

[Company name]

# KD8CEC V2.0 User Manual

A Guide to using KD8CEC V2.0 Software

Mark Hatch (AJ6CU)  
3-12b-2023

## Revision History

March 12b, 2023:	Discovered additional feature where you can string messages
March 12, 2023:	Initial Release of Memory Keyer Instructions

## Preface

In porting the KD8CEC to new hardware, it became very clear that there was limited information on how to actually use the software. Rod Davis (KM6SN) wrote a very good manual to start with that is available at:

[https://ubitx.net/wp-content/uploads/2018/04/ug1072\\_087.pdf](https://ubitx.net/wp-content/uploads/2018/04/ug1072_087.pdf)

However, there were several sections that were listed as “Section to be completed” and the manual mainly focused on the use of the software thru the “LCD overlay Menu System” (short press of the VFO knob) and did not address the use via the Nextion interface. This decision is understandable as the Nextion interface was never really standardized and there were multiple variations of the user interface created by various Hams.

The goal of this manual is to provide a more complete manual that also embraces the Nextion. This will probably be a work-in-progress for a long time. Initially, the lack of documentation for the Memory Keyer is the driving force for the first efforts. Other areas will be added as developed.

This manual should be used as a companion to the uBITX Settings Editor manual. The current draft is available at:

<https://github.com/AJ6CU/uBITX-EEPROM-Manager/releases/download/V2-beta-1/uBITX.Settings.Editor.User.Manual.pdf>

If you have suggestions of other areas of KD8CEC that could use additional instructions, please let me know. I am “good on QRZ” and frequently monitor the BITX20 group on groups.io. I look forward to your feedback!

73

Mark (AJ6CU), March 12, 2023

## Introduction

The uBITX is manufactured by HF Signals (<https://www.hfsignals.COM/>) using an Arduino Nano V3 MCU. This Nano is the heart of the uBITX. It has 32kbytes of flash memory and 1024 bytes of Electrically Erasable Programmable Read-Only Memory (EEPROM). The flash memory holds the firmware or software that runs the uBITX where as the EEPROM is used to store settings (e.g. calibration information, last used frequencies and modes, tuning rates, etc.).

Although functional, the original firmware provided by HF Signals was pretty basic and the plan all along was for other Hams to take the lead to extend it. As a result, Dr. Ian Lee (KD8CEC) developed an enhanced version of the firmware (we will refer to it as KD8CEC throughout this manual) that extended the original LCD offering and in later releases, provided a graphical user interface based on Nextion touchscreens. His efforts are documented in blog format at [www.hamskey.COM](http://www.hamskey.COM).

The “eye candy” and enhanced functionality of the KD8CEC software is what first catches the attention of most Hams. However, what many miss is that the KD8CEC software is of limited value without a tool to easily tailor it to the users needs.

Dr. Lee wrote the uBITX Memory Manager (available at: <https://github.COM/phdlee/ubitx> ) as a companion tool for customizing the KD8CEC firmware. This companion tool has been rewritten and a beta of the uBITX Settings Editor is available at:

