

# W0EB/W2CTX Alternate Firmware for the Micro BITX 80-10 meter Transceiver

---



uBITX built by W0EB

**Instruction manual for the alternate uBITX transceiver software  
written by Ron Pfeiffer, W2CTX and Jim Sheldon, W0EB,  
including the minor hardware changes needed to make it run  
properly on CW.**

**Use this software AT YOUR OWN RISK.**

Every effort was made by the authors to ensure the software is trouble free and safe to use. However, NO WARRANTY is provided or implied and it is up to the person(s) using this software and this manual to read and understand the following before proceeding. Minimal support is available from the authors. See credits section at the end of this document.

## INTRO:

uBITX - Necessary information for programming and utilizing this software:

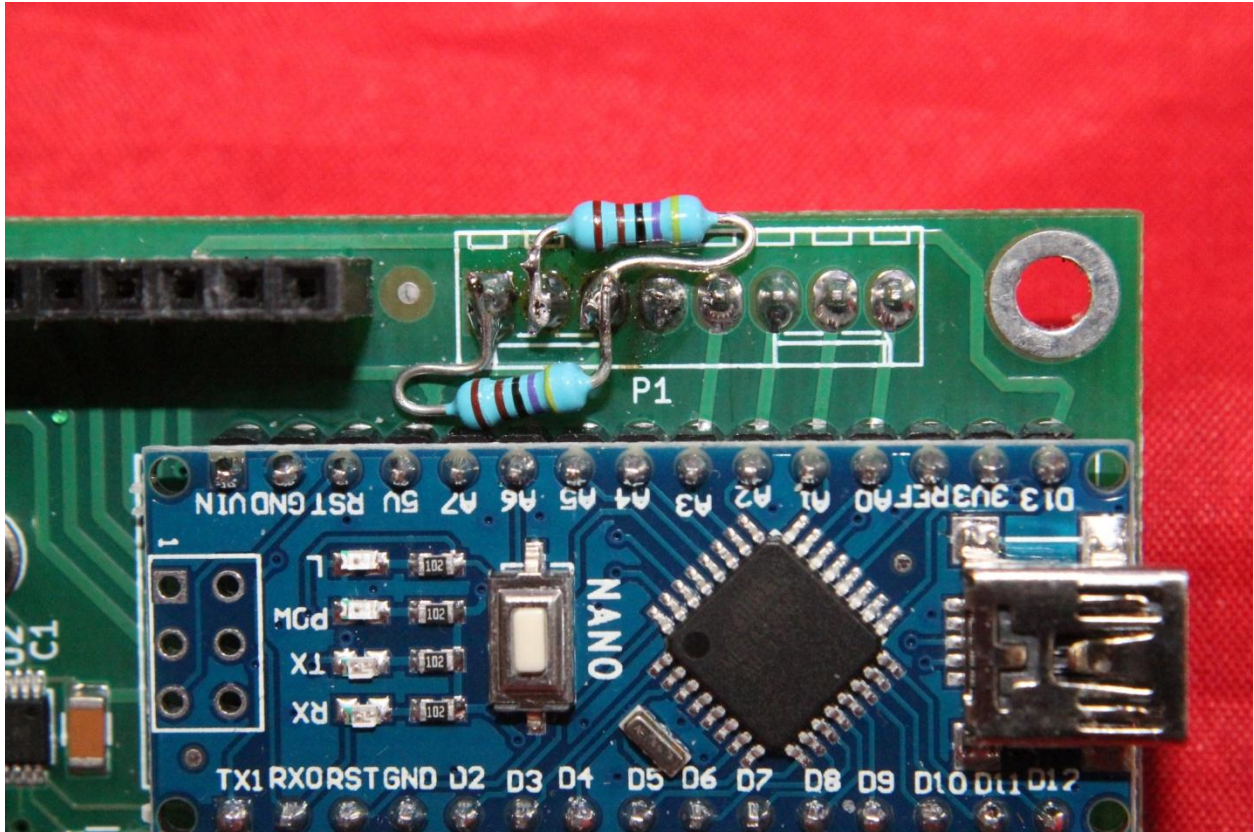
Three things need to happen for the user to be able to load and operate this software with the Micro BITX transceiver or as it has been called, the uBITX.

First, we assume the user knows how to connect to the Arduino “Nano” on the “Raduino” card that controls the transceiver and has installed the Arduino “IDE” to allow for program compiling and uploading software to the card. There is an excellent tutorial available via a link on the HF Signals website. You must become familiar with this part before any reprogramming of the uBITX can take place.

