of persons who contributed to the code, and then a statement which version of piHPSDR is working here, and when it has been compiled. Here you also find the version number of the WDSP library which is the ,,engine" running under the hood, and which does nearly all of the signal processing. Finally, there is some data on the radio, namely the device type and version numbers, and which protocol is running. For diagnostic purposes, you also see the MAC address of the radio, its IP address and the IP address of the computer running piHPSDR. The MAC address is of interest since the radio-specific preferences are stored in a file whose name is derived from the radio's MAC address.

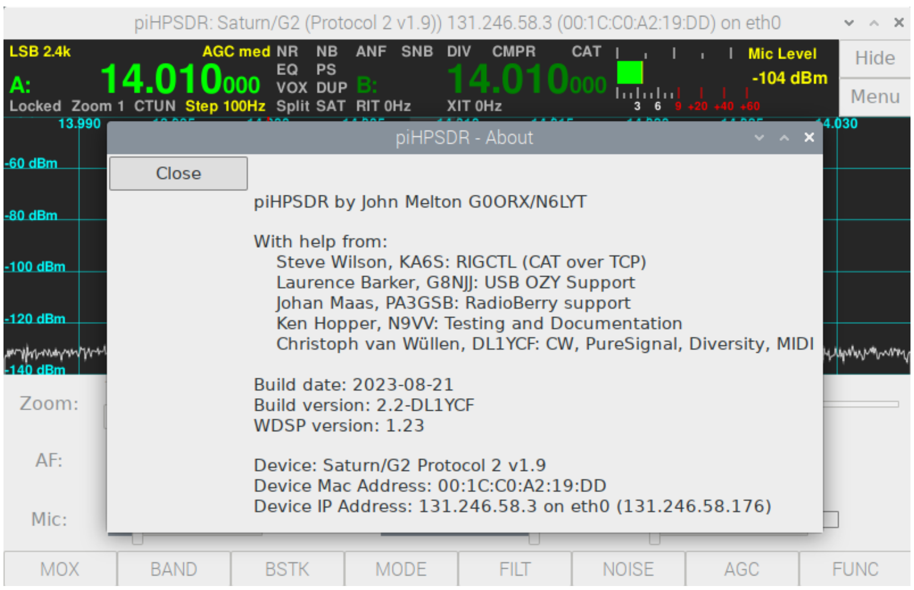


Fig. 4.3: The `About` menu.

Chapter 5

The Main Menu: Radio-related menus

5.1 The Radio Menu

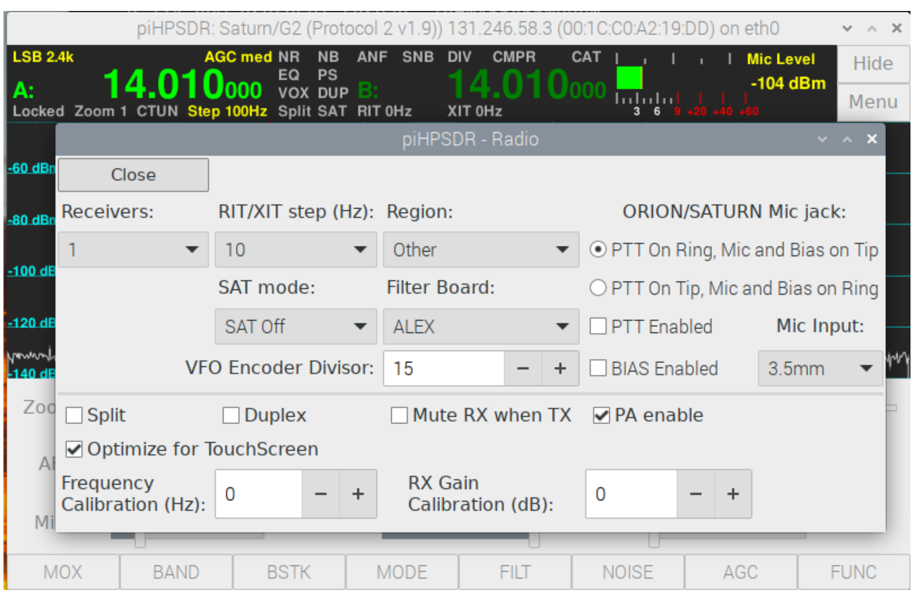


Fig. 5.1: The Radio menu.

The Radio menu lets you make settings which affect the general setting, and the hardware of the radio. The following figure (Fig. 5.1) shows the menu as it opens on an Anan G2 radio. Note this menu looks slightly different for different radios and protocols, this will be discussed at the end of the section. First, we go through all the elements we see in Fig. 5.1, they will be colored red in the following list.

`Receivers:` In the pop-down menu (GTK combo-box) below this string, you can select the number of receivers that are running (well, you can choose between 1 and 2). When the number of receivers change, the radio communication will shortly be stopped and then resumed, so do not be surprised if the spectrum scope freezes for a second or so.

`RIT/XIT step:` In the pop-down menu you can choose among three (1 Hz, 10 Hz, 100 Hz) step sizes for RIT and XIT. For example, if the RIT step is 10 Hz, then you can change the RIT offset in steps of 10 Hz with the RIT+ or RIT- buttons in the toolbar or on the GPIO/MIDI controller.

`Region:` Although not obvious, this selects settings for the 60m band. Possible choices are "Other", "UK" and "WRC15". The `Other` and `UK` choices implement the channel structure of the 60m band according to the regulations valid in the USA and Great Britain. "WRC15" gives you a small (15 kHz wide) 60m band according to the WRC15 (World Radio Conference 2015) document, which is now implemented in many countries.

`Orion/Saturn Mic jack:` This part of the menu is not shown for pre-Orion boards. The Orion, Orion2, and Saturn boards can switch the connections of the TRS microphone jack in software (hardware jumpers had to be used previously). While the ring of the TRS plug is always connected to ground, the microphone and PTT connections are on the ring an tip and you can choose which one is on the ring and which one on the tip. You can then separately enable the PTT function of the jack, and select whether a bias (DC offset) is applied to the mic connetion (this is necessary for condensor microphones and detrimental if a dynamic mirophone is connected without a blocking capacitor).

`Mic Input:` This is only shown for Saturn boards. These radios have two jacks for connecting a microphone, either a 3.5mm TRS jack in the front panel, or an XLR connection in the back panel. The pop-down menu lets you choose between these two options.

`SAT mode:` Here you can choose between `SAT off`, `SAT`, and `RSAT`. In SAT mode, frequency moves applied to one of the two VFOs are applied to the other VFO as well. This is convenient for cross-band operation over satellites with (normal) linear transponders. In RSAT mode, frequency moves applied to one of the two VFOs are applied to the other with the sign inversed, that is, if for example you move the frequency of VFO A up by 3 kHz, the frequency of VFO B moves down by the same amount. This is convenient for cross-band operation over satellites with inverted transponders. Inverted transponders are sometimes find in low and fast moving satellites because this leads to some Doppler correction.

`Filter board:` Normally SDRs have some sort of built-in PA with a filter board. Filters in the TX path between the PA and the antenna are always required, and filters in the RX path provide some protection against ADC overloads from strong out-of-band signals. Here you can choose between `NONE`, `ALEX`, `APOLLO`, `CHARLY25`, and `N2ADR`. Choose `NONE` if none of the other cases apply, and hope your radio does things right automatically. `ALEX` is the most frequent choice and applies to the largest part of current HPSDR radios. `APOLLO` is an early design of a PA/filter combination for Hermes boards, choose this if you have one. `CHARLY25` is a filter board used in some RedPitaya based radios (STEMlab and HAMlab). If you choose this, the Attenuator slider will disappear from the Slider area (because this design does not have a step attenuator), instead, you get a combined Attenuator/Preamp check-box which lets you choose between zero, preamp values of 18 and 36 dB, and attenuation values of 12, 24, and 36 dB. `N2ADR`, finally, is the filter board usually used in combination with a HermesLite-II radio. It is controlled by the OC (open collector) bits in the HPSDR protocoll. This means if you use `N2ADR`, this will override your OC settings upon program startup. It is possible to change the OC settings in the `OC` menu, and these settings are saved with the preferences. Upon next program start, however, these preferences will again be overwritten as long as the `N2ADR` filter board is chosen.

`VFO Encoder Divisor:` This option is normally only used for GPIO controllers. Often, the encoders of the main VFO dial generate too many ticks per revolution, such that it is difficult to fine tune on a signal. If the VFO Encoder Divisor, as shown in the example, has a value of 15, only every 15th tick will we processed. The Divisor is also effective if you control piHPSDR with an ANDROMEDA console. So, if the frequency moves too fast if you turn the VFO knob, you have to increase the Divisor, and if it moves too slowly, decrease it.

`Split` Use this checkbox to enable/disable Split mode. In Split mode, the frequency of the non-active receiver (when using two receivers) or the frequency of VFO-B (when using one receiver) controls the TX frequency. In normal (non-Split) mode, it is the frequency of the active receiver (2 RX) or the frequency of VFO-A (1 RX) that matters.

`Duplex` Use this checkbox to enable/disable Duplex mode. In Duplex mode, the receiver(s) continue working during TX. In normal setup, this is detrimental since the very strong signal that originates from the crosstalk at the T/R relay will lead to AGC pumping, making your receiver(s) essentially deaf for a short period after TX/RX switching. However, when using different and well-decoupled antennas for RX and TX (this is typical for some satellite operations), Duplex mode gives you important information, as you can see your own downlink signal. In contrast to what is often stated, Duplex mode does not affect the data stream between the computer and the radio, it *only* determines whether the receivers (within the WDSP library) are shut down during transmit or not.

`Mute RX when TX` This option mutes the RX audio while transmitting. It is important to note that the RX continue to work, so you can see the signals on the RX panel, the S-meter works, etc. This option is largely equivalent to moving the AF slider to the minimum position while transmitting.

`PA enable` This enables/disables the PA in the radio. In addition to this global flag, there is a per-band PA enable option for the transverter bands (see the `XVTR` menu).

`Optimize for TouchScreen` The normal procedure to make a selection from a pop-down menu (such as the `Receivers:` button on this screen) is to click (and hold) it with a mouse, then drag the mouse to your choice, and then the selection is made by releasing the mouse button. This is very difficult to achieve on a touch screen. Therefore, if `Optimize for TouchScreen` checkbox is checked, the pop-down menus are modified as follows: You click and release on the menu button, then it pops down and stays open. Then you make your selection by a second click/release sequence on your choice. While this is (only a little bit) more involved than the normal procedure when using a mouse, it is a great helper when using a touch screen. Therefore this option is set by default, but you can uncheck it here if you prefer normal mouse operation. Note that this option becomes effective when the next menu is opened.

`Frequency Calibration` Here you can set a frequency offset (in Hz). This offset will be added to all frequencies sent to the radio. This means that if you discover that a reference signal occurs in your RX panel at 10001 kHz where it should occur at 10000 kHz, you have to set the calibration value to -1000. Note it is an absolute value, which will we applied to all frequences.

`RX Gain Calibration` Here you can calibrate your RF front end. To this end, you need a highly accurate signal source of, say, -73 dBm. Connect this source to your radio and tune on the signal. If the signal appears at, say, -70 dBm, decrease the calibration value. At -3 dBm, your S-meter should then state the correct signal strength. The value is the amplification/attenuation of a virtual device you would need in your RF front end. Therefore, you need a negative value (attenuation) if the signal shown is too strong. For normal HPSDR radios, the default value is zero. For the HermesLite-II and other radios using the AD9866 chip, the default value is 14.

There are some further check boxes in the `Radio` menu that you cannot see in Fig. 5.1, since they only appear for specific radio hardware. They appear to the right of the touch screen optimization check box and will be listed here.

`HL2 audio codec` This box only appears if the radio is a HermesLite-II (HL2). Some of these radios are equipped with an audio codec and a modified firmware. The audio codec must be enabled by setting a specific bit in the HL2 protocol, and this check box enables/disables this bit. Furthermore, if this check box is enabled, RX audio data is sent to the HL2.

`Anan-10E/100B` This box only appears for Hermes boards. While the Anan-10E and the Anan-100B identify themselves as Hermes boards, they have a FPGA with limited resources and this affects the allocation of PureSignal feedback channels. To make PureSignal work on these machines, you have to check this box in the `Radio` menu.

`Swap IQ` This box only appears for radios connected via the SoapySDR library. If checked the I and Q samples are exchanged, both in the receivers and in the transmitter. An indication that this is necessary is if you see signals with a frequency above your dial frequency in the left half of the RX panel, or if you have to go to LSB to receive USB signals. If you observe this behaviour, check this box.

`Hardware AGC` This box only appears for radios connected via the SoapySDR library. If checked, automatic gain control (AGC) that is implemented in hardware in the radio is enabled.

ATLAS bus options. For legacy ATLAS bus radios, a number of additional settings have to be made. Therefore, the area where we have seen the Orion microphone options now contains ATLAS bus settings, as shown in Fig. 5.2. You will see this only if the radio identifies itself as a METIS board.

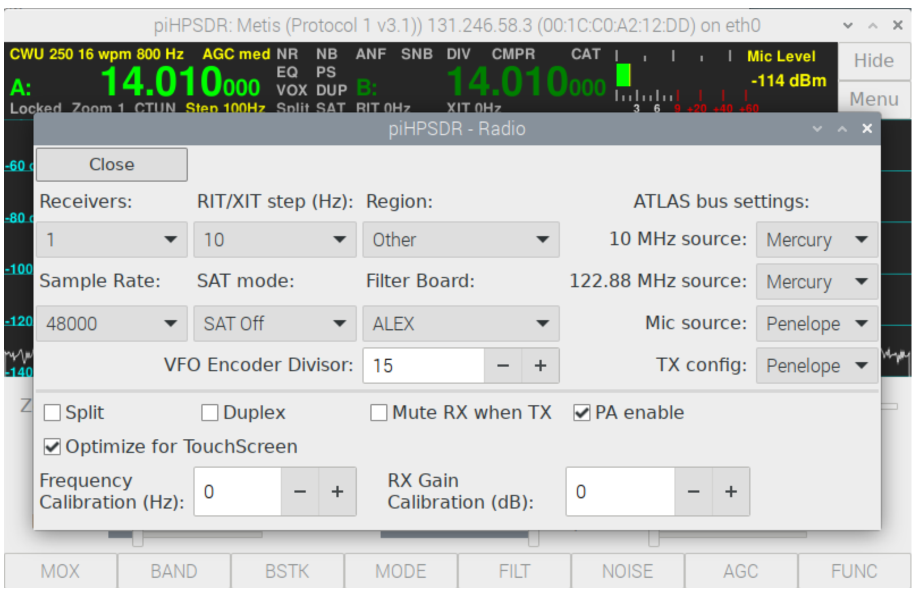


Fig. 5.2: The `Radio` menu for a legacy HPSDR board.

The first difference you notice is that at the left edge of the menu, in the middle, there is a new `Sample Rate:` pop-down menu. This has nothing to do with the ATLAS bus, this occurs if the radio is connected via P1, as it is often the case for legacy radios. In P1, all receivers share the sample rate, therefore it is set in the `Radio` menu. The same applies for SoapySDR radios. In P2 on the other hand, each of the two receivers can have its own sample rate, therefore the sample rate is specified in the `RX` menu. The ATLAS bus settings are at the right edge of the menu (see Fig. 5.2. The ATLAS bus has separate receiver and transmitter plug-in boards. To build a radio, they must be synchronized somehow, and therefore their clocks cannot run independently, but there must be a master clock.

10 Mhz source: This selects the 10 Mhz master clock, which can be either ATLAS (the bus itself is the source), Penelope (the transmitter board is the source), or Mercury (the receiver board is the source).

122.88 MHz source: This selects the 122.88 Mhz master clock, which can be either Penelope or Mercury.

Mic source: This selects where the microphone samples that are sent to the computer originate, that is, where your microphone has to be connected. The default is Penelope, this means the microphone is connected to the transmitter board. The other choice is Janus. The Janus board is simply an ADC/DAC board (not a radio) and used in some very early setups.

TX config: This indicates which transmitter board is present on the bus. It can be No TX, if this is a receive-only radio, Penelope or Pennylane. The Pennylane is a later version of the Penelope transmitter board, the essential difference is that it can control the output signal level. In the Penelope case, piHPSDR will scale the IQ samples to provide TX drive control.

Janus Only This box is for ATLAS systems that only have an OZY and a Janus board, and will only appear for OZY (USB-connected) boards. While this hardware is not a radio, external hardware such as the SDR-1000 can be connected to the Janus interface. If this option is checked, piHPSDR assumes that the radio is controlled outside piHPSDR, and will thus only process the data stream but not try to send any commands to the radio.

Note that I have no access to such legacy radios, so the piHPSDR code to handle these radios is partly built on speculation (that is, studying the specs) and exchanging e-mails with people who still run such hardware. If you meet any inconsistencies, please contact the author.

5.2 The Screen Menu

The Screen menu lets you dynamically change the size of the piHPSDR main window, and choose between different VFO bar layouts. The possibility to adjust the screen size has been the most frequent feature request in the last years, so I finally decided to implement it. The menu is opened via the main menu and the Screen button and is shown in Fig. 5.3.

The window width and height can be chosen with the spin buttons shown.

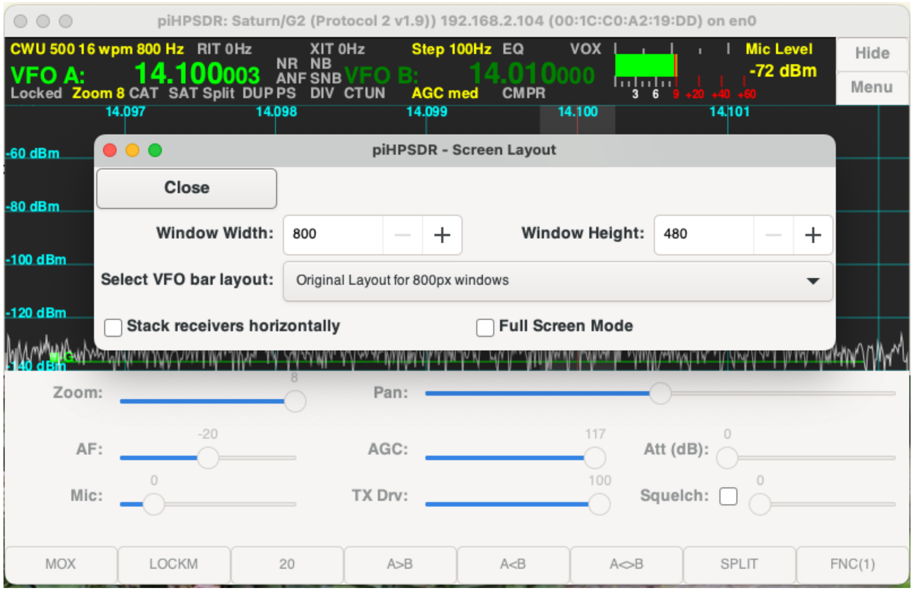


Fig. 5.3: The Screen menu.

The minimum values for width and height are 800 and 480, the maximum values are determined by the resolution of the monitor. Changes made in the spin buttons become effective immediately. If piHPSDR is in full screen mode (see below), which implies that the width and height equal the screen dimensions, and you descrease either width or height, the piHPSDR window auomatically switches from full screen into window mode.

If the window width is decreased such that the VFO bar chosen does no longer fit, the first one in the list that does fit is automatically selected, and the current choice printed in the pop-down menu `Select VFO bar layout` is updated. This menu lets you choose the layout of the VFO bar. In Fig. 5.4 the pre-defined layouts are shown.

The two largest ones require a window width of about 1000 and 900 pixels, while the two small ones can be used with a screen only 800 pixels wide. The VFO bar shown at the bottom of Fig. 5.4 represents the original piHPSDR layout, the second one from the bottom has larger characters for the VFO dials and therefore a few more pixels height. The VFO bar has been described in detail in chapter 3.7. If you choose a VFO bar layout that is wider than the current window width allows, the window with is automatically adjusted.