<https://github.com/AJ6CU/uBITX-EEPROM-Manager/releases/tag/V2-beta-1>

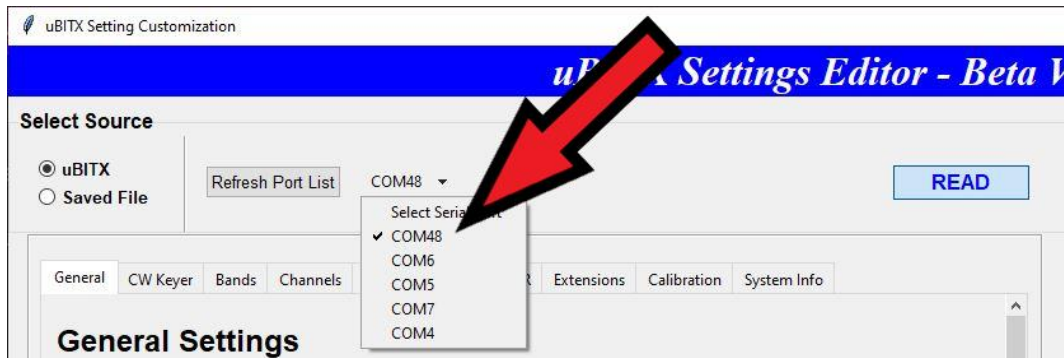
## The Plan for our Journey

This document is the user manual for the port of the actual KD8CEC software to new processors by AJ6CU. It is likely to be a work-in-progress for a long time as new sections are added when they become available.

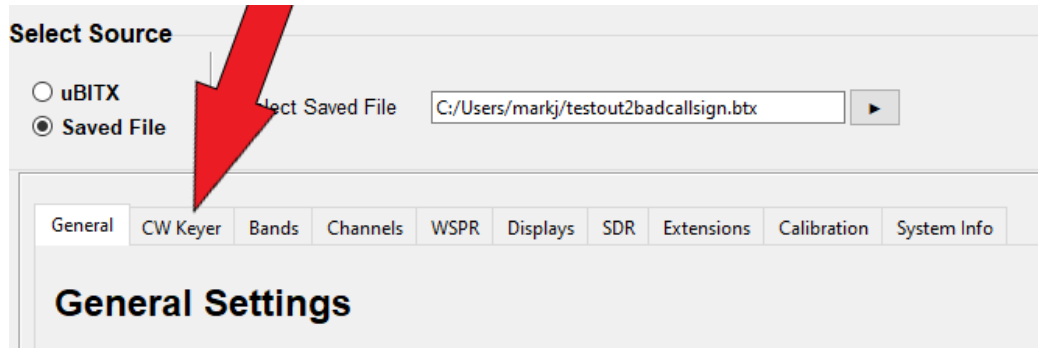
## Setting up the CW Keyer in the Settings Editor

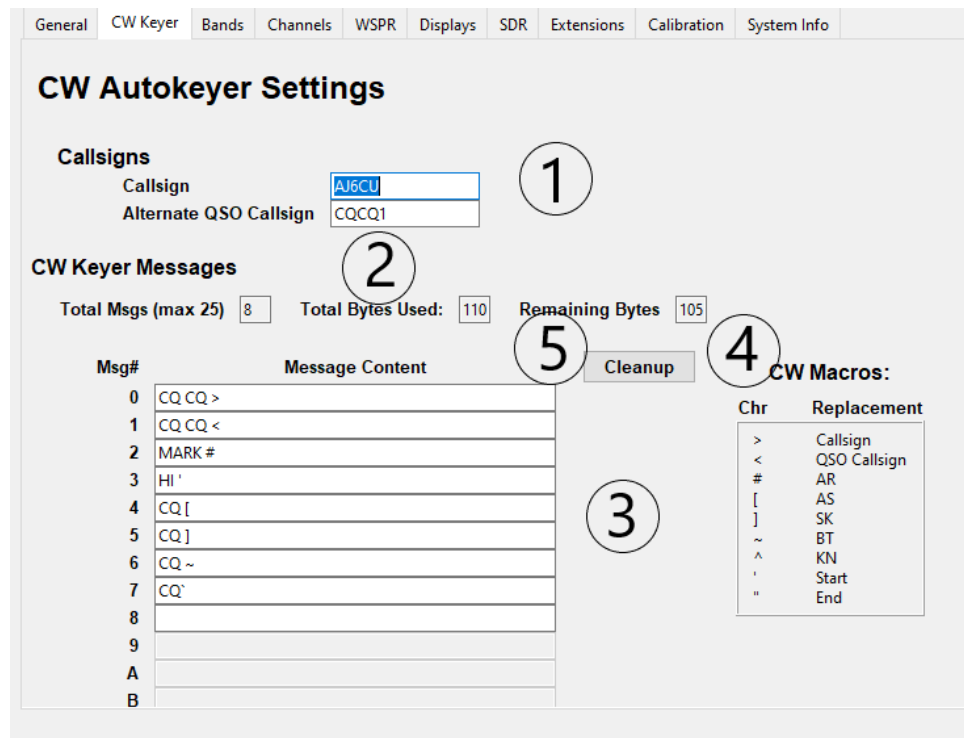
The KD8CEC software includes a CW Memory Keyer that many will find useful for basic contesting. It does not have the extensive set of macro instructions that you might find built into your commercial radio or into the WinKeyer ([www.hamcrafters.com](http://www.hamcrafters.com)) but the limited macros are simple to use and will work for most.

