

***WKmini* USB**

WinKeyer3
Morse Interface
User Manual v1.3



K1EL
HAM RADIO KITS

www.k1el.com

Introduction

This document will describe *WKmini* and its various interfaces. All the information you will need to connect *WKmini* to your station will be included in this manual along with specific guides for getting it working with N1MM+ Logger, HRD/DM780, N3FJP AC Log, FLdigi, and MRP40.

What is WinKeyer and why do I need it ?

Due to timing latencies inherent in multi-threaded operating systems like MS Windows, it is very difficult to generate accurately timed Morse in software. All it takes is a higher priority task to demand service and Morse generation will be delayed which results in wrongly timed dits, dahs, or the spacing between them. A WinKeyer based device, like *WKmini*, attaches to a PC's USB port and appears as a serial com port to an application running on the PC. The application, instead of generating Morse in software, sends the letters to WinKeyer. The application has full control over sending speed as well as weighting, spacing, and PTT generation. *WKmini* has a paddle interface so a user can break in immediately to handle fast exchanges in contest situations. Electrically isolated keying outputs provide safety and prevent ground loops. Two separate keying outputs allow the user to run two different radios from one keyer without swapping cables. The feature list below shows some of the advanced options supported by WinKeyer. Some WinKeyer products also provide a speed potentiometer interface so that an operator can instantly dial any speed desired.

WKmini Features

- USB 2.0 Interface with Mini type B connector
 - Iambic CW Paddle Interface
 - Iambic A, B, Ultimatic & "Bug" paddle modes
 - Two separate keying outputs OR
 - One key output with PTT
 - Solid state relay output rated at 60V @ 200 mA
 - 160 character input buffer
 - External power not required, runs off USB power
 - ESD protection on paddle input
 - Firmware can be updated over USB
 - Embedded commands
 - Meets FCC and CE emissions requirements
 - RTTY FSK transmit with WK3.1
 - Utilizes K1EL's latest WK3 IC
 - Adjustable Speed 5-99 WPM
 - Adjustable Weighting
 - Adjustable Keying Compensation
 - Adjustable Letterspacing
 - Adjustable PTT lead in and tail delays
 - Optional autospacing
 - Metal enclosure with RFI filtering
 - Auxiliary input for future enhancements
 - 160 character input buffer
 - Inexpensive cabling options
 - Adjustable dit/dah ratio
- Works with any logging or contest application that supports WinKeyer !!

Product Warranty, Support, and Liability

WKmini is fully warranted to the original purchaser against defects in materials and workmanship for one year after purchase. This warranty does not cover damage caused by accident, improper care, or lightning damage. Please contact us before returning your *WKmini* for repair and we will issue an RMA.

Please submit support questions by e-mail to k1el.kitsinfo@gmail.com

While every effort has been made to insure that the *WKmini* design is safe and the documentation is clear and accurate, it is still possible to cause equipment damage or incur personal injury if:

***WKmini* is not used as intended.
WKmini is connected incorrectly.
Safety guidelines outlined in this document are not followed.
WKmini is modified in any way.**

K1EL or Hamcrafters cannot be held responsible in these or other similar events.

Let's Get Started !

USB drivers

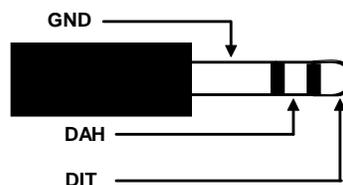
Don't plug WKmini into your PC until you have read this paragraph !!

WK uses a USB interface IC manufactured by FTDI. This is one of the most popular and reliable USB devices in existence. We depend on FTDI's expertise and they do an excellent job in maintaining reliable USB drivers that run on various platforms and interface to their controllers. Most current model PCs now ship with an FTDI USB driver preinstalled so you can just plug WKmini in and you are good to go. However, we highly recommend reading through Appendix A for our recommended install procedure which may save you some trouble. Just to be clear, there is nothing special about installing a USB driver for WKmini. If you have USB device experience and are familiar with the process, feel free to proceed on your own.

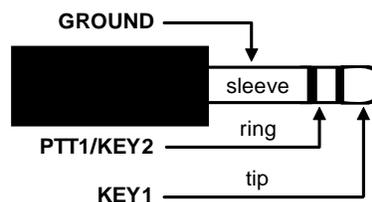
WKmini Connections



There are four connectors on WKmini, two on the left side and two on the right, The left facing connectors are labeled PDL and OUT. The PDL connector is an 1/8" stereo jack which accepts paddleset input. The paddle can be an iambic or side swiper type. A bug is not acceptable. The paddle inputs are activated when switched to common ground (GND). The paddle connector needs to be wired this way:



The OUT connector uses the same type connector as the paddle input. It provides two switched outputs. These outputs are isolated from WKmini ground and all other WKmini signals by an optically coupled solid state relay. The outputs can switch voltages up to +/- 60 V at up to 200 mA and provide a 5000 volt isolation from WKmini electronics. The output connector needs to be wired this way:



Note that the ring connection is dual purpose, it can act as a PTT output tied to KEY1 or it can act as a separate second key output KEY2. The connections are controlled by software configuration.



The right facing side connectors are labeled **USB** and **AUX**. The USB connector is designed to mate with a Type A to Mini-B cable. This cable is shown at left. The interface will run at USB 1.0 and 2.0 speeds but is not compatible with USB 3.0.

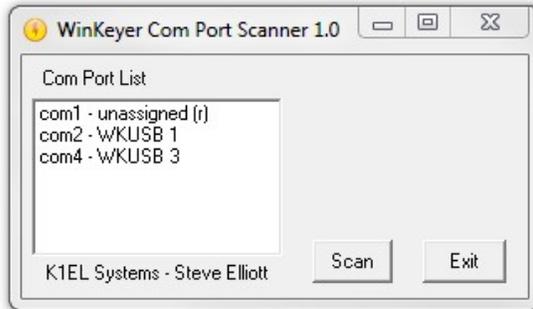
The AUX connector is reserved for future enhancements to the WKmini product. Currently this connector is not used.

WKmini contains no user serviceable or configurable jumpers, Please do not tamper with or modify the internal circuitry of WKmini as this will void the unit's warranty and may result in unsafe operating conditions.

Let's actually run WKmini in CW mode !

