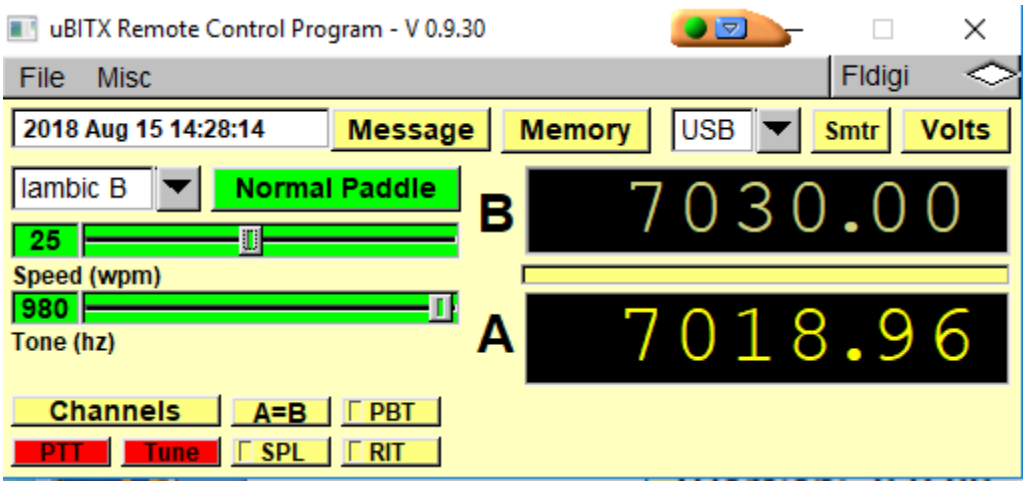
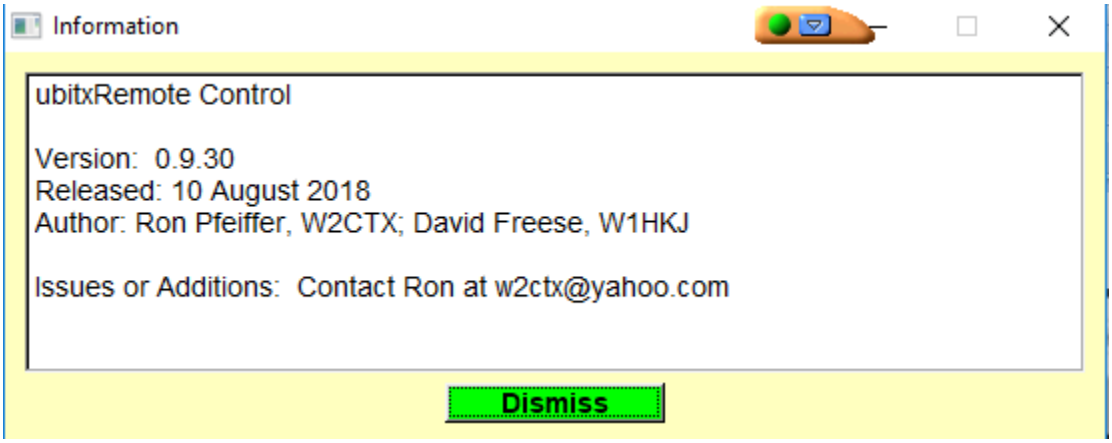
Remote Control Program for the Micro BITX Transceiver

Requires W0EB/W2CTX uBITX control software V3.00R or later

Manual written by Jim Sheldon, W0EB





Introduction:

Almost all modern Amateur Radio transceivers have the ability to be remotely controlled through a PC's serial or USB ports and most front panel functions of the transceiver can be operated through a specially designed program on some sort of an external computer (or in some cases a dedicated external control panel).

When the team of W0EB, W2CTX and N5IB (which we playfully called the "Triumvirate Skonk Worx") began implementing our own innovations for operation and control of the recently released Micro BITX (abbreviated uBITX) transceiver by Ashhar Farhan, VU2ESE, and released for sale by his company HF Signals out of Hyderabad, India, we did so for our own pleasure and benefit. These innovations and the resulting software went through a few growing pains like all projects of this nature are wont to do but they turned out to function extremely well, especially in the later versions so we decided to release them to the general uBITX community. (We are up to release Version 3.00R and are feverishly working on V4.00R at the time this manual was being written.)

Our software was designed to implement smoother tuning and a fully functional CW Keyer that didn't depend on the limited ability of the original Arduino NANO micro controller to read the difference in voltage across a resistive voltage divider to determine dot/dash paddle closures or a hand key closure. Jim, N5IB and I figured out a way to pick off and use the I2C serial bus on the original "Raduino" (Radio/Arduino) card that controls the uBITX and provides the RF oscillator signals along with other controls. The reason for this was the ability to use a serial I2C display controller and display instead of the 16 pin parallel display that was attached to the

Raduino and took up a full six digital I/O lines on the Arduino NANO controller. Having those extra digital lines to work with, Ron, W2CTX rewrote the CW Keyer routines to use Interrupts for dot/dash transition memories and implemented selectable Iambic keyer modes A and B (the most common) that could function without worrying about things happening external to that routine and actually allowed limited multitasking within the entire control program.

This gave rise to fairly easy implementation of PC Control (CAT) functions via the NANO's USB programming/communication port that could function simultaneously with the CW Keyer and not cause the keying to become erratic.

Next, we decided that since Push To Talk (PTT) wasn't needed in CW operation and the Hand Key wasn't normally needed in SSB operation that the same key line could be used for both PTT and Hand Key. This made a separate jack for the Hand Key something that most would want to add. The other option would be to wire the hand key's plug to use the PTT line of the microphone jack and plug the key into that jack. There was an added benefit that came out of doing things this way. You now can use the PTT button on your microphone as an emergency hand key should that become necessary.