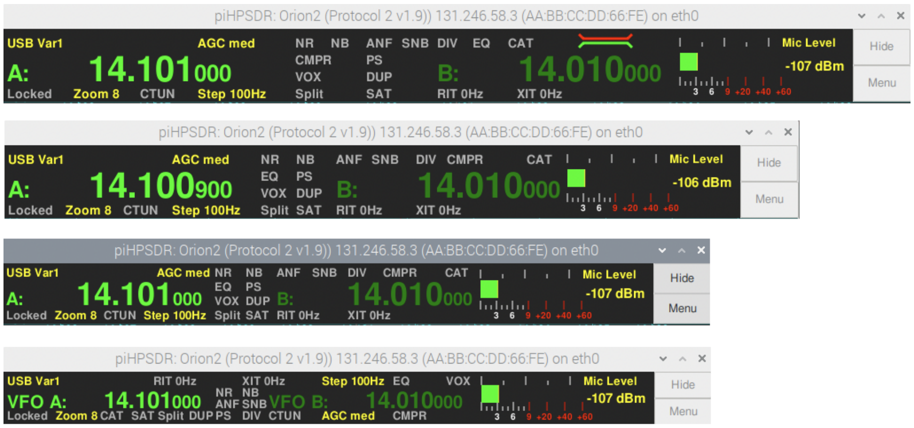


Fig. 5.4: Four choices for the VFO bar built into piHPSDR.

Stack receivers horizontally. If checked, this puts the panels of the two receivers (if two receivers are used) side-by-side instead of on top of each other.

Full Screen Mode. If you check this option, the window width and height is set to the screen width and height, and then the piHPSDR window is put into ,,full screen mode". This means that there is no piHPSDR window title bar and no OS menu bar, and the piHPSDR window area covers the whole screen. For small screens, this gives some extra space to the piHPSDR window, since the maximum window dimensions (such that the whole window can be displayed) are smaller than the screen size, typically, the height is about 30 pixels less, and the width about 10 pixels less than the screen size. If piHPDR is is full screen mode and you *uncheck* this box, this gives you a pihpsdr window with the size of the screen, which is slightly too large to be displayed completely on the screen.

5.3 The Display Menu

The Display menu is used to customize the overall layout of the piHPSDR window and the RX panadapters. Adjustments for the TX panadapter must be done in the TX menu. The menu is shown in Fig. 5.5.

Frames Per Second: This adjust how often the RX and TX (!) displays are

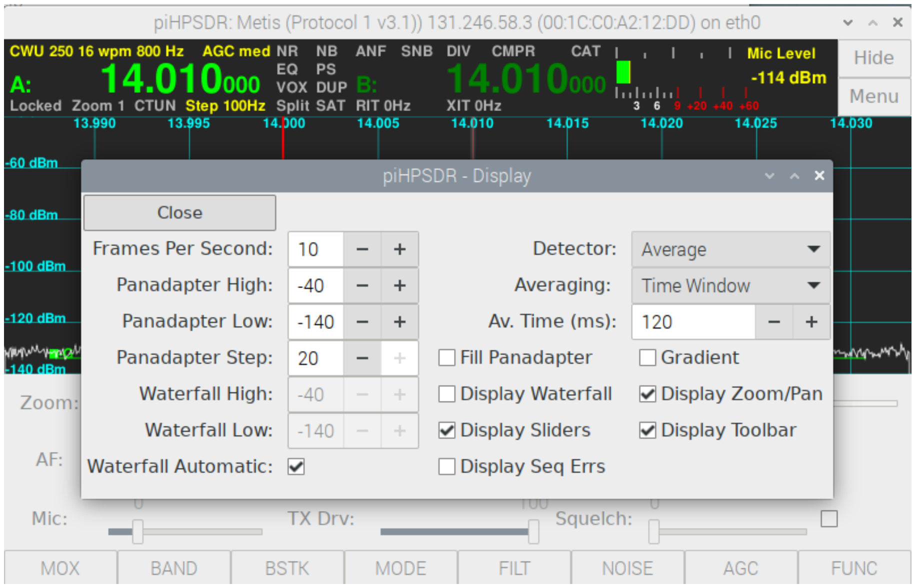


Fig. 5.5: The `Display` menu.

re-drawn. 10 frames per second (the default) is a good value.

`Panadapter High:` This value is the dBm value of the RX signal strength at the top of the RX spectrum scope. A value of -40 dBm corresponds to S9 + 33 dB for HF signals.

`Panadapter Low:` This value is the dBm value of the RX signal strength at the bottom of the RX spectrum scope. A value of -140 dBm is usually low enough such that the noise floor is still on display.

`Panadapter Step:` This value is the spacing of the horizontal lines on the spectrum scope. Lines are drawn at dBm values that are multiples of the step size.

`Waterfall High:` This is the RX dBm value that will lead to the brightest color (yellow) in the waterfall.

`Waterfall Low:` This is the RX dBm value below which the waterfall will be black.

`Waterfall Automatic:` If this box is checked, the `Waterfall High:` and `Waterfall Low:` controls are inactive and the values are not used. Instead, the lowest and highest signal strength in the RX spectrum are automaticall determined and are then used instead of the waterfall High/Low control values.

`Detector:` Here one can choose between Peak, Rosenfell, Average and Sample. The Rosenfell detector is probably closest to what one knows from a spectrum analyzer. the Average detector is usually preferred since it is less „nervous".

`Averaging:` Here the possible choices are None, Recursive, Time Window and Log Recursive. For the details, see the WDSP manual.

`Av. Time (ms):` If averaging is used for the spectrum scope, the time constant involved in averaging can be set here.

`Fill Panadapter` This is used to enable/disable the „Filling" option for the RX and TX spectrum scopes (see chapter 3.2).

`Gradient` This is used to enable/disable the „Gradient" option (color coding) for the RX spectrum scope (see chapter 3.2). Note color coding is not available for the TX panel.

`Display Waterfall` This option enables/disables the waterfall display of the RX panels. Note the spectrum scope is always shown.

`Display Zoom/Pan` This option can be used to show/hide the Zoom and Pan slider below the RX or TX panel. If you do not use Zoom, or control Zoom via an external GPIO or MIDI controller, this can be used to save some vertical space.

`Display Sliders` This option can be used to show/hide the slider area (AF through Squelch sliders). Hiding them makes little sense unless you have a GPIO or MIDI controller. For temporarily gaining vertical space, use the `Hide` button at the top right of the main window.

`Display Toolbar` This option can be used to show/hide the toolbar.

`Display Seq Errs` All data packets from the radio to the PC contain a sequence number that is increased for each packet sent. piHPSDR checks for each incoming data packet whether the sequence number is exactly larger by 1 than the number of the previous packet. If this is not the case, this is a *sequence error*. Sequence errors may be caused by loosing packets on the way from the radio to the computer, or (this is what is usually the reason) packets lost within the computer due to the computer being busy with something else (e.g. writing to disk). If the Display Seq Errs is checked, a red „Sequence error" message appears in the RX panel for two seconds. Sequence errors during RX often cause cracks in the RX audio. Unless this happens regularly, it is no reason to worry.

5.4 The Meter menu

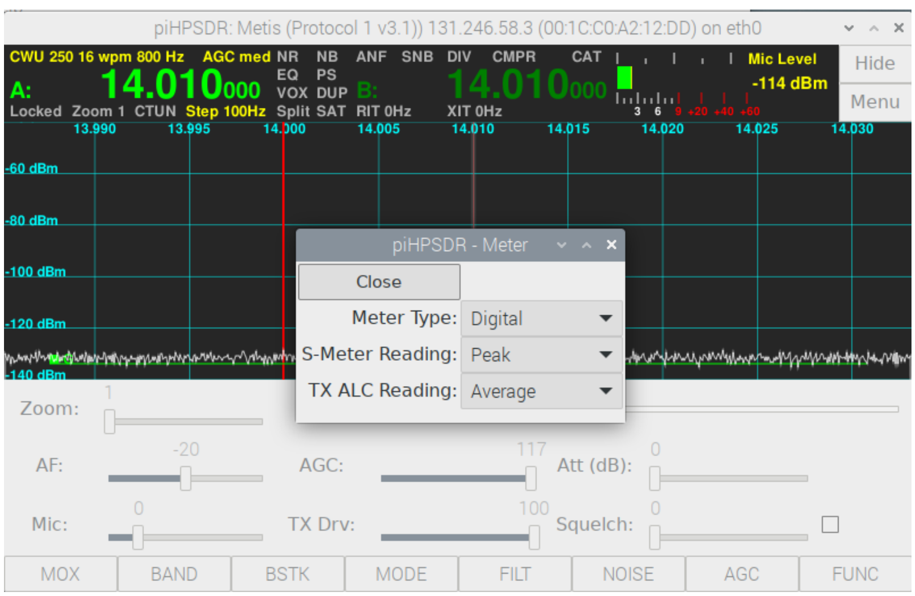


Fig. 5.6: The Meter menu.

The Meter can either be opened simply by clicking in the meter area, or through the main menu. Only few choices can be made here.

Meter Type: Here you can select between a digital and an analog meter. The four different designs (either analog or digital, and during RX or TX) have already been shown in Sec. 3.8.

In both cases, there is a choice between Peak and Average reading, which refers to peak envelope power and average power. Here averaging is done over relatively short times. For a two-tone signal for example, the peak reading is 3 dB above the average reading.

S-Meter Reading: Here you can choose whether the S-meter reports a Peak or an Average value (default is Average). Note, however, that in order to make the display less ,,nervous" a moving average with a rather long time constant (about 0.5 sec) has been implemented on top of both the Peak and Average S-meter readings.

`TX ALC reading:` Here, the possible values are Peak, Average, and ALC gain. For a two-tone signal with maximum audio amplitude the Average ALC value is -3.0 dB, while the Peak value is 0.0 dB. Therefore I personally prefer the Peak value here and made it the default in piHPSDR: if the value is less than zero, one can and should increase either the amplitude of the incoming audio signal (e.g. boost the microphone preamp) or move the Mic gain slider to the right. The reason is, that PureSignal only works if the TX audio input has maximum amplitude, so you can put the drive slider to zero, then put the radio into TX mode, whistle into the microphone and slowly increase the Mic gain until the ALC value shown is only slightly less than zero.

5.5 The `XVTR` (Transverter) Menu

The `XVTR` menu lets you define additional bands that you can work on using transverters. The bands should normally be beyond the standard frequency range of the radio, otherwise the the calculation of a band from a given frequency will sometimes not work. To give an example, suppose you have a transverter which you can drive with frequencies between 28 and 30 MHz and which will convert them to the frequency range 144 to 146 MHz, and which will receive frequencies in that range and mix them down to the 10m band. The data you have to enter in the `XVTR` menu (use the first free entry) are as follows:

`Title` In this column, enter a name for your band. You can choose whatever name you want, this is the one that will be displayed in the `Band` menu. In the present example, use "144" or "144 MHz" or "2m".

`Min Freq` Enter the lowest frequeny of the transverter band in MHz, in the present case, 144.

`Max Freq` Enter the highest frequeny of the transverter band in MHz, in the present case, 146.

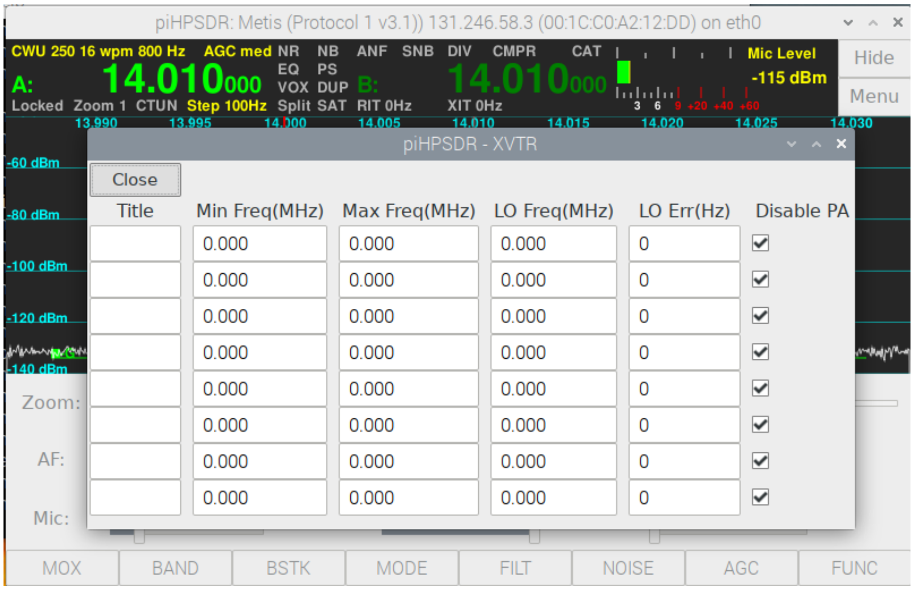


Fig. 5.7: The XVTR (transverter) menu.

LO Freq This is the frequency offset (in MHz) between the radio frequency and the operating frequency. In this case, use 116. From this offset, radio frequencies between 28 and 30 MHz will be used for operating frequencies between 144 and 146 MHz.

LO Err This entry can be used for a fine calibration of the frequency. The value (in Hz) is added to the local oscillator (LO) freq in Mhz.

Disable PA This checkbox indicates that the PA of the radio should be disabled when using the transverter band. This implies that the radio has some sort of low-power output that is used to drive the transverter.

Chapter 6

The Main Menu: VFO-related menus

In this chapter we discuss the menus from the second column of the main menu. These are all VFO-related menus.

6.1 The `VFO` menu

The `VFO` menu can be used for direct frequency entry and to enable/disable some frequently used options. If the menu is opened, it refers either to VFO-A or VFO-B. If opened via the main menu, it automatically refers to the VFO controlling the active receiver. The easiest (and therefore recommended) way to open the `VFO` menu is just to make a mouse click (or a touch screen press) into the VFO bar. If clicked in the left half of the VFO bar, the menu is opened for VFO-A and if clicked in the right half, it is opened for VFO-B. The `VFO` menu is shown in Fig. 6.1.

The ,,keypad" is used for direct frequency entry. You can enter digits and a decimal point. While entering a number the string entered so far is not only shown in the upper part of the `VFO` menu, but also (in yellow digits) in the VFO bar. The other buttons of the ,,keypad" have a special meaning:

`BS` Backspace. This cancels the last entered character (digit or decimal point).

`Hz` This enters the frequency ,,as is".

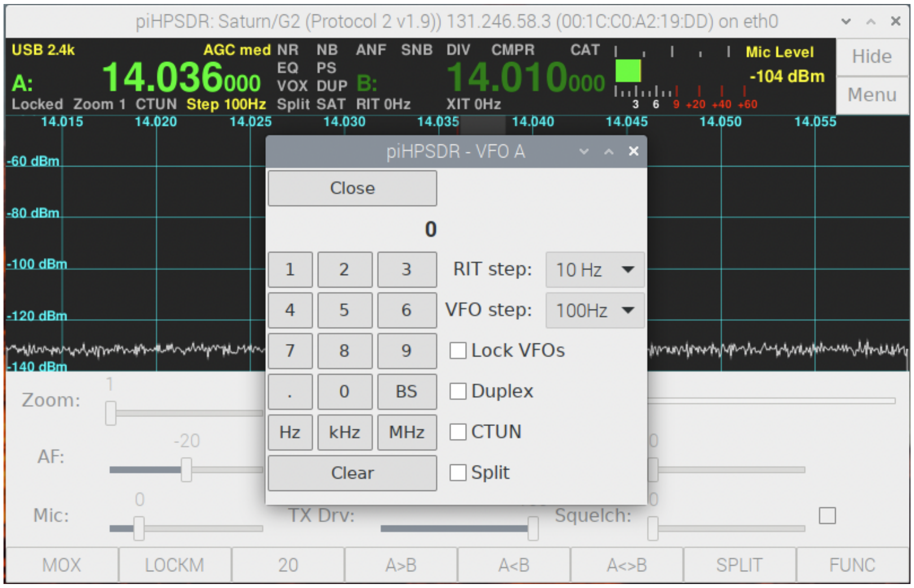


Fig. 6.1: The VFO menu.

kHz This multiplies the frequency just entered with 1000 and enters it. This means, the number entered is interpreted as a frequency in kHz.

MHz The string typed in so far is interpreted as a frequency in MHz and this frequency is transferred to the VFO.

Clear The string typed in so far is deleted, the VFO frequency is not updated.

The commands entered by clicking the buttons of the keypad in the VFO menu can also be entered by push-buttons from a GPIO or MIDI controller, see the NumPad commands in Appendix A.

In addition to frequency entry, the VFO menu offers a convenient way of changing some piHPDSR settings, simply because the VFO menu can be opened by a simple mouse click into the VFO bar.

Rit Step: In this pop-down menu, the RIT/XIT step size can be chosen (1/10/100 Hz).

VFO Step: In this pop-down menu, the VFO step size can be chosen. The VFO step sizes range from 1 Hz to 1 MHz.

Lock VFOs With this check box enabled, VFO frequencies cannot be changed by turning a VFO dial (GPIO or MIDI controller), or by clicking/dragging in the RX panel. Band changes (via the Band menu) and other VFO related functions still work.

Duplex and Split. With these check boxes, you can put the radio in Duplex or Split mode, see the Radio menu.

CTUN. With this check-box, you can put the VFO this menu is referring to into CTUN mode. In CTUN mode, the spectrum scope does not move when changing the frequency, rather, the RX ,,window" moves. CTUN mode does not affect TX operation.

6.2 The Band menu

The Band menu lets you change the band of the active receiver. It is shown in Fig. 6.2. When the menu opens, the button of the current band is highlighted.

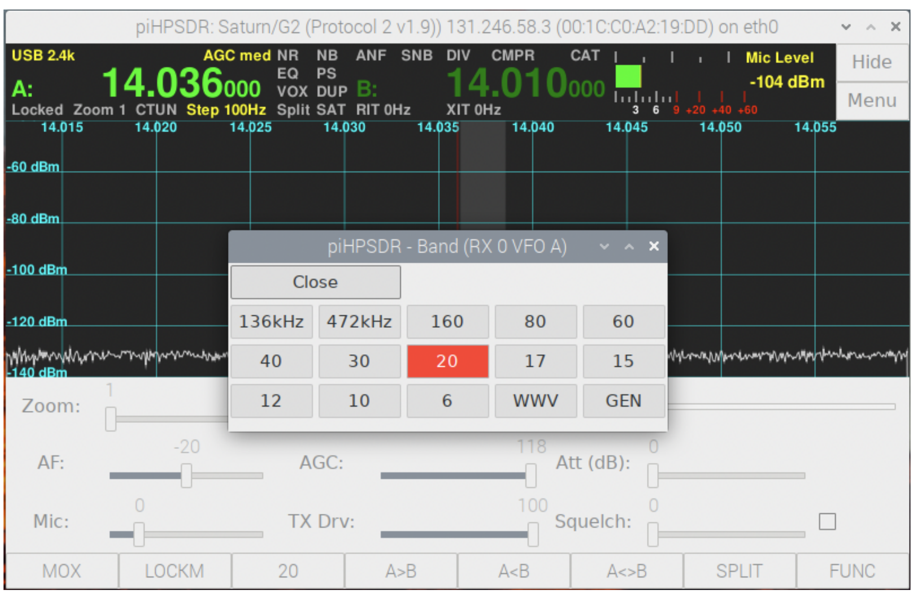


Fig. 6.2: The Band menu.

Pressing a button corresponding to another band, two things happen: first, if the active receiver is controlled by VFO-A, the current frequency is stored in the current bandstack (which is thus updated). Then, the new band is chosen, the frequency and mode are from the active bandstack entry of the new band. This means that if you switch to another band and shortly thereafter switch back to the original band, the frequency and mode is restored to what you had before.

If you hit the highlighted button, you will not change the band (since you hit the button of the current band) but instead will cycle through the band stack of that band (see the `BndStack` menu).

