CONTENTS

Introduction

The zBitx is an open source, multi-band, multi-mode, SDR platform covering the HF bands. At its heart is the popular Raspberry Pi Zero 2 single board computer that processes the signals digitally, allowing the user to not only experience high performance radio reception but also to “hack” the radio. This makes the zBitx “future-proof”, which means that the user can work any new, state-of-the art digital mode by just installing the required software and configuring it to work with the zBitx.

The radio covers frequencies from 3 MHz to 30 MHz with maximum power output of 5W (varies according to the input voltage) with a visible spectrum displaying a 25 KHz of the selected band.

Weighing in at just around 500 grams, the zBitx is pocket friendly and can be set up for operation from anywhere in a matter of a few minutes.

The zBitx operates off 6-9 V, and comes equipped with a 18650 Li-Ion battery holder for portable operation.

What’s in the box?

The package contains:

- zBitx with 32GB microSD card (pre-installed)
- Stylus for the touchscreen
- M3 Allen Key
- Micro USB cable
- Micro USB OTG adapter

Initial Setup

The zBitx requires minimal setup to get on the air.

Set your shack power supply to 9V and use a DC barrel connector (5.5x2.1mm) to plug it into the radio. Ensure that the power supply can source a minimum of 3A of current.

Figure 01 - DC Power connector

To use the radio out in the field, insert two 18650 Li-Ion batteries, paying attention to the polarity of the cells.

Figure 02 - Battery holder with polarity markings

Attach a 50 Ohm antenna to the BNC connector. To power up the zBitx, simply plug in the DC power connector into the DC power port, located on top of the radio.

If the user wishes to use a morse key/paddle, an external microphone, an external earphone/amplified speaker, they can do so by plugging them into the **K**ey, **M**icrophone, **E**arphone jacks on the top of the radio. The **HDMI** port can be used to connect to an external display.

There are two micro USB ports on the top - one marked “**USB**”, for use with USB peripherals, and the other marked “**CAT**”, for plugging into the shack computer for external CAT control andto update the front panel firmware.

Figure 03 - zBitx side panel connectors

Once powered up, the display comes on as the Pi Zero begins to boot up.
You should see the message displaying the firmware version and a message stating that the zBitx is starting up.

Once the Pi Zero boots up, the waterfall is displayed and you should hear the band noise on the speaker.

Setting up your zBitx

To set up the zBitx, press the **SET** button on the top right corner of the display.

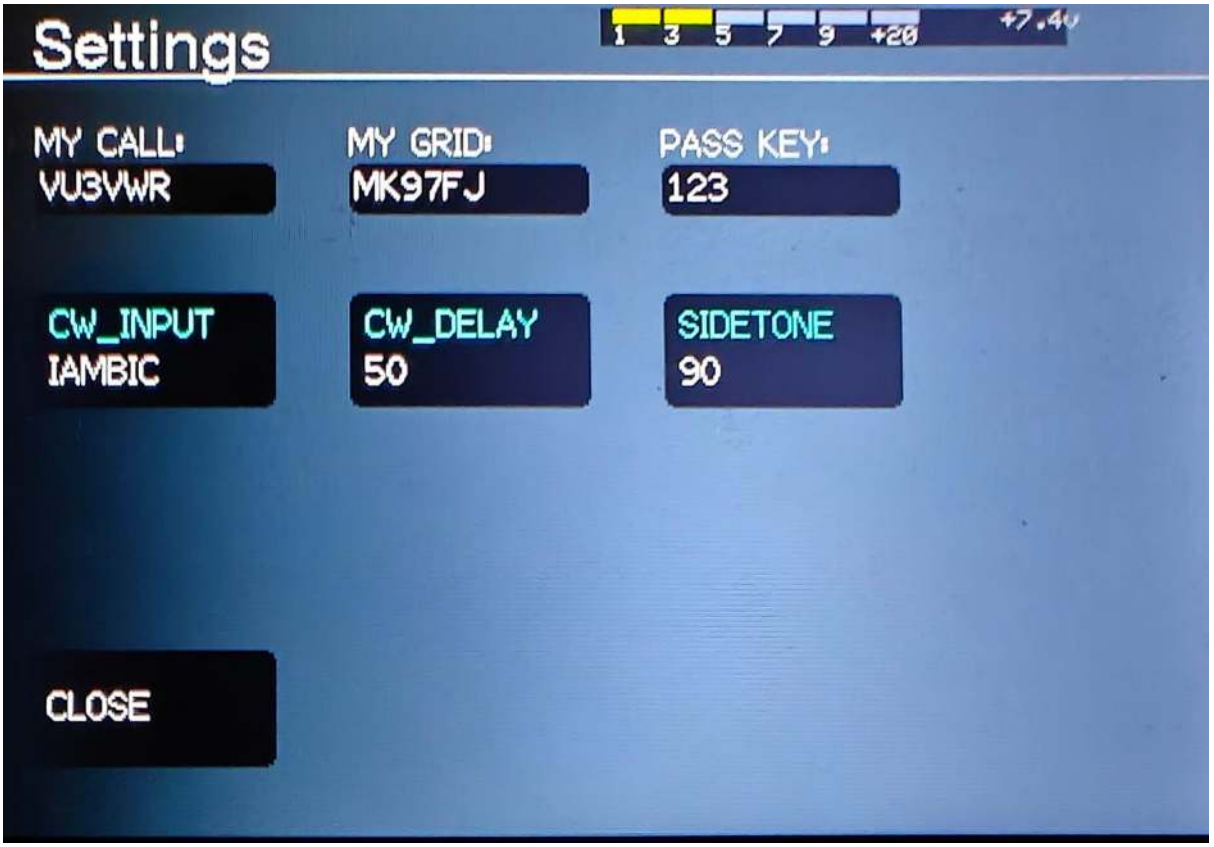

Figure 04 - Settings Menu

Enter your callsign in the **MYCALL** field by pressing the black text input field.
This should bring up the onscreen keyboard that should allow you to input the text.

Enter your grid locator in the **MYGRID** field. The zBitx uses the callsign and grid to create preset messages for use in CW and FT8 modes.

Press the **PASS KEY** field to set the password for the web interface.

The **CW INPUT** option is used to tell the zBitx the intended method of sending CW - straight key, Iambic A, or Iambic B. Select this by pressing the button and rotating the knob or by pressing it multiple times.

The **CW DELAY** option is used to set the hang time, that is, how long the radio stays in transmit before switching to receive.

