User Manual for KD8CEC uBITX firmware rev 1.072

Table of Contents

Introduction

This manual describes features of the KD8CEC firmware (software) revision 1.072 running on a uBITX transceiver.

The KD8CEC firmware ("CECFW" herein) was written by Dr. Ian Lee, KD8CEC. Ian has graciously provided the CECFW to the uBITX community under the GNU public license.

Ian maintains a website http://www.hamskey.com/ which contains up-to-date information on CECFW.

This manual is written for people who have some understanding of the uBITX transceiver hardware as described at http://hfsignals.com/

As shipped, a uBITX has factory firmware ("Factory FW" herein) installed, which was written by Ashar Farhan, designer of the uBITX hardware.

CECFW is an optional alternative to the Factory FW. Many features of CECFW are similar to the Factory FW, both at the operational level, and at the EEPROM byte storage level.

If you install and test CECFW, and then decide to revert to the Factory FW, it is simple to re-install the Factory FW. See
http://ubitx.net/how-to-load-up-a-sketch-on-your-raduino/
for instructions to re-install the Factory FW.

CECFW is open-source, and you can change and/or modify it to suit your needs, per the GNU public license.

This manual was written by Rod Davis, KM6SN. Please send any errata or feedback to genuse22@yahoo.com. Please indicate what revision level of this document you are reporting about.

Please note this manual is not an “official” manual, and it is incomplete, and may contain errors.

This manual is provided “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Acknowledgments

Thanks to Dr. Ian Lee for his generous contribution to the uBITX community.

Many thanks to Mike, ZL1AGX, for his website http://ubitx.net/, which is a central repository of uBITX information, including information on KD8CEC firmware.

I am grateful for the valuable proofreading help, and constructive suggestions from (in no particular order): Jon KK6LVO, Al N1AW, Vince N2AIE, Ralph Kuehn, Paul K0ZYV, Dennis KG4RUL, Richard WB8YXF. If I have accidentally omitted your name from this list, please pardon me, and send me an email correcting my ways!

New Features

Quite a few new features are available, in addition to the features provided in previous releases. Some of the new features are:

Four different types of LCD display are supported.

A. The original 16 character 2 line display using parallel wiring
B. A 16 character 2 line display using I2C® bus wiring
C. A 20 character 4 line display using parallel wiring
D. a 20 character display using I2C bus wiring

See the section Software Options for further details.

Optional Factory Calibration

Many users do not use The Factory Calibration feature; instead, they use the Setup Menu calibration features. In this CECFW revision, you can elect NOT to compile the Factory Calibration functions. See the section Software Options for more details.

SDR Support

See ttp://www.hamskey.com/2018/04/how-to-connect-sdr-rtl-sdr-to-ubitx-and.html

S Meter support

Using a 20 character LCD, S meter software support is provided:

Instructions for this will be provided in a later revision of this manual.

Support Extended Buttons

If you use an I2C LCD display you can use the newly-freed-up I/O pins for new front-panel pushbutton functions. See section Software Options for more details.

Software Installation

CECFW is available in two forms:

1. a .hex file which can be directly loaded onto the uBITX, or
2. a collection of software source files with .ino as a file extension, which need to be compiled and uploaded onto the uBITX.

Installing a .hex file

Instructions for installing a .hex file in a Windows® PC can be found here:

http://www.hamskey.com/2018/01/how-to-upgrade-ubitx-firmware.html

On a Linux® PC system the procedure is
:

1. install the `avrdude` software package on the PC.