Once the USB driver is installed and you have plugged WK into your PC, you can determine WK's port assignment using K1EL's WKscan app: <http://www.hamcrafters2.com/WKscan.html>

WKscan Tutorial



After you have installed WKscan, start it up and you will see the dialog box shown at left. Click Scan and WKscan will display all of the serial ports found on the system. The ports will either be identified as unassigned, in use, or attached to a WK device. In this example we have a WinKeyer1 device on COM2 and a WinKeyer3 device on COM4. Yes, multiple WKs can be operated from a single PC using two applications.

Once you know WK's com port, you are ready to run WKdemo.

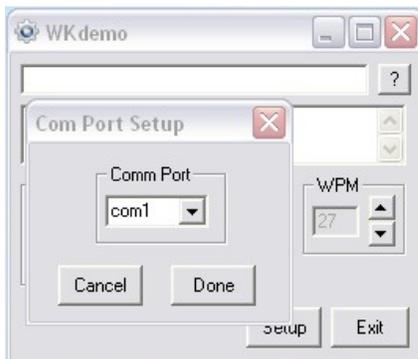
WKdemo tutorial



WK3DEMO is a general purpose test application that can exercise all of WKmini's capabilities:

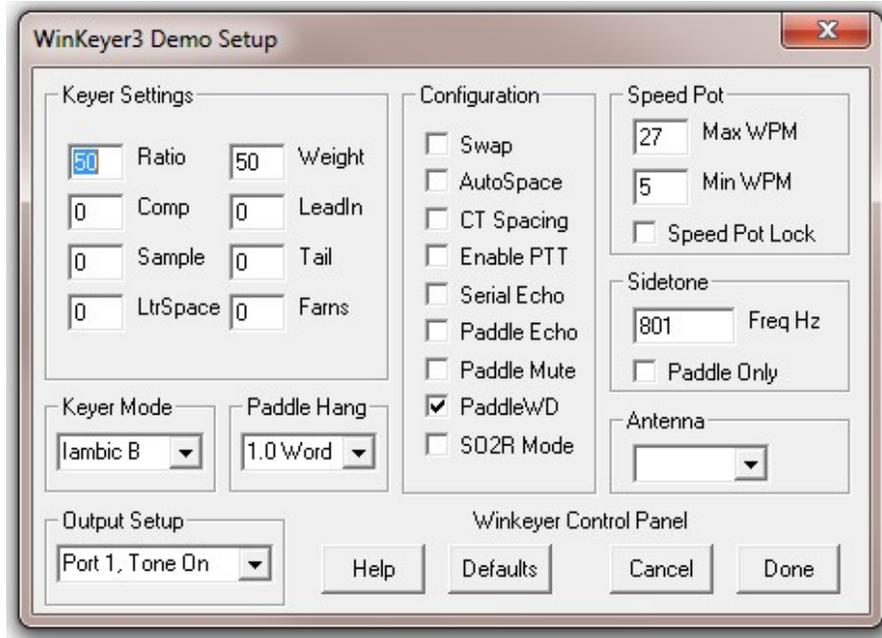
<http://www.hamcrafters2.com/WK3demo.html>

Once the application is installed, start it up and you will see the dialog box show at left. All you need to do to get started is to configure the serial communications port. Click on Setup and the com port selection list is shown. Only the currently active USB com ports are shown. Select a WK com port identified by WKscan.



You can click on ? In the upper right for a very brief help menu.

Click on Open to connect to WKUSB. If all is well, the Open pushbutton will change to Close and the WK IC version will be displayed. Click on Setup again and this time the Setup dialog box will be displayed as shown on the next page. This presents all of WK3's various configuration options which can be modified to your requirements.



Click Cancel for now and go back to the main dialog box. Type into the lower keyboard entry window and the test LED light will flash through the hole in the side of the enclosure. WKmini does not have a local sidetone speaker so the LED is included for testing purposes.



For this reason, the sidetone options on the WKdemo setup display are disabled. If serial echo is on in Setup then what you type will be echoed in the upper text window as it is sent.

Now hook up some paddles and test paddle entry. If Paddle Echo is enabled, letters entered on the paddles will displayed in the upper text window. It's tricky to do this without an audible sidetone. Use the Setup Dialog to try different keying options. Note that if you change settings on the Setup dialog box, you have click Done and return to the main application to transfer those new settings to WKmini.

Like sidetone, WKmini does not have a speed pot so those controls related to speed pot use are disabled. That means it will function as if Speed Pot Lock was disabled. When Pot Lock is unchecked, the Up/Down arrows next to the WPM readout on the main dialog box will control the speed or you can directly enter a speed in the WPM edit box. In other words, speed is controlled solely by the host application. This is how the majority of PC logging and contest applications operate. Speed is controlled by keyboard commands, macros, or messages with embedded speed commands. The name of the game in contesting is to have one central user focus, that being the keyboard.

Although WKtest packs a lot of power in a small app, it is meant to be a hardware test tool; it gives you a good idea of what WKmini is capable of and assures you that it is operating properly. The next logical step is to move on to a 'real' application, such as N1MM+. Before that we will briefly look the WK3tools application.

WK3tools application

Normally when a WinKeyer device is not connected to a host application it will run in standalone mode. Since WKmini has been designed to be a cost effective host only device, it does not provide a useful standalone mode. It will operate when connected to a USB port, but since there are no pushbuttons or a sidetone speaker, the user has no control over its operation. In other words you can send at a fixed speed using a set of paddles but that's about it. You really need to run it from a PC application for it to be effective.

<http://www.hamcrafters2.com/WK3tools.html>

The WK3tools application is used to customize standalone settings, you can use this app to set up a standalone configuration and it will work but as previously stated there aren't many options here. Any controls related to messages, users, and speed control are disabled. WK3tools is most useful for updating the firmware on WK3mini. Current firmware versions are posted here:

<http://www.hamcrafters2.com/WK3upgrade.html>

If you find there is a newer firmware version available for your device, send in an upgrade request and we will send you the upgrade files and directions on how to perform the upgrade.