The **SIDETONE** option is used to set the volume of the sidetone that is heard on the speaker when keying the radio on CW and FT8 modes.

Once these options are set, press the **CLOSE** button to return to the main screen.

Synchronising time

The zBitx comes factory synchronized. However, all the internal real time clocks tend to drift and they may need to be synchronized time servers on the Internet.In order to work FT8 and record the correct time when logging a contact, the time on the Real Time Clock needs to be updated.

To do this,

- Plug in a mini HDMI cable into the **HDMI** port and connect it to an external display. A mouse and keyboard needs to be connected to the **USB** port.
- Apply power to the zBitx. Once the radio boots up, the zBitx desktop is displayed.
- Connect to the local WiFi network by selecting it from the WiFi menu. The zBitx will automatically update the time in a few minutes, and the updated time should now be seen.
- Once the time has been updated, shut down the device and remove the HDMI and USB cable.

Figure 05- WiFi Menu

Controls on the zBitx

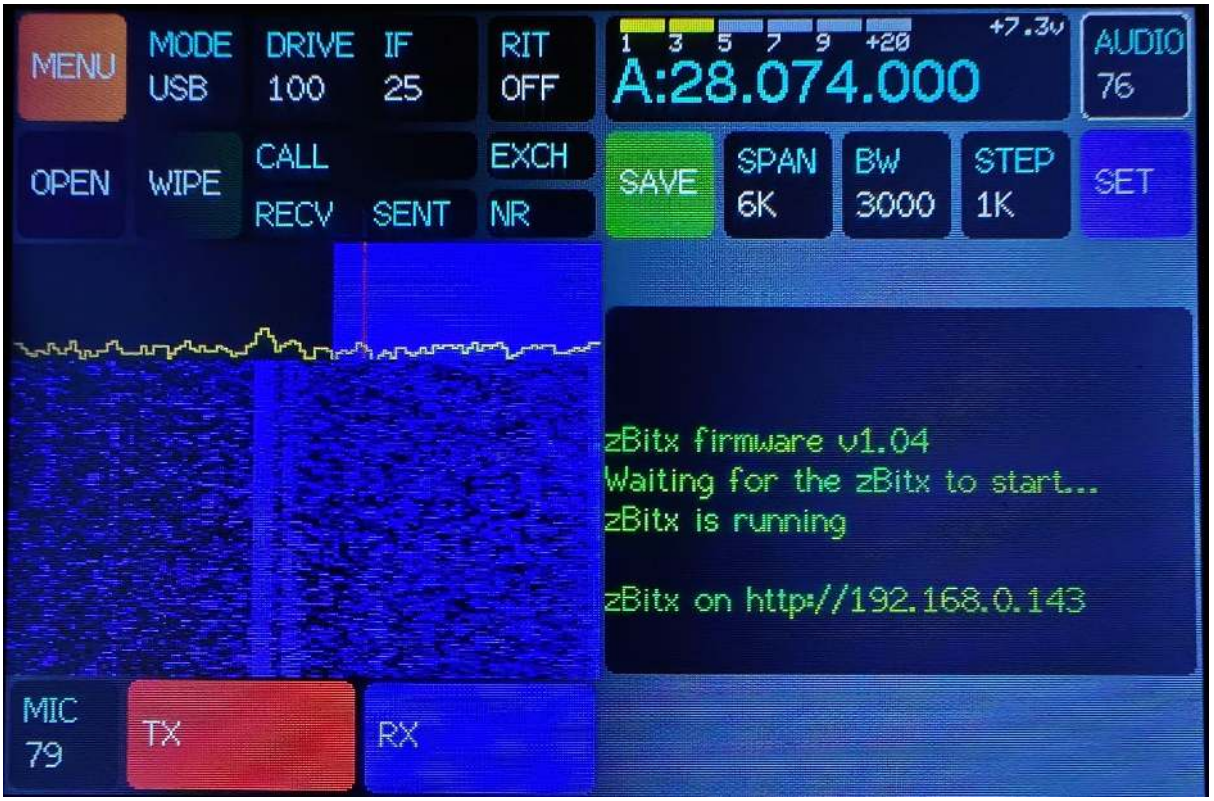

Figure 06 - zBitx main screen

To interact with the various controls on the zBitx, tap it once to select it. The feature is now highlighted in white. Rotate the tuning knob to change the value. For example, to increase or decrease the volume, select **AUDIO** and rotate the knob.

Some controls can also be changed on the fly by pressing on it. For example, to switch on the RIT function, simply press on the **RIT** button once. To switch it off, press the **RIT** button again.

Let’s explore the controls on the radio.

1. **MENU**: The menu option is used to select the band, set the AGC, switch between VFO A and B, and to set the VFO to operate in split mode.

Figure 07 - Menu screen

a. **BAND SELECTION**: To select the band, press on the relevant band button once. For example, to switch to 7MHz, press the **40M** button once. The radio immediately goes back to the main screen and the VFO displays the frequency of the radio.

b. **BAND STACKING**: The zBitx allows band stacking, I.e. the user can set four different frequencies for different modes, which allows the user quick access to certain frequency and mode.

   For example, for the 10M band, the first stack can be set to **CW**, second for **USB**, third for **FT8**, and the fourth for **AM**.

   To do this,

   - Press the **10M** button once. This will select the first stack and the radio returns to the main screen. Select the desired mode and set the frequency.
   - Press the **MENU** button and press the **10M** button again. This will select the second stack. Select a different mode and set the frequency.

- When you open the **MENU**, you should see “2” displayed under the selected band this time. Press the **10M** button again to select the third stack. Repeat the procedure to set the four stacks.
- Once the four band stacks are set, you can cycle through them by tapping the **10M** button. Any changes made to the stack are stored and will be recalled when the stack is selected again later.

c. **AGC**: The automatic gain control, or the AGC, automatically adjusts the amplification of a signal to maintain a constant level of volume, regardless of the signal strength.

There are four settings - OFF, SLOW, MEDIUM, and FAST. These determine the speed at which the AGC reacts to a change in signal level.
When you're dealing with a loud voice signal, the SLOW setting will help maintain a consistent gain and filter out most background noise that’s much quieter than the main signal. On the other hand, the FAST and MED settings allow for quicker adjustments, making them ideal for scenarios where you want the AGC to closely track the primary signal within the passband. The AGC can be switched off for modes such as FT8 where quick changes can cause data loss or when hunting for weak signals on CW.

d. **VFO**: VFO A and VFO B are two separate tuning controls that enable you to set the receive and transmit frequencies independently. This allows you to switch between two different frequencies quickly. Typically, VFO A is the primary frequency on which you transmit and VFO B is used to listen for replies. You can change the VFO by pressing on the **VFO** button once.

e. **SPLIT**: Enabling **SPLIT** allows you to transmit on one frequency and receive on another. This is very useful when working pileups and allows you to pick out individual stations from multiple calling stations.