2. add your username to the group dialout via the command

```
    sudo adduser yourusername dialout
```

3. check to see whether ttyUSB0 exists in the directory /dev/ via the command `ls /dev/ttyU*`

if ttyUSB0 does not exist:

a. create a file ubitx.rules, and place the following into it:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="1a86",
ATTRS{idProduct}=="7523", MODE:="0666"
```

(should be all on one line)

b. copy ubitx.rules to /etc/udev/rules.d/ via the command

```
    sudo cp ubitx.rules /etc/udev/rules.d/
```

c. reboot your PC

The process described above should be done one-time only.

Then, to upload a file called myfile.hex onto the uBITX, in a command-line window, use this command:

```
avrdude -p m328p -P /dev/ttyUSB0 -c arduino -b 57600 -U
flash:w:myfile.hex
```

(the above should be all on one line)

In the above example, change myfile.hex to the applicable .hex file name.

Installing from source files

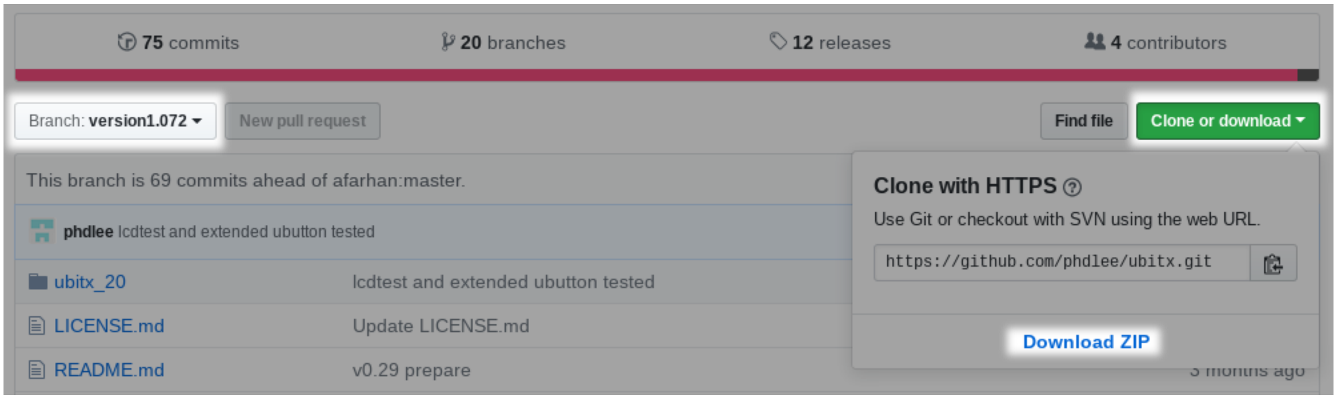
The CECFW source is a collection of text files formatted as Arduino® programming code, each with an .INO file extension. This collection of files is called an Arduino “sketch”, and will be compiled with the Arduino IDE, Arduino Integrated Development Environment.

CECFW installation:
(Although the following description may seem lengthy, it is not complex. Do one step at a time.)

A. Use a USB cable and connect your PC to the uBITX.

B. Download and install the appropriate Arduino IDE for your platform from the main site https://www.arduino.cc/en/Main/Software . Arduino IDE versions 1.8.5 is preferred, but version 1.8.3 may also work.

C. Download the CECFW source code from github https://github.com/phdlee/ubitx
For this example we will be using version 1.072.



This branch is 69 commits ahead of afarhan:master.

phdlee lcdtest and extended ubutton tested

ubitx_20	lcdtest and extended ubutton tested
LICENSE.md	Update LICENSE.md
README.md	v0.29 prepare

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

`https://github.com/phdlee/ubitx.git`

3 months ago

Click the “Branch” button to select version 1.072, then click “Clone or download” and click “Download ZIP”. Unzip the downloaded file and make a note of the folder location, or move the unzipped directory into your arduino sketch folder.