The evolution of the programs and desire for a bit more hardware control availability lead to the design and offering for sale of our "RadI2Cino" (pronounced rad ee too CEE no) plug-in replacement for the original Raduino card. This was necessary to allow using the I2C bus to operate with display controllers that needed to have pull ups to +5 volts on the I2C bus' SDA and SCL lines. Since the Si5351a Clock chip (all 3 oscillators

that drive the uBITX) uses 3.3 volts on the SDA and SCL I2C communication lines and cannot tolerate voltages higher than about 3.5 volts on those I/O pins, some means of protecting it from the display controllers using 5 volts was needed. We put a 3.3 to 5 volt bi-directional level shifter on the I2C bus AFTER it went to the Si5351a so that no 5 volt transitions on the SDA and SCL lines by display controllers could get back to the clock chip and damage it thus allowing just about any I2C enabled display to work with the RadI2Cino card.

Now that we had that working, Ron, W2CTX wrote this Remote Control Program (RCP). Since it only ran under Linux Ron teamed up with Dave, W1HKJ and he converted it to run under windows.

There are two CAT protocols contained within the uBITX software. A limited version of the Kenwood protocols to allow logging programs such as the N3FJP, N1MM (not tested yet) and others that can read frequency and mode information from a radio via the serial/USB port, and our own unique uBITX commands used by this program to provide full control of all the functions in the radio that can be controlled via the serial port on the NANO (or any other processor system we subsequently may implement). We released a Programmers Reference Manual describing these CAT commands.
As originally distributed, and to keep from having some email programs and providers reject it as an attachment, the file was first named “ubitx.fred” (this was just an arbitrary extension).

To surmount this problem, the program is now hosted on the www.w0eb.com website in the files section as a zip file and is now called appropriately ubitxRCP.exe.

This program needs no installer and you can put it in any directory on your computer you desire. We recommend putting it in a directory by itself (name that directory whatever you want) rather than just leaving it on the desk top as it does write several files for frequency and CW message memories that are usually hidden by the Windows system's protocol as they are initialization files for this program and normally don't show up in the directories unless the "Show Hidden Files" check box in the directory's "Properties" menu is checked.

To do this, you have to create a directory and move this file into it as there is no installer program. ubitxRCP.exe does not install into the Windows Registry so if you need to uninstall it, for some reason, simply delete the entire directory you placed it in. (That's why we recommend putting it in its own directory – you will uninstall the program and all files it creates by simply deleting that directory.)

Once you have "ubitxRCP.exe" located in its own directory on your computer, you need to create a shortcut on your desktop to run it.

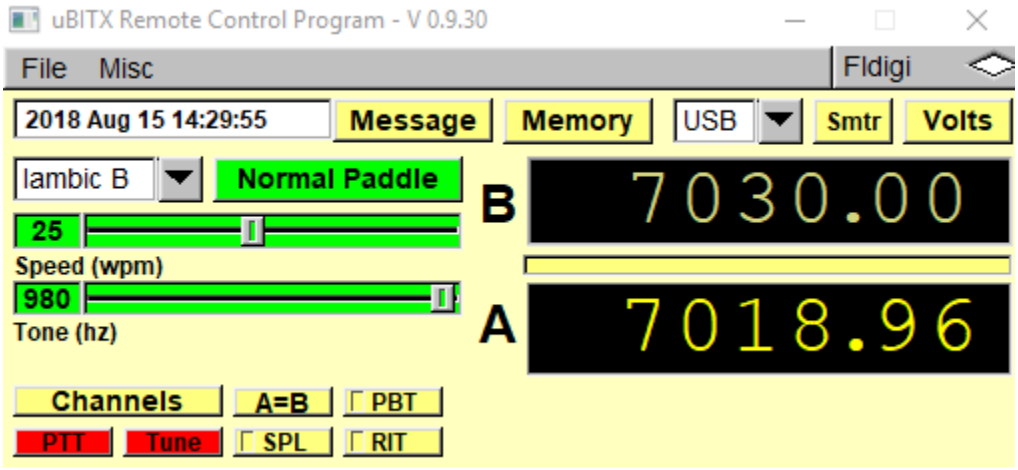
Now that you have the remote control program in its own directory, hook up the USB cable from your computer to the NANO (or other processor if utilized) via the MINI (or MICRO) B connector on the processor. Determine which COM port is being used (you should write down the one that shows up in the Arduino IDE when you program your NANO). The program defaults to 38400 baud and right now this is not changeable so make sure the software in your uBITX is set to 38400 baud as well or, of course the program won't be able to communicate with the radio. (Instructions to set this parameter are contained in the instructions for whichever

W0EB/W2CTX software version you are using so it's not covered here as it may vary slightly within versions.)

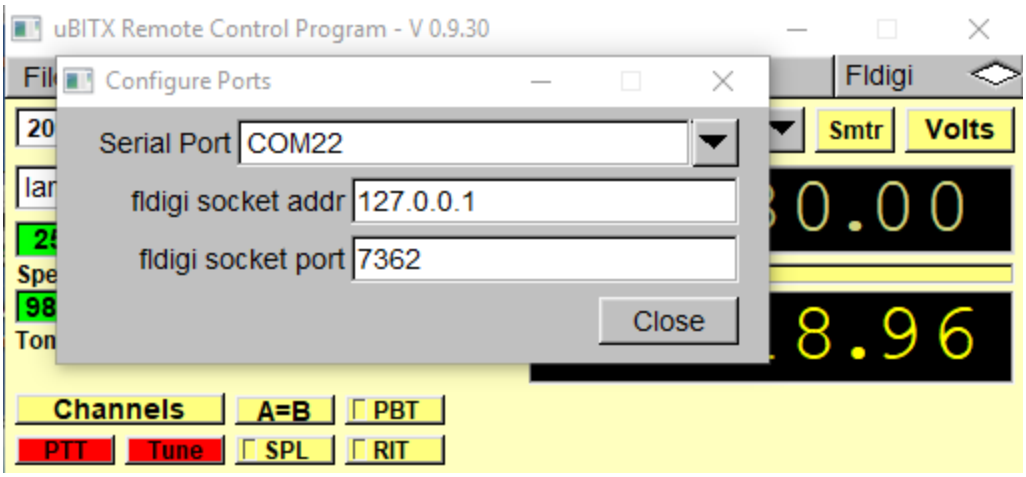
Turn the radio on with the USB cable connected to both the uBITX and your computer's USB port before you start this program.