2. **MODE**: This is used to select the mode of operation - LSB, USB, 2TONE, DIGITAL, AM, FT8, CW, CWR. To select a mode, you can either press the **MODE** button multiple times, or press it once to highlight it and rotate the knob to select the desired mode.

3. **DRIVE**: This sets the output power of the zBitx. Lowering the drive value reduces the output power. This is useful when operating QRPp or when connecting to an external amplifier or transverter which require low power input. When the drive is set to zero, the radio doesn’t transmit any signal.

4. **IF**: This sets the receiver gain of the zBitx. You can increase the IF gain according to band conditions to hunt for weak signals and lower it to reduce the strength of strong signals. If this is set correctly, the waterfall should look blue with some black spaces interspersed between. If the IF is set too high, the waterfall turns orange.

5. **BW**: The bandwidth control sets the bandwidth of the filter. The user can lower this all the way to 50Hz for modes such as CW and increase it to 5000Hz for modes such as AM or SSB. This allows you to enhance the selectivity, improving the received signal quality, and eliminate interference from other stations.

6. **AUDIO**: This is used to set the audio volume of the zBitx.

7. **SET**: This is used to access the settings menu. This menu is used to input the user's callsign, maidenhead locator grid, web interface password, CW input, CW delay and the sidetone volume.
The CW delay option lets the user set the hang time for when the radio switches from transmit to receive.

8. **STEP**: This is used to select the step size for tuning. A high step size allows the user to quickly tune across a band. A lower step size can be used to fine tune into a signal. Lower step sizes are typically preferred when working narrow bandwidth modes such as CW to allow for fine tuning and to avoid skipping past a signal.

9. **RIT**: This switches the Receiver Incremental Tuning on or off. The RIT option is to fine-tune the received frequency without changing the main operating frequency. This is useful when working signals that are drifting away from the transmit frequency.

10. **SPAN**: This sets the width of the waterfall. For digital and CW modes, a span of 10KHz allows you to clearly spot individual signals. For LSB, USB, and AM, a span of 25KHz allows you to see how busy the band is.

Logging

Figure 08 - Logger

The zBitx interface comes with an easy to use logger that can be used to log contacts on the radio. This eliminates the need for an external logging setup.

To log a contact :

1. Enter the callsign of the station into the **CALL** field.
2. Enter any additional information such as the grid locator received from a station in the **EXCH** field.
3. Enter any additional information such as a contest exchange sent by the user to a station in the **NR** field.
4. Enter the signal report sent to a station in the **SENT** field. This could either be two or three digits based on the mode of operation.
5. Enter the signal report received by the user in the **RECV** field.
6. Pressing **SAVE** will log the contact. If the user wishes to reset all fields, simply press **WIPE**.
7. Pressing **OPEN** will open the list of contacts logged on the zBitx.

The log is saved in the sbitx.db file on the /home/Pi/sbitx/data folder.

VFO, S-meter AND Voltage Monitor

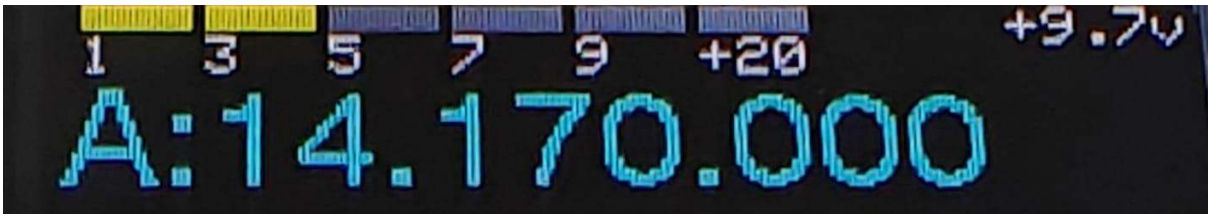

Figure 09 - VFO, S-Meter & Battery Voltage Monitor

The S-Meter is displayed on top of the VFO frequency. The user can refer to this when giving a signal strength report to the station they are working.

To the right of the S-Meter is the battery voltage monitor. This is used to quickly gauge the condition of the battery. It's recommended not to operate the radio below 6V to avoid damaging the batteries.

When the radio goes into transmit, the SWR and output power are shown instead of the S-Meter.

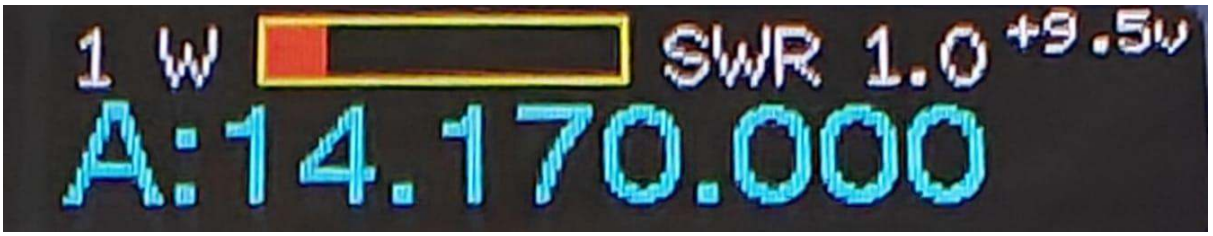

Figure 10 - Power and SWR meter

The VFO frequency is displayed below the S-Meter. It indicates the VFO in use (A/B), and the frequency that the radio will transmit on.

When RIT is switched on, the VFO displays the RX frequency, while maintaining the previously chosen transmit frequency. Use this when Dxing or contesting to receive stations that are slightly away from your own transmitting frequency.