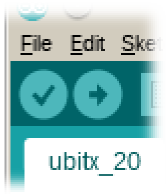
D. Start the Arduino IDE, choose File, Open and navigate to the directory where you unzipped the 1.072 version and continue into the ubitix_20 directory. Select the “`ubitx_20.ino`” file, click Open and thirteen tabs will open, one for each file in the directory.

The downloaded code for 1.072 is set for a default of 20x4 Parallel LCD.

IT IS NECESSARY TO CHANGE THE ubitx.h FILE TO ACCOMODATE A STANDARD UBITX RADUINO WITH A 16X2 PARALLEL LCD.

See the section on “Software Options, Display Types” for making necessary edits in the ubitix.h file according to the type of LCD display being used on your system.

To compile the code and upload, first verify the type of board, processor and serial port in use.
Select Tools from the IDE menu, Board, and choose “`Arduino Nano`”.
Select Tools from the IDE menu, Processor, and choose “`ATMega328P`”.
Select Tools from the IDE menu, Port, and choose the appropriate port for the board. Linux systems use `/dev/ttyUSB0,` or possibly `/dev/ttyUSB1`
Windows systems will show a numbered COM port. Use the Windows device manager to identify the COM port associated with the uBITX.



To compile and verify the code, click the “Verify” button, a check-mark button at the top-left corner of the screen.

To compile and upload the code, click the “Upload” button, an arrow button, at the top-left corner of the screen.

If the board type and processor and serial port are configured correctly, the process should complete without any errors. Monitor the output of the compile and upload process in the lower part of the screen. Status and progress messages will appear here, and you can increase verbosity by checking the verbose messages option under File, Preferences.

When the upload is complete, you can remove the USB cable and start operating the uBITX transceiver.

Software Options

Display Types

Four different types of LCD display are supported.

A. The original 16 character 2 line display using parallel wiring
B. A 16 character 2 line display using I2C® bus wiring
C. A 20 character 4 line display using parallel wiring
D. a 20 character display using I2C bus wiring

Edit the ubitx.h file and select the type of LCD display to be used. The lines of interest are 24..27.

THE ubitx.h FILE , AS DOWNLOADED, LOOKS LIKE THIS:

```
//#define UBITX_DISPLAY_LCD1602P      //LCD mounted on unmodified uBITX
//#define UBITX_DISPLAY_LCD1602I      //I2C type 16 x 02 LCD
#define UBITX_DISPLAY_LCD2004P      //24 x 04 LCD (Parallel)
//#define UBITX_DISPLAY_LCD2004I        //I2C type 24 x 04 LCD
```

because it is for a 20x4 display.

TO USE THIS CODE ON AN ORIGINAL UBITX WITH A 16X2 DISPLAY, CHANGE THE FOUR LINES TO LOOK LOOK LIKE THIS:

```
#define UBITX_DISPLAY_LCD1602P      //LCD mounted on unmodified uBITX
//#define UBITX_DISPLAY_LCD1602I      //I2C type 16 x 02 LCD
//#define UBITX_DISPLAY_LCD2004P      //24 x 04 LCD (Parallel)
//#define UBITX_DISPLAY_LCD2004I        //I2C type 24 x 04 LCD
```

Be aware the hash-character ‘#’ must be in the first column of the line.

PLEASE NOTE THE LCD SELECTION IN SOFTWARE MUST AGREE WITH THE HARDWARE DISPLAY TYPE AND WIRING YOU ARE USING.

A parallel LCD display will use up to 16 data pins and the I2C display will use 3 data pins.

Using an I2C display frees up 5 Raduino pins that can be used for different functions.

S meter

This section is to be completed.

Factory Calibration

Many users do not use The Factory Calibration feature; instead, they use the Setup Menu calibration features. In this CECFW revision, you can elect NOT to compile the Factory Calibration functions,, by modifying the file ubitx.h, line 34 which reads:
#define ENABLE_FACTORYALIGN

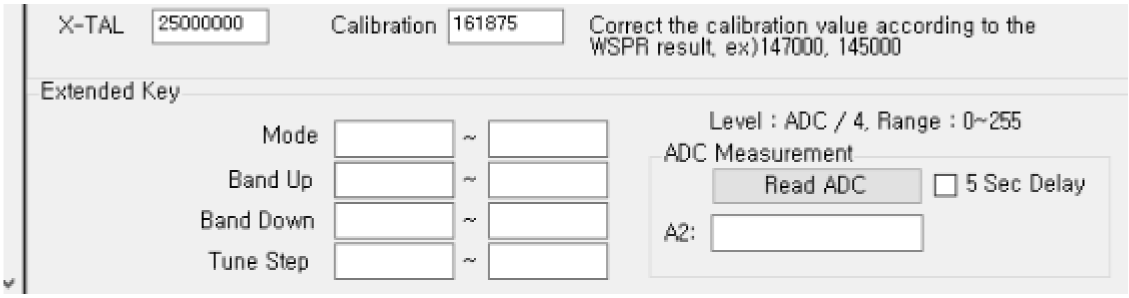
Place the two characters ‘//’ in front of the # character to disable Factory Calibration features and free up some program space. Save the file, compile and upload.

Support Extended Buttons

If you use an I2C LCD display you can use the newly-freed-up I/O pins for new front-panel pushbutton functions. You can add frequently used Mode Change, Band Up, Band Down and Select Tune Step buttons.

Function keys can be mounted using resistors and switches.

The following image is from the uBITX Memory Manager software which runs on the PC:


X-TAL 25000000

Calibration 161875

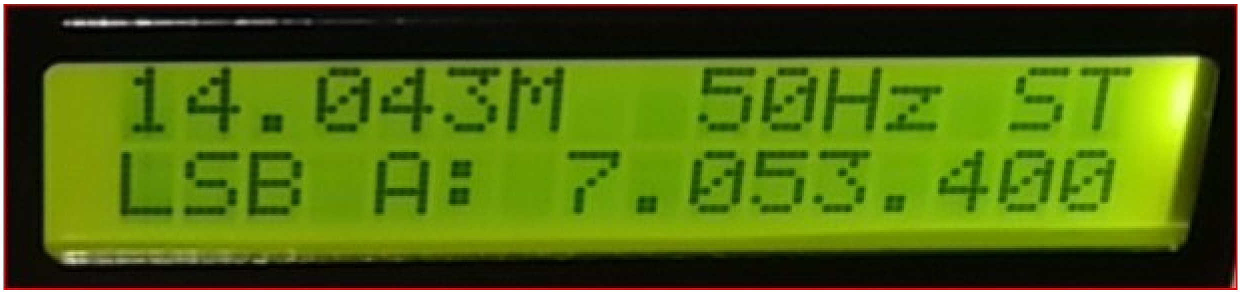
Correct the calibration value according to the WSPR result, ex)147000, 145000

Level : ADC / 4, Range : 0~255

This also registers key values in uBITX Manager for various product support. Details on how to do this will be posted separately after releasing the firmware.

Initial display

Using the CECFW, when powered-on, the uBITX will initially display something like this:



```
14.043M  50Hz ST
LSB A: 7.053.400
```

THE TOP LINE MEANS:

1. CECFW provides two VFOs, VFO A and VFO B. On the top line, 14.043M shows the frequency VFO B is currently tuned to. This is just a reminder. VFO A is the active VFO, shown by the “A:” on the next line.

2. 50Hz - shows the current tune rate; for each click of the encoder the VFO frequency will change by 50 Hz.

3. ST - means the CW mode is currently Straight Key. That means only the key jack tip and sleeve are monitored for key closures.

THE BOTTOM LINE MEANS:

1. LSB – Lower Sideband is the receive and transmit operating mode.

2 A: - VFO A is in use

3. the uBITX is currently tuned to 7,053,400 Hz.

CECFW Menu system

The uBITX has a rotary encoder with a push-switch. In this document, that push-switch is called the “Function SW”.

Be aware there are three different time-durations of Function SW push that are important:

Short duration: a quick click; approximately one or two tenths of a second

Medium duration: approximately one second

Long duration : approximately two seconds

There are four different operational menus that can be selected by pushing the Function SW.

1. Short duration push: invokes the MAIN MENU

2. Medium duration push: invokes the TUNING RATE MENU

3. Long duration push: locks/unlocks the VFO.

4. Push and hold the Function SW during uBITX power-up: this invokes Factory Calibration mode. This menu is practically obsolete, and you may remove it from the firmware; see the section Software Options. That process frees up program memory space on the Nano.

MAIN MENU

When you have done a short push on the Function SW, MAIN MENU options appear. You can scroll through the options by turning the encoder knob.
The options, are:

Band Select

Select VFO B? (or, if you are currently using VFO B, this will show VFO A)

Select Mode

RIT: Off, On

IF Shift: Off, On

ATT On

CW: 20WPM Change

Split On

VFO to Channel

Channel to VFO

Memory Keyer

WSPR Beacon

SDR Mode On

Setup On (Setup off)	(reveals/hides previously hidden menu items pertaining to setup)

Exit Menu

When you find the menu item you want, push the Function SW. Each menu item will be described in the following section.

Band Select.

This menu displays “Press to confirm”. Turn the encoder until the band you want shows up on the bottom line. Then press the Function SW. Note that you can get to a quick exit from this menu by rotating the encoder counter-clockwise (anti-clockwise).

When Band Select is displayed, by applying a long-press to the Function SW you can toggle between Ham Band and General Coverage.

Select VFO x.

Selecting this menu option changes which VFO is selected. If VFO A was active, the display will change to VFO B. Conversely, if VFO B was active, the display will change to VFO A.

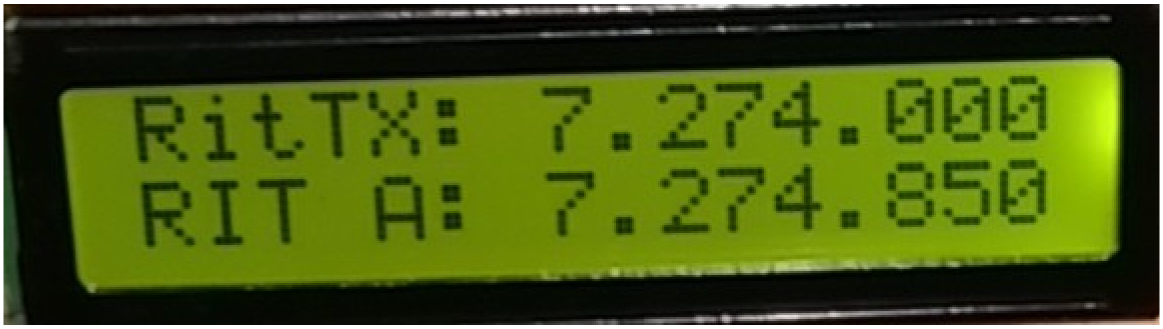
Select Mode.

Use this menu option to select the transceiver mode of operation. Turn the encoder until the “>” symbol points to the mode you want, then push the Function SW again. The available modes are LSB, USB, CWL, and CWU.

Rit Off On.

Select this menu option to start RIT (Receive Incremental Tuning). Turn the encoder to change the receive frequency. When the frequency is what you want, push the Function SW. **THEN TURN THE ENCODER UNTIL Exit Menu? Is shown,** and push the Function SW again.

In this example:



```
RitTX: 7.274.000
RIT A: 7.274.850
```

VFO A is in use, and the uBITX will transmit on 7.274.000, and it will receive on 7.274.850

IF Shift: On , IF Shift Change

This menu entry adjusts the BFO so that voices sound more natural. Tune the uBITX to listen to sideband voice prior to selecting this menu entry. Then

1. Select the IF Shift:ON

2. Turn the encoder knob until the voice sounds the most natural, then

3. Carefully push the Function SW to save the setting.

Note when the IF shift value is non-zero, this menu entry will read
IF Shift Change? A long-press on the Function SW while **IF Shift Change?** is showing will turn IF Shift Off.

Similarly, a long-press on the Function SW while changing the IF Shift value will turn IF Shift Off.

ATT On

This section to be provided.

CW 20WPM Change

Turn the encoder until the desired CW speed is displayed, then press the Function SW.

Split On

Press the Function SW to turn Split operation on. To turn Split operation off, go to the Main Menu, and select menu Split Off? And press the Function SW.

Split operation means using one VFO for receive, and the other VFO for transmit.

VFO to Channel

The CECFW provides the ability to save the current operating environment (VFO selection, mode, etc) in “Channels”. You can think of a Channel as a separate and special memory place that persists and endures across power off/on cycles. You can save all of your favorite operating frequencies and modes in these Channels, and recall them later at your convenience.

There are 20 Channels. Turn the encoder to select the Channel you want to use. Channels which have something in them will show the frequency which was saved. Channels which have never been used show as blanks. When you are on the channel you want to use, press the Function SW to store the current environment into the Channel.

Channel to VFO

Use this function to recall the operating environment stored in a Channel. Turn the encoder to show the Channel you want, and then press the Function SW.

Memory Keyer

This section is to be completed.

WSPR Beacon

This section is to be completed.

SDR Mode On

See http://www.hamskey.com/2018/04/how-to-connect-sdr-rtl-sdr-to-ubitx-and.html

Setup On

The CECFW has a hidden menu extension to the MAIN MENU functions shown above. Usually they are not accessible. To access these hidden functions, when in the Setup ON menu, press the Function SW.

*Then, the next time you press the Function SW to invoke the Main Menu, turn the encoder past the **Setup Off** function, and continue to turn the encoder past the **Exit Menu** function to reveal additional Setup Menu items.*

TUNING RATE MENU

Note that this is not a part of the Main Menu; to invoke the Tuning Rate Menu hold the Function SW in for medium duration.

This menu entry selects the number of Hz that main tuning will change for each pulse from the encoder.

Turn the encoder until you see the tuning rate you desire, then push the Function SW.

HARDWARE ARCHITECTURE

A word about uBITX hardware architecture is needed here.

The uBITX is a double-conversion transceiver, with a First IF of 45 MHz, and a Second IF of 12 MHz.

UBITX architecture is shown here:

http://www.hfsignals.com/index.php/ubitx-circuit-description/

The CECFW configures the uBITX as a 12 MHz USB rig, with two front-end mixers to cover the HF ham bands.

For receive operation, starting at the antenna, the **First Mixer** converts the ham band of interest to 45 MHz. The signal is passed through a 45 MHz filter which is about 15 kHz wide, and then on to the **Second Mixer**. The Second Mixer converts the 45 MHz signal to 12 MHz. On transmit, the signal directions are reversed.

The **Third LO** is what is commonly known as the **BFO.** The **Third LO** feeds the **Product Detector**. The **Third LO** frequency is always lower than 12 MHz. ,which is the approximate center frequency of the crystal filter. Therefore, the **Product Detector is always operating in the USB mode**.

In the following examples assume the uBITX is tuned to 7.150 MHz. Frequencies in these examples are approximate.

USB Example

7.150 USB

The First Local Oscillator (**First LO**) feeds the **First Mixer**. The **First LO** always runs at a higher frequency than the ham band being used. In this example the **First LO** runs at approximately 52.140 MHz. Because the First LO is higher than the incoming signal, the First Mixer causes sideband inversion. Therefore, the signal coming out of the First Mixer is now LSB at 45 MHz.

The **Second LO** feeds the **Second Mixer**. Sideband selection is done by a judicious choice of **Second LO** frequencies. In this example, the **Second LO** runs at 56.990 MHz, which is higher than 45 MHz, and the sidebands are inverted by the **Second Mixer.** Therefore, the output of the **Second Mixer** is USB once again, and at 12 MHz.

LSB Example

7.150 LSB