The first time you run it the communications port will not be selected and after about 10 seconds the MISC pull down tab in the program where you will select the COM port under "Configure I/O" is displayed. Select the appropriate COM port. The exit and restart RCP. Connection should occur within 5-6 seconds.

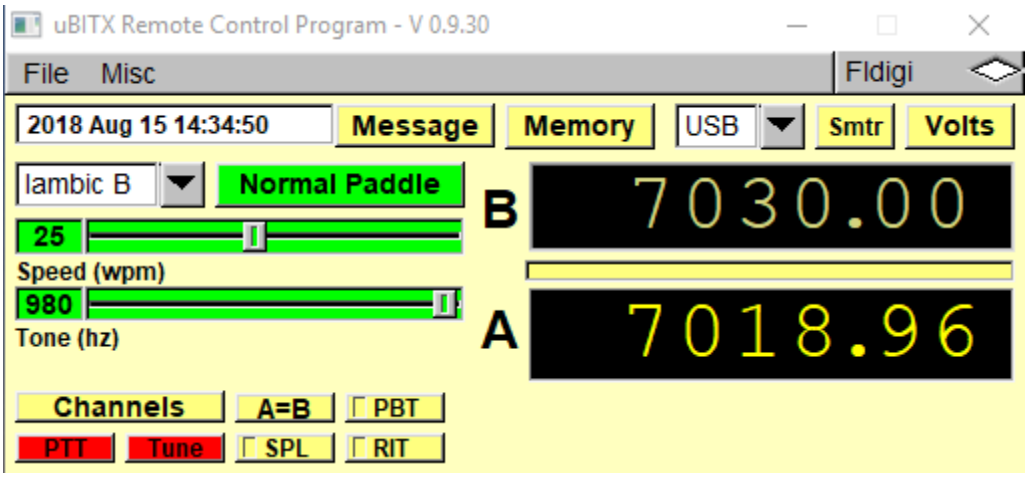
To manually change the COM port, do the following:



Click on Misc and then click on Configure I/O which will bring up the dialog box where you will see COM ports listed. The COM Port that your uBITX is connected to should appear in the list. Click on it and click the CLOSE button that will appear at the bottom of the list once the port is selected.



Once the connection is made between the program and your uBITX (may take several tries the first time you connect to get everything set right) you should see the following window on your screen.

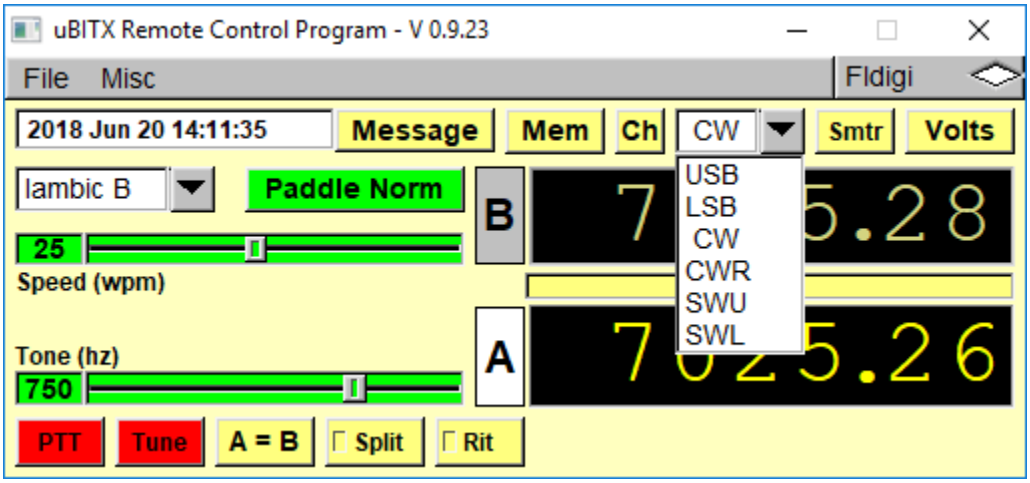


This is your main control window and from here all available functions can be operated. Everything is pretty self-explanatory and you should play with the features to get used to them – CAUTION! The red PTT and TUNE buttons WILL place the uBITX in transmit so while becoming familiar with the program, the radio should be connected to a resonant antenna or preferably a dummy load to prevent damage if either of these buttons is clicked!

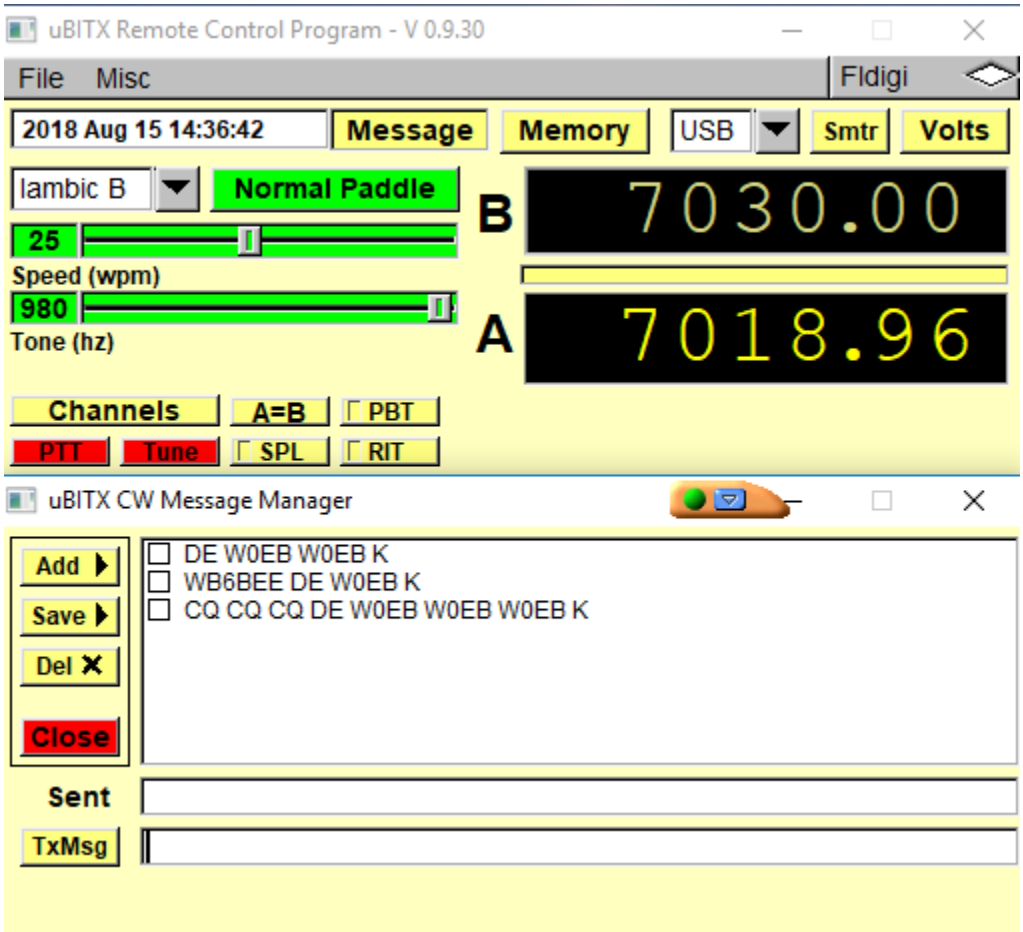
Both the A and B VFO frequencies are displayed. The active VFO is indicated by the frequency being displayed in Yellow and the non-selected on is displayed in Grey. Select either VFO by clicking on the A or B letter to the left of the frequency window.