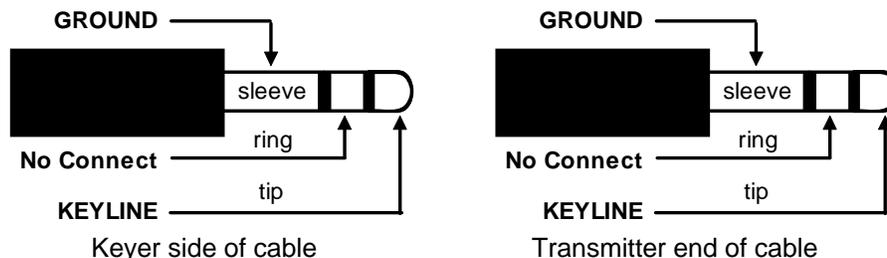
Connecting WKmini to your station

Minimally all you need to do is connect the USB cable from *WKmini* to a USB port on your PC and connect the key output to the keying input on your rig. Since WKmini performs the keying function it drives your rig in the same manner as a straight key. In other words there are only two leads, ground and key control. For that reason a keying cable is quite easy to build.

Hamcrafters sells a universal keying cable kit that covers 95% of keying requirements.

Keying cable details (also see page XX)

An 1/8" stereo phone is recommended for the WKmini end of the keying cable and the other end will have a connector that mates to your rig's key input. Usually this is a 1/4 inch phone plug. A two wire cable is all that is required to key a typical transceiver. Wire the key line to the tip of the 1/8" connector and ground to the sleeve as shown here.



The nice thing about using a stereo connector on the keyer side is that you can easily add PTT or a second key line in the future. On the transceiver end, the key line goes to tip and ground goes to the sleeve. Make sure you disable the radio's internal keyer (if it has one) and set it up for

straight key operation. It's always a good idea to test your keying cable before using it with WK. Plug the cable into your transceiver and leave the keyer end unconnected. Now set up your transceiver for transmit and then short the keyer cable end, tip to sleeve, with a clip lead. If this doesn't work then either the cable is not made correctly or the radio is not in break in mode. If you get dits or dahs, then the keyer in the radio is still enabled. Some transceivers want you to use a stereo plug even if the internal keyer is disabled, it's a good idea to use a stereo connector at the radio end no matter what and leave the ring unconnected.

If you want to key a transmitter with vacuum tube finals, WKmini will work fine because it has a solid state keying relay rated for $\pm 60V$ switching at 200 ma. This may cover grid block keying but will NOT be suitable for cathode keying. In any case always check the transmitter's manual for keying requirements before plugging WKmini in.

WKmini's keying connector is electrically isolated from WKmini's enclosure and ground. This means that you can directly connect your transceiver to WKmini with PC attached and not worry about ground loops or RF getting into your PC through WKmini.

PTT Connection

PTT will not be used in most radio setups. It is provided to run an external antenna switch or to pre-key a linear amplifier. If you are just driving a barefoot transceiver, PTT will not be required.

If you do need PTT, WKmini provides useful functionality. PTT is asserted before KEY and is held for the entire transmit interval. Both lead-in and tail delays are adjustable. Just wire PTT to the ring of the keyer connector and configure WKmini for single radio keying (KEY1/PTT1). Make sure that PTT is enabled in your configuration settings.

Paddle Connection

If you decide to use a set of keying paddles with WKmini, they plug into the paddle jack input on the left side of the unit. An 1/8 inch stereo jack is used on WKmini, if your paddles have a 1/4 inch size plug an adapter cable will be necessary.

WK3demo is useful for verifying that WKmini is keying your rig properly. If you run into problems, make sure you have selected the correct key output port. This is done in the Output Config control in the WKdemo setup dialog box. Also check to make sure the connector is fully plugged into the WKmini jack. The tune function is useful for signal tracing.

A cheap path to a keying cable

In today's world of iPhones and portable music players, 1/8" stereo to 1/8" stereo connectors are very easy to find and are inexpensive. If your transceiver requires an 1/8" keying plug you are done. If it needs a 1/4" plug, purchase an 1/8" to 1/4" stereo adapter.



Alternatively, an 1/8" stereo plug to dual RCA cable can also be used. In this case get two RCA to 1/4" connectors and you have a keying cable that can key two different radios.



The point is that you can come up with just about any cable you need by juggling cable types and adapters. Or buy one of our cable kits from us.

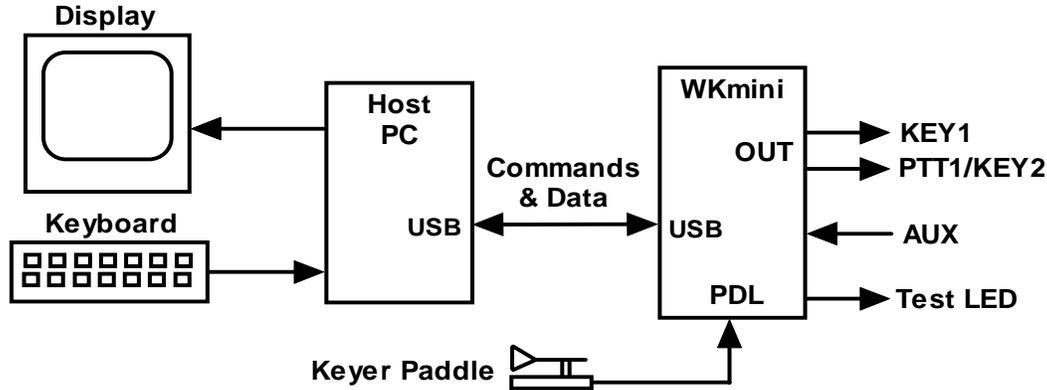
Note that 3.5mm and 1/8" connectors are the same size, so don't let that confuse you.

Troubleshooting hints

- 1) *WKmini won't key my radio.* First check to make sure your keying cable is wired correctly and then check to make sure it is plugged into the KEY output. If you have it plugged into AUX or PDL it won't work.
- 2) I get dits or dahs only or the CW output is garbled. This usually happens when you have not disabled your transceiver's internal keyer. You need to enable straight key mode. In most transceivers this function is controlled by menu selection, older radios may employ a dip switch or jumper. Please check your radio's user manual.
- 3) When I plug WKmini into the USB port, I see an R flashed on the debug LED but nothing else works after that. This is a sign that one of the dit or dah paddles are stuck low. Remove the paddle connector and see if that clears the problem. If so you may have a short in your paddle cable or paddle.
- 4) *When I try to run WKmini from WK3demo I get the error message Attach WK: No Echo.* This means that the com port selected was a valid one but not the one that WKmini is connected to. Recheck your port selection with WKscan.
- 5) *WKmini is plugged in but WKscan does not show a com port with a WK device.* This is a sign that either a USB driver has not been installed or the USB hardware on WKmini is not working. It could be a bad USB cable, or bent pin inside the USB connector. Note that the USB interface is factory tested before a board is shipped so if it doesn't work the places to look for trouble are limited to the connector and cable. It could be that the USB cable is not fully inserted at either WKmini or at the PC.