Note that the band menu may look different from the one shown here: there are many bands (24 bands plus up to 8 transverter bands) defined in piHPSDR. However, the bands that are outside of the radio's frequency limits are not shown. For example, a radio whose maximum frequency is 30 Mhz will not show the 6m band. The `GEN` (General) band encompasses the whole frequency range of the radio. If you set the frequency (e.g. via the `VFO` menu) to a frequency outside of any of the other bands, you will end up in the General band. If you have defined transverter bands (see the `XVTR` menu) they will be shown, with the title you have chosen, in the `Band` menu.

6.3 The `BndStack` (Bandstack) menu

For each band, the band stack is collection of frequency/mode pairs. The idea is that you can have preferred or recently visited frequencies to which you can easily come back. If you open the `BndStack` menu (Fig. 6.3), you see all these frequency/mode pairs, the band stack entry currently selected is highlighted.

If you press the highlighted button, the current frequency/mode is stored in that band stack entry. If you press another (non highlighted) bandstack entry, the current frequency/mode is stored in the highlighted band stack entry, and then the frequency/mode of the new entry becomes effective. Note that bandstack entries are only changed if the active receiver is controlled by VFO-A.

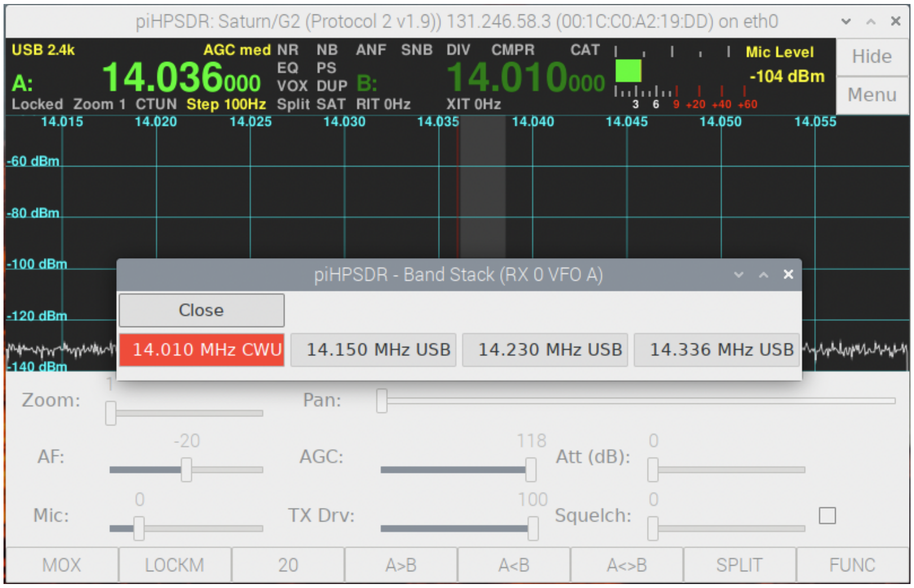


Fig. 6.3: The BndStack (Bandstack) menu.

6.4 The Mode menu

The Mode menu lets you change the mode of the active receiver, so you can switch, say, from LSB to CWU or DIGU. The mode menu simply lists the available modes, the current one is highlighted (Fig. 6.4).

Settings stored with the mode. Many settings such as filter choices, noise reduction and equalizer settings, and TX compressor settings are only reasonable with a specific mode in mind. Therefore, these settings are stored with the mode. If you later switch back to this mode, the settings that were effective the last time you used that mode are restored. This is the main reason why the USB and DIGU modes are separate (although technically they are the same), the same applies for LSB and DIGL. For digital operation, you will normally choose DIGU and have no noise reduction, no TX compressore, and no equalizer. For voice operation (USB/LSB) you may then have the noise reduction, equalizer and TX compressor settings as you like it. Then you can easily switch between DIGU and USB/LSB and have the correct settings automatically. The same applies to CWL/CWU where you will normally use different settings compared to both USB/LSB and DIGU.

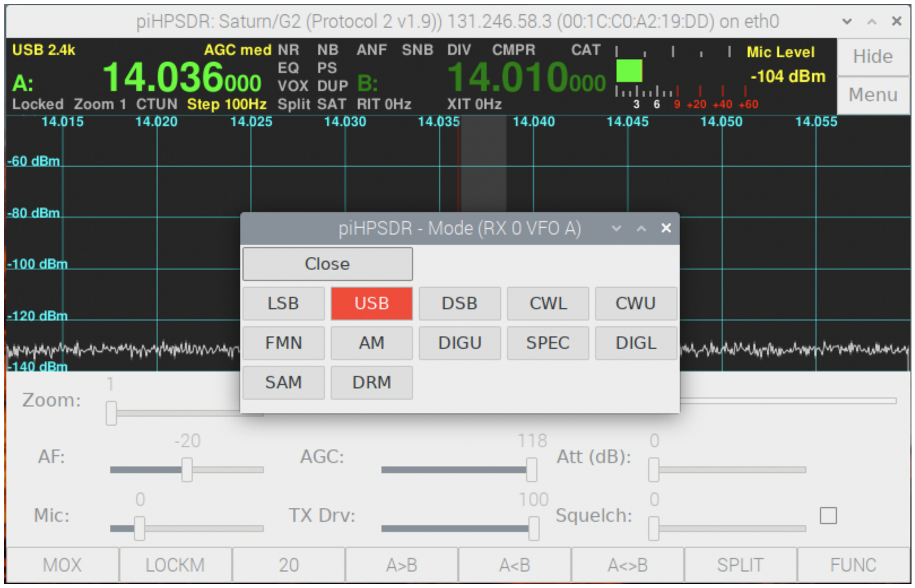


Fig. 6.4: The `Mode` menu.

6.5 The `Memory` menu

The `Memory` menu gives you access to five memory slots. The menu is shown in Fig. 6.5. You can store the current frequency and mode of the active receiver in any of the five slots by clicking a button in the leftmost columns (e.g. "Store M2"), or you can restore data from any slot. The current memory contents are shown in the right column (frequency, mode, and filter width) and clicking on of the five buttons there will restore the data. So if you have some often used frequencies (e.g. for a net), the `Memory` menu allows you to become QRV there with only few mouse clicks.

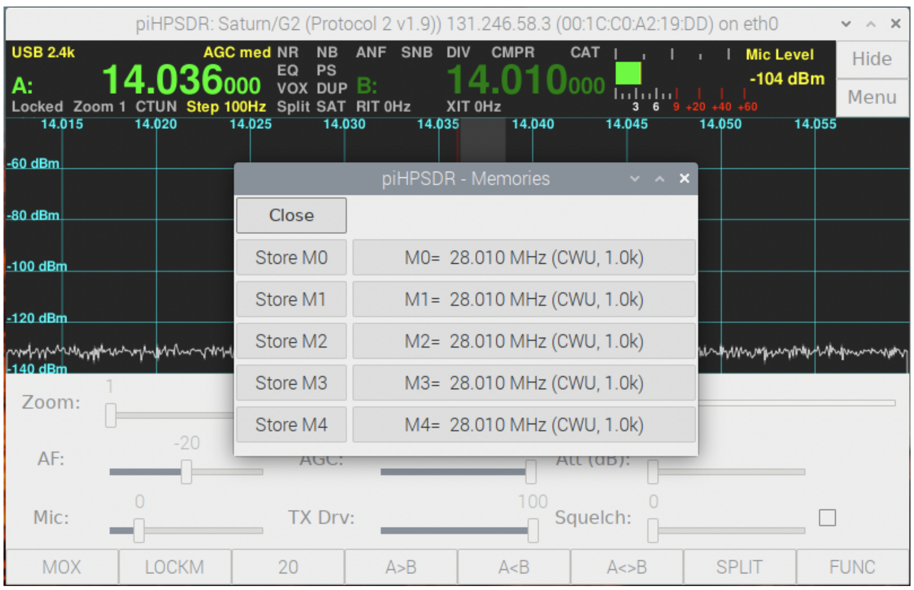


Fig. 6.5: The `Memory` menu.

Chapter 7

The Main Menu: RX-related menus

The second column of tghe main menu contains menus which allow you to change receiver settings.

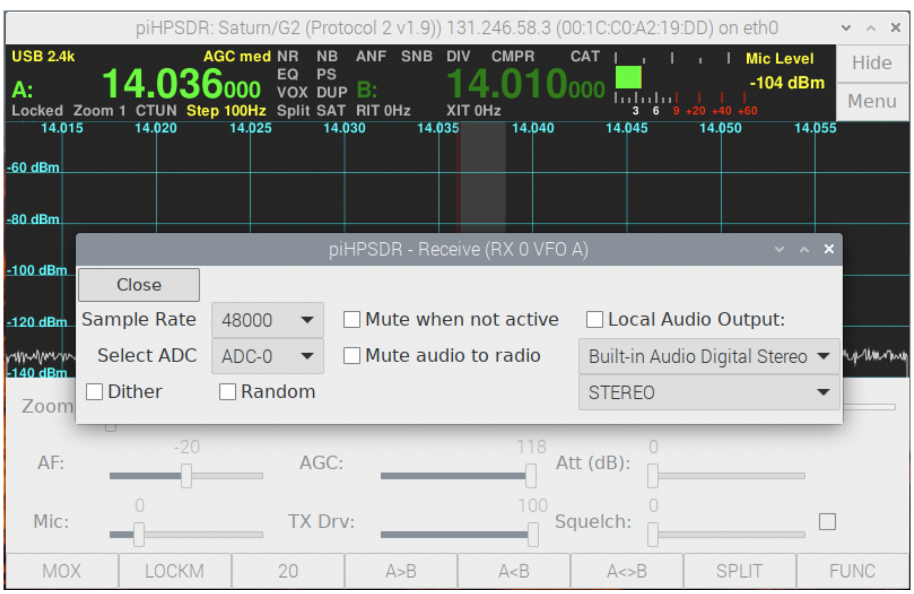


Fig. 7.1: The RX menu.

7.1 The RX Menu

Invoking the RX menu through the main menu always implies that the settings of the active receiver are to be modified. Using a mouse, you can also open the menu by a secondary click (using the right mouse button) into the receiver panel. This way, if piHPSDR is running two receivers, you can open the RX menu for both receivers (the active receiver as well as the other one), depending on in which panel you have right-clicked. Note secondary clicks are usually not possible with a touch screen. The menu is shown in Fig. 7.1.

`Sample Rate` This box is only shown for radios running P2, since only there the receivers can have an individual sample rate. For radios running P1 or radios accessed through the SoapySDR library, the sample rate is a global quantity that is modified throught the `Radio` menu (see above).

`Select ADC` This box is only shown if the radio has more than one analog-to-digital converter (ADC), such as Orion, Orion-II and Saturn boards. These radios have two ADCs so you can choose whether the receiver gets data from ADC0 or ADC1. For these radios, nearly all antenna jacks go to ADC0, while there is a jack denoted "RX2" (or similar) that is connected with ADC1. In most cases, ADC0 is used for normal operation, while ADC1 can be used for connecting a dedicated RX antenna.

Note: Diversity. When using `Diversity` reception, the ADC setting is overridden, since there data streams from ADC0 and ADC1 are combined (mixed).

7.2 The Filter menu

With the `Filter` menu, you can change the filter of the active receiver. There are ten fixed filters and two variable filters, see Fig. 7.2. It depends on the current mode which filters are at your disposal, and Fig. 7.2 is what you see for USB and LSB modes. The filter currently active is highlighted, and you can choose another filter simply clicking the button. For USB and LSB, the filters are such the low-frequency cut (in the audio domain) is at 150 Hz, so a 2.7k filter actually encompasses audio frequencies from 150 to 2850 Hz. With the variable filters (Var1 and Var2) you can be more flexible in the low audio frequency range. Here you can individually select the low- and high

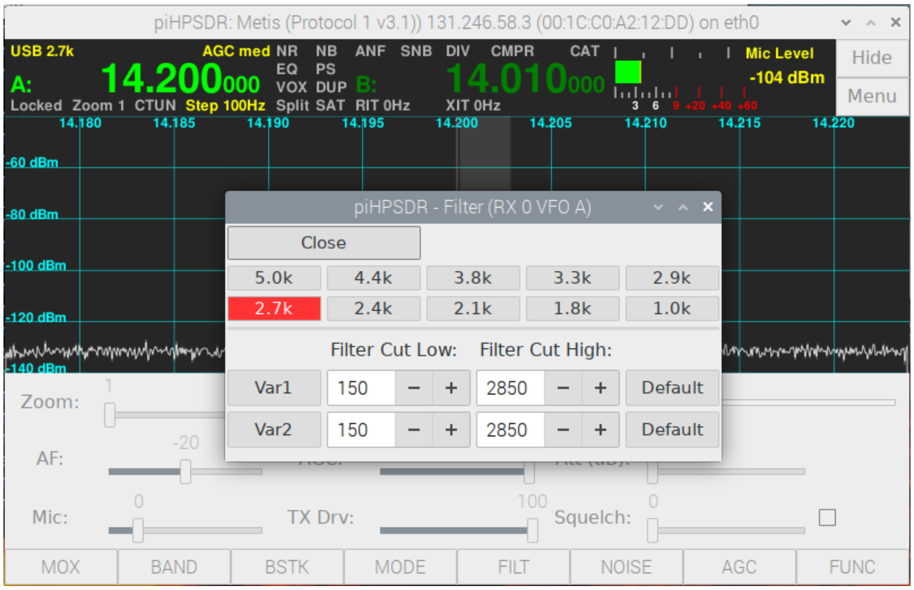


Fig. 7.2: The Filter menu (single side band modes).

frequency cut (both frequencies refer to the audio domain, and are thus both positive value for USB and LSB).

The pre-defined filters for the digital modes DIGU and DIGL are a little bit different. For filter widths up to 3 kHz, the filter is centered around 1500 Hz. For example, a 1.0k filter for DIGU/DIGL passes audio frequencies between 1000 and 2000 Hz.

For modes such as CW and AM, low/high cutoff frequencies have little meaning, so the Filter menu looks slightly different (Fig. 7.3). The fixed filters are designated by their width, they are centered around zero (for AM) or around the CW side tone frequency (for CWU and CWL). For the variable filters Var1 and Var2, the spin buttons can set the filter width and the filter shift. Normally you will not want to change the filter shift, but it may help in special cases.

Enable additional CW Audio peak filter. If the mode of the active receiver is CWL or CWU, there will be an addition check box in the top row of the menu. Here you can enable/disable an audio peak filter that is applied to the final audio output of the receiver, that is, on top of the regular filtering. The audio peak filter will only be effective in the CW modes, its center fre-

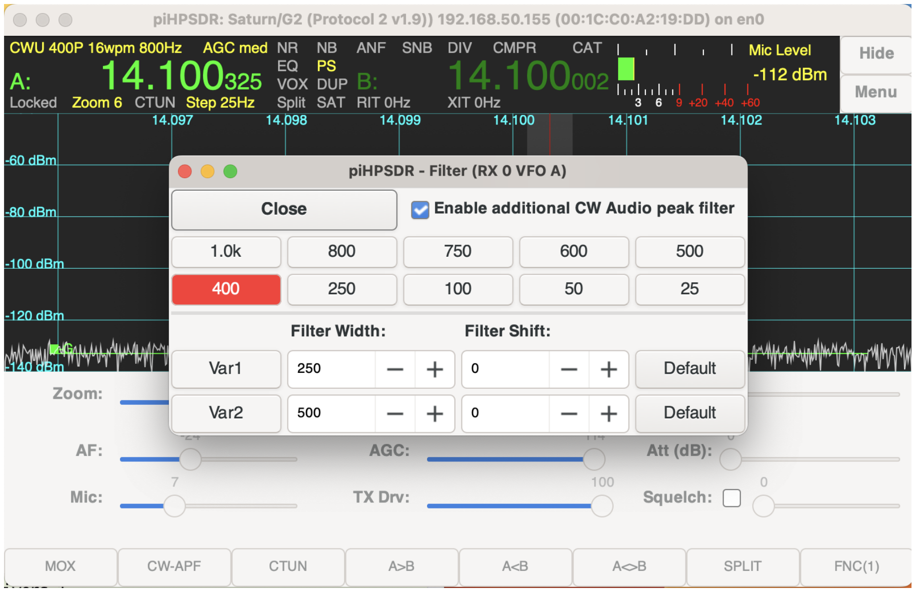


Fig. 7.3: The `Filter` menu for CWL/CWU.

quency is given by the CW side tone frequency and its width is automatically calculated, depending on the width of the primary filter. The main use of the audio peak filter is *not* fighting QRM, but to help tuning on a signal: suppose you use a CW side tone frequency of 800 Hz and a 500 Hz filter. Now you hear a CQ call, and if you do not have absolute hearing, it may happen that you tune such that the CQ call in your headphone sounds at 650 Hz, which means that you are 150 Hz off the frequency of your prospective QSO partner. If the other stations uses a very narrow CW filter, your signals may be be list un-heard since they are outside the pass-band of the other station. The reason why this easily happens is that the normal RX filters have a flat passband, that is, the signal you hear at 650 Hz is as strong as it would be for correct tuning. This is where the audio peak filter comes in: with the audio peak filter engaged, the received signal appears much louder when it is at 800 Hz (correct tuning) than if it were at 650 Hz. This implies that the width of the audio peak filter should be such that signals at the edges of the passband of your normal filter should still be audible. piHPSDR uses a width that equals one fourth of the width of the normal filter.

There is the function `CW Audio Peak Filter` that can be mapped on toolbar buttons or GPIO/MIDI buttons so you can quickly enable/disable the audio peak filter. In Fig. 7.3 you see the sign CW-APF (which is the short string for CW Audio Peak Filter on the second button from the left, and with this button you can quickly disable/enable the peak filter. I prefer to have the audio peak filter engaged when I answer to CQ calls, but to have it disabled when doing CQ calls myself.

7.3 The Noise Menu

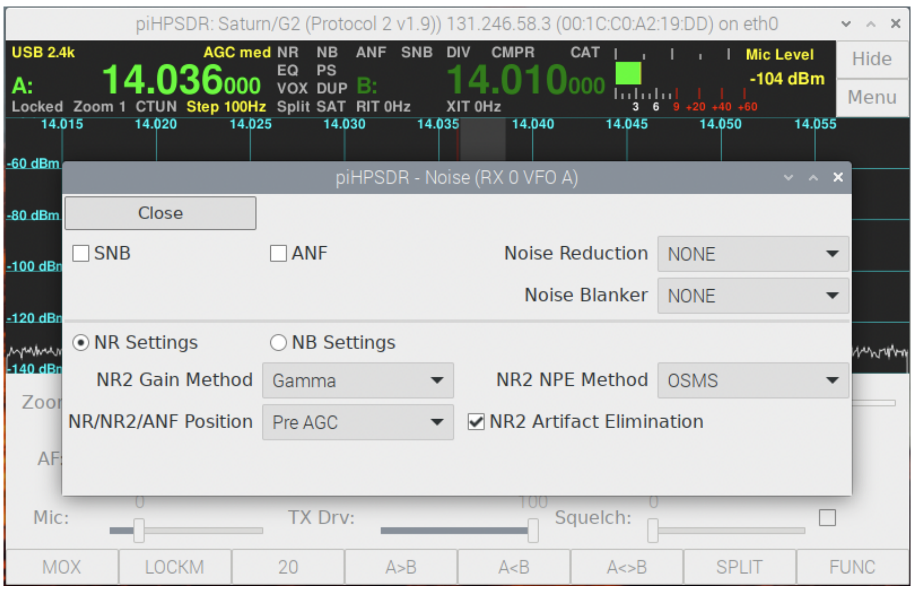


Fig. 7.4: The Noise menu (with NR settings).

With the Noise menu you can select a variety of noise reduction and/or noise blanker capabilities (Fig. 7.4). The upper part of the menu always looks the same, the lower part lets you fine-tune noise reduction or noise blanker parameters. For an in-depth explanation of the noise reduction and noise blanker capabilites, the reader is referred to the WDSP manual.

SNB This check box lets you enable/disable the spectral noise blanker.

ANF This check box enables/disables the automatic notch filter. The ANF is very good at eliminating a single-tone QRM carrier in SSB modes. It goes without saying that activating the ANF in CW is detrimental rather than beneficial, because here the signal is of the type the ANF tries to eliminate.