You can tune the radio by placing your mouse cursor near the top or bottom of any digit and clicking the left button will increase that digit by one if near the top of the digit or decreasing it by one if near the bottom of the digit. You can also tune by placing the mouse anywhere in the digit and use the mouse wheel to scroll up or down. (The encoder and frequency increment on the radio are still active though the menu system is disabled during remote control operation to prevent accidental conflicting commands.)

The operating modes are USB, LSB, CW, CWR, SWU and SWL. USB, LSB and CW are self explanatory. CWR is CW but receiving on the opposite sideband (Default is LSB). SWU and SWR are for short wave listening and select either USB or LSB and while these modes are active the transmitter is inhibited to prevent accidentally transmitting on these frequencies. They are selected from the pull down mode menu and when you click on one to select it, the uBITX will change to that mode.



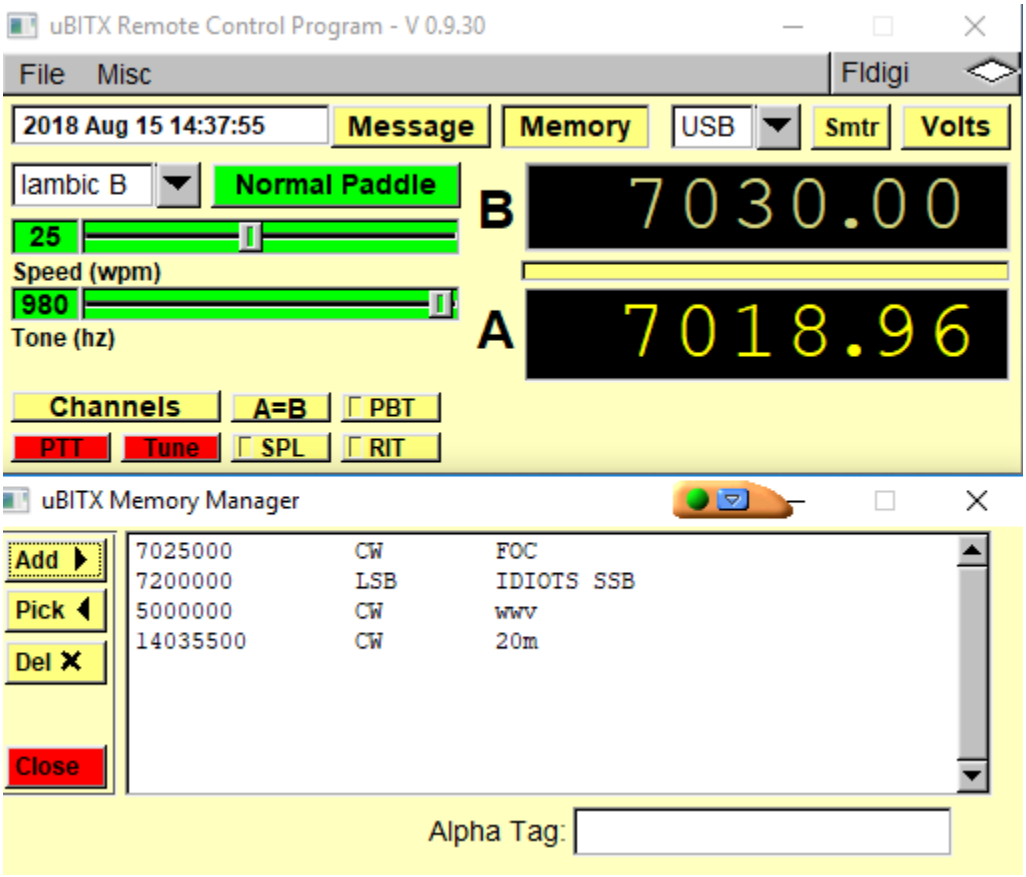
In CW and CWR modes, you have a “Message” memory function that can be accessed by clicking on the Message button which will bring up a special dialog box where you can enter, select and send short messages on CW (there is no capability for voice recording so this only works on CW).



In the above example, you see 3 messages, each with a square “check box” next to them. To enter a new message, type the new message in the bottom box next to the TxMsg button. Then click on the Add> button. Your message can’t exceed 255 characters and remember it has to be transmitted at your current Keyer speed.

If you want to send a message but not save it then just type the message into the same line as above but end it with the <ENTER> key and it will automatically be sent (as long as you are in CW mode). What was sent will appear in the “Sent” window line. To delete a saved message, left click on the check box next to that message and then click on the Del X button. When finished with message operation, click on the red close box and the message dialog window will close, returning you to the main control display again (you can hold the left mouse key down and pull any of the large dialog windows to whatever position you want them on your screen.