Theory of Operation

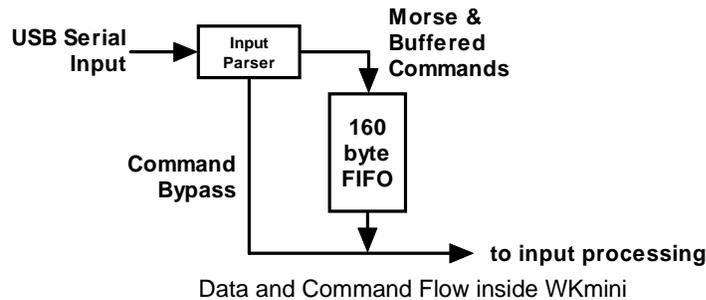
This section will briefly describe how WKmini works. As shown in the block diagram below, the host PC is connected to WKmini over a USB com port. WKmini receives commands and Morse letters from the PC. Letters will be sent in accurately timed Morse.



PC to WKmini Block Diagram

Commands modify WKmini's operation in some way, for example changing operating speed, pausing transmission, or asking for status. Commands and Morse letters are processed differently in WKmini. Morse is put into a FIFO* buffer memory that allows the host to send data ahead of the Morse being sent. The size of this buffer is 160 bytes. Since there can be a considerable delay from host input to Morse output, commands bypass the input FIFO and are acted upon immediately. This allows changes to be made while sending is underway.

*First In First Out; bytes are taken out in the order they were put in.



Since there are times when you don't want commands to take effect immediately, WKmini allows commands to be buffered. This means that the command is placed in the serial buffer mixed with Morse and won't be acted on until it comes out of the buffer. An example would be sending two words at two different speeds, the first at 15 WPM and the second at 20 WPM. By placing a buffered speed command between the words, the speed will not be changed until the first word is completely sent.

WKmini will send data back to the host for processing. There are four reasons this will happen:

- 1) Inform the host of a WKmini status change.
- 2) Inform the host of a speed pot or pushbutton change.
- 3) Respond to a request for status from the host.
- 4) Echo Morse as it's being sent by message or by paddles

Paddle Input Priority

WKmini accepts input from either its host port or iambic paddle port. Paddle input will always take priority and will interrupt host data, automatically clearing WKmini's serial input buffer. When a paddle break-in occurs, any additional serial data that arrives from the host will be processed, but will be ignored unless it is an immediate command. After paddling ceases, WKmini will pause for one word space time before it resumes serial data transmission.

USB Sense

WKmini is able to sense when the host PC it is connected to goes into standby. This causes WKmini to also go into a standby state and it will stay in this state until the PC comes out of standby. It will wake up and be ready to resume. The caveat is that the host application must be able to properly handle standby and not close any of its active serial ports. If the host app does not reliably handle standby then set the PC's settings so that it will not be allowed to time out and go into standby.

SO2R Mode

WKmini is not able to support SO2R mode like WKUSB and other K1EL keyers. It doesn't have enough outputs to support it properly

Serial Baud Rate

WKmini's standard baud rate is 1200 baud with an alternate baud rate of 9600 selected by command. This is all handled by the host application so there is no need or requirement to set the USB port's baud rate, stop bit, or parity settings. All of these things are configured automatically by the host application when it opens WKmini for use.

Reset WKmini/Restore Factory Defaults

There are cases when a host app misbehaves and can corrupt WKmini's basic setup parameters. We don't know of any specific app that does this, but under heavy use with RF floating around unexpected things can happen, ESD is another issue that can cause this to happen. WKmini has very good ESD suppression on the paddle inputs so again, not likely. The easiest way to restore WKmini's EEPROM settings is to run WK3tools. First unplug WKmini from the PC and then plug it back in after 10 seconds. Then start up WK3tools, set the com port and click WKtest to be sure it can connect. Then simply click Reset to Defaults followed by WriteWK. That will restore WKmini to factory fresh settings.

What about RTTY transmit, how does work ?

Starting with WK version 3.1, we added RTTY transmit capability to WK which allows you to run hardware FSK RTTY with N1MM+, FLdigi, and WK3demo. In depth details on how the RTTY API works can be found in Appendix A on page 35. Setting up RTTY for N1MM, FLdigi, and WKdemo are covered in this document on pages 19, 29, and 31.

RF Immunity

There are several RF filters included in the WKUSB circuit design. Bypass capacitors on the keying and PTT lines prevent RF from coming back through the keying lines, Bypass capacitors shunt any RF arriving over the paddle leads, and an LC filter is employed on the USB power input. These components also reduce WKUSB radiation.

PCs and associated peripherals can produce RF emissions on their own that may cause RFI to your radio receiver. In addition, most keyboards have poor conducted RF emissions immunity and operation can be compromised in high RF environments. To guard against this, additional filtering may be required on the keyboard, usually a ferrite bead on the keyboard cable will address this.

FCC Regulations Compliance

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

Radio and Television Interference

This equipment has been tested and found to comply with the limits for a [Class B digital device](#) pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against [harmful interference](#) in a residential installation. This equipment generates, uses and can radiate [radio frequency energy](#) and, if not installed and used in accordance with the instructions, may cause [harmful interference](#) to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause [harmful interference](#) to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

You may also find helpful the following booklet, prepared by the FCC: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the U.S. Government Printing Office, Washington D.C. 20402.

Changes and Modifications not expressly approved by the manufacturer or registrant of this equipment can void your authority to operate this equipment under Federal Communications Commission's rules.

In order to maintain compliance with FCC regulations shielded cables must be used with this equipment. Operation with non-approved equipment or unshielded cables is likely to result in interference to radio & television reception.