Figure 11 - VFO with RIT enabled

Basic Operation

Operating CW

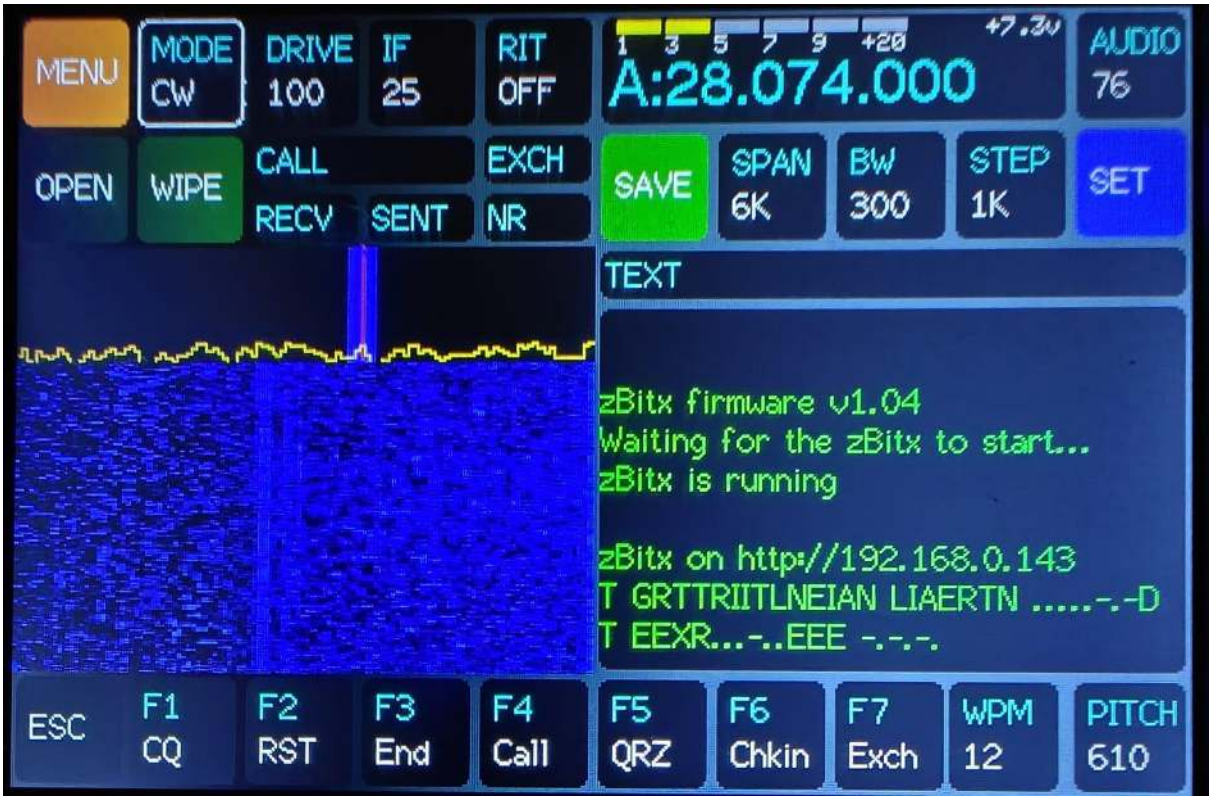

Figure 12 - CW Screen

In order to work CW,

1. Choose the band you wish to operate on, by selecting it from the MENU.
2. Set the **AGC** to slow. You can turn off the AGC when hunting for weak signals.
3. Set the **MODE** to **CW**.
4. Set the **BW** to 300Hz. You can set the bandwidth to 1KHz to hunt for weak stations and reduce it to 300Hz to reduce the noise and increase the readability of the signal.
5. Set the **STEP** size to 100Hz. Using smaller step sizes reduces the chances of skipping past a weak station.
6. Set the **SPAN** to 6KHz to narrow down the spectrum on the waterfall. This allows you to look for weak CW signals.
7. Tap on the **VFO** and set the frequency you wish to operate on.
8. Adjust the **IF** setting according to the prevailing band conditions. A high IF value results in overloading, while a very low IF value would result in poor sensitivity.

9. Set the **PITCH** to your preference. Changing the pitch value moves the red line on the passband of the filter. The **PITCH** control determines the tone of the received signal and interacts with the bandwidth function. For example, if the pitch is set to 600 Hz and the BW is set to 200, you can hear from 500Hz to 700Hz (200Hz centered on 600Hz).
10. If you are trying to work a station that is already transmitting, set the **WPM** value to a comfortable speed. If you wish to use the decoder function, center the blue passband strip on the waterfall display over the signal you wish to work by tuning into the signal. The decoder relies on the **WPM** setting to decode CW, so if it's having trouble decoding, try to match the speed at which you think the other station is transmitting.
11. In order to send CW, you can either use the onscreen keyboard, the preset macros, or use a paddle/straight key.
    a. To use the onscreen keyboard, tap the **TEXT** field and input your message and press **Start** to begin sending the message. To send CW in real time, press **Start** before you begin typing the message. Press **SYM** to bring up the extended keyboard. Here you will find prosigns, punctuations, and symbols. To stop sending a message, press **Stop**.
    b. To use the preset macros, press the button with the message you wish to send. You can also use the macro keys in the onscreen keyboard, by pressing the **SYM** button. The macros automatically fetch the details such as callsign of the station, signal report, etc from the logger. If you want to halt transmission mid way, press the **ESC** button. This will also clear the text fields on the logger.
    c. To use a paddle/ straight key, plug it into the key jack, set the **CW INPUT** in the **SET** menu and begin sending out your message.

Figure 12 - Onscreen keyboard

Figure 13 - Extended keyboard

In **CWR** mode, the zBitx tunes the receiver to the lower side band allowing you to have a cross-mode contact on CW with a station that is operating on LSB.

EDITING AND CREATING MACROS

Each set of macros is stored as a text file in the directory /home/Pi/sbitx/web with the filename extension ".mc".
They are easy to understand and the user can create a new macros file with any text editor. The macro files are text files written in the N1MM format.

The * is a shortcut for MYCALL (your own callsign). The ! is a shortcut for CALL (the callsign of the station you are working). The # is used to send the serial number for the current QSO. {SENTRSTCUT} will send numbers in cut format (5NN, instead of 599).

Please look at the N1MM help file on
https://n1mmwp.hamdocs.com/setup/function-keys/#macros for a list of all commands.

For example,
F7 Exch, * {SENTRST} # 73 would assign F7 for the macro button named "Exch" and would send the callsign of the station you are working, their RST report followed by the contest serial number followed by a 73 when pressed.