`Noise Reduction` With this pop-down menu, you can choose the type of noise reduction (no noise reduction, NR1 or NR2).

`Noise Blanker` With this pop-down menu, you can choose the type of noise blanker (no noise blanker, the preemptive wideband blanker NB or the interpolating widetband blanker NB2).

`NR Settings/NB Settings` Choosing one of the two buttons determines whether the lower part of the menu offers fine-tuning of the noise reduction or noise blanker settings. The set up for changing the noise reduction settings is shown in Fig. 7.4, below (Fig 7.5) you find the set up for changing the noise blanker settings. We discuss the noise reduction settings first, but note again for the details you have to study the WDSP manual.

`NR2 Gain Method` The available choices for the NR2 noise reduction here are Linear, Log, and Gamma, where Gamma is the default.

`NR2 NPE Method` The available choices for the NR2 noise reduction here are OSMS and MMSE, where OSMS is the default.

`NR...Position` In the RX chain, the noise reduction can be placed before or after the automatic gain control (AGC). The choice here refers to *all* noise reduction capabilities (SNB, ANF, NR1, NR2).

`NR2 Artifact Elimination` The NR2 noise reduction algorithm is prone to producing artifacts, so there is an option to reduce such artifacts which should normally be checked (artifact elminiation ,,on").

Note the noise blanker works very different from the noise reduction, since noise blanking is applied to the original RX IQ samples before any frequency shifts etc. take place. If you have a single source of noise (e.g. a Plasma TV) that drives you crazy, it is worth the effort to play around with the NB2 parameters, especially the timings. Different QRM sources will require different parameters! The default parameters have been proven useful for many situations, but for you a different setting may produce better results! The options to control the noise blanker algorithms are:

`NB2 Mode` The available choices for the interpolating NB2 noise blanker here are Zero, Sample&Hold, Mean Hold, Hold Sample, and Interpolate.

`NB Slew time/Lag time/Lead time/Threshold` These parameters apply both to NB and NB2. piHPSDR currently does not allow to have a separate set of parameters for NB and NB2.

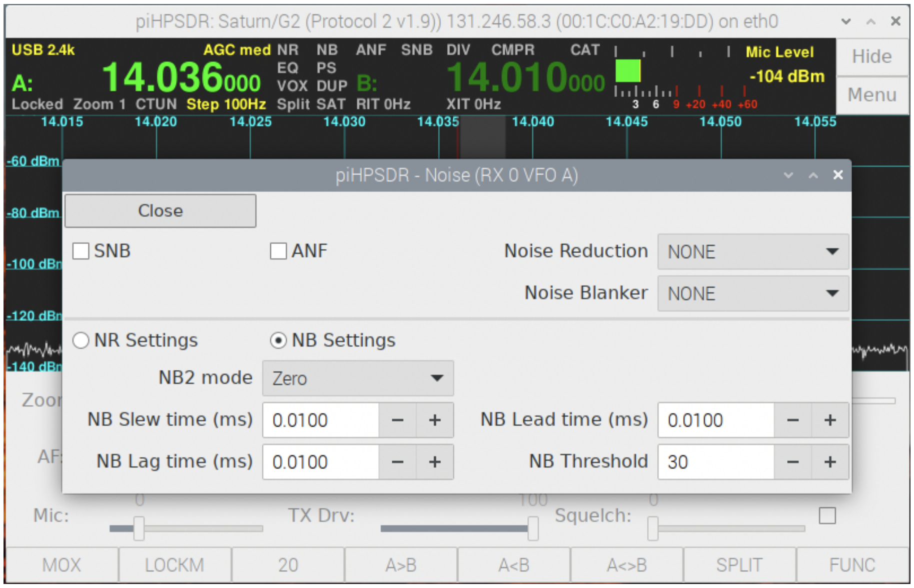


Fig. 7.5: The Noise menu (with NB settings).

7.4 The AGC Menu

Only few parameters can be controlled via the automatic gain control (AGC) menu. The first is the AGC time constant, which can be Off (no AGC), Long, Slow, Medium, and Fast. A very long AGC time constant protects your ears, but it also means that the receiver becomes „deaf" for a rather long time after a strong QRM burst. This phenomenon is known as "AGC pumping". Generally, if you do SSB on a quiet band, the AGC time constant can be longer, for CW on the other hand, I personally prefer short time constants (Medium or Fast).

The AGC Hang Threshold is only effective if the AGC time constant is Long or Slow, since the AGC hang time is turned off for Medium and Fast. In this case, the RX spectrum scope not only shows the „normal" AGC line in green, but also the hang threshold line in orange.

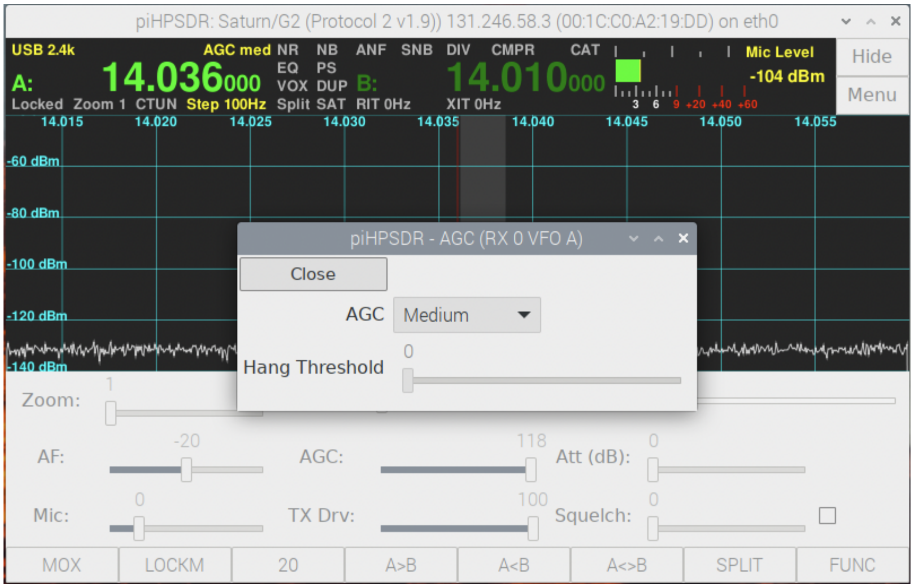


Fig. 7.6: The `AGC` menu.

7.5 The `Diversity` Menu

`Diversity` is a very powerful tool to improve reception by using two different antennas and two ADCs. To explain how it works, suppose you live in a house which produces a lot of local QRM. Your ,,normal" antenna will pick up the wanted DX signals, but also a lot of noise that originates somewhere in your house. Now suppose you have a second receive-only antenna placed in your house that will predominantly pick up your local QRM and only very little DX signal.

Of course, this RX-only antenna does not deliver anything useful at first sight. But, imagine you could shift the phase and the amplitude of the signal of the in-house antenna such that it exactly opposes the local QRM picked up by your DX antenna! Adding this (phase shifted and amplitude adjusted) signal from your in-house antenna to what comes from your DX antenna will produce a signal where the local QRM is largely eliminated while the DX signal is only weakly affected. This is what `Diversity` is all about.

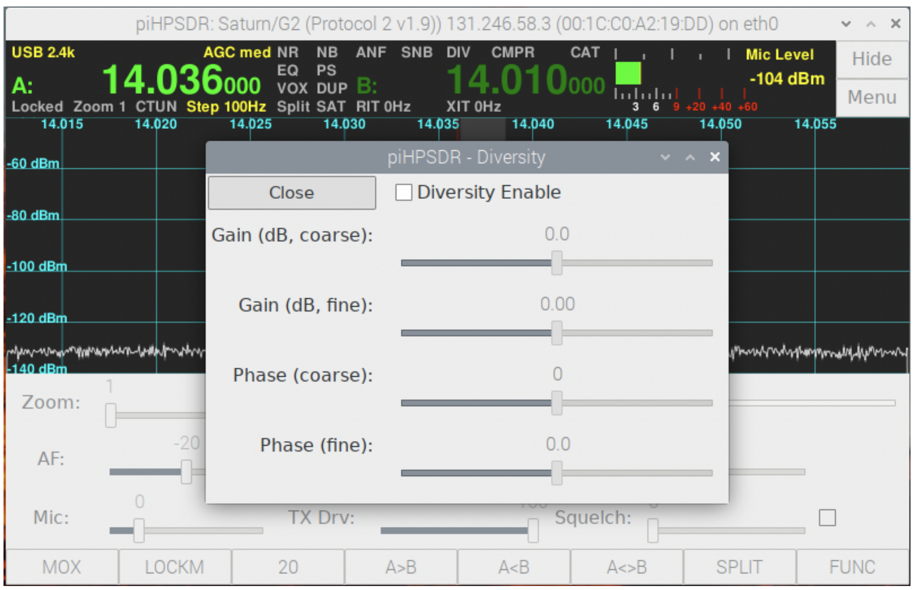


Fig. 7.7: The `Diversity` menu.

Chapter 8

The Main Menu: TX-related menus

8.1 The TX Menu

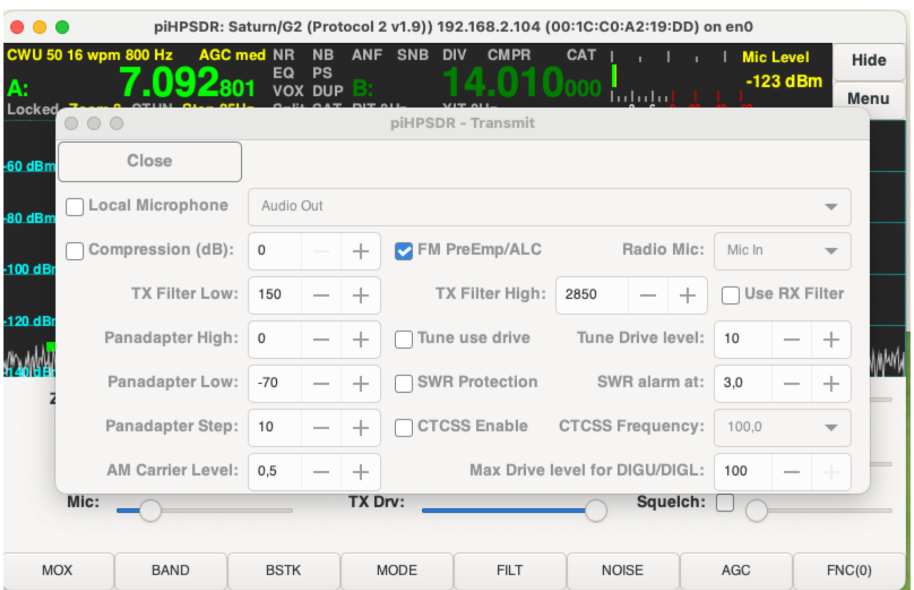


Fig. 8.1: The TX menu.

The TX menu can be opened from the main menu, or just by a secondary mouse click into the TX panadapter (while transmitting). The menu is shown in Fig. 8.1.

`Local Microphone`. If this box is checked, the TX audio samples come from a soundcard attached to the host computer, or from a virtual audio cable. The sound device can be selected from the pop-down menu to the right. This menu is intentionally very broad since audio device names can be very long. This check box, and the pop-down menu, is absent if there are no output sound devices available.

Note: If the radio has the possiblity to connect a microphone, and if PTT comes from the radio, the radio microphone samples and the local (sound device) microphone samples are mixed (added). This is very convenient if one does SSB with a microphone attached to the radio, and digital modes with a local sound device or virtual audio cable: when doing SSB, the local sound device normally produces no audio, and while pressing PTT at the microphone, you can work SSB normally. So you can go from digital mode to SSB without the need to change the microphone setup in the TX menu.

`Compression (dB):` With this box the TX compressor can be enabled/disabled. The compression level (0-20 dB) can be chosen in the spin button to the right.

Note: The compressor on/off flag, as well as the compression level, is ,,stored with the mode". So it is possible to have the compressor enabled for SSB (LSB/USB) and disabled for digi modes (DIGL/DIGU), and when switching modes, the compressor settings for the new mode are automatically restored.

`FM PreEmp/ALC`. When transmitting FM, the audio input signals are ,,emphasized". This means that from 300 to 3000 Hz (the usual range of audio frequencies in amateur radio FM), there is a 6 dB per octave (20 dB per decate) that leads to a 20 dB damping of an input signal at 300 Hz (and no damping ad 3000 Hz). This manipulation can be done before or after the TX ALC. If it is done *after* the TX ALC, the transmitter cannot be over-driven but you also cannot drive the transmitter to full power with a low frequency signal. Many operators consider the resulting signal ,,too thin" and prefer that the pre-emphasis is done *before* TX ALC, together with a general overall amplification of the audio signal. These two possibilities can be selected with this check box. If checked, pre-emphasize takes place *before* ALC. This makes the FM signal ,,thicker" because the TX audio input is boosted by 15 dB when doing FM. If the checkbox is not checked, pre-emphasize takes place *after* ALC, which makes the 15 dB audio boost ineffective because it is ,,corrected" by the ALC. So if you want more ,,punch" in FM, check this box, but be aware that strong high-frequency audio input signals may lead to TX over drive. With normal speech this is not likely to happen because a large part of its power is in the low-frequency region.

`Radio Mic:` This text, and the pop-down menu to its right, only occurs if a microphone can be connected to the radio. The pop-down menu lets you choose between `Mic In` which means that a microphone can be connected to the microphone input jack, `Mic Boost`, which additionally switches on a hardware 20 dB mic amp, and `Line In` which means that the "Line In" jack of the radio is used for the audio samples transferred from the radio to the host computer. This is part of the HPSDR protocol, it may well happen that your radio has a microphone jack but no line-in input. The optional 20 dB preamp may be necessary when connecting a dynamic microphone whose input level (few mV) is considerably lower that that of a condensor (electret) microphone, or a dynamic microphone with built-in preamp.

`TX Filter low:` With this spin button you can set the low cut of the TX filter. The frequency refers to the audio domain.

`TX Filter high:` With this spin button you can set the high cut of the TX filter. The frequency refers to the audio domain.

`Use RX Filter` If this check box is enabled, the TX filter low/high cuts are ignored, and the filter edges of the current RX filter are used instead.

`Panadapter Low:` This spin button sets the lower edge (in dBm) of the TX panadapter.

`Panadapter High:` This spin button sets the upper edge (in dBm) of the TX panadapter.

`Panadapter Step:` This spin button determines how many horizontal lines are drawn on the TX panadapter. If set to 10, for example, there will be a horizontal line for every multiple of 10 dBm.

`AM carrier level:` This sets the AM carrier level for the AM modulator. If set to zero, there is no carrier and the signal is a DSB signal. A reasonable value is 0.5 which leads to 100% modulation. Values larger than 0.5 have less than 100% modulation. This means too much power goes into the carrier.

`Tune use Drive` If this box is checked, TUNEing will be done with the power that corresponds to the current position of the drive slider, and the tune drive level is ignored.

`Tune Drive Level` The value that can be adjusted with this spin box is the virtual position of the drive slider while TUNEing. This value is ignored if the `Tune use Drive` box is checked.

`SWR protection` If this box is checked, a very simple SWR protection is enabled. If the SWR exceeds the threshold value (see next point), the drive slider is set to zero. The SWR protection is disabled while TUNEing.

`SWR alarm at` The spin button to the right determines the SWR threshold. If the SWR is beyond the threshold, the SWR reported in the meter turns red. If SWR protection is enabled, the drive slider is set to zero if the SWR exceeds the threshold.

`CTCSS Enable` (FM only!) This checkbox enables/disables CTCSS (continuous tone coded squelch system). If enabled, a low-frequency tone is transmitted together with the normal TX audio. This can be used to trigger repeaters, or any other function implemented on the other side. The frequency itself can be chosen with the following menu point:

`CTCSS Frequency` This pop-down menu lets you choose the CTCSS frequency. This choice has no effect if CTCSS is disabled. The frequency list includes 38 standard TIA/EIA-603-D CTCSS frequencies between 67.0 and 250.3 Hz.

`Mac Drive level for DIGU/DIGL` This spin button restricts the range of the drive slider from 0 to the chosen value for the DIGU and DIGL modes. If the value is 100, this has no effect. The primary use of this menu point is PA protection, since many digital modes (unlike SSB voice) are constantly transmitting at full power.

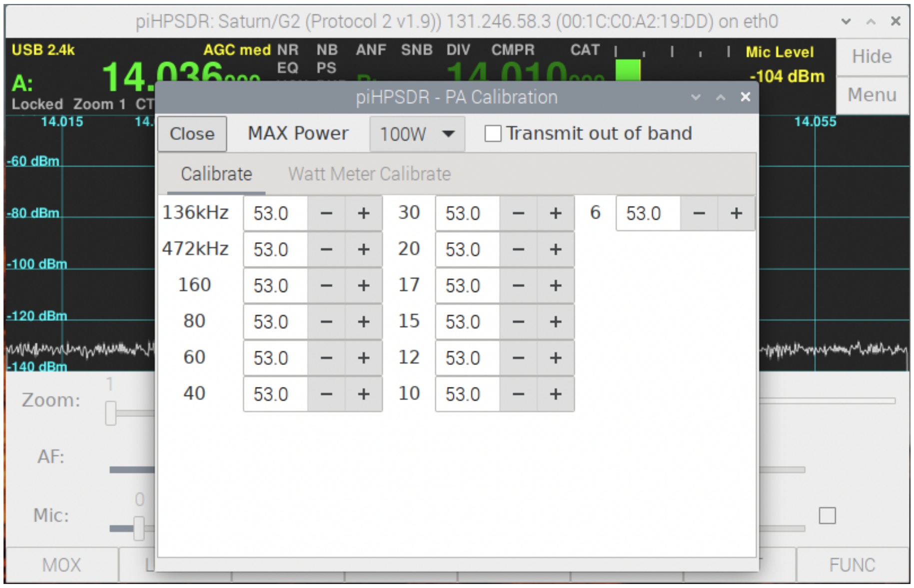
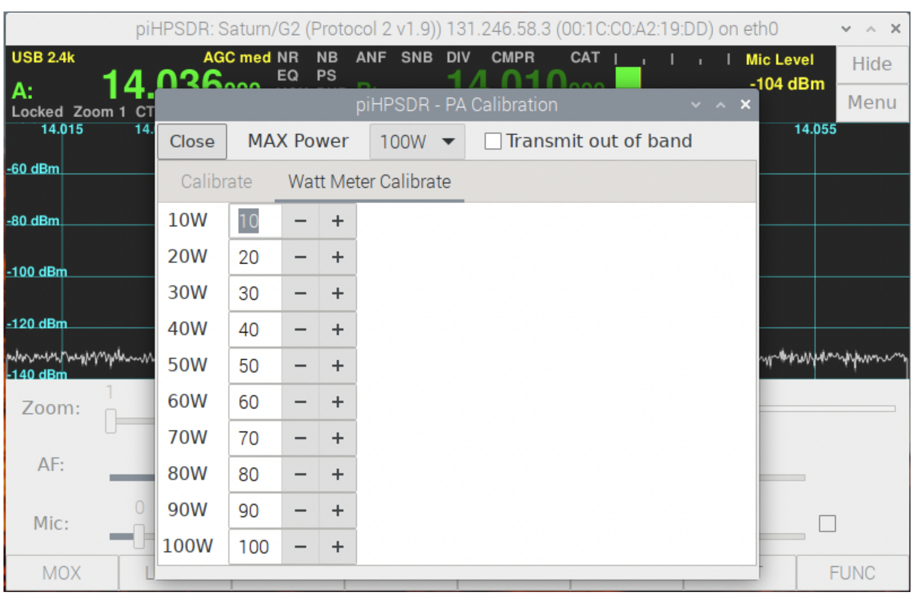
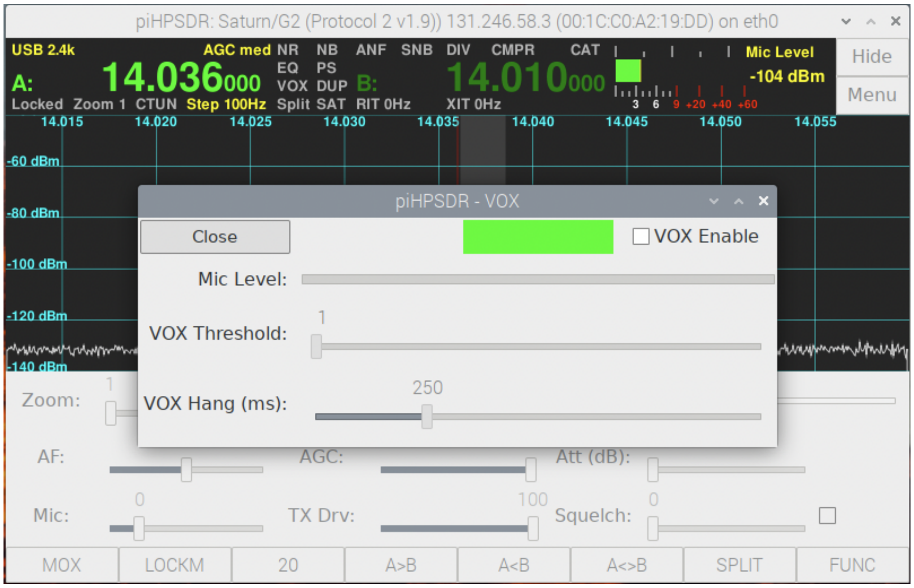