ASCII Code Assignments, Prosign Mapping

WKmini maps several unused ASCII codes to standard prosigns as shown in the table below. Additional prosigns can be generated using the merge character command. Some abbreviations are not mapped (null), WK will ignore these if received. Codes below 0x20 are commands, 0x30 to 0x39 are numerals 0-9, 0x41 to 0x5A are letters A-Z. All codes above 0x5D are ignored except for | (0x7c) which is the ½ dit pad code.

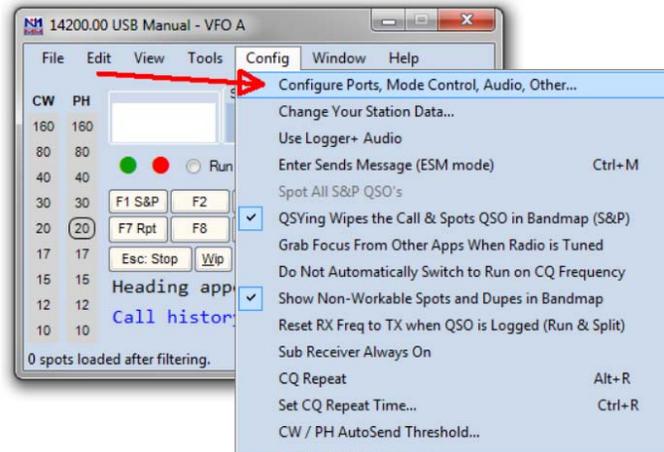
| ASCII | Hex | | Prosign | ASCII | Hex | | Prosign |
|-------|------|---------------------|---------|-------|------|---------------------|---------|
| | 0x20 | <i>Is space</i> | | . | 0x2E | <i>Is period</i> | |
| ! | 0x21 | <i>Is mapped to</i> | (null) | / | 0x2F | <i>Is mapped to</i> | DN |
| " | 0x22 | <i>Is mapped to</i> | RR | : | 0x3A | <i>Is mapped to</i> | KN |
| # | 0x23 | <i>Is mapped to</i> | (null) | ; | 0x3B | <i>Is mapped to</i> | AA |
| \$ | 0x24 | <i>Is mapped to</i> | SX | < | 0x3C | <i>Is mapped to</i> | AR |
| % | 0x25 | <i>Is mapped to</i> | (null) | = | 0x3D | <i>Is mapped to</i> | BT |
| & | 0x26 | <i>Is mapped to</i> | (null) | > | 0x3E | <i>Is mapped to</i> | SK |
| ' | 0x27 | <i>Is mapped to</i> | WG | ? | 0x3F | <i>Is question</i> | |
| (| 0x28 | <i>Is mapped to</i> | KN | @ | 0x40 | <i>Is mapped to</i> | AC |
|) | 0x29 | <i>Is mapped to</i> | KK | [| 0x5B | <i>Is mapped to</i> | AS |
| * | 0x2A | <i>Is mapped to</i> | (null) | \ | 0x5C | <i>Is mapped to</i> | DN |
| + | 0x2B | <i>Is mapped to</i> | AR |] | 0x5D | <i>Is mapped to</i> | KN |
| , | 0x2C | <i>Is comma</i> | | | 0x7C | <i>Is ½ dit pad</i> | |
| - | 0x2D | <i>Is mapped to</i> | DU | | | | |

Table 1 – Prosign/Abbreviations Assignments

N1MM Interfacing

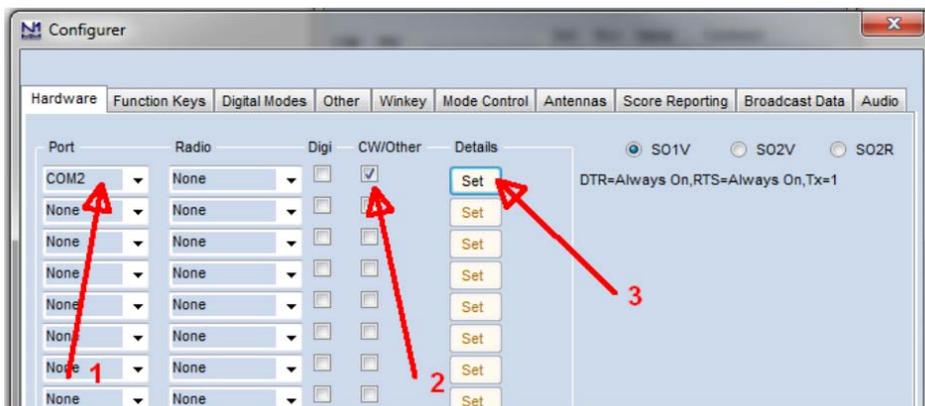
You will first need to know your WKmini's comport assignment, if you don't know it, follow the WKscan Tutorial on page 4.

Now start up N1MM+ and click on the Config Tab and then select the entry as shown:

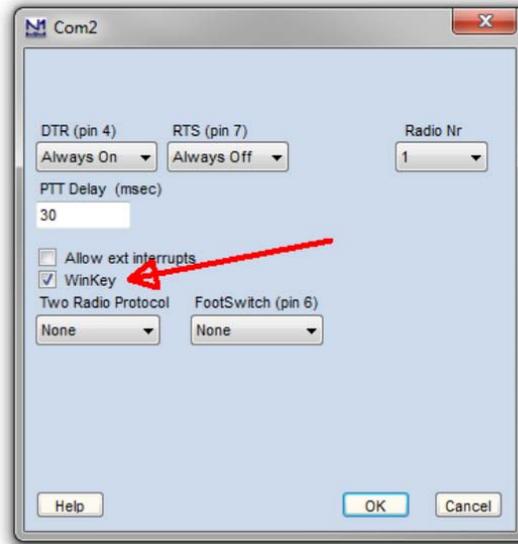


This brings up the Configurer dialog box. This is where devices attached to N1MM+ are set up. For this example we will set up a WinKeyer device on COM2.