Sample macro file for regular dxing
F1 CQ, cq cq cq de {MYCALL} {MYCALL} {MYCALL} ar k
F2 RST, {CALL} ur rst {SENTRST} {SENTRST} kn
F3 End,! de {MYCALL} . tnx fer rpt . 73 es cu agn kn sk
F4 Call, *

F5 QRZ, qrz?
F6 Chkin, ! de {MYCALL} ur rst {SENTRST} es tnx fer call . hw cpy? AR K
F7 Exch,! de * ur rst {SENTRSTCUT} {EXCH} K
F8 Agn?,agn?
F9 Zone?,zn?
F10 About, ! de * . my name is farhan qth hyderabad . rig is zbitx es dipole . hw cpi? ! de * k
F11 QRZ,qrz?
F12 -,-

The macros are mapped to the function keys of any attached keyboard and can also be sent by tapping the macro button. A maximum of 12 macros can be defined by the user.

- In the web mode, all the macro files available are listed in the dropbox next to the macro button. The user can choose a different macro file depending upon the kind of operation they prefer. CW1 macro file is a good option to begin with.
- The macros will substitute the user's callsign, grid, the contact's filled callsign, report (from the fields on the logger).

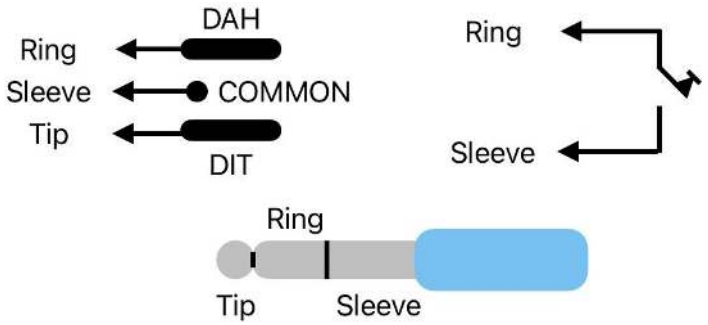

Figure 14 - Paddle/Key wiring

If you intend to use a paddle or a straight key for sending CW, follow the wiring diagram as shown in figure 14. The zBitx accepts a standard 3.5mm TRS stereo plug for the key input.

Operating FT8

The FT8 relies on starting the transmission/reception at every 15 second boundary. It is vital that the zBitx Real-time clock is properly synchronized within a second of the UTC. The RTC synchronizes itself using the NTP to within a few milliseconds each time there is Internet connectivity.

To do this, connect a HDMI display, an external keyboard, and mouse to the radio before powering it up. Once the desktop environment loads, select and login to your local WiFi network from the WiFi menu.

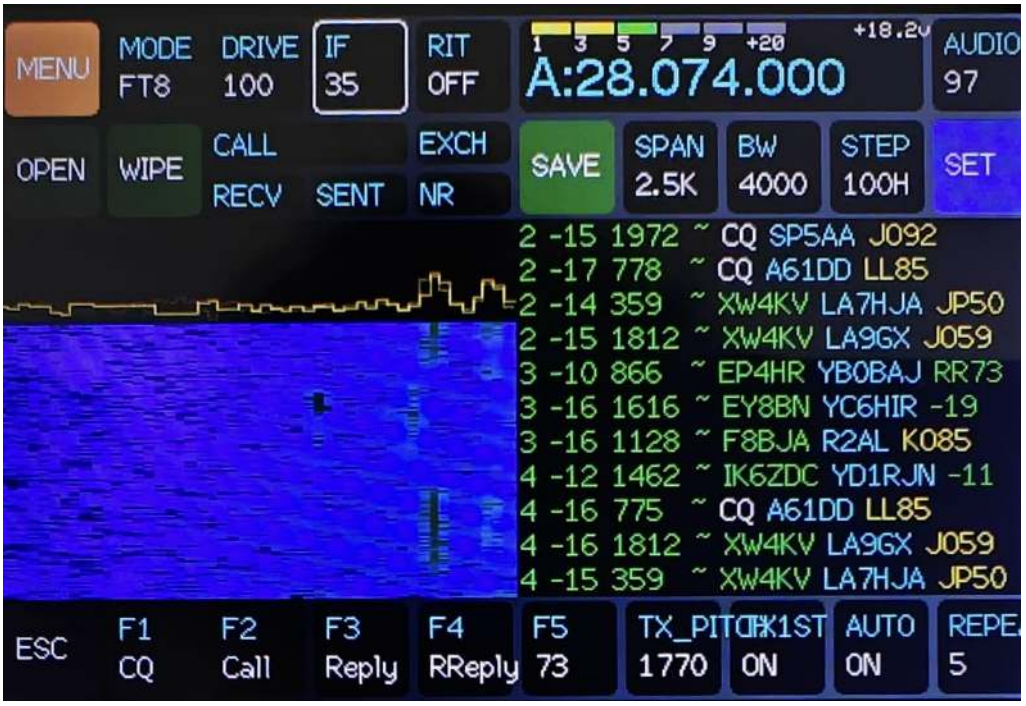

Figure 15 - FT8 Screen

To work FT8,

1. Choose the band you wish to operate on, by selecting it from the **MENU**.
2. Switch off the **AGC** from the **MENU**.
3. Tune the radio to the FT8 calling frequency for the band.
4. Set the MODE to **FT8**.
5. Set the **BW** to 4000. This wide bandwidth setting enables the zBitx to receive almost twice as many signals as a conventional analog radio. It also eliminates the need to use the **SPLIT** to transmit or receive at greater offset from the base frequency.
6. Set the **SPAN** to 10KHz.
7. Set the transmit frequency by adjusting the **TX PITCH**.
8. Listen for a while. You should begin to see messages on the band activity pane on the left of the waterfall.

Figure 16 - Waterfall and Band activity panes on FT8 mode

9. To browse through the list of decodes, press on the band activity pane and rotate the main knob to scroll through the band activity. The format of the decoded message is time slot, signal strength, the received signal offset and the text transmitted.