The “Memory” button opens a dialog window where you can set frequency and mode memories for later recall and you can recall them easily into either the A or B VFO depending on which VFO is selected as active on the main screen. Add>, Pick< and Del X buttons work similarly to the message memory window. Set the frequency and mode in the main display. Open the Memory dialog and click the Add> button – that frequency and mode will be added (if you want to name the memory, type the name in the “Alpha Tag” box at the bottom of the screen before clicking add and that’s all there is to it.



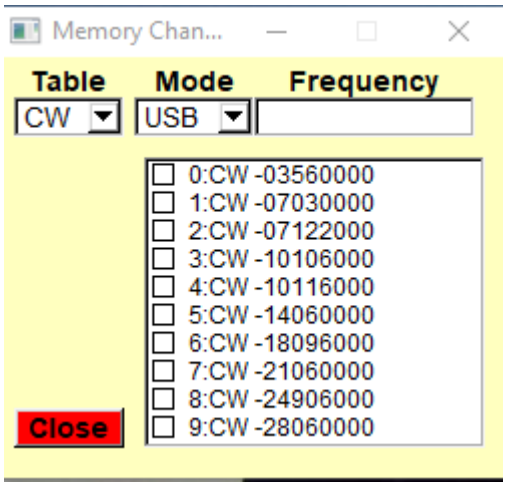
To recall a memory to a particular VFO (A or B) first make that VFO active on the main screen. Open the Memory box, click on the line containing the frequency/mode you want in that VFO and click on the Pick< button. You can also just double-click on the line containing the frequency/mode you want and it will be entered into the selected VFO.

To delete a memory, click on the line to be deleted and then click on the Del X button. That's all there is to it.

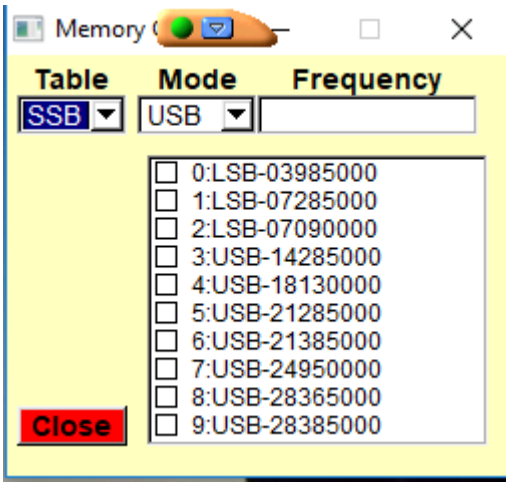
In addition to the MEM button a Memory Channel and Bank select button has been added that adds quite a bit of versatility to this program. If you click on the "Ch" button, it opens up a dialog box that will allow you to select between 3 pre-loaded memory banks and one user defined memory bank of 10 channels each (more may be added later). The banks are

named CW, SSB, DIG and USR. Frequencies and modes contained within the various banks can be easily edited from within the RCP.

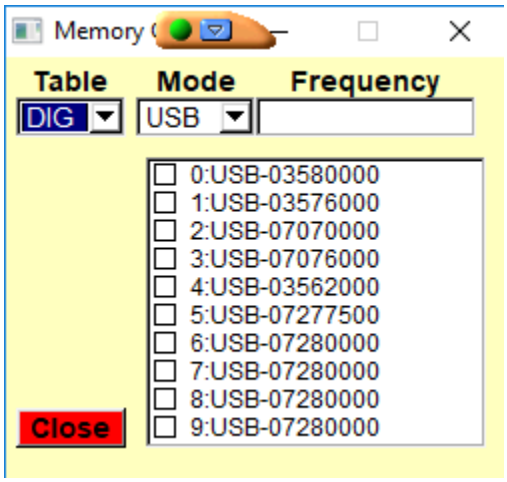
The CW bank comes pre-loaded with the QRP CW calling frequencies for the various bands.



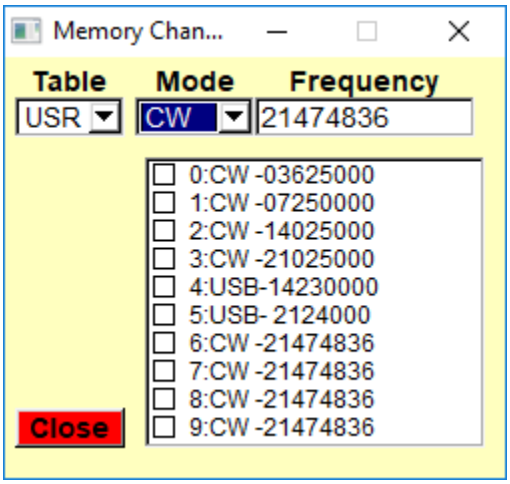
The SSB memory bank contains the SSB QRP calling frequencies for the various bands.



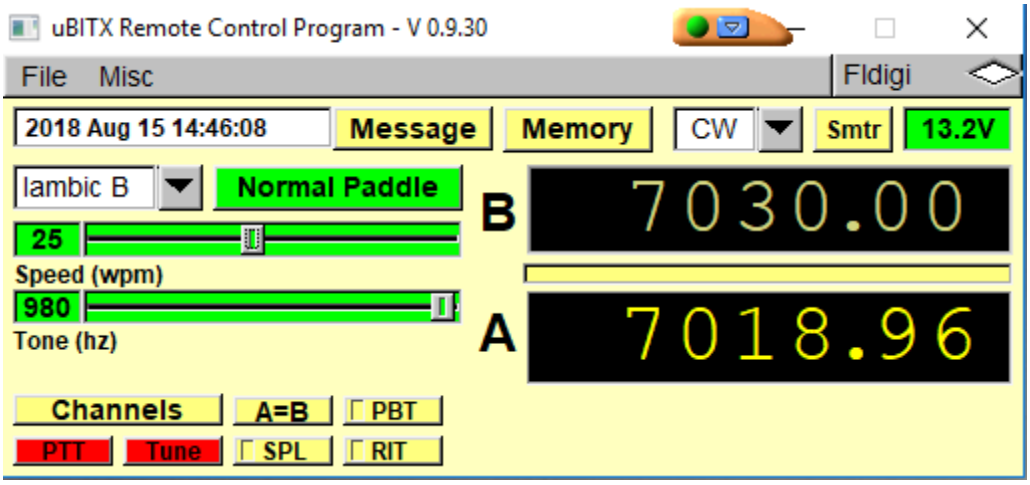
The Digital bank contains the standard Digital calling frequencies for the various bands