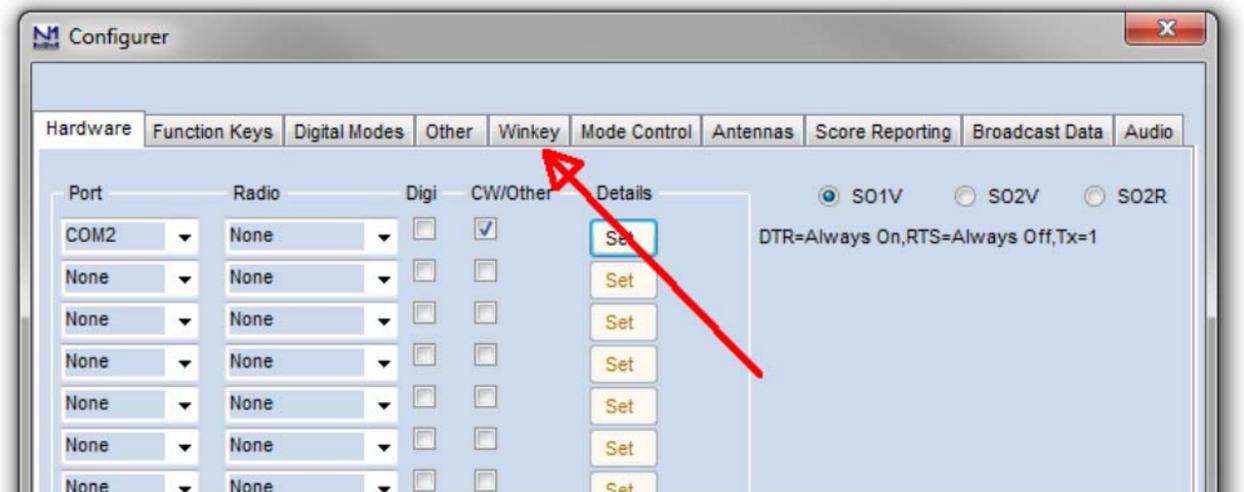
- 1) Select COM2 from the drop down box.
- 2) Specify that it is a CW device by clicking the CW/Other checkbox
- 3) Continue on to the Com Port setup dialog box by clicking Set



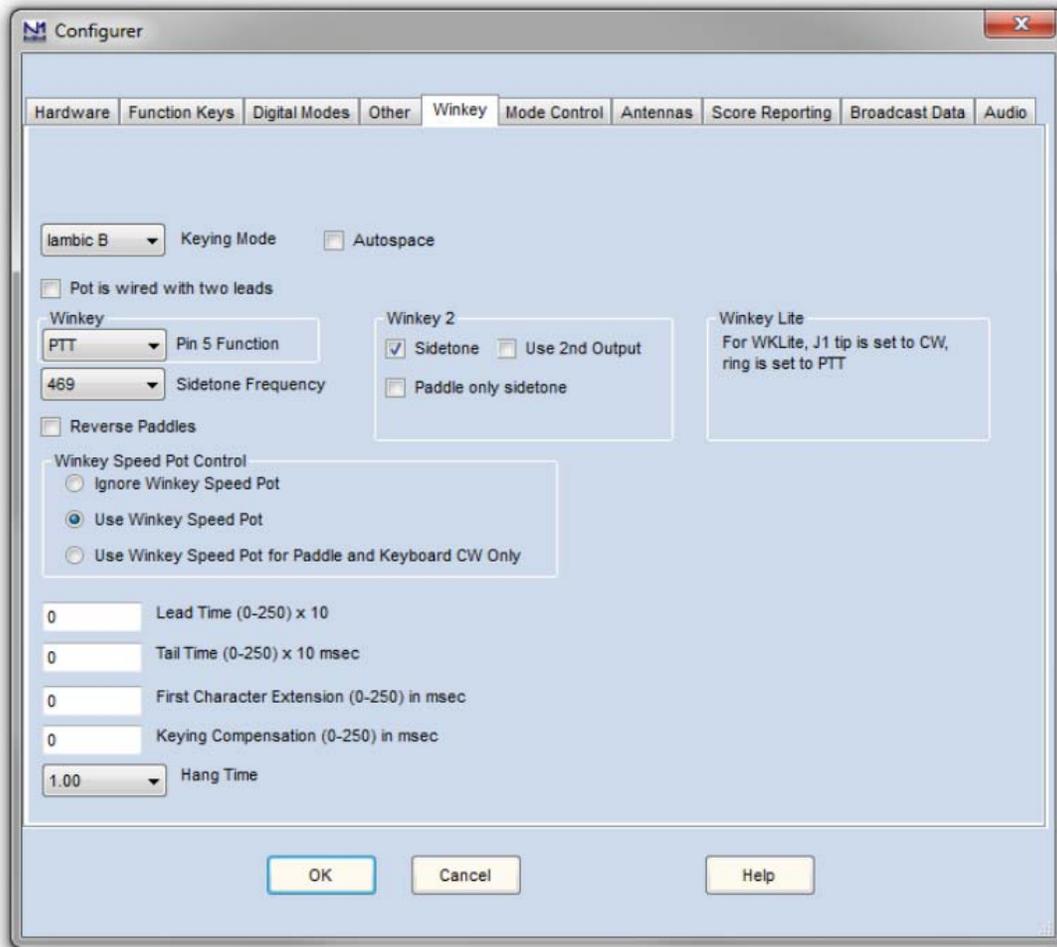
Clicking Set brings up the Com2 configuration dialog box. All we need to do here is click the WinKey checkbox, leave all the other settings as is.



Now click OK to close the Com2 configuration dialog box which brings us back the Configurer dialog box. Here we select the Winkey tab as illustrated below:



Clicking on the Winkey tab brings up the WinKeyer specific configuration dialog box which is shown on the next page.



There are very descriptive hints on most of the controls.

Winkey 2 options also apply to WK3 based keyers.

Disable Paddle Only Sidetone for initial testing purposes.