Fig. 8.2: The PA menu, PA calibration screen

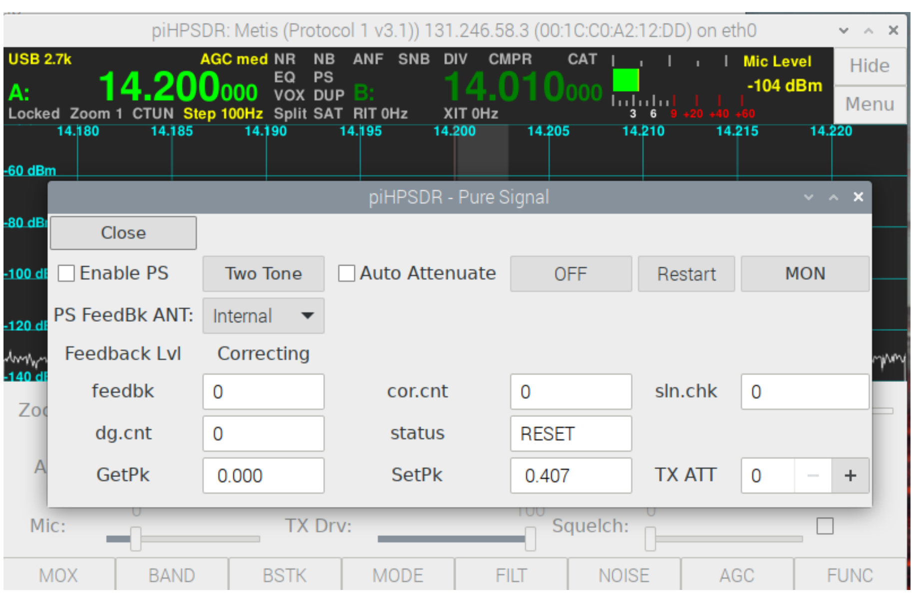
8.2 The PA Menu

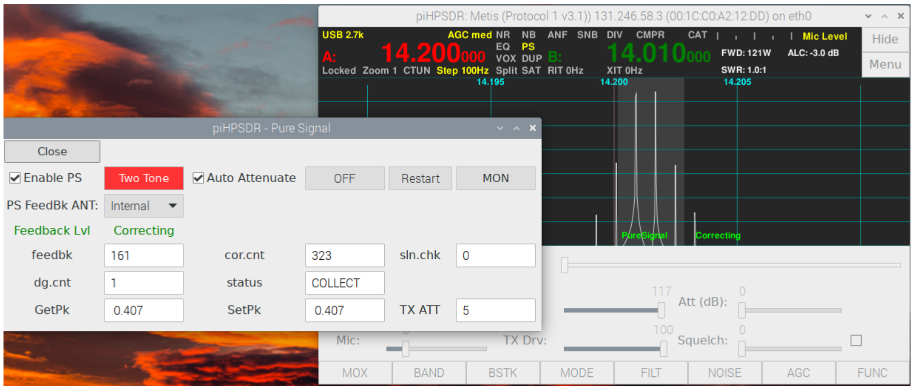


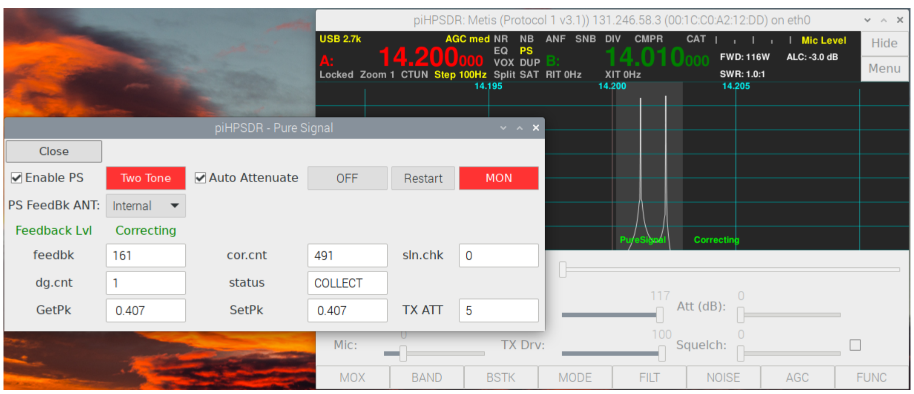
8.3 The VOX Menu

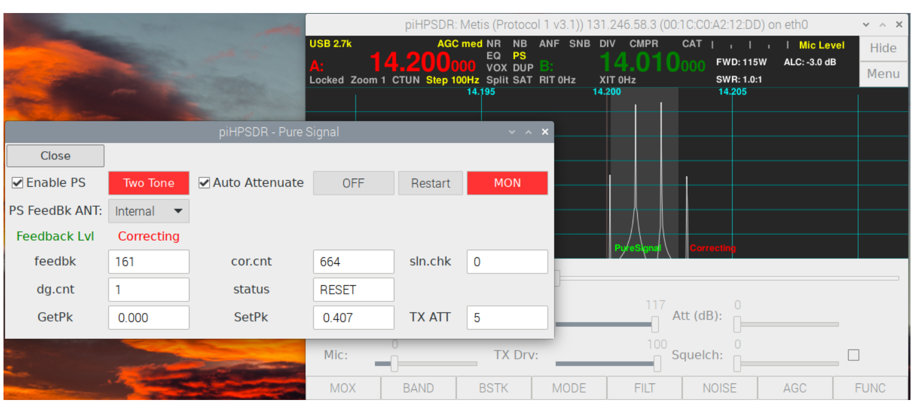


8.4 The PS (PureSignal) Menu

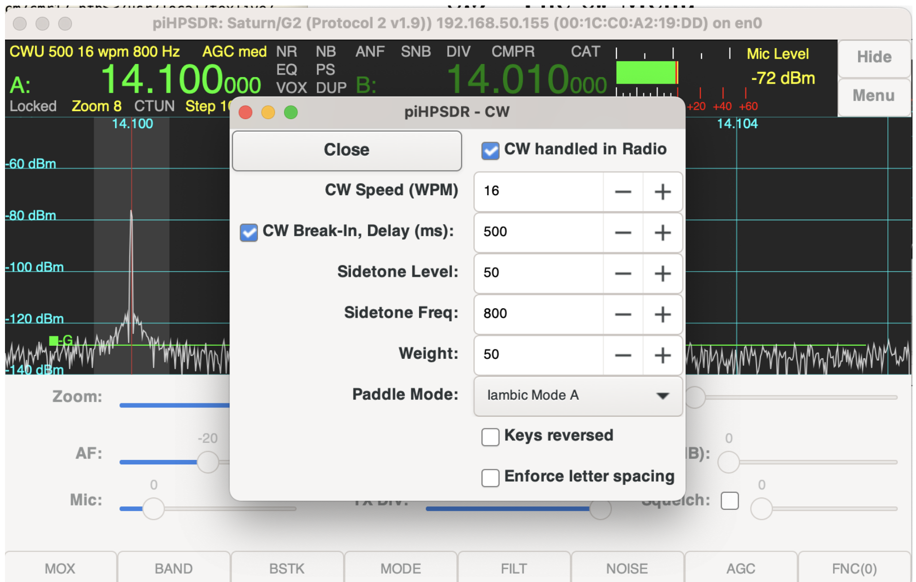



piHPSDR: Metis (Protocol 1 v3.1)) 131.246.58.3 (00:1C:C0:A2:12:DD) on eth0
piHPSDR - Pure Signal
Close
Enable PS
Two Tone
Auto Attenuate
OFF
Restart
MON
PS FeedBk ANT:
Internal
Feedback Lvl
Correcting
feedbk
161
cor.cnt
323
sln.chk
0
dg.cnt
1
status
COLLECT
GetPk
0.407
SetPk
0.407
TX ATT
5
MOX
BAND
BSTK
MODE
FILT
NOISE
AGC
FUNC


piHPSDR: Metis (Protocol 1 v3.1)) 131.246.58.3 (00:1C:C0:A2:12:DD) on eth0
piHPSDR - Pure Signal
Close
Enable PS
Two Tone
Auto Attenuate
OFF
Restart
MON
PS FeedBk ANT:
Internal
Feedback Lvl
Correcting
feedbk
161
cor.cnt
491
sln.chk
0
dg.cnt
1
status
COLLECT
GetPk
0.407
SetPk
0.407
TX ATT
5
MOX
BAND
BSTK
MODE
FILT
NOISE
AGC
FUNC


piHPSDR: Metis (Protocol 1 v3.1)) 131.246.58.3 (00:1C:C0:A2:12:DD) on eth0
piHPSDR - Pure Signal
Close
Enable PS
Two Tone
Auto Attenuate
OFF
Restart
MON
PS FeedBk ANT:
Internal
Feedback Lvl
Correcting
feedbk
161
cor.cnt
664
sln.chk
0
dg.cnt
1
status
RESET
GetPk
0.000
SetPk
0.407
TX ATT
5
MOX
BAND
BSTK
MODE
FILT
NOISE
AGC
FUNC

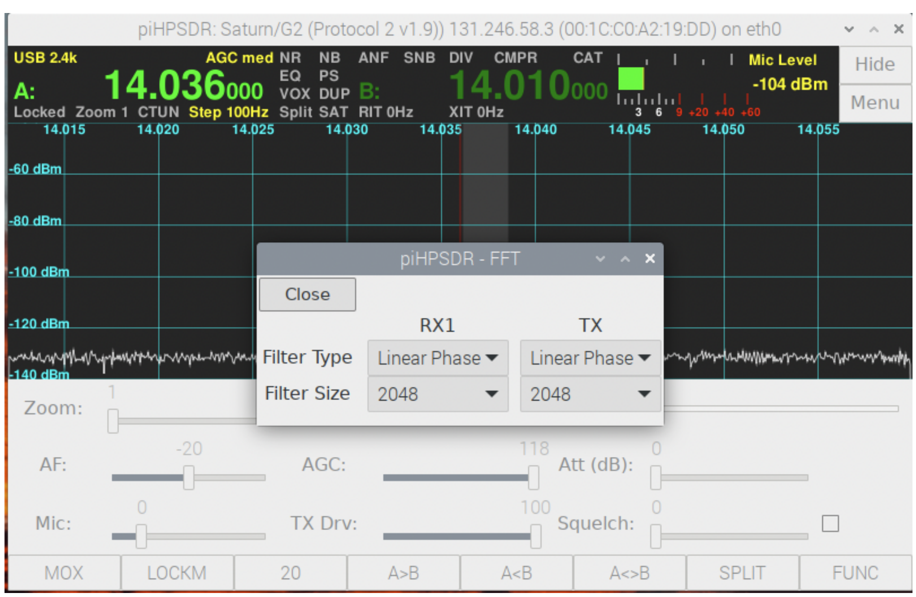
8.5 The CW Menu



Chapter 9

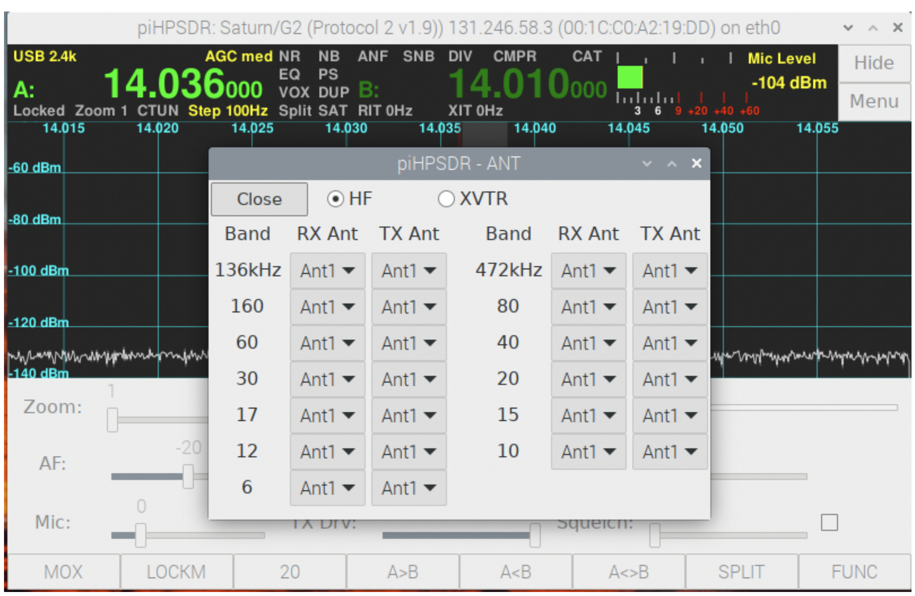
The Main Menu: menus for RX and TX

9.1 The FFT (Signal Processing) Menu

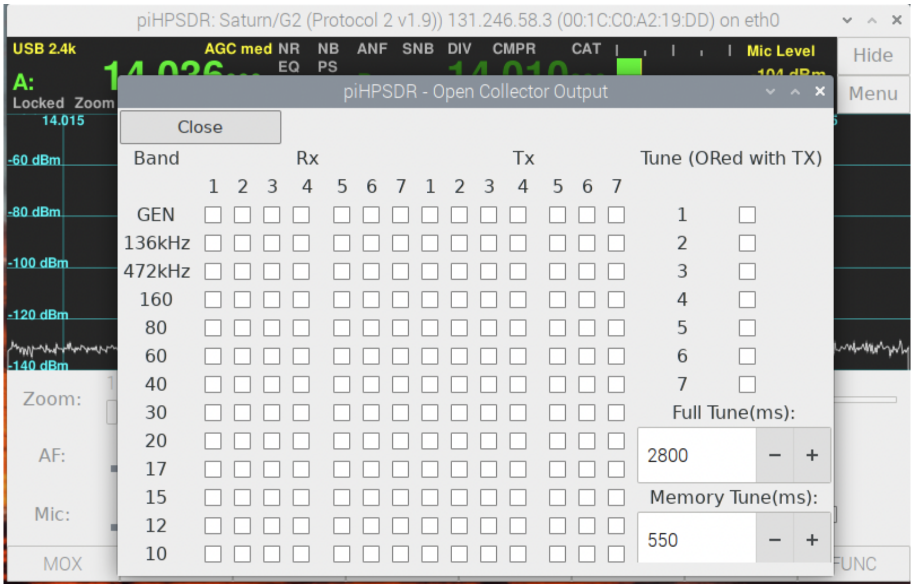


9.2 The Equalizer Menu

9.3 The Ant (Antenna) Menu



9.4 The `OC` (OpenCollector) Menu



Chapter 10

The Main Menu: controlling piHPSDR

In this chapter, the customization of the toolbar (at the bottom of the piHPSDR window), as well as how to configure GPIO and MIDI controllers, is described. Furthermore, in this chapter we discuss the RIGCTL menu which allows controlling piHPSDR by some external program such as a logbook or contest program, via standardized CAT commands that can be sent to piHPSDR either over a serial line or via TCP.

Note for Controller1 owners: The eight switches (push-buttons) of the controller, that a positioned below the screen, are bound to the eight toolbar buttons on the screen. Therefore, there is no "Switches" menu for this controller, and the switches are implicitly configured via the Toolbar menu.

10.1 The `Toolbar` Menu

We start with the "Toolbar" menu, that can be found at the top of the rightmost column in the main menu. The toolbar consists of eight buttons that can be assigned to a set of eight functions. There are six such sets, and pressing the FUNC button cycles through these six sets. If you click the Toolbar button, a menu pops up as shown in Fig. 10.1.

The six lines denoted `F0` through `F5` indicate the six different layers. The

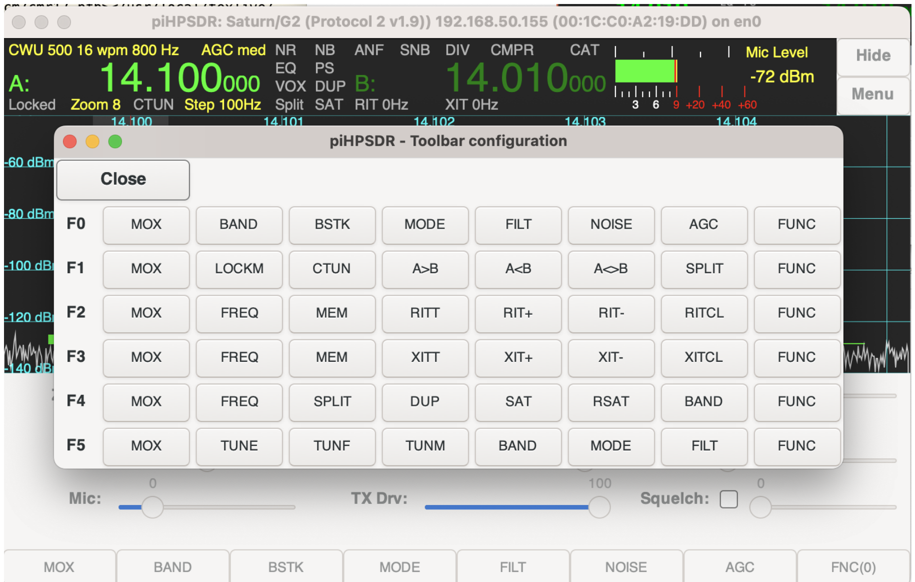


Fig. 10.1: The Toolbar menu, just opened.

text on the rightmost toolbar button, `FNC(0)`, indicates that the set 0 is the one that is currently active, and the labels in the `F0` line exactly match the labels in the toolbar. In this menu, the possible actions (that can be bound to a button) are written with the short text (see Appendix A), since this is the text that is printed on the toolbar buttons. If one now clicks (just an example) the `CTUN` button in the line `F1`, an ,,action dialog" pops up that looks as follows:

The current action selected (`CTUN`) is high-lighted. Lists of possible actions can be rather long, so it might be necessary that you have to scroll up or down in such an action dialog until you have found what you were looking for. Now (again just an example) the button `Band 20` has been clicked in the action dialog, such that it gets high-lighted:

If one now closes the action dialog by clicking the `OK` button, the third button in the `F1` line of the toolbar menu has changed, it now gives the short text (`20`) of the action, which will switch the active receiver to the 20m band (see the explanation of all the actions in Appendix A).