Second, you will need to unzip the ubitx\_V1\_14.zip file into your Arduino directory. This file contains ALL the necessary files to allow the program to be compiled and uploaded to the uBITX by the Arduino IDE program.

Third, you will need to add a resistor to the originally unused spare pin on the Raduino’s “Digital” connector. This is pin 1 of the connector and has a violet wire that was originally indicated as a “spare” and is connected to the A7 (analog input #7) on the Nano.

When you wired up your uBITX in the first place, you were told you MUST add a 4.7K resistor between the A6 (Blue wire) input and +5 volts (Green wire). We now require a second 4.7K resistor to be wired between the +5 volt (Green wire) and the A7 (Violet wire) input to make the CW Keyer module of the software operate properly. It IS okay to install these resistors directly on the Raduino card’s terminal pads (reverse side of the “Digital” connector).



The above picture shows the two resistors soldered in place on the Raduino Board.

The key jack also needs to be rewired slightly – the Blue wire needs to be connected to Pin1 (tip) of the key jack and the Violet wire to Pin2 (ring) of the key jack. The tip connection becomes the dot paddle and hand key (or bug or external keyer) input to the keyer and the ring connection is now the dash paddle to the keyer.

Once this is done, no special contact conditioning is needed for either the paddle contacts or a hand key/bug's contacts as instead of reading several voltage levels created by the extra 2.2K and 10K resistors (no longer needed) between the dot/dash contacts of the paddles we now use separate lines to read the paddles. Since only analog input lines were available the 4.7K pull up resistors were needed. Now, when we close a contact on the paddles or the hand key, we only need to determine Key UP

or Key DOWN conditions and the key down voltage does not have to be completely at Zero volts to indicate Key DOWN. This makes reading the contacts much easier in the software and ends the criticality of having “squeaky clean” contacts on your key(s) though this is still recommended on principle.

It is no longer possible to easily detect the difference between paddles or other key types just by plugging them in without more code than would fit in the available programming space so the selection has been moved to an encoder selectable “Menu” that contains multiple selections for various functions on the uBITX and a subsequent chapter will deal with navigating those menus.

Another minor hardware change the user might want to make is to provide a second function pushbutton separate but in parallel with the button on the encoder. When selecting menu items with the button switch on the encoder it is easy to accidentally turn the encoder slightly when pressing this. That can have the effect of selecting something you didn't really want and cause you to have to go back and do some operations more than once. With a separate switch, pushing the button will be done with your hand away from the knob on the encoder and make life a bit easier. This is not absolutely necessary and it's up to the user to decide whether or not he or she wants to add this modification.

That covers the hardware changes necessary for making the keyer work properly and the convenience modification the user may want to make.

## uBITX Operational Modes:

We have created 3 operational modes in an attempt to optimize the basic HF SSB/CW operations. The modes are selected during power-up via the encoder (or separate) push button.

The three modes are: Normal, EEPROM UPDATE and ALIGNMENT. (ALIGNMENT is the original "Factory Alignment" programming).

All mode selections occur ONLY at power-up. They are mutually exclusive and when you are finished using the EEPROM UPDATE and ALIGNMENT modes you must cycle the power (Turn the uBITX off and then back on again).

## SELECTING MODES:

**Normal Mode:** General radio operations. Power up your uBITX and do nothing. The normal radio features will be defaulted. This is the mode you will use to operate the uBITX on the air.

**EEPROM UPDATE Mode:** This mode will accept EEPROM reads/writes via Ian Lee's (KD8CEC) Windows EEPROM Manager program. (In the future there will be a Linux EEPROM manager).

**\*\*\*\*\* NOTE: CAT is NOT supported in this version.**

In order to use Ian's EEPROM Manager:

1. Power up your uBITX.
2. When the Version banner is displayed, push the encoder button.
3. Keep the encoder button pushed until "EEPROM" is displayed.
4. When "EEPROM" is displayed release the encoder button. You may now use Ian's program (follow his directions) to read/write the

EEPROM memory locations. (Not for the faint hearted, but it will allow you to change values in the various locations.)

To exit this function close down Ian's program and depress the encoder button until "EXIT" appears on the top line of the uBITX display. Tap the button to Exit. Power cycle the uBITX and it will come up in "Normal" mode with the changes you made applied.

ALIGNMENT Mode: Will present the menu for performing a factory alignment (Ashhar Farhan's original factory alignment code has not been changed so follow the original uBITX instructions for setting Calibration and BFO.)