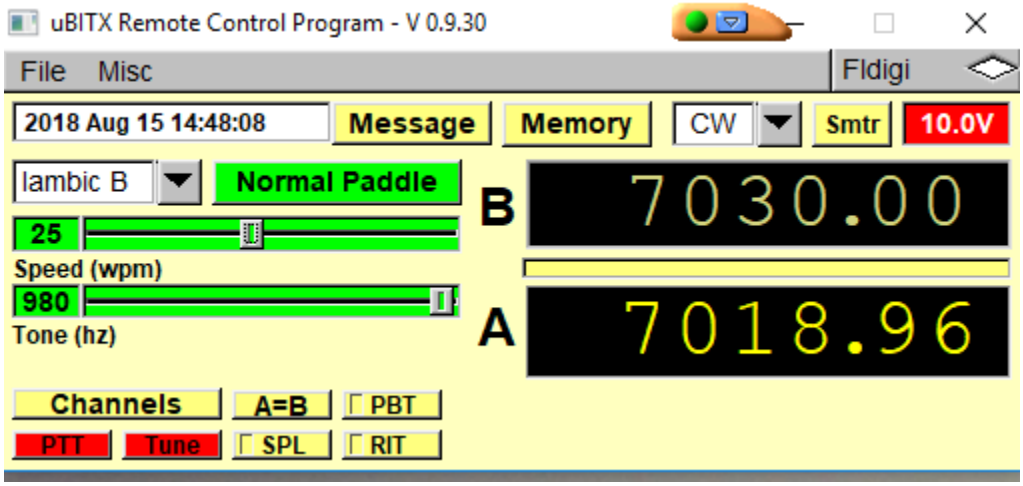
The User bank in this picture shows the way W0EB has his user bank configured.



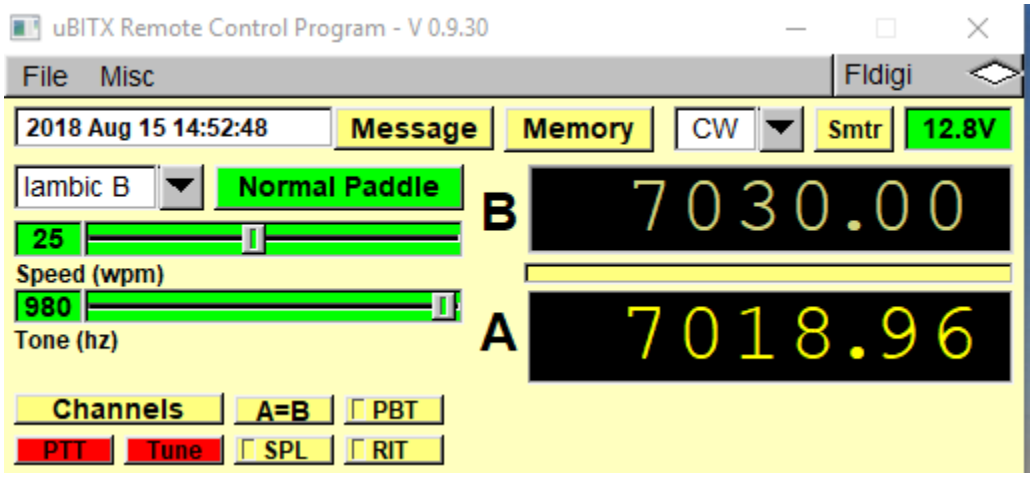
To change Mode, Frequency or both in a given table, click on the check box next to the memory channel you wish to change then click on mode which changes to whichever mode (CW, USB, LSB, SWU or SWL) you want for that channel. Type in the frequency (no decimal point and a leading zero below 10 MHz), hit enter and that memory location is updated to the new mode and frequency. Click on “Close” to close the window.



The “Volts” button is sort of a special case. It only works with our RadI2Cino and newer cards that have an added voltage divider connected to the A6 analog input and is used to monitor the incoming +12 Volt power so when on battery you can see when you need to change or charge your battery. If the voltage falls below 11.0 volts (i.e. 10.9) both the “Volts” button and the green voltage gauge change to RED indicating the battery may be damaged if you discharge it below that mark.



The next couple of functions pertain to the CW keyer and how your key paddles work. The window showing (currently) Iambic B is a small pull-down and you can switch the keyer type to either Iambic mode A or B just by clicking on it here. The green Paddle Norm button will change to red with white letters when pressed and the dot/dash paddles will be reversed.



Clicking the button again will put them back to what most people call “normal” with the dots on the left paddle, dashes on the right.

The green slider below the paddle controls is your keyer speed control. The slider will control the speed from 5 WPM up to 50 WPM and you can either click on the “knob” and slide it or place the mouse cursor on the line and use the scroll wheel (if you have one) to roll the speed up or down.

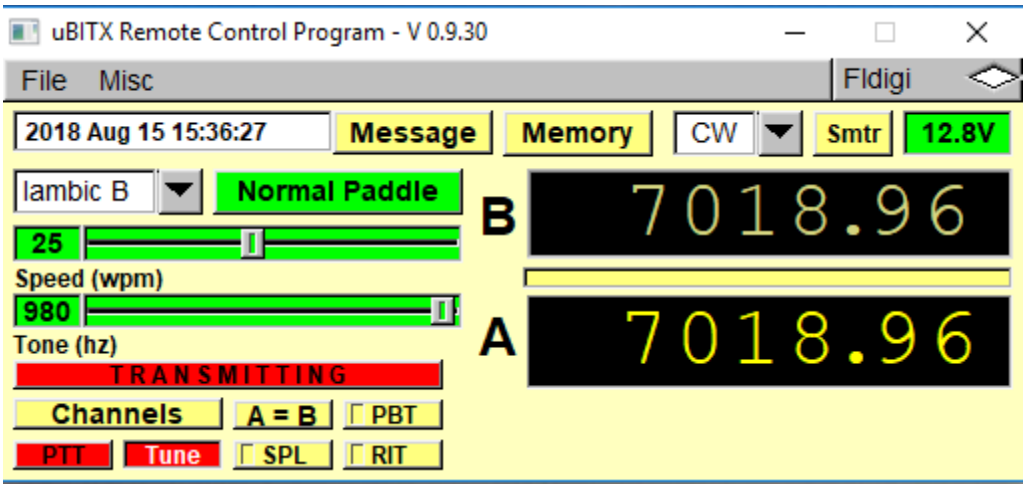
The Tone slider below that functions the same way but controls the side tone frequency. The ST does NOT sound while using it however.