You also see that the toolbar has not changed, because we have just changed the `F1` set, while currently the `F0` set is active. If one now, however, clicks

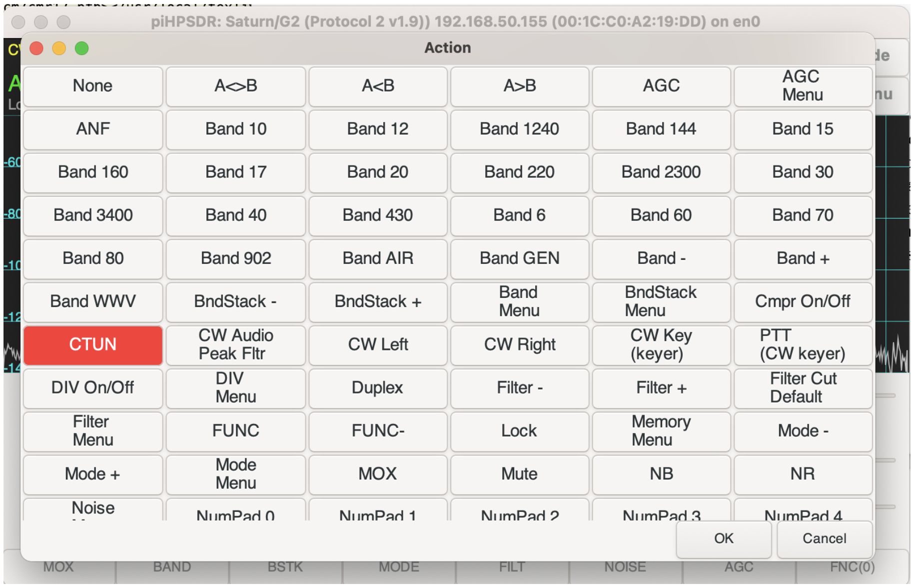


Fig. 10.2: Toolbar menu. Changing second button in `F1` layer.

the rightmost toolbar button with the text `FNC(0)` one advances to the `F1` set and the toolbar labels are updated:

It can be seen that the text of the first seven toolbar buttons has changed to reflect the functions of the `F1` set, and also the rightmost button (which is always mapped to `Function`) has changed to `FNC(1)` in oder to indicate the `F1` layer is now active. For mouse users (only), a secondary click on the rightmost toolbar button cycles through the layers in in reverse order.

Note that it is not possible to change the assignment of the rightmost button of the toolbar, it will always be assigned to `FUNC`, since if one has no access to this function, one no longer cycle through the sets.

Fig. 10.3: Just selected `Band20`.

10.2 The `RigCtl` (Rig control, or CAT) Menu

10.3 The `MIDI` Menu

10.4 The `Encoders` Menu

10.5 The `Switches` Menu

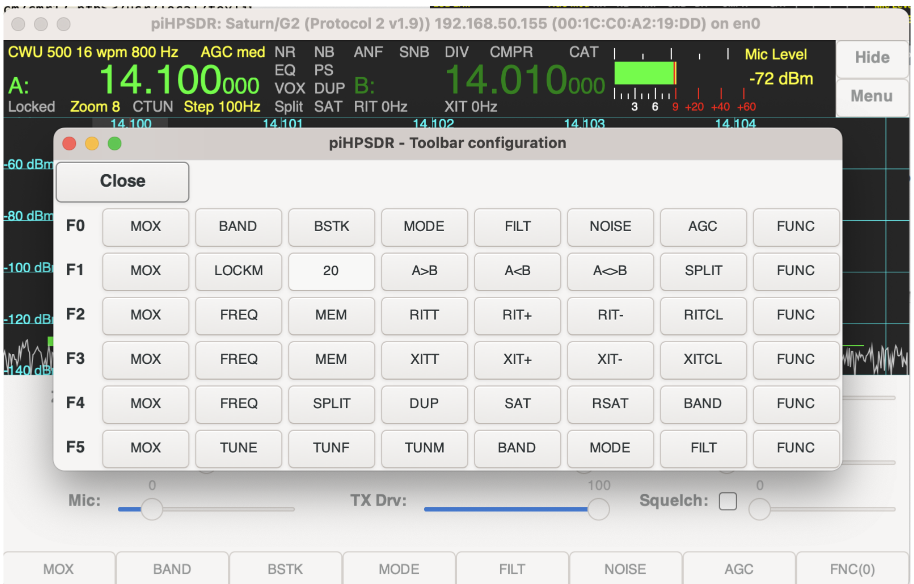


Fig. 10.4: Toolbar assignment accomplished.

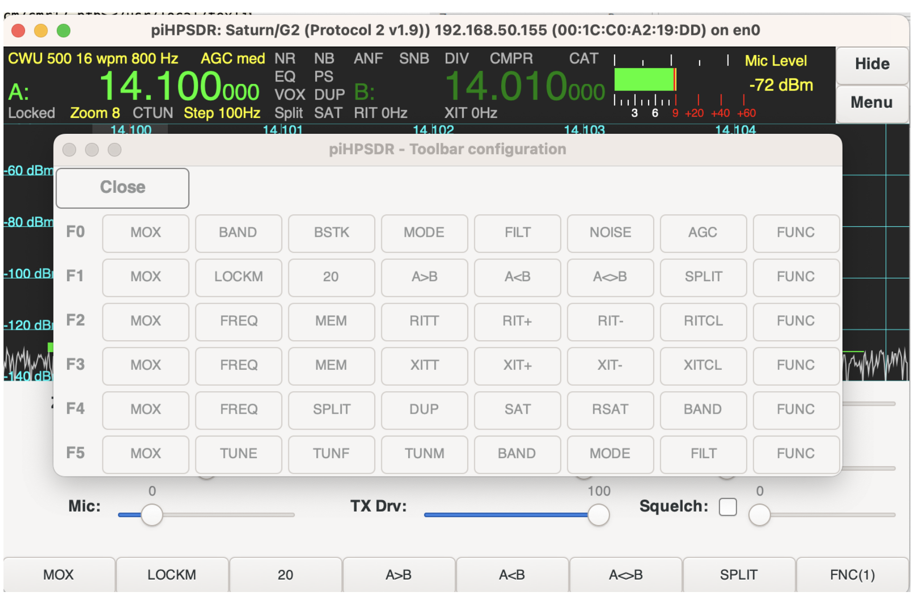
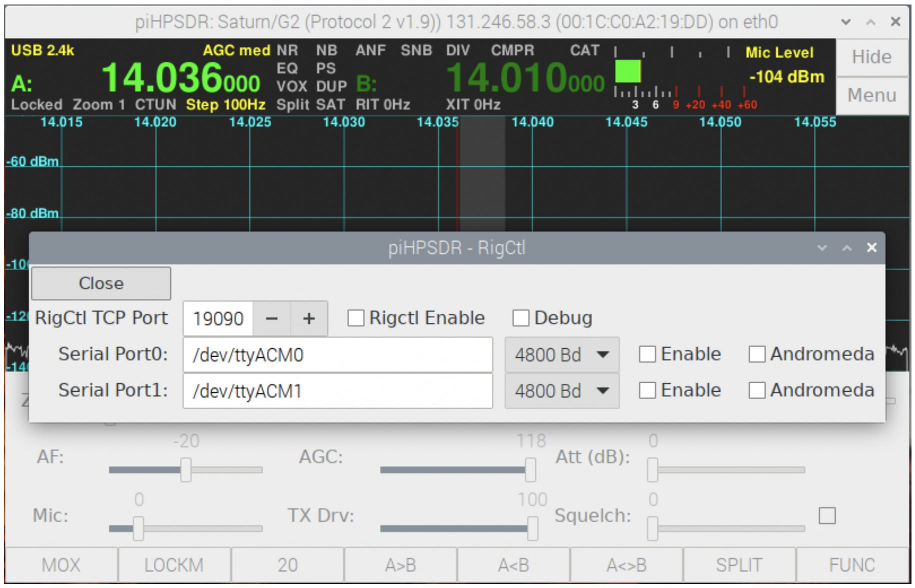
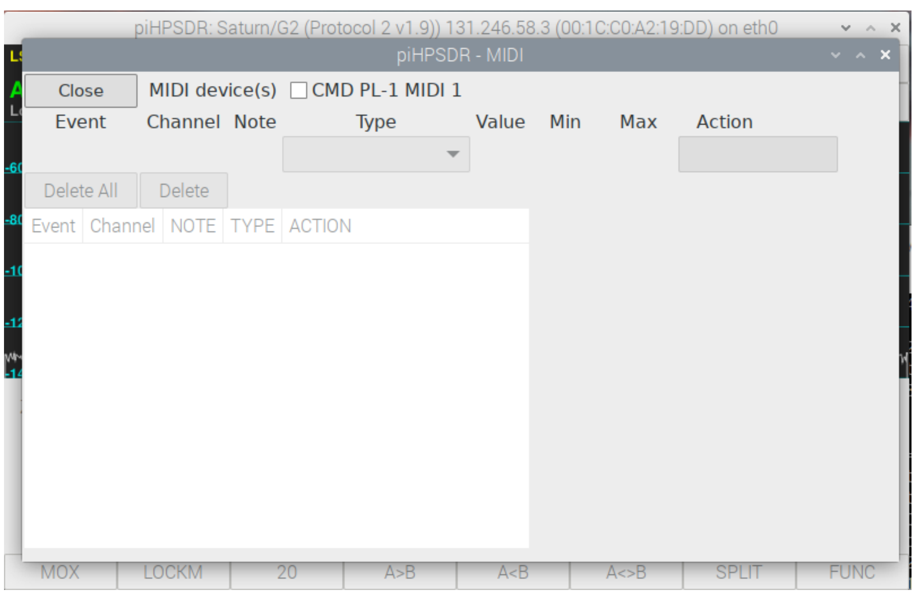
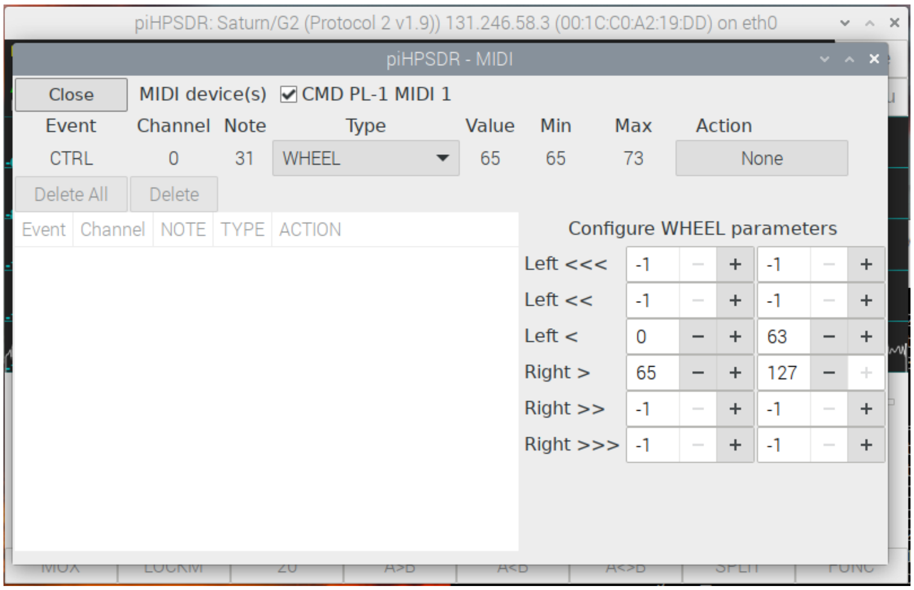
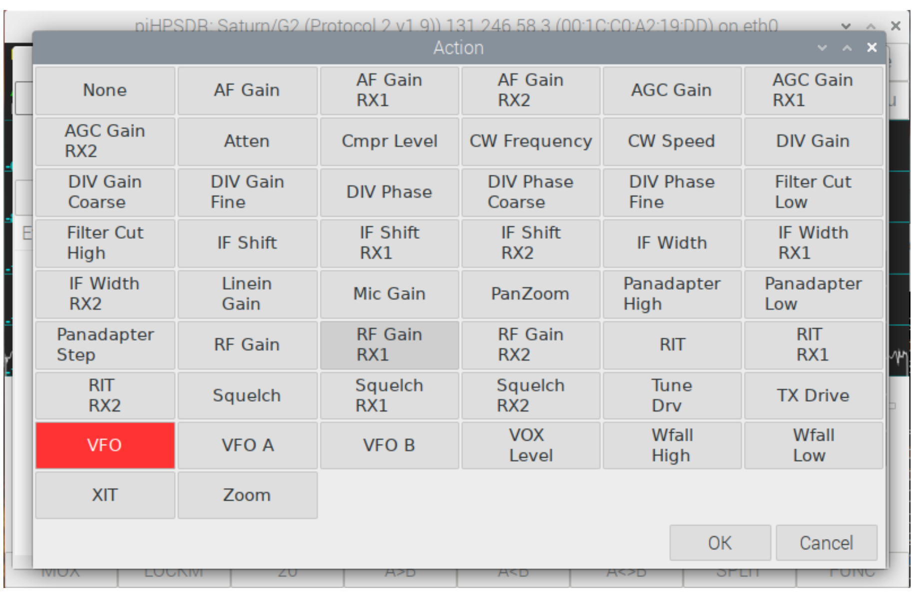


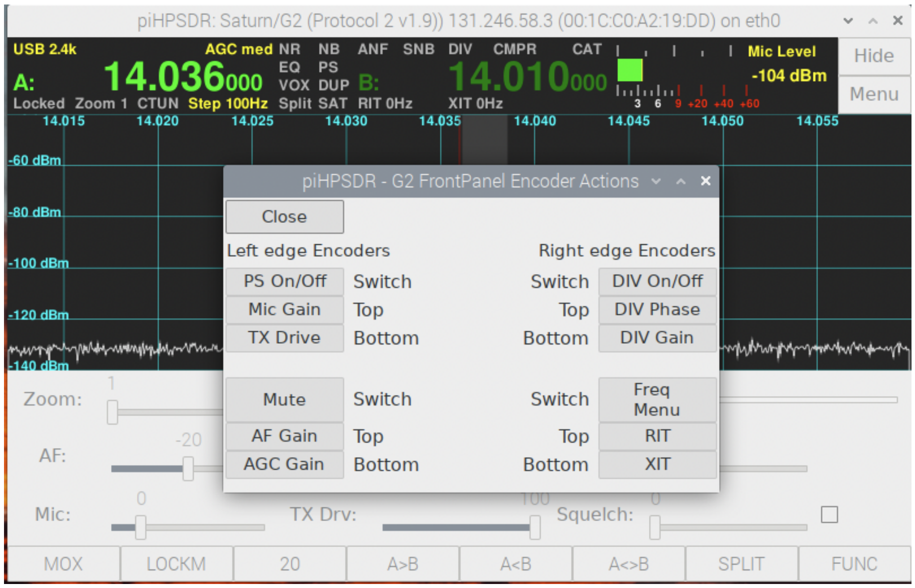
Fig. 10.5: The new `F1` layer is operative.

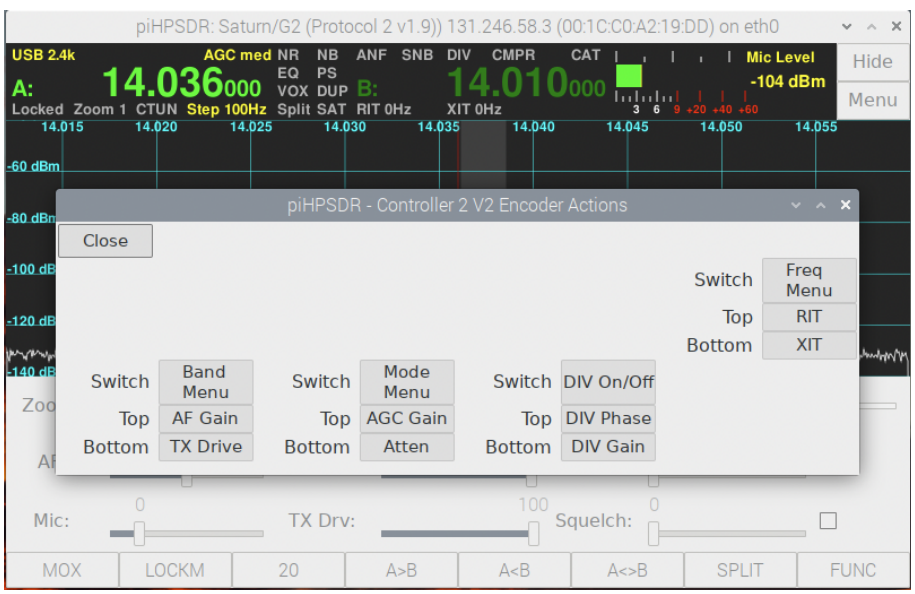

piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
piHPSDR - RigCtl
Close
RigCtl TCP Port
19090
Rigctl Enable
Debug
Serial Port0:
/dev/ttyACM0
4800 Bd
Enable
Andromeda
Serial Port1:
/dev/ttyACM1
4800 Bd
Enable
Andromeda
AF:
AGC:
Att (dB):
Mic:
TX Drv:
Squelch:
MOX
LOCKM
20
A>B
A<B
A<>B
SPLIT
FUNC

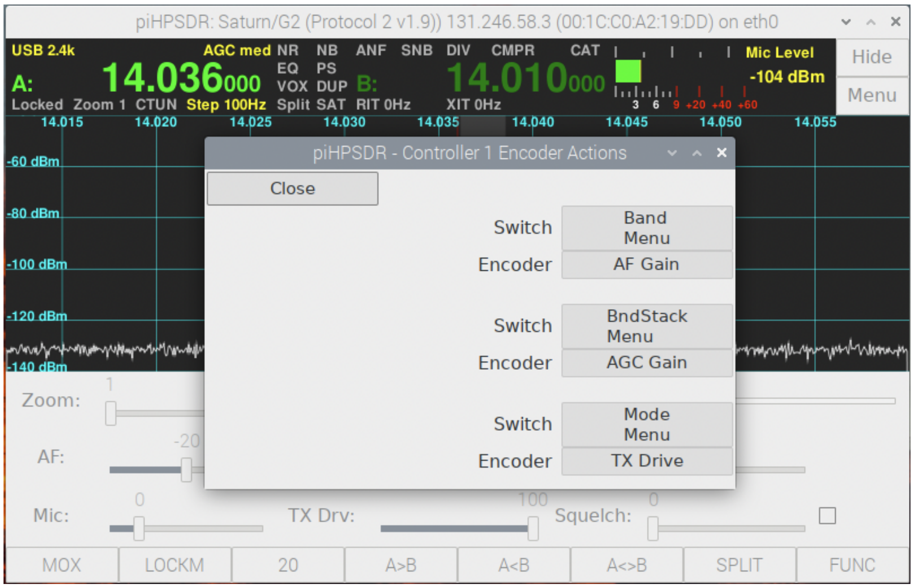

piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
piHPSDR - MIDI
Close
MIDI device(s)
CMD PL-1 MIDI 1
Event
Channel
Note
Type
Value
Min
Max
Action
Delete All
Delete
Event
Channel
NOTE
TYPE
ACTION
MOX
LOCKM
20
A>B
A<B
A<>B
SPLIT
FUNC

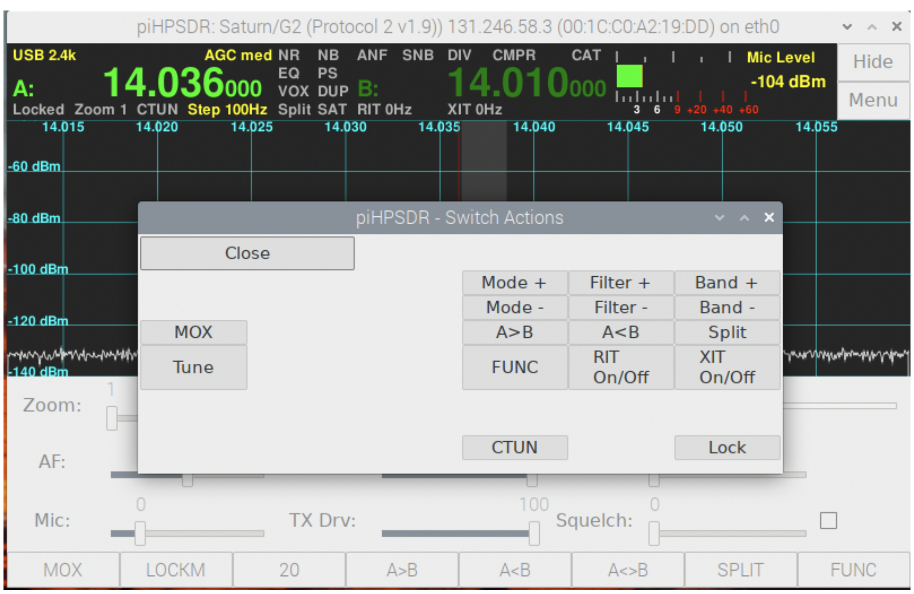

piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
piHPSDR - MIDI
Close
MIDI device(s)
CMD PL-1 MIDI 1
Event
Channel
Note
Type
Value
Min
Max
Action
CTRL
0
31
WHEEL
65
65
73
None
Delete All
Delete
Event
Channel
NOTE
TYPE
ACTION
Configure WHEEL parameters
Left <<<
Left <<
Left <
Right >
Right >>
Right >>>