   For example the first decode in figure 16 indicates that the transmission occurred during the second time slot, with a signal strength of -15dB at an offset of 1972 Hz with the message CQ SP5AA JO92

10. If you wish to answer a CQ call, select the station calling CQ and press the encoder knob once.
11. If the **TX1ST** option is enabled, the radio will begin transmitting a CQ during the next even time slot. If this is disabled, it will transmit during the next odd time slot.
12. If the **AUTO** option is enabled, the zBItx will automatically respond to any replies heard for your callsign. If this is disabled, you will have to manually select the reply from the macros.
13. The **REPEAT** function sets the number of times the zBitx sends out a message, if no reply is heard. For example, if the REPEAT value is set to 5, the zBitx will send the message five times if no response is heard.
14. If you wish to call CQ, press the **CQ** macro button and the zBitx begins calling CQ till someone responds to your call. Any responses to your call are shown in white color. If no response is heard, the zBitx will call CQ based on the **REPEAT** value. If a response is heard and the **AUTO** setting is enabled, the zBitx will respond to the reply, complete the QSO, and log it.
15. A quick glance at the logger allows you to monitor the progress of a QSO. As the QSO progresses, the zBitx automatically updates the fields on the logger.
16. Pressing the **ESC** key will halt the transmission and clear the text fields in the logger.

Operating USB/LSB/AM

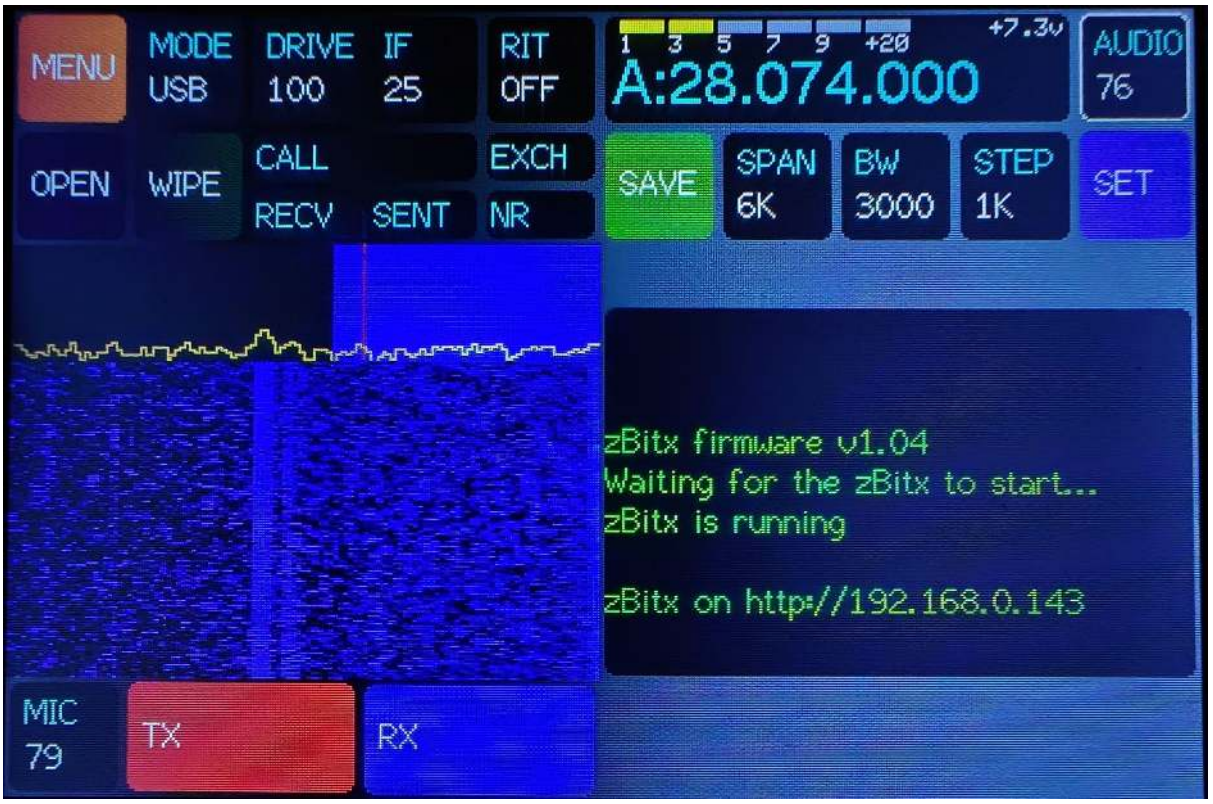

Figure 17 - USB Screen

To work USB/LSB/AM,

1. Choose the band you wish to operate on.
2. Set the frequency you intend to transmit on.
3. Set the mode to **LSB**/**USB**/**AM**. The convention is to use **LSB** below 10MHz and **USB** above 10MHz.
4. Set the **BW** to 2.5KHz. This is the standard SSB bandwidth of most commercial radios.You can reduce this if there is interference from stations near the operating frequency.
5. Set the **SPAN** to 25KHz or 10KHz, depending on how crowded the band is.
6. Set the **STEP** size to 1KHz if you wish to slowly tune across the band. You can set this to 10KHz to quickly scan a band.
7. Set the **AGC** to **SLOW** if the band is quiet. Set it to medium, if the band is noisy. You can also switch off the AGC if you are trying to copy a weak station.
8. If you plan to use an external microphone, plug it into the microphone jack on top of the radio.
9. Set the **MIC** gain depending on the operating environment. You can refer to the modulation monitor that appears above the waterfall to set the **MIC** gain. Set the gain so that the radio is modulating within the levels of the monitor when

you’re speaking into the microphone. Higher values can result in distortion, so be careful not to set this too high.

10. Press the **TX** button to begin transmitting and speak into the microphone. The radio remains in TX until the **RX** button is pressed. This allows for hands-free operation.
11. When listening to a station working split or drifting in frequency, enable the **RIT** option by pressing the RIT button. Now you can change the RX frequency without changing the TX frequency.
12. If you are facing a pileup, you can spread out overlapping callers over a few kilohertz by enabling the **SPLIT** option in the **MENU**. This allows you to set VFO A to a transmit frequency and VFO B to receive frequency. Now, you press the TX button, the radio will transmit VFO A’s frequency and switch to VFO B on receive.

Figure 18- USB/LSB/AM controls

If you plan to use an external hand microphone, please ensure that the microphone and the PTT switch is wired as shown in figure 17.