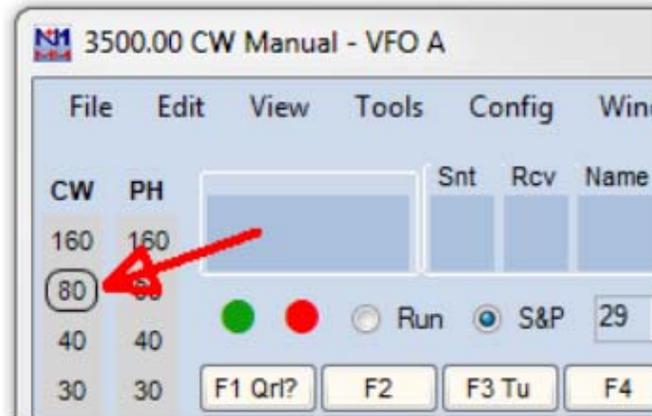
Winkey Speed Control has three options:

- 1) Ignore Speed Pot which only allows the host to control sending speed
- 2) Use Speed Pot which tells the host that the speed pot will set sending speed
- 3) Paddle & Keyboard where the host controls only the message speed

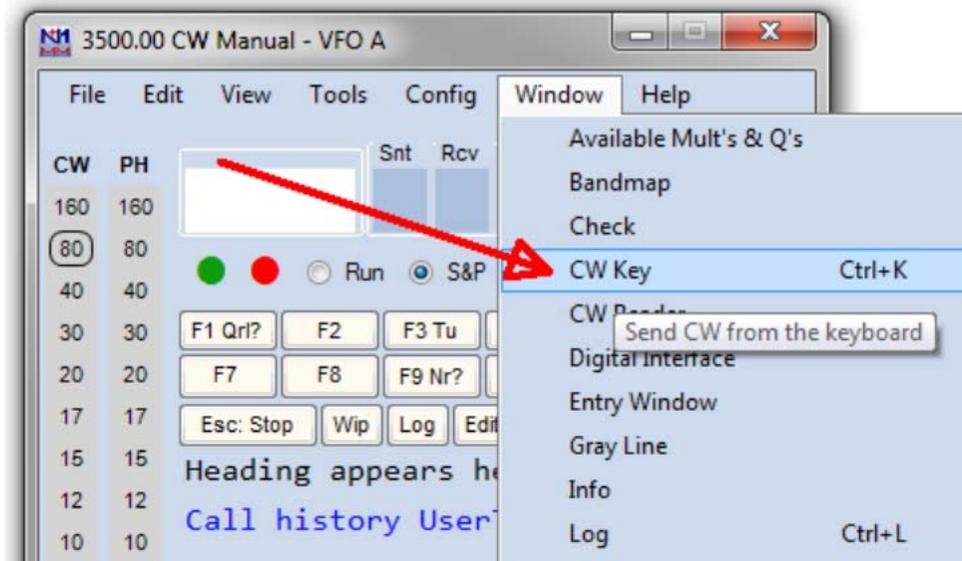
Lead and tail settings apply to PTT delay Keying compensation adjusts dit and dah spacing relative to sending speed.

Hang Time is similar to PTT tail except that it is proportional to sending speed.

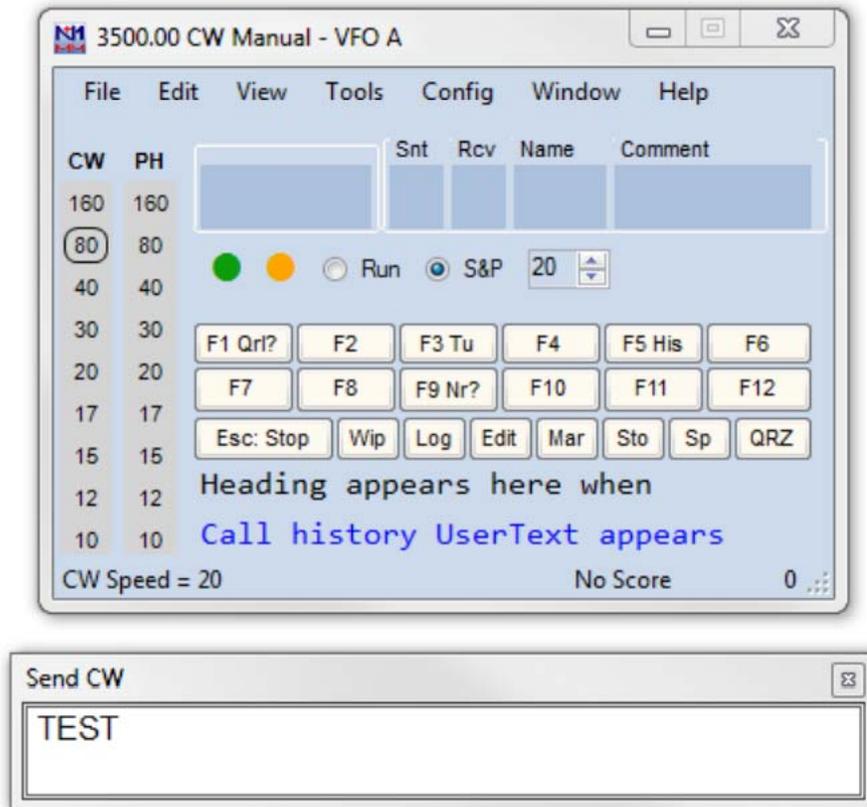
When you are finished, click OK and then another OK to close the Configurer. Now that we are back to N1MM+ main control dialog box we can do some WKmini testing. First off select CW mode by clicking on a band in the CW column:



Now on the PC's keyboard enter CTL-T to put WKmini into tune mode. Cancel tune mode by entering CTL-T a second time. With tune on, your radio will be keyed to allow tuning. If you don't have a radio connected, the test LED will be lit and can be seen through the hole in the side. Now either enter CTL-K or select the CW Key Window as shown:



The CW keyboard window is now displayed and you can type into it to send letters to WKmini. Note that when the Send CW window is up, the main N1MM+ dialog box is disabled.



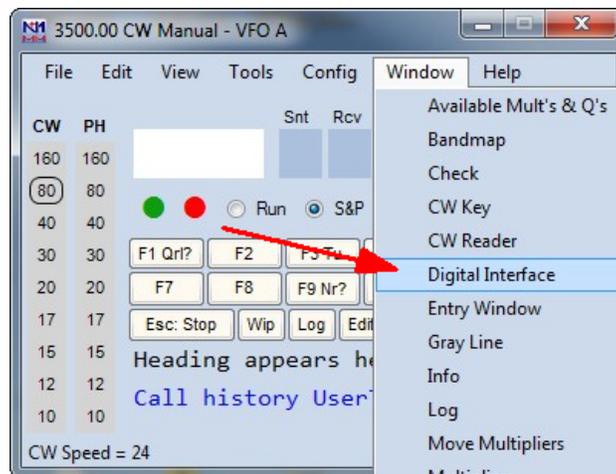
That's it for basic N1MM+ and WKmini CW testing. Now it's time to learn all the amazing features in N1MM+ for setting up messages and getting up to speed with automatic CW contest exchanges.