To set up the CQ Memory Keyer, start the uBITX Settings Manager ( <https://github.com/AJ6CU/uBITX-EEPROM-Manager/releases/tag/V2-beta-1> ). Select com port on which your uBITX is attached and click the READ button.



Select the tab that says “CW Keyer”





The KD8CEC software provides a fairly functional keyer with up to 25 messages! But from a practical viewpoint, 25 is probably far too many to handle given the limits of the user interface handle and the total memory available for keyer messages is on the order of 210 characters. So realistically, you will run out of memory before you run out of available messages.

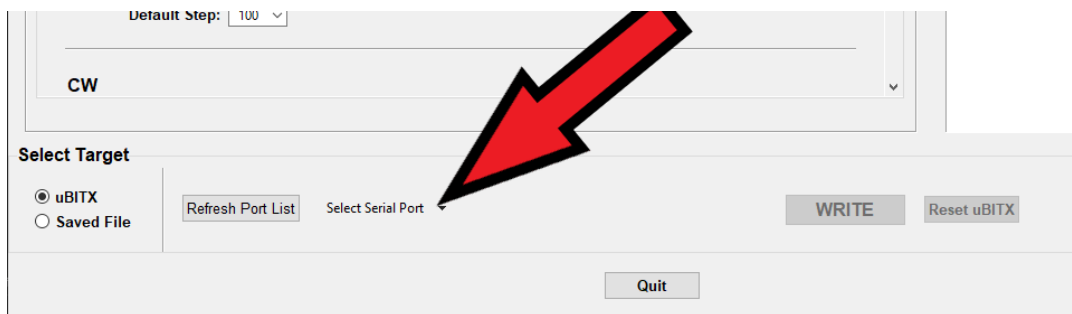
### By the Numbers

1. The callsign you entered in the General Settings is repeated up here at the top. This tab also provides the option of an alternative callsign that you can use. Perhaps your contesting from a remote site and want to signify that with something like AJ6CU/5. Like your normal callsign it is limited to 18 characters.
2. Since EEPROM memory is limited, the line identified by "2" tracks 3 key aspects: 1. How many messages are active, 2. Total bytes used, and perhaps most importantly 3. Remaining bytes
3. These are the CW messages that have been saved. Every time you enter the last one (for example we just entered the "CQ" in message #7, the next line (#8) becomes open. This was done this way to help you save memory since every active message has a 2 byte overhead.
4. This is the list of macros you can use in your CW message. Even though a ">" expands to your full callsign, you only use one byte (for the character ">") in your message. Although these macros are convenient, they are especially useful for saving Keyer memory.

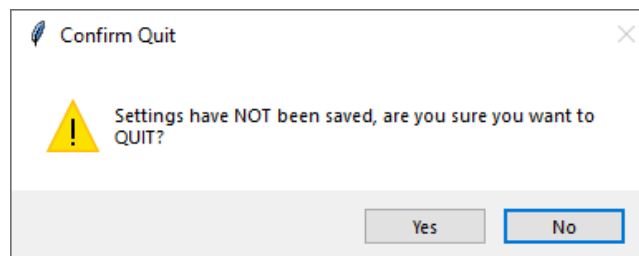
5. This button (CLEANUP) will eliminate empty messages and move them all up to the front. For example, suppose I decided to delete message 5 “CQ J”. This would leave an empty slot in message #5. Hitting the CLEANUP button, will move 6 into slot 5, 7 into slot 6, delete the blank line at 8, and recovery the extra overhead bytes. You get a neater looking set of messages (especially important if you are keying via the rotary encoder), and you free up some bytes.