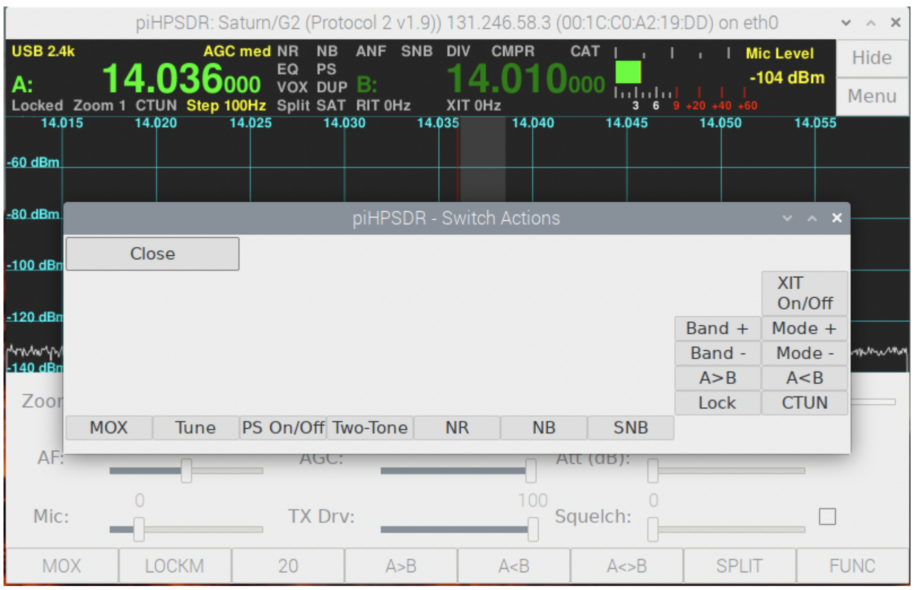

Action
None
AF Gain
AF Gain RX1
AF Gain RX2
AGC Gain
AGC Gain RX1
AGC Gain RX2
Atten
Cmpr Level
CW Frequency
CW Speed
DIV Gain
DIV Gain Coarse
DIV Gain Fine
DIV Phase
DIV Phase Coarse
DIV Phase Fine
Filter Cut Low
Filter Cut High
IF Shift
IF Shift RX1
IF Shift RX2
IF Width
IF Width RX1
IF Width RX2
Linein Gain
Mic Gain
PanZoom
Panadapter High
Panadapter Low
Panadapter Step
RF Gain
RF Gain RX1
RF Gain RX2
RIT
RIT RX1
RIT RX2
Squelch
Squelch RX1
Squelch RX2
Tune Drv
TX Drive
VFO
VFO A
VFO B
VOX Level
Wfall High
Wfall Low
XIT
Zoom
OK
Cancel


piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
USB 2.4k
AGC med NR NB ANF SNB DIV CMPR CAT
EQ PS
VOX DUP
A: 14.036000
B: 14.010000
Locked Zoom 1 CTUN Step 100Hz Split SAT RIT 0Hz XIT 0Hz
Mic Level
-104 dBm
Hide
Menu
piHPSDR - G2 FrontPanel Encoder Actions
Close
Left edge Encoders
Right edge Encoders
PS On/Off Switch
Mic Gain Top
TX Drive Bottom
Switch DIV On/Off
Top DIV Phase
Bottom DIV Gain
Mute Switch
AF Gain Top
AGC Gain Bottom
Switch Freq Menu
Top RIT
Bottom XIT
Zoom:
AF:
Mic:
TX Drv:
Squelch:
MOX LOCKM 20 A>B A<B A<>B SPLIT FUNC


piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
USB 2.4k
AGC med NR NB ANF SNB DIV CMPR CAT
A: 14.036000
B: 14.010000
Locked Zoom 1 CTUN Step 100Hz Split SAT RIT 0Hz XIT 0Hz
Mic Level
-104 dBm
Hide
Menu
piHPSDR - Controller 2 V2 Encoder Actions
Close
Switch Freq Menu
Top RIT
Bottom XIT
Switch Band Menu
Top AF Gain
Bottom TX Drive
Switch Mode Menu
Top AGC Gain
Bottom Atten
Switch DIV On/Off
Top DIV Phase
Bottom DIV Gain
Mic:
TX Drv:
Squelch:
MOX LOCKM 20 A>B A<B A<>B SPLIT FUNC


piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
USB 2.4k
AGC med NR NB ANF SNB DIV CMPR CAT
A: 14.036000
B: 14.010000
Mic Level
-104 dBm
Hide
Menu
Locked Zoom 1 CTUN Step 100Hz Split SAT RIT 0Hz XIT 0Hz
piHPSDR - Controller 1 Encoder Actions
Close
Switch
Band Menu
Encoder
AF Gain
Switch
BndStack Menu
Encoder
AGC Gain
Switch
Mode Menu
Encoder
TX Drive
Zoom:
AF:
Mic:
TX Drv:
Squelch:
MOX
LOCKM
20
A>B
A<B
A<>B
SPLIT
FUNC


piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
USB 2.4k
AGC med NR NB ANF SNB DIV CMPR CAT
A: 14.036000
B: 14.010000
Mic Level
-104 dBm
Hide
Menu
piHPSDR - Switch Actions
Close
Mode +
Filter +
Band +
Mode -
Filter -
Band -
A>B
A<B
Split
MOX
Tune
FUNC
RIT On/Off
XIT On/Off
CTUN
Lock
Zoom:
AF:
Mic:
TX Drv:
Squelch:
MOX
LOCKM
20
A>B
A<B
A<>B
SPLIT
FUNC


piHPSDR: Saturn/G2 (Protocol 2 v1.9)) 131.246.58.3 (00:1C:C0:A2:19:DD) on eth0
USB 2.4k
AGC med
A:
14.036000
B:
14.010000
Locked Zoom 1 CTUN Step 100Hz
Mic Level
-104 dBm
Hide
Menu
piHPSDR - Switch Actions
Close
XIT On/Off
Band +
Mode +
Band -
Mode -
A>B
A<B
Lock
CTUN
MOX
Tune
PS On/Off
Two-Tone
NR
NB
SNB
MOX
LOCKM
20
A>B
A<B
A<>B
SPLIT
FUNC

Appendix A

List of piHPSDR „Actions”

In this chapter, we give a list of „actions” implemented in the piHPSDR program. These actions can be assigned to toolbar buttons on the screen, or pushbuttons/encoders of a GPIO-connected or MIDI controller. Not all actions can be assigned to all control elements. Changing the AF volume, for example, can only be assigned to a knob which you can turn, while switching RIT on/off can only be assigned to a button that you can push. For each action in the following table, there is a long and a short string assigned. The long string will be used when there is enough space, while the short string is used for small buttons and to store actions in preference files (therefore the short strings never contain a blank character or a line break). Then, for each action we give the type of control element allowed for this action as a combination of the letters B, P, E, which stand for

`B` ”Button”: A button in the toolbar, or a push-button or switch on a GPIO or MIDI connected console

`P` ”Potentiometer”: A potentiometer or a slider on a MIDI connected console

`E` ”Encoder”: A rotary encoder on a GPIO or MIDI connected console

The main difference between a ”potentiometer” and an ”encoder” is, that the former has a min and max position, while an encoder can be turned in either direction without stopping. This means that a potentiometer reports a value between min and max, while an encoder reports an increment, that is, whether it has been turned clock wise or counter clock wise. The existing GPIO consoles do not have potentiometers (most likely because of the lack of analog inputs), but many MIDI consoles do have, and Arduino-based MIDI controllers might have it because there analog inputs to read out potentiometers are available.

To give an example, controlling the TX drive can be down both with a slider and with an encoder. While for a slider/potentiometer, the values from min to max are simple mapped to the TX drive values from 0 to 100, the signals from an encoder will just increase or decrease the value until one of a limits has been reached.

In the following, the actions are alphabetically sorted by their long name, with the "empty" action listed first.

`NONE`	`NONE`	`BPE`
This is an action which does nothing. It can be assigned to buttons or encoders that are often accidentally operated. Some MIDI consoles, for example, report a button press event if the VFO knob is touched, and this we want to ignore.		

`A<>B`	`A<>B`	`B`
Swap VFOs A and B. This will not only swap the frequencies, but also all other settings associated with that VFO, such as mode, filter, CTUN, and RIT settings.		

`A<B`	`A<B`	`B`
Copy VFO B to VFO A.		

`A>B`	`A>B`	`B`
Copy VFO A to VFO B.		

`AF Gain`	`AFGAIN`	`PE`
Change the AF gain (headphone volume) of the active receiver.		

AF Gain RX1	AFGAIN1	PE
Change the AF gain (headphone volume) of the RX1 receiver.		

AF Gain RX2	AFGAIN2	PE
Change the AF gain (headphone volume) of the RX2 receiver.		

AGC Menu	AGC	B
Opens the `AGC` menu.		

ANF	ANF	B
Toggels the state (on/off) of the automatic notch filter for the active receiver.		

Atten	ATTEN	PE
Changes the value (0-31 dB) of the step attenuator of the active receiver. This funciton is only available for radios that have such an attenuator.		

Band 10	10	B
Change band of the active receiver to the 10m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 12	12	B
Change band of the active receiver to the 12m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 1240	1240	B
Change band of the active receiver to the 1240 MHz (23 cm) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 144	144	B
Change band of the active receiver to the 144 MHz (2m) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 15	15	B
Change band of the active receiver to the 15m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 160	160	B
Change band of the active receiver to the 160m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 17	17	B
Change band of the active receiver to the 15m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 20	20	B
Change band of the active receiver to the 15m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 220	220	B
Change band of the active receiver to the 220 MHz (1.25 m) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 2300	2300	B
Change band of the active receiver to the 2300MHz (13 cm) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 30	30	B
Change band of the active receiver to the 30m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 3400	3400	B
Change band of the active receiver to the 3400 Mhz (9 cm) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 40	40	B
Change band of the active receiver to the 40m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 430	430	B
Change band of the active receiver to the 430 MHz (70 cm) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 6	6	B
Change band of the active receiver to the 6m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 60	60	B
Change band of the active receiver to the 60m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 70	70	B
Change band of the active receiver to the 70 MHz (4m) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 80	80	B
Change band of the active receiver to the 80m band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band 902	902	B
Change band of the active receiver to the 902 MHz (33 cm) band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band AIR	AIR	B
Change band of the active receiver to the 108 MHz band, used for aircraft communication. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band GEN	GEN	B
Change band of the active receiver to the current bandstack entry of the "general" band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

Band -	BND-	B
Change band of the active receiver to the next lower band in the list of bands. If already at the lowest band, switch to the highest band (including transverter bands which have been defined) whose frequency is with the radio's frequency range.		

Band +	BND+	B
Change band of the active receiver to the next higher band in the list of bands (including transverter bands that have been defined). If already at the highest band, switch to the lowest band whose frequency is with the radio's frequency range.		

Band WWV	WWV	B
Change band of the active receiver to the current bandstack entry of the WWV band. If already on that band, move to the next bandstack entry. This action is a no-op if the frequency of the band falls outside the frequency range of the radio.		

BndStack -	BSTK-	B
Cylcle backward through the bandstack entries of the active receiver.		

BndStack +	BSTK+	B
Cylcle forward through the bandstack entries of the active receiver.		

Band Menu	BAND	B
Open the `BAND` menu.		

BndStack MENU	BSTK	B
Open the `BANDSTACK` menu.		

Cmpr On/Off	COMP	B
Toggle the state (on/off) of the compressor used in the TX audio input.		

Cmpr Level	COMPVAL	PE
Change the value of the compressor (0-20 dB) used in the TX audio input. The compressor is automaticall switched on (off) if the ”new” value of the compressor is larger then (equal to) zero.		

CTUN	CTUN	B
Toggle the state (on/off) of the CTUN state of the active receiver. CTUN stands for ”click to tune”. In CTUN mode, you can move the RX frequency over the whole spectrum scope, whose center then remains at a fixed frequency.		

CW Audio Peak Fltr	CW-APF	B
Toggle (on/off) the CW audio peak filter for the active receiver. Note that the width of this filter (default: 75 Hz) can only be modified through the CW menu.		

CW Frequency	CWFREQ	PE
Change the CW side tone frequency in the range 300-1000 Hz. This also changes the BFO frequency upon receive.		

CW Left	CWL	B
This action indicates the closure/opening of the left paddle of a CW key. It is usually assigned to a GPIO line or a MIDI controller to which a Morse paddle is attached, and works with the iambic keyer that is built into piHPSDR. This keyer is only active if CW is *not* handled in the radio (see CW menu).		

CW Right	CWR	B
This action indicates the closure/opening of the right paddle of a CW key. It is usually assigned to a GPIO line or a MIDI controller to which a Morse paddle is attached, and works with the iambic keyer that is built into piHPSDR. This keyer is only active if CW is *not* handled in the radio (see CW menu).		

CW Speed	CWSPD	PE
Change the CW side tone frequency in the range 1-60 wpm. This affect the built-in iambic keyer or the keyer inside the radio, depending on whether CW is handled in the radio or not (see CW menu).		

CW Key (keyer)	CWKy	B
Straith key key-down or key-up event. Usually assigned to a GPIO line of MIDI controller to which a straight key or an external keyer is attached. Note that this action does not automatically switch to TX, so it must be used together with either manual RX/TX switching, or with the ”`PTT (CW Keyer)`” action.		

PTT (keyer)	CWKyPTT	B
This very similar to the `PTT` action (see below) with the exception that CW handling in the radio is temporarily disabled (thus, CW handling in piHPSDR is enabled). This allows to have, e.g. a paddle attached to the radio while a contest logging program ,,talks” to piHPSDR.		

DIV On/Off	DIVT	B
Toggles (enabled/disabled) DIVERSITY reception.		

DIV Gain	DIVG	E
Adjust DIVERSITY gain. One tick of the encoder increments of decrements the gain by an amount of 0.5		

DIV Gain Coarse	DIVGC	E
Adjust DIVERSITY gain (coarse adjustment). One tick of the encoder increments of decrements the gain by an amount of 2.5		

DIV Gain Fine	DIVGF	E
Adjust DIVERSITY gain (fine adjustment). One tick of the encoder increments of decrements the gain by an amount of 0.1. Since adjusting the DIVERSITY gain (or phase) is sometimes difficult, assigning one encoder to a coarse and another encoder to a fine adjustment may help in locating the „sweet spot”.		

DIV Phase	DIVP	E
Adjust DIVERSITY phase (fine adjustment). One tick of the encoder increments of decrements the gain by an amount of 0.5		

DIV Phase Coarse	DIVPC	E
Adjust DIVERSITY gain (coarse adjustment). One tick of the encoder increments of decrements the gain by an amount of 2.5		

DIV Phase Fine	DIVPF	E
Adjust DIVERSITY gain (coarse adjustment). One tick of the encoder increments of decrements the gain by an amount of 20.1		

DIV Menu	DIV	B
Open the DIVERSITY menu.		

Duplex	DUP	B
Toggle (on/off) DUPLEX status. IN the DUPLEX mode, the receivers continue to work during TX, and the RX panels are not removed during TX. Instead, a separate TX window opens during transmitting. Generally, DUPLEX only make sense when using different and well decoupled RX and TX antennas.		

Filter -	FL-	B
Cycle forward (!) through the list of filters for the current mode of the active receiver. Normally, this means switching to a narrower filter (hence the name `FILTER -`). When reaching the last filter in the list, further cycling switches to the first (widest) filter.		

Filter +	FL+	B
Cycle backward (!) through the list of filters for the current mode of the active receiver. Normally, this means switching to a wider filter (hence the name `FILTER +`). When reaching the first filter in the list, further cycling switches to the last filter which is the variable `Var2` filter.		

Filter Cut Low	FCUTL	E
Adjust the low-cut of the current filter. Note that the notion of ,,low" edge of the filter refers to audio frequencies for the single side band modes LSB, CWL, DIGL. This action is a no-op unless the current filter is one of the two variable filters `Var1` or `Var2`.		

Filter Cut High	FCUTL	E
Adjust the high-cut of the current filter. Note that the notion of ,,high" edge of the filter refers to audio frequencies for the single side band modes LSB, CWL, DIGL. This action is a no-op unless the current filter is one of the two variable filters `Var1` or `Var2`.		

Filter Cut Default	FCUTDEF	B
Reset the low and high cut of the current filter to the default values. This action is a no-op unless the current filter is one of the two variable filters `Var1` or `Var2`.		

Filter Menu	FILT	B
This opens the `Filter` menu.		

VFO Menu	FREQ	B
This opens the `FREQ` (VFO) menu.		

Function	FUNC	B
Cycle through the six toolbar sets. For the piHPSDR GPIO controller V1, where the eight switches follow the toolbar buttons, this also affects the function of the switches. Note that this action is *always* connected with the right-most toolbar button.		

FuncRev	FUNC-	B
Cycle backwards through the six toolbar sets. For the piHPSDR GPIO controller V1, where the eight switches follow the toolbar buttons, this also affects the function of the switches. When using a mouse, this action can be invoked by a secondary mouse click on the rightmost toolbar button.		

IF Shift	IFSHFT	E
This command is effective only if one of the variable filters Var1 or Var2 is currently used in the active receiver, and shifts the filter, that is, it affects the low and high cut in the same way.		

IF Shift RX1	IFSHFT1	E
This command is effective only if one of the variable filters Var1 or Var2 is currently used in VFO-A, and shifts the filter, that is, it affects the low and high cut in the same way.		

IF Shift RX2	IFSHFT2	E
This command is effective only if one of the variable filters Var1 or Var2 is currently used in VFO-B, and shifts the filter, that is, it affects the low and high cut in the same way.		

IF Width	IFWIDTH	E
This command is effective only if one of the variable filters Var1 or Var2 is currently used in the active receiver, and changes the filter width, that is, it affects the low and high cut in an opposite way.		

IF Width RX1	IFWIDTH1	E
This command is effective only if one of the variable filters Var1 or Var2 is currently used in VFO-A, and changes the filter width, that is, it affects the low and high cut in an opposite way.		

actionIF Width RX2IFWIDTH2EThis command is effective only if one of the variable filters Var1 or Var2 is currently used in VFO-B, and changes the filter width, that is, it affects the low and high cut in an opposite way.

Linein Gain	LIGAIN	PE
Change the line-in gain of the radio. If the radio does not have a line-in input, this control has no effect.		

Lock	LOCK	B
Lock the VFOs. A locked VFO will not accept VFO frequency steps in either direction, and cannot be moved by dragging with the mouse. Band changes etc. are still possible, though. The command is intended to guard against accidentally moving the VFO dial.		

Memory Menu	MEM	B
Open the `MEM` (Memory) menu.		

Mic Gain	MICGAIN	PE
Change the mic gain (from -12 to 50 dB). The amplification of the microphone audio data is done in software, and applies to the TX audio input samples whereever they come from. (See the discussion of local microphones in the `TX` menu.		

Mode -	MD-	B
Cycle backwards through the list of modes for the active receiver. When the first mode (LSB) has been reached, jump to the last one (DRM). Note that when changing the mode, the current filter, noise reduction, equalizer, VFO step size, and TX compressor settings are stored for the old mode, and the settings last used with the new mode are restored. This allows to quickly switch between SSB and CW, or between SSB and digi modes, without re-adjusting these settings.		