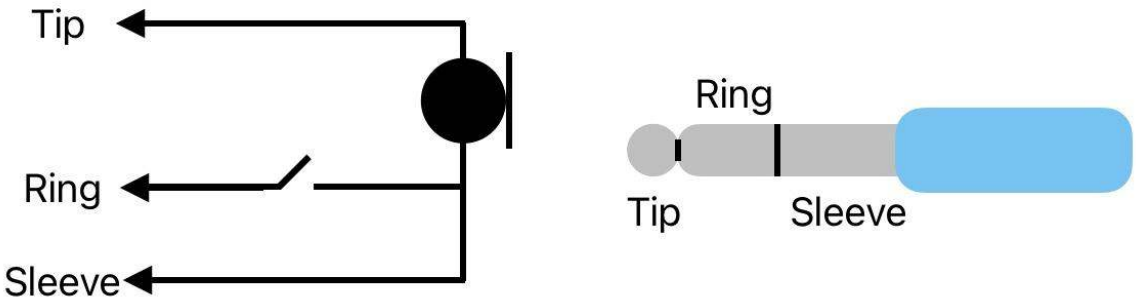

Figure 19 - PTT microphone wiring

DIGITAL MODES

The Digital mode is used to route the audio and PTT signals from an external third party program such as QSSTV or JS8Call. The zBItx needs to be connected to an external display, keyboard, and mouse in order to work with third party software.

To operate third party digital modes,

1. Choose the band you wish to operate on.
2. Set the frequency you intend to transmit on.
3. Set the mode to **DIGI**.
4. Enable Hamlib in the third-party software's settings..
5. From the third party software, choose **Hamlib NET rigctl** as the radio.
6. Set the Network server to 127.0.0.1, port 4532 (standard for NET rigctrl)
7. Set the **PTT** Method to **CAT**
8. Set the **MODE** (if prompted) to **USB**
9. Choose **plughw:CARD=Loopback_DEV=1** as the Input/Record Audio. Alternatively, you can also choose **Loopback: PCM(hw:1,1)**
10. Choose **plughw:CARD=Loopback_DEV=0** as the Output/Playback Audio. Alternatively, could also choose **Loopback: PCM (hw:2,0)**

2TONE

The 2TONE mode is used for testing the zBitx transmitter's IMD by transmitting a 2 tone signal. While most users won't need to use this mode for testing, it can be used to quickly check the output power of the radio, the SWR of the antenna, or to tune an ATU connected to the zBitx.

Press the TX button to begin transmitting a signal and make the adjustments. Press the RX button once you are finished.

UPDATING SOFTWARE AND FIRMWARE

Updating the zBitx

The zBitx contains a Raspberry Pi Zero that does the signal processing. The software for the Pi Zero needs to be updated once in a while to ensure that bug fixes are applied and the radio functions properly. It's good practice to update the software once a month, so that new features and bug fixes are implemented in the radio.

To update the software on the Pi Zero,

1. Remove power to the zBitx.
2. Connect an external HDMI display, a USB keyboard and mouse to the zBitx.
3. Power on the zBitx. You should see the zBitx desktop environment load on the external HDMI display.
4. Once the desktop environment loads, the zBitx application will automatically load up in a few seconds. Close the applications on screen.
5. Ensure that the zBitx is connected to the internet. If not, use the WiFi menu to login to your local network.
6. Open a terminal window. In the terminal window, type the command "cd sbitx" and press enter. Follow this up with the command "./update" and press the enter key. This begins the automatic software update. The zBitx will fetch the latest files from the internet and update the software on your radio.

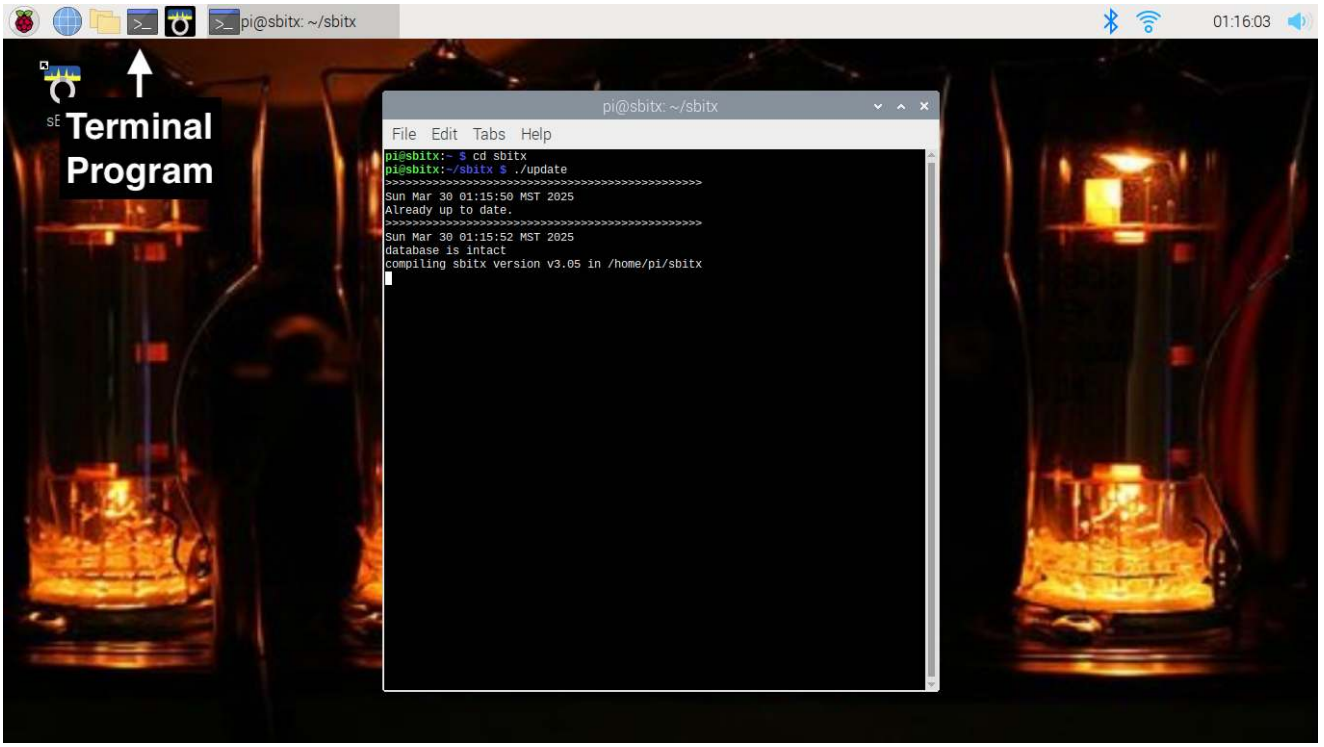

Figure 20 - Terminal window for updating the software

7. Once the update is complete, you can restart the radio and unplug the external display and USB peripherals.

Updating the firmware

The front panel of the zBitx functions as an interface to the Raspberry Pi Zero, that does the processing of the signals.