The PTT and TUNE buttons both place the uBITX into TX.

PTT is used in SSB mode and when you click on it, the red “Transmitting” banner will display and the rig will stay in transmit until the button is clicked

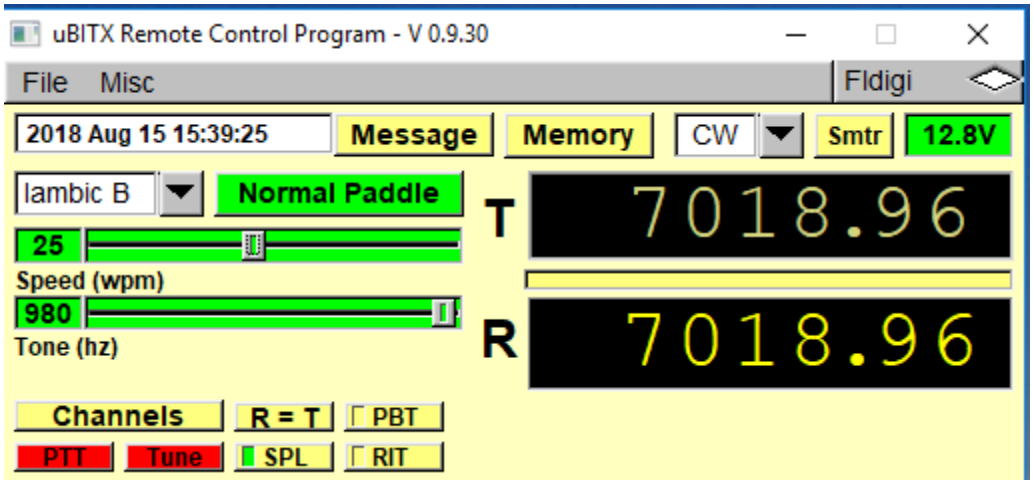
again when the uBITX will return to RX and the red “Transmitting” banner will disappear again.

Tune is used to place the rig into transmit and regardless of the selected mode (except for the SWL and SWU short wave listener modes) an on-frequency CW carrier will be transmitted to allow for adjusting antenna tuners, etc. The red “Transmitting” banner will display and the side tone will also sound during the duration “Tune” is active.



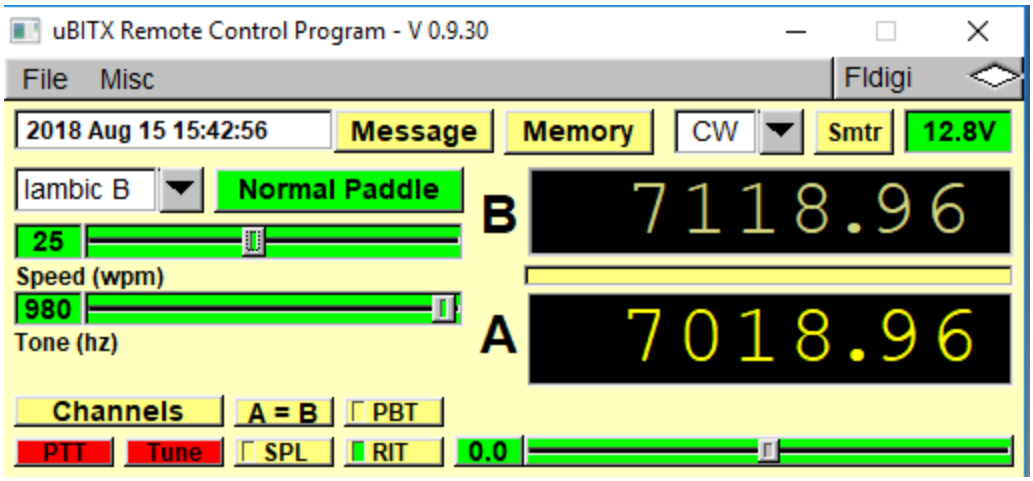
The A=B button, when clicked simply copies the frequency in VFO A into VFO B so they are equal.

The Split button, when clicked puts the uBITX into “Split” mode where the transmit frequency is controlled by VFO B and receive is controlled by VFO A. When active, the little square window to the left of “Split” in the button will change to a green square. VFO B is also set to VFO A’s frequency to make it so you have a good starting point to set your split transmit frequency.

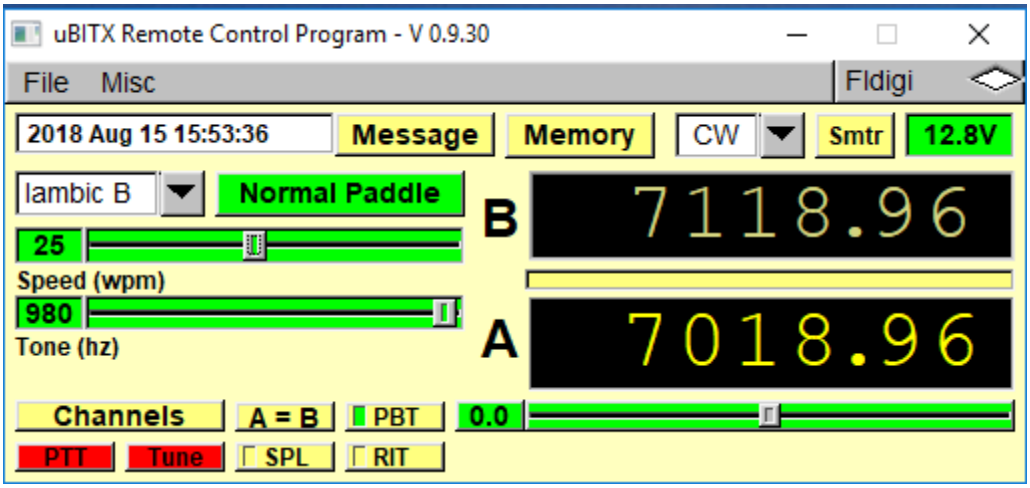


The designator buttons for the A and B VFO's will change to "R" and "T" respectively and the one being tuned will have the frequency displayed in Yellow while the other frequency will be grey. To turn off Split, just click the button again and the little green box will clear as well as the VFO indicators will return to A and B again.