The First Local Oscillator (**First LO**) feeds the **First Mixer**. The **First LO** always runs at a higher frequency than the ham band being used. In this example the **First LO** runs at approximately 52.140 MHz. Because the First LO is higher than the incoming signal, the **First Mixer** causes sideband inversion. Therefore, the signal coming out of the **First Mixer** is now USB at 45 MHz.

The **Second LO** feeds the **Second Mixer**. Sideband selection is done by a judicious choice of **Second LO** frequencies. In this example, the **Second LO** is runs at 32.995 MHz, which is lower than 45 MHz, and the sidebands are NOT inverted by the **Second Mixer.** Therefore, the output of the **Second Mixer** is still USB, but at 12 MHz.

SETUP MENU

The following functions alter uBITX fundamental setups. The reader is strongly encouraged to use the KD8CEC Memory Manager software on the PC to save the current uBITX settings for a backup prior to modifying any critical items. See section **uBITX Memory Manager** for details.

Set Calibration

The uBITX uses an SI5351 synthesizer chip to generate 3 frequency sources. The SI5351 has a main reference crystal oscillator on-board that can be accurately calibrated using this menu.

The procedure is:

1. put the ubitx in USB mode

2. connect an antenna to the uBITX to listen 10 MHz WWV; alternatively connect a calibrated signal source at 10 MHz to the antenna input

3. navigate the menus to get to the Set Calibration entry, and press the Function Sw. The uBITX will tune to 10,000,000 Hz, but the display will not show that.

4. rotate the encoder until you hear the zero beat

5. Carefully push the Function SW to save the calibration

As a result of this process, uBITX frequencies will be within a few Hz of the dial setting.

For those who have access to a high resolution frequency counter, as an alternative to the above, you can:

connect the uBITX to the counter via a pad of at least 20 db, and hold the CW key down while adjusting the uBITX encoder until you see on the frequency is exact. Then carefully push the Function SW to save the calibration.

Set the BFO

Set the BFO frequency used in USB and LSB. Rotate the encoder to vary the BFO frequency, then press the mic PTT switch to confirm. If you do not want to change the BFO frequency, press the function switch to abort.

Set CW RX BFO

Set the BFO frequency used in USB and LSB. Rotate the encoder to vary the BFO frequency, then press the mic PTT switch to confirm. If you do not want to change the BFO frequency, press the function switch to abort.

Change CW Tone

Set the CW sidetone frequency. Rotate the encoder to vary the CW sidetone frequency, then press the mic PTT switch to confirm. If you do not want to change the BFO frequency, press the function switch to abort.

CW TX-RX Delay

To be written.

CW Start Delay

To be written.

Change Key Type

This menu entry controls how the key contact closures are recognized by the firmware.

Rotate the encoder to chose one of Straight key, or keyer IambicA mode, or keyer IambicB mode, then push the Function SW.

ADC Line Monitor

This function will become obsolete.

TX OFF

To be written.

Exit Menu?

uBITX CAT control

The CECFW emulates a subset of the Yaesu FT-817 CAT protocol.

To use PC software with uBITX CAT:

1. use a USB cable to connect the PC to the uBITX

2. in PC software, select Yaesu FT-817

3. select 38400 baud, 8 bits, no parity, 2 stop bits.

uBITX Memory Manger

The uBITX Memory Manger software (“MMSW” herein) runs on a PC that is connected to a uBITX.

There is a group of data on the uBITX that is stored in a re-writeable memory called EEPROM. The MMSW provides the functions to:

1. Connect to the uBitx, and

2. read the uBITX EEPROM data from the uBITX, and

3. decode the EEPROM data into human-recognizable entries, and

4. change the EEPROM data, and

5. encode the human-reconizable entries into hexadecimal, and

6. write the EEPROM data to the uBITX, and

7. write the EEPROM data to a file on the PC (with a .btx file extension), thereby providing a backup system for the uBITX settings, and

8. load a .btx file into the MMSW, allowing one to encode the data and then write it to the uBITX to restore uBITX settings.

A separate manual will explain the features of the Memory Manager in more detail.

Installing Memory Manager

on Microsoft® Windows® PCs, go to

https://github.com/phdlee/ubitx/releases/download/v1.06/ubitx_Manager1.01.zip

on Linux® PCs, go to

http://www.hamskey.com/2018/02/running-ubitx-manager-on-linux.html

On Linux systems, you must install Mono, an the open-source software that provides a .NET framework.

Then, invoke the Memory Manager using the command-line like this:

```
mono uBITX_Manager.exe
```

Linux users- note that you must check the Linux Compatibility Mode box in the upper left corner. Also be aware in the serial port selection window you may have to use the down-arrow to scroll far enough down to find /dev/ttyUSB0

Linux users- also note it may be necessary to execute the procedure described in the section **Installing a .hex file** before running the Memory Manager.