Mode +	MD+	B
Cycle forward through the list of modes for the active receiver. When the last mode (DRM) has been reached, jump to the first one (LSB). Note that when changing the mode, the current filter, noise reduction, equalizer, VFO step size, and TX compressor settings are stored for the old mode, and the settings last used with the new mode are restored. This allows to quickly switch between SSB and CW, or between SSB and digi modes, without re-adjusting these settings.		

Mode Menu	MODE	B
Open the `Mode` menu.		

MOX	MOX	B
Toggle between TX and RX. Unlike the PTT action, which puts the radio into TX when pressed and into RX when released, this button toggles the PTT state when pressed.		

Mute	MUTE	B
Toggles the „mute” state of the active receiver. If a receiver is muted, it produces zero-amplitude audio output.		

NB	NB	B
Cycles through the noise blanker states (NB off/NB1/NB2).		

NR	NR	B
Cycles through the noise reduction states (NR off/NR1/NR2).		

Noise Menu	NOISE	B
Opens the `NOISE` menu.		

NumPad 0	0	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,0" in the `VFO` (VFO) menu.		

NumPad 1	1	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,1" in the `VFO` menu.		

NumPad 2	2	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,2" in the `VFO` menu.		

NumPad 3	3	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,3" in the `VFO` menu.		

NumPad 4	4	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,4" in the `VFO` menu.		

NumPad 5	5	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,5" in the `VFO` menu.		

NumPad 6	6	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,6” in the VFO menu.		

NumPad 7	7	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,7” in the VFO menu.		

NumPad 8	8	B
(Used for direct frequency entry. This is the same as hitting the corresponding button ,,8” in the VFO menu.		

NumPad 9	9	B
Used for direct frequency entry. This is the same as hitting the corresponding button ,,9” in the VFO menu.		

NumPad BS	BS	B
Used for direct frequency entry (BS = backstep). This is the same as hitting the corresponding button in the VFO menu. It cancels the last-entered digit.		

NumPad CL	CL	B
Used for direct frequency entry (CL = clear). This is the same as hitting the corresponding button in the VFO menu. It cancels all entered digits so far.		

NumPad Dec	DEC	B
Used for direct frequency entry (DEC = decimal point). This is the same as hitting the corresponding button in the VFO menu.		

NumPad kHz	KHZ	B
Used for direct frequency entry. This is the same as hitting the corresponding button in the VFO menu. The VFO frequency is changed to the value entered so far, multiplied with 1,000. For example, to go to 7.040 MHz, one can enter the sequence ,,7”, ,,0”, ,,4”, ,,0”, ,,KHZ”.		

NumPad MHz	MHZ	B
Used for direct frequency entry. This is the same as hitting the corresponding button in the `VFO` menu. The VFO frequency is changed to the value entered so far, multiplied with 1,000,000. For example, to go to 7.040 MHz, one can enter the sequence „7”, „DEC”, „0”, „4”, „MHZ”.		

NumPad Enter	EN	B
Used for direct frequency entry. This is the same as hitting the corresponding button in the `VFO` menu. The VFO frequency is changed to the value entered so far. For example, to go to 7.040 MHz, one can enter the sequence „7”, „0”, „4”, „0”, „0”, „0”, „0”. This is rarely used but offers Hz-resolution for the direct frequency entry.		

PanZoom	PAN	E
Change the Pan value. This control is only effective when the Zoom value is larger than 1.		

Pan-	PAN-	B
Decrease the PAN value by 100. This control is only effective when the Zoom value is larger than 1.		

Pan+	PAN+	B
Increase the PAN value by 100. This control is only effective when the Zoom value is larger than 1.		

Panadapter High	PANH	PE
Change the dBm value (from -60 to +20) at the top of the spectrum scope of the active receiver. Values outside this range can be set in the `DISPLAY` menu.		

Panadapter Low	PANL	PE
Change the dBm value (from -160 to -60) at the bottom of the spectrum scope of the active receiver. Values outside this range can be set in the DISPLAY menu.		

Panadapter Step	PANS	PE
Change the step size (from 5 to 30) of the panadapter of the active receiver. This is the spacing of the thin horizontal lines in the spectrum scope.		

Preamp On/Off	PRE	B
Toggle the preamp of the active receiver. Although the preamp switching is part of the HPSDR protocol, this has no effect in current radio models since the preamp is hard-wired „on”.		

PS On/Off	PST	B
Toggle (on/off) adaptive predistortion (PureSignal).		

PS Menu	PS	B
Open the PS (PureSignal) menu.		

PTT	PTT	B
Put the radio into TX mode when the button is pressed, and go back to RX when the button is released. This is one of the few actions where a button release event is significant. When attaching, say, the PTT contact of a microphone to a GPIO line for this purpose, take care of proper debouncine, since piHPSDR is not good at debouncing switches where both the press and release events are significant.		

RF Gain	RFGAIN	PE
Set the gain of the RF front end of the active receiver. Only effective for radios that have such a gain control. Most HPSDR radios do not have RF gain, they have a step attenuator in the RF front end instead. Small SDR radios using the AD9866 chip (HermesLite, RadioBerry) and radios connected via the SoapySDR library usually do have an RF gain control.		

RF Gain RX1	RFGAIN1	PE
Set the gain of the RF front end of RX1. Only effective for radios that have such a gain control. Most HPSDR radios do not have RF gain, they have a step attenuator in the RF front end instead. Small SDR radios using the AD9866 chip (HermesLite, RadioBerry) and radios connected via the SoapySDR library usually do have an RF gain control.		

RF Gain RX2	RFGAIN2	PE
Set the gain of the RF front end of RX2. Only effective for radios that have such a gain control. Most HPSDR radios do not have RF gain, they have a step attenuator in the RF front end instead. Small SDR radios using the AD9866 chip (HermesLite, RadioBerry) and radios connected via the SoapySDR library usually do have an RF gain control.		

RIT	RIT	E
Change the RIT value of the active receiver in the range -9999 to 9999 Hz. If a zero value is set, RIT is automatically disabled, if a non-zero value ist set, RIT is enabled.		

RIT Clear	RITCL	B
Set the RIT value of the active receiver to zero. As a side effect, RIT is disabled for the active receiver		

RIT On/Off	RITT	B
Toggle RIT (enabled/disabled) for the active receiver. Note the RIT value is not changed, so you can temporarily disable RIT, and then enable it with the same offset (RIT value) used before.		

RIT -	RIT-	B
Decrement the RIT value of the active receiver by the RIT step size, in the range -9999 to 9999 Hz. If a value of zero is reached, RIT is automatically disabled, and if a nonzero value is reached, RIT is automatically enabled. Note that this action belongs to the few ones for which a button release event has an effect. If you press and hold RIT- (either on the toolbar, or on a GPIO or MIDI console), there is an auto-repeat such that the action will be repeated every 250 msec until the RIT- button is released.		

RIT +	RIT+	B
Increment the RIT value of the active receiver by the RIT step size, in the range -9999 to 9999 Hz. If a value of zero is reached, RIT is automatically disabled, and if a nonzero value is reached, RIT is automatically enabled. Note that this action belongs to the few ones for which a button release event has an effect. If you press and hold RIT+ (either on the toolbar, or on a GPIO or MIDI console), there is an auto-repeat such that the action will be repeated every 250 msec until the RIT+ button is released.		

RIT RX1	RIT1	E
Change the RIT value of RX1 in the range -9999 to 9999 Hz. If a zero value is set, RIT is automatically disabled, if a non-zero value ist set, RIT is enabled.		

RIT RX2	RIT2	E
Change the RIT value of RX2 in the range -9999 to 9999 Hz. If a zero value is set, RIT is automatically disabled, if a non-zero value ist set, RIT is enabled.		

RIT Step	RITST	B
Cycle through the possible values (1 Hz, 10 Hz, 100 Hz) of the RIT step.		

RSAT	RSAT	B
If the SAT mode is either Off or SAT, change it to RSAT. If the SAT mode is RSAT, change it to Off. In RSAT mode all VFO frequency *changes* applied to one of the two VFOs will be applied to the other VFO with the sign reversed.		

SAT	SAT	B
If the SAT mode is either Off or RSAT, change it to SAT. If the SAT mode is SAT, change it to Off. In SAT mode all VFO frequency *changes* applied to one of the two VFOs will be applied to the other VFO as well.		

SNB	SNB	B
Toggle (enable/diable) the spectral noise blanker for the active receiver.		

Split	SPLIT	B
Toggle (on/off) the split status of the radio.		

Squelch	SQUELCH	PE
Change the squelch threshold value of the active receiver. Squelch is automatically enabled (disbled) if the resulting value is non-zero (zero).		

Squelch RX1	SQUELCH1	PE
Change the squelch threshold value of RX1. Squelch is automatically enabled (disbled) if the resulting value is non-zero (zero).		

Squelch RX2	SQUELCH2	PE
Change the squelch threshold value of RX2. Squelch is automatically enabled (disbled) if the resulting value is non-zero (zero).		

Swap RX	SWAPRX	B
Make the inactive receiver the active one. This action is only effective if piHPDSR is running two receivers.		

Tune	TUNE	B
Toggle (on/off) TUNE. If selected in the OC menu, an OC output will become active (low). This can then be used to start an external automatic tuner.		

`Tune Drv`	TUNDRV	E
Change the drive level (0-100) used for TUNEing. This is equivalent to changing the ”Tune drive level” spin button in the `TX` menu **and** to check the ”Tune use drive” box.		

`Tune Full`	TUNF	B
Set the ”full tune” flag and clear the ”memory tune” flag. If an OC output is assigned to the TUNE state, it will be cleared (go high again) 2800 msec after starting TUNE (this time can also be adjusted in the OC menu).		

`Tune Mem`	TUNM	B
Set the ”memory tune” flag and clear the ”full tune” flag. If an OC output is assigned to the TUNE state, it will be cleared (go high again) 550 msec after starting TUNE (this time can also be adjusted in the OC menu).		

`TX Drive`	TXDRV	PE
Set the TX drive level (0-100).		

`Two-Tone`	2TONE	B
Toggle (on/off) the two-tone state of the transmitter. If the two-tone state is engaged, the radio will go TX and emit a two-tone signal.		

`VFO`	VFO	E
This is the VFO frequency control of the active receiver.		

`VFO A`	VFOA	E
This is the VFO frequency control of VFO-A.		

`VFO B`	VFOB	E
This is the VFO frequency control of VFO-B.		

<table>
<tr><td><code>VOX On/Off</code></td><td><code>VOX</code></td><td>B</td></tr>
<tr><td colspan="3">Toggle (on/off) vox status. If vox is enabled, you can automatically key the transmitter by talking into the microphone, without the need to press a PTT button. See the <code>VOX</code> menu.</td></tr>
</table>

<table>
<tr><td><code>VOX Level</code></td><td><code>VOXLEV</code></td><td>E</td></tr>
<tr><td colspan="3">Change the VOX level threshold. If you operate vox, and the radio does not go TX while talking into the microphone, decrease the VOX threshold. If the radio goes TX simply because the neighbour's hound starts barking, increase the VOX threshold.</td></tr>
</table>

<table>
<tr><td><code>Wfall High</code></td><td><code>WFALLH</code></td><td>E</td></tr>
<tr><td colspan="3">Change the "high" level (-100 dBm ... 0 dBm) of the waterfalls. Signal levels between low and high are colour coded from black to yellow, while signals above "high" are yellow and signals below "low" are black. This value has no effect if the automatic waterfall coloring is chosen ("waterfall automatic"), which is usually preferable.</td></tr>
</table>

<table>
<tr><td><code>Wfall Low</code></td><td><code>WFALLL</code></td><td>E</td></tr>
<tr><td colspan="3">Change the "low" level (-150 dBm ... -50 dBm) of the waterfalls. Signal levels between low and high are colour coded from black to yellow, while signals above "high" are yellow and signals below "low" are black. This value has no effect if the automatic waterfall coloring is chosen ("waterfall automatic"), which is usually preferable.</td></tr>
</table>

<table>
<tr><td><code>XIT</code></td><td><code>XIT</code></td><td>E</td></tr>
<tr><td colspan="3">Change the XIT value of the transceiver in the range -9999 to 9999 Hz. If a zero value is set, XIT is automatically disabled, if a non-zero value ist set, XIT is enabled.</td></tr>
</table>

<table>
<tr><td><code>XIT Clear</code></td><td><code>XITCL</code></td><td>B</td></tr>
<tr><td colspan="3">Set the XIT value of the transmitter to zero. As a side effect, XIT is disabled.</td></tr>
</table>

XIT On/Off	XITT	B
Toggle XIT (enabled/disabled) for the transceiver. Note the XIT value is not changed, so you can temporarily disable XIT, and then enable it with the same offset (XIT value) used before.		

XIT -	XIT-	B
Decrement the XIT value of the transmitter by the RIT (!) step size, in the range -9999 to 9999 Hz. If a value of zero is reached, XIT is automatically disabled, and if a nonzero value is reached, XIT is automatically enabled. Note that this action belongs to the few ones for which a button release event has an effect. If you press and hold XIT- (either on the toolbar, or on a GPIO or MIDI console), there is an auto-repeat such that the action will be repeated every 250 msec until the XIT- button is released.		

XIT +	XIT+	B
Increment the XIT value of the transmitter by the RIT (!) step size, in the range -9999 to 9999 Hz. If a value of zero is reached, XIT is automatically disabled, and if a nonzero value is reached, XIT is automatically enabled. Note that this action belongs to the few ones for which a button release event has an effect. If you press and hold XIT+ (either on the toolbar, or on a GPIO or MIDI console), there is an auto-repeat such that the action will be repeated every 250 msec until the XIT+ button is released.		

Zoom	ZOOM	PE
Change the ZOOM value (1...8) of the active receiver.		

Zoom -	ZOOM-	B
Decrease the ZOOM value of the active receiver by one. If the ZOOM value was already 1, this is a no-op.		

Zoom -	ZOOM-	B
Increase the ZOOM value of the active receiver by one. If the ZOOM value was already 8, this is a no-op.		

Appendix B

piHPSDR keyboard bindings

Appendix C

piHPSDR CAT commands

The CAT model of piHPSDR largely follows that for other SDR programs. It is based upon the Kenwood TS-2000 CAT command set, which can easily be found on the internet (see the Appendix of the Kenwood TS-2000 instruction manual) and will not be reproduced here for copyright reasons. So if you want to connect a logbook or contest logging program to piHPSDR, you will normally tell this program that it has to control a Kenwood TS-2000.

In the SDR community, there exist a heavily extended TS-2000 CAT command set known as the ,,PowerSDR CAT command set", the original source is probably

`https://www.flexradio.com/documentation/powersdr-cat-command-reference-guide/`

Many (but probably not all) of the commands listed there are implemented in piHPSDR, because it seems that there exist SDR controllers which communicate over the serial line. In recent years, such an open-source controller, the ANDROMEDA controller, has been developed by Laurence Barker G8NJJ, see

`https://github.com/laurencebarker/Andromeda_front_panel`

This controller uses some additional CAT commands to communicate with the radio, and these commands have also been implemented into the RIGCTL module of piHPSDR by Rick Koch N1GP (thanks Rick). These are the ZZZD and ZZZU commands for moving the VFO frequency down/up, and the ZZZP and ZZZE commands for sending information about push-buttons and encoders, and a ZZZS command which contains information on the ANDROMEDA version. Furthermore, if "Andromeda" is selected in the RIGCTL menu, piHPSDR will constantly *send* status information to the ANDROMEDA controller using a ZZZI command. Status information is sent if something changes (active receiver, diversity status, PTT status, TUNE mode, PS status, CTUN mode, RIT and XIT status, and LOCK status), such that the ANDROMEDA controller can update the corresponding LEDs.

Appendix D

How to connect a Morse key/paddle

Appendix E

Running piHPSDR alongside with DigiMode programs

Appendix F

RaspPi: Preparations for using GPIO controllers

If you do not run piHPSDR on a Raspberry Pi, or if you do not use a GPIO controller connected to the Raspberry Pi GPIO I/O lines, there is no reason to read this chapter. There are only two aspects that need be covered here.

Setting the direction of the GPIO I/O lines. The first one is the switching of the direction (input or output) of the I/O lines, which has changed when the RaspPi3 was replaced by the RaspPi4. So for some time, it was necessary to put all the GPIO lines used by piHPSDR „manually" into an INPUT state (with pull-ups) when booting the RaspPi. The necessity to do so should fade out very quickly now, but just in case, the procedure is decribed here. Setting the GPIO lines used by piHPSDR $(4 - 27)$ to input state with pull-up can simply be done by adding a line

```
gpio=4-27=ip,pu
```

at the end of the file `/boot/config.txt` (this requires administrator privileges). For the ease of the user, a shell script with name `release/gpio.sh` has been provided which exactly does this. So it should be enough (this manual assumes that the pihpsdr directory is at top level within the users's home directory) to open a terminal window and issue the commmands

```
cd $HOME/pihpsdr
release/gpio.sh
```

and this will update your `/boot/config.txt` file. Note that the command `sudo` is used to actually perform the tasks that need administrator privileges, so make sure that you can use this command from your user account. Depending on the installation, you might be asked an administrator password when executing the shell script.

Activating the I2C interface. This is only necessary if you use a ,,version 2" or a G2 front-panel controller. These controllers extend the number of available input lines (and thus, the number of switches piHPSDR can control) by an i2c device, so i2c must be enabled. This can be done via the ,,Raspberry" → ,,Preferences" → ,,RaspberryPi Configuration" menu. In the ,,Interfaces" tab, make sure that I2C is enabled.

Appendix G

Binary installation of piHPSDR (RaspPi only)

Note: binary installation is not recommended, you should rather try to compile from the sources. A binary installation can only work if the operating system used to produce the binaries is more or less the same as the one where the binaries will be run. OS upgrades, a switch from a 32-bit to a 64-bit OS, and other changes can have the consequence that the binaries will not run on your system. This problem can be completely avoided if you compile and link piHPSDR from the sources on your system, and it is then also recommended to repeat this each time you do a major OS upgrade. For the same reason, **only binaries for the RaspberryPi** are provided, they have been produces on a RaspberryPi using the latest RaspberryPi OS. Note there exists a very detailed document on setting up a RaspPi and compiling piHPSDR from the sources, and several absolute Linux beginners have successfully followed these instructions. Another advantage when compiling from the sources is, that you always get the most recent version of piHPSDR. The binaries are only occasionally updated.

So, if you continue reading here despite the long list of caveats, you are sure you have a RaspPi and want a binary installation. Such an installation is provided for the RaspPi only. To make this manageable and minimize the dependence on third-party libraries, the binary version does not contain support for SoapySDR devices or USB OZY devices, and also cannot start RedPitaya SDR apps through the web interface. If you need any of these features, you have to compile from the sources with the SOAPSDR and/or the USBOZY and/or the STEMLAB options activated.

The easiest way to obtain the ,,tar-file" which contains the installation. Probably the easiest way to download the file is using a brower (you can open the browser on your RaspPi or on any other computer) and go to the following URI (this is one single string without blanks which is broken over two lines here)

```
https://github.com/dl1ycf/pihpsdr/blob/master/
release/pihpsdr.tar
```

and then click the ,,Raw" button. This should make your browser download a file with name "pihpsdr.tar". Put this file into your home directory. If you have down loaded the file on a different computer, you have to transfer it to your RaspPi. Then, in a terminal window issue the three commands

```
cd $HOME
tar xvf pihpsdr.tar
pihpsdr/install.sh
```

These commands will ,,extract" the tar file. At the end you have created a folder names pihpsdr in your top level home directory which contains lots of files, and the third command as put these files in the right location and should also create a piHPSDR desktop icon. Please read Appendix F and possibly configure GPIO and/or I2C if you want to use GPIO controllers.

The preferred way of starting piHPSDR is then double-clicking the piHPSDR desktop icon. As an alternative, you can open a terminal window and issue the two commands

```
cd $HOME/pihpsdr
pihpsdr
```

Appendix H

Installation of piHPSDR from the sources (Linux, RaspPi)

Appendix I

Installation of piHPSDR from the sources (MacOS)