1. Power up your uBITX.
2. When the Version banner is displayed push the encoder button.
3. Keep the encoder button pressed until "Alignment" is displayed.
4. Once "ALIGNMENT" is displayed, release the encoder button – Turn the encoder knob to select the desired alignment function, either Calibration or Set BFO. After these functions are complete, turn the encoder knob to the left until "Exit" appears and press the button to exit. Even though the uBITX will appear to come up in "Normal" mode on exit, you will still need to power cycle the radio to make sure the alignment changes are applied.

#### NORMAL MODE FEATURES:

Many people expressed the desire for a different method of changing bands and tuning speed as the original method was sometimes difficult to control.

We removed the “faster the knob is turned, the faster the frequency changes” as the timing implemented the speed-up far too quickly for many people and it was hard to set an exact frequency if you happened to slip. The frequency would sometimes jump by 2-300 KHz and then you had to crank forever to get back to where you were. Instead, we created a function where a short press (tap) of the encoder button would change the position of an underscore cursor and allow you to select the digit of the frequency you wished to change. This allows for more precise quick setting of the frequency.



The cursor under the digit can be moved by tapping the encoder button.

## MENUS:

Now for the menus I mentioned near the beginning of this manual.

To enter the Menus, power up your uBITX and wait for it to boot up into the default which is “Normal” mode.

Now, press and hold the encoder button until “Menu” is displayed. Once “Menu” is displayed, release the encoder button and “Exit Menu?” will display. Originally “Band Select?” was the first item and exit was the last. This occasionally created a problem if you happened to accidentally tap the encoder button again after entering the menus or if there was a bit of contact bounce due to a faulty switch, you would be in the “Band Select” portion of the menu and the only way out would be to select a band. This got frustrating too so the order was changed a bit.

Menu items are selected by rotating the encoder knob to one of the following and briefly pressing (tap) the encoder button to select the item. Each item is explained in detail following the list.

1. Exit Menu?
2. Band Select?
3. RIT:Off, On?
4. Pdl:WPM Change?
5. Select VFO X? (X is either A or B)
6. Select SSB? (or CW depending on the mode you are in).
7. Select USB? (or LSB)
8. Select PADDLE (or KEY)
9. Change CW Tone
10. Version?



The first Menu item is “Exit Menu?” so an accidental extra press of the button will get you back to Normal operating mode.

Once you are in the “Menus”, turn the encoder knob to the right (it’s slowed down a bit to make finding the selections a bit easier) and the next selection will be “Band Select?”. A short press (tap) of the encoder button will enter the band selection menu with “80” displayed on the top line of the display. Turn the encoder knob to the right until the band you want is displayed and then tap the encoder button. The radio will tune to the normal QRP operating frequency for that band and return to “Normal Operation” without further operator intervention.

**NOTE:** If you are in SSB mode when you enter the menus, a “Band Select” will set the frequency to the QRP SSB calling frequency for the band and if you are in CW mode, the QRP CW frequency will be set. Since only CW/DATA is allowed on 30 meters, only the QRP CW frequency will be set on that band.

Next menu item is “RIT:Off, On? Which means the RIT is off do you want it on?. Tap the button to turn RIT on. Normal mode will be resumed but now there will be a lower case “r” displayed in front of the VFO A or B (whichever is selected) and the encoder knob will only tune the receive frequency, keeping the transmit where it was before you turned on the RIT. To turn RIT off, enter the menus and select RIT mode again – if it is ON when you turn to that item in the menu it will show “RIT:On, Off?” indicating that RIT is ON and asking do you want to turn it Off?. Press the encoder button, RIT will be turned off and Normal Operation will be resumed. The displayed frequency will be reset to what it was before you turned RIT on and that will still be your transmit frequency.

Next is the keyer speed setting. It will show “Pdl:WPM Change?” and when you press the encoder button, the top line will display “WPM:XX where XX is the current keyer speed. Use the encoder knob to set it to what speed you want. The bottom line of the display reads “button to set” indicating that once you have your desired speed set, tap the button and you will return to Normal Operation with the keyer speed set to whatever value you changed it to.