So eventually, you have tweaked every setting, and you want to save your work. Saving is just a reverse of the reading process. The box at the bottom of the software controls the writing.

In the screenshot, the red arrow points to the selection drop down for a list of COM ports. Select the one (generally the same as the input) and click WRITE.

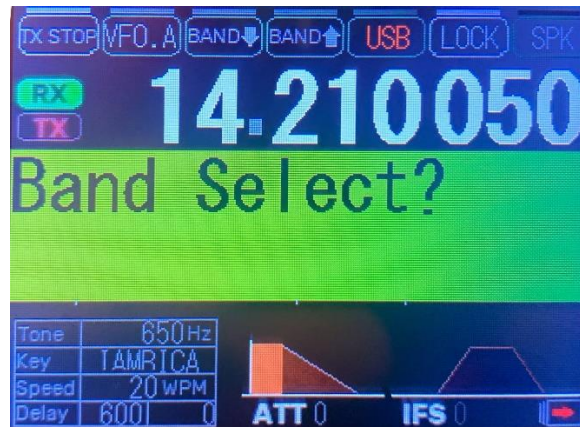


**Remember to CLICK WRITE TO ACTUALLY SAVE THE SETTING TO THE FILE.** If you forget to click WRITE, you will be greeted with the following message when you try to exit:



## Operating the CW Keyer From Your uBITX

Currently, the CW Keyer is not accessible thru the Nextion user interface. Instead, you effectively have to go into an “LCD Overlay Mode” with a **short press of the VFO dial**. The following screen should appear:



If you were to do another **short press of the VFO dial**, you could change the bands. But instead, rotate the VFO dial clockwise until you see the following screen:





Now perform a short press on the VFO dial to get to the Memory Keyer submenu.



A couple things to notice here. First is the “0” around the middle and left side of the screen. This is the number of the message you created a little while ago in the Settings Editor. You can easily switch messages by rotating the VFO knob until the desired message starts scrolling in from the right.

Second, the instructions to “PTT to Send” are misleading. A short press on PTT (I use the microphone button as my PTT ), you will move to the next option which is to adjust the TX frequency. (More later, but a LONG press will send the current message.)

Note that from this point, **we do not Push the VFO dial**. We will turn it to select an option. If you push the VFO dial, you will exit the menu system completely!



At this point, we can rotate the VFO dial to select a frequency to send. As can be seen in this screen shot, I am way off of the traditional CW area on 20m. So I adjusted the VFO to a more traditional frequency and ended with the screen below.



So, how do we send. Try a quick push on the PTT. You will end up back in the message selection menu like the following screen shot. Another quick push will take you back to the Freq adjust menu option.



So how do we finally send the message? A LONG PUSH of PTT **starts the sending of the selected message** at the selected VFO. (BTW, after you are done, the VFO remains at the frequency you selected for the Memory keyer.)

There are a couple other options you have **while your message** is being sent:

1. To HOLD sending – PTT LONG PUSH
2. To Repeat the current message – PTT SHORT PUSH
3. To string messages (up to 3):
  - a. PTT LONG PUSH
  - b. Select next message with VFO dial
  - c. PTT SHORT PUSH to confirm
4. VFO Dial SHORT PRESS – stops sending immediately and leaves LCD Overlay Mode.

The sequence of LONG PUSH followed by selecting the next message, and SHORT PUSH does not seem reliably consistent. You might have to play around here to get this to work as you want.

## Last Words

As mentioned, this manual will continue to be a work in progress with more sections added as time passes and I feel that an underutilized feature of KD8CEC needs to be made more visible.

73

Mark