A Raspberry Pico microcontroller board acts as the intermediary between the Pi Zero and the front panel controls and needs to be updated, whenever there is change in the user interface.

To upgrade the firmware on the Pico,

1. Download the Pico firmware file (uf2 file) to your computer from the zBitx product page (https://www.hfsignals.com/index.php/zbitx/).
2. Unplug the power to the zBitx.
3. While keeping the main knob button pressed, attach the supplied USB cable between the PC and the CAT port on the zBitx. Remember, you have to attach to the CAT port, not the one marked USB.

Figure 21 - USB Cable connected to CAT connector

4. Now, let go of the knob. The zBitx display will light up with an all grey screen. The Pico controller will show up as a removable drive on your PC.
5. Copy the .uf2 firmware file from the PC to the Pico controller.
6. The Pico microcontroller is now updated. Remove the USB cable.
7. Restart the zBitx. The latest firmware should show up on the display.

Web Mode

The zBitx can be remotely controlled by a PC, phone, or tablet connected to the same WiFi network as the zBitx.

To connect to the zBitx over WiFi,

1. Power on the zBitx and ensure that it is connected to the local WiFi network.
2. Connect your PC/phone/tablet to the same WiFi network as the zBitx.
3. Open the browser on your PC/phone/tablet and navigate to http://sbitx.local.
4. The zBitx web interface will now be displayed. Login with the pass key you had set earlier in the **SET** menu. The default pass key is 123.
5. Once you log in, you should see the web interface with all the controls on the zBitx here. Any change you make on the web interface is mirrored on the zBitx. The audio from the zBitx can be heard on your PC/phone/tablet, allowing you to work CW and FT8 remotely.
6. In CW, can either use the built in macros or type out a message in the TEXT/CMD field. In FT8, you can use the preset macros to make contacts.
7. It is possible to work USB/LSB/AM via the web interface, but an external microphone needs to be connected to the zBitx as the radio does not support voice transmission from a PC/phone/tablet at the moment.

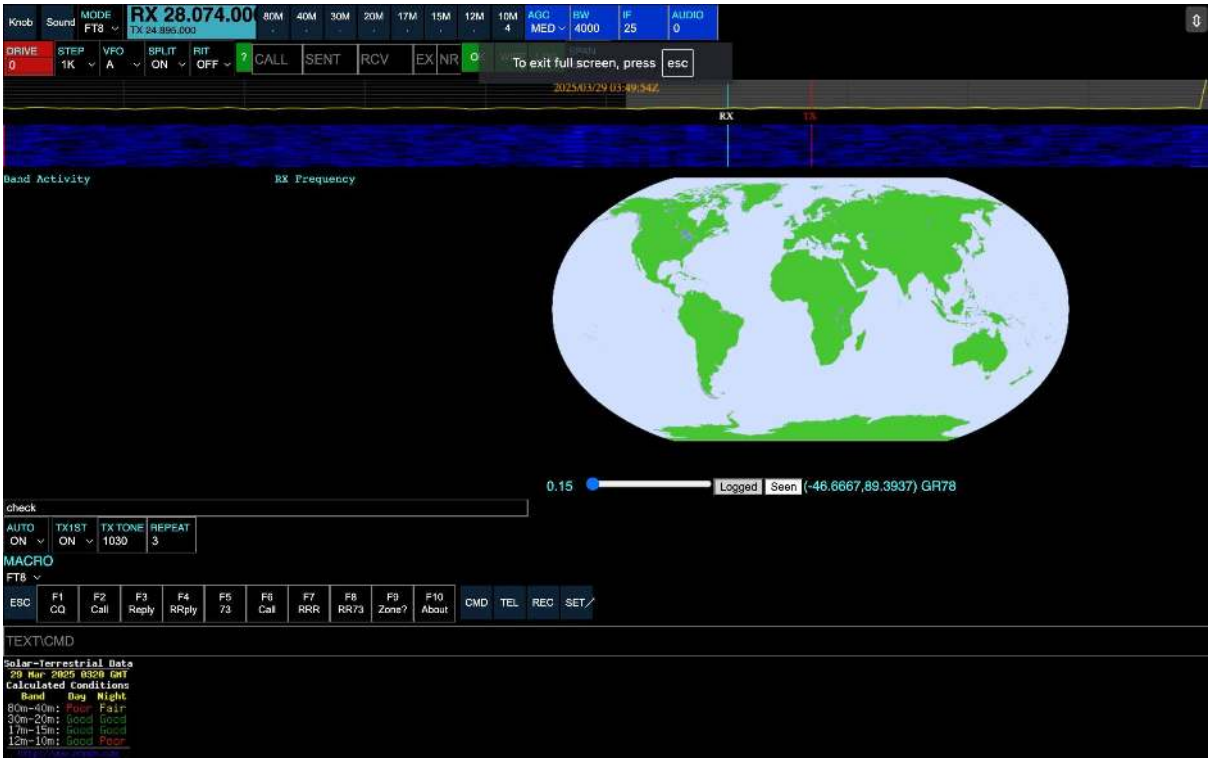

Figure 22 - Web Interface

The controls on the web interface are similar to the zBitx interface, except for a few additions.

1. **KNOB** - This bring up the virtual tuning knob. This is useful when you don't have a mouse. With a mouse attached to a PC/tablet, click on the VFO and use the scroll function on the mouse (when operating the interface from a PC) to set the frequency.
2. **SOUND** - This mutes and unmutes the audio from the zBitx on the PC/phone/tablet.
3. **CMD** - Used to bring up the command terminal for use with text commands
4. **TEL** - This brings up the Telnet window.
    - To telnet into a server, use the command TELNETURL [server:port] to store the URL of the telnet server.
      For example, TELNETURL dxc.g3lrs.org.ul:7300, connects to the g3lrs server.
    - Once this is set, press **START** to begin the telnet session and press **STOP** to terminate the telnet session. You can minimize the telnet window by closing it and reopening it, as long as the session isn't terminated. Check the telnet server's helpfile to figure out the commands for the server.
5. **REC** - This function allows you to record a QSO in progress.
6. **SET** - This brings up the settings window for settings your web interface pass key, callsign, grid, and telnet server.
7. **MACRO** - This brings up a list of available macros in CW. This is useful during contesting and Dxing, as it allows the user to quickly select and send regularly sent exchanges.
8. **TEXT** - This field allows the user to quickly type out a message when in CW mode. The text is transmitted in real time by the zBitx when it is entered here.