The next item is the VFO A or B select. If you are currently using VFO A, The top line will show “Select VFO B?” pressing the button will select VFO B, set the uBITX to VFO B’s frequency and mode. Additionally it will save the frequency and mode that VFO A was set to in EEPROM. The next time you select VFO A it will tune to that frequency (and mode). If you want to save a particular frequency & mode in either VFO A or B, first tune that VFO to the freq/mode you want, enter the menus, select the VFO change item and change to the opposite VFO. Then tune that VFO to what you want it to be (or leave it along if it’s already what you want) and again change to the opposite VFO. A little cumbersome but it becomes intuitive once you get used to it.

Next is the SSB/CW operating mode select. If you are in CW mode the item will indicate “Select SSB?” on the top line. Pressing the encoder button will switch to SSB mode (BE CAREFUL if you have phone sub bands in your country, as it will switch to SSB even in the CW sub bands). The next time you enter this menu item it will display “Select CW?” on the top line and another button press will switch to CW mode.

NOTE about CW Mode – In SSB mode, the dial frequency is both the Receive AND transmit frequency of the uBITX and when you tune the

receiver to best sounding signal you are zero-beat with the sending station's suppressed carrier. In CW mode, the display MUST show the actual frequency of our transmitter but if we were exactly zero-beat with the other station's CW signal we would not hear it. Therefore, in CW mode we must offset the RECEIVER (not the transmitter) by whatever value our keying sidetone is so even though we are transmitting zero beat to the other station, and our display MUST indicate the actual transmit frequency, we are not able to hear the other station's signal because OUR RECEIVER has been shifted slightly to allow receiving that signal in the center (approximately) of our crystal filter's passband for best signal. Sounds complicated but this is the way it works on all modern transceivers. You offset the receiver, not the transmitter.

Next menu item is where we select USB or LSB. This selection works both in CW and SSB. CW is normally defaulted at power up to LSB to be received with the uBITX set to LSB (this has been the convention for most transceivers since transceivers started replacing the discrete receivers & transmitters in ham (and commercial) stations, so we implemented that here as well. However, sometimes we can eliminate hearing an interfering CW signal that's in our receiver's passband by switching to the opposite sideband and using our filter to remove (or drastically reduce) the interfering signal so the ability has been provided via this menu item to change the sideband for CW operation. In SSB, of course, we have to select which sideband is being used by the other station or we will not be able to properly tune in the signal and it will be unintelligible. The convention (and uBITX default) is Lower Sideband (LSB) below 10 MHz and Upper Sideband (USB) above 10 Hz. Also, the convention is that all

CW is normally LSB so if you are in CW mode the uBITX defaults to all LSB unless you change it via the menu.

Next, since we can't detect the type of key plugged into the key jack in this software, we need a way to tell the keying circuits in the uBITX whether we are using a hand key (or another external key type such as a bug or an external paddle keyer) or if we are using the internal paddle keyer routine in the software. If you are set to "Paddle" the display will come up and indicate "Select KEY?" – a press of the button sets hand /external key mode. If you are in "KEY" it will read "Select PADDLE?".

Moving on, the next item is setting the side tone frequency. The display here is "Change CW Tone". A tap on the button changes the top line of the display to read "XXX" (where XXX is the frequency in Hertz you wish the tone to be) and the bottom line reads "Button to set". Another tap on the button sets the sidetone to whatever value is indicated on the top line and exits to normal operating mode.

The final menu item displays the installed software version and date/time the distributed program was created.

## CREDITS:

The authors of this program and manual are Ron Pfeiffer, W2CTX and Jim Sheldon, W0EB. Photos by Jim Sheldon, W0EB.

We have tried to ensure that the program functions properly in a standard Micro BITX transceiver after the required hardware modification (adding a 4.7K resistor to the A7 line in addition to the 4.7K resistor that already should have been installed on the A6 line and connecting the A7 line to the “Ring” pin of the Key jack) has been completed. The program will not function properly with CW Paddles unless the additional resistor is installed.

The authors are releasing this software into the Open Source community for the Micro BITX transceivers and you are responsible for it's use in your own Micro BITX transceiver. We, the authors are offering absolutely NO Warranty as to the program's suitability and you must act accordingly if you use it. Also, you are responsible for determining the laws of your country governing Amateur Radio operating frequencies and modes and operating radios using this software in strict accordance with those rules. The authors assume absolutely NO responsibility and may not be held responsible for the actions of any/all third parties using this software.

The authors' contact email addresses are provided below for bug reporting and feature requests.

Jim Sheldon, W0EB - email address [w0eb@cox.net](mailto:w0eb@cox.net)

Ron Pfeiffer, W2CTX - email address [w2ctx@yahoo.com](mailto:w2ctx@yahoo.com)