The Rit button allows for fine tuning the receiver frequency independent of the transmitter to help when someone calls you but slightly above or below your frequency. When you click on the Rit button, the little square to the left of the R changes to green indicating that Rit is active and an adjustment slider similar to the previously mentioned Keyer Speed and Tone sliders appears to the right of the Rit button. You can use your mouse wheel to operate this slider or you can click and drag the slider however you like. Clicking the Rit button a second time turns the feature back off. The Rit's range is + or – 5 KHz from the indicated frequency of the active VFO.

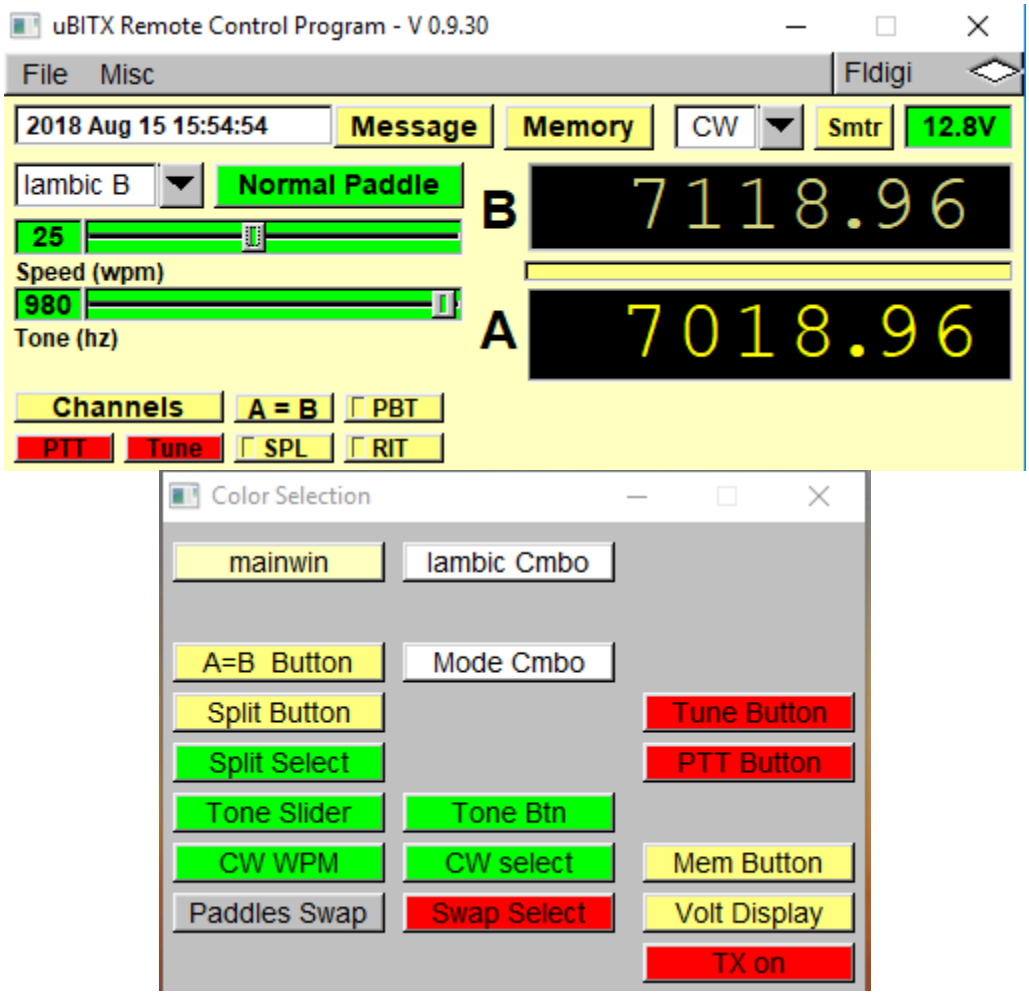


The PBT button selects “Pass Band Tuning” mode where the IF and BFO frequencies can be shifted in unison to move the passband of the crystal filter up or down from the normal values to place a given signal optimally on the filter’s skirts to help eliminate interference. When selected, the slider below the VFO A frequency display turns green and becomes the tuning slider for PBT. Click PBT again to turn it off and the IF/BFO frequencies return to the normal calibrated values.



The final item is the ability to customize the look of the main program window. Under the Misc pull-down menu, is an item you can use to set the

colors of all the different buttons, windows, sliders and other features of the program. You will have to experiment with this a bit to find the color combination that suits you if it differs from the “Factory” settings. Some of these settings are for future use and not yet implemented. Also, the Time Set feature is not implemented for any card running our software that has an Arduino NANO as the micro-controller as the NANO does not have a built-in “Real-Time-Clock”



The various buttons allow you to customize the colors for the various items and buttons on the display to whatever you personally prefer.

Also under configure I/O you will see references to FLDIGI with an “fldigi socket addr (defaulted to 127.0.0.1) and an fldigi socket port (defaulted to port 7362). These are for use with the FLDIGI program and though this remote control program does not allow FLDIGI to control the uBITX, it will pass frequency and mode information to FLDIGI from the uBITX via this program to aid in FLDIGI operation.

Please contact Ron Pfeiffer, W2CTX (w2ctx@yahoo.com) to report any bugs or concerns regarding this program. – Enjoy –

Document author/editor, Jim Sheldon, W0EB, Program author and primary editor, Ron Pfeiffer, W2CTX.