Running WK 3.1 with N1MM+ in RTTY mode

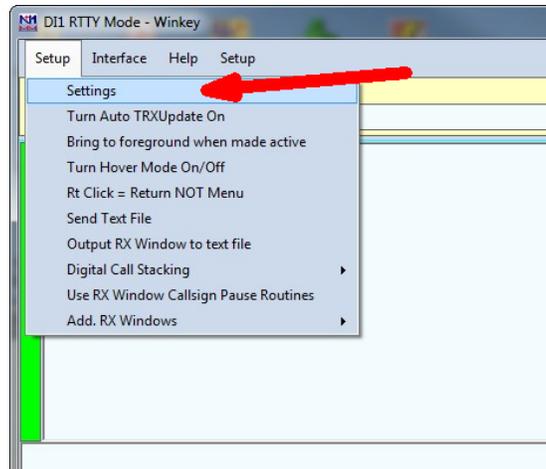
If you happen to have a WKUSB that uses the WK 3.1 controller IC, you can run it with N1MM+ in RTTY mode. Bring up the Configurer dialog for Winkeyer and you will see a new RTTY mode selection option:



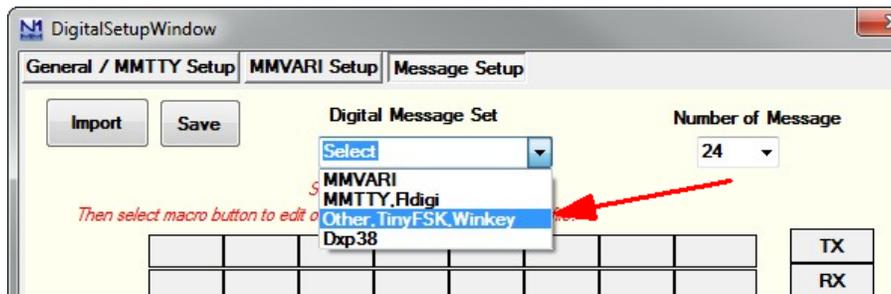
Click on Enable RTTY and then set a delay in both Lead Time and Tail Time; 40 mSec in each is a good starting point. Now click on the Configurer Digital Modes tab and in the left most drop down, select Winkey. Now close the Configurer and go back to the main N1MM+ window. From there we will open up the digital interface window:



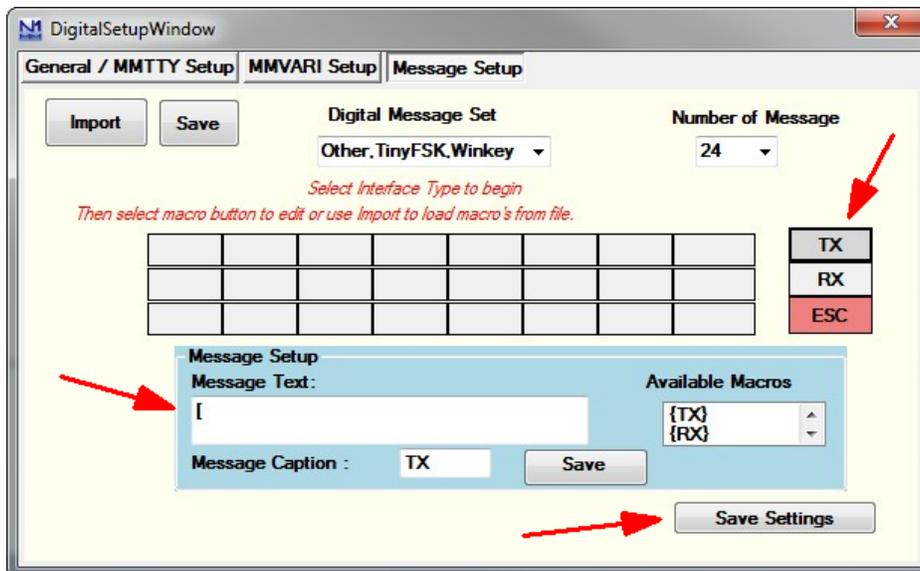
There are several things to set up in the digital interface window, first we will setup the macro keys for Tx, Rx, and ESC. Carefully follow the next instructions.



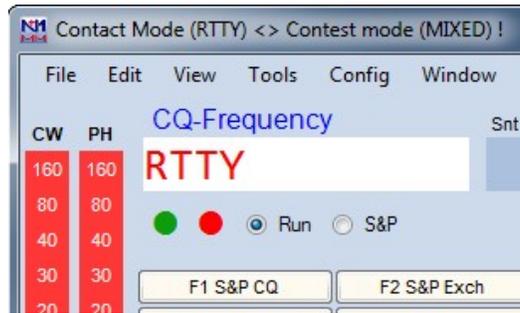
Click *Setup->Settings* to bring up the Digital Setup Window, click on the Message Setup tab and then select "Other, TinyFSK, Winkey" from the drop down menu.



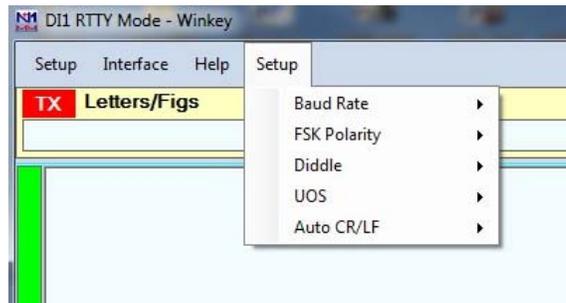
Using the picture below as a guide, click **Tx**, then enter [in the message Text Window followed by **Save Settings**. Repeat this process by entering] for **Rx** and then \ for **ESC**. Note that you have to re-select "Other, TinyFSK, Winkey" each time.



The fastest way to get into RTTY mode is to enter RTTY in the entry box as shown and hit Enter.



Now go back to the Digital Interface window, and select your desired RTTY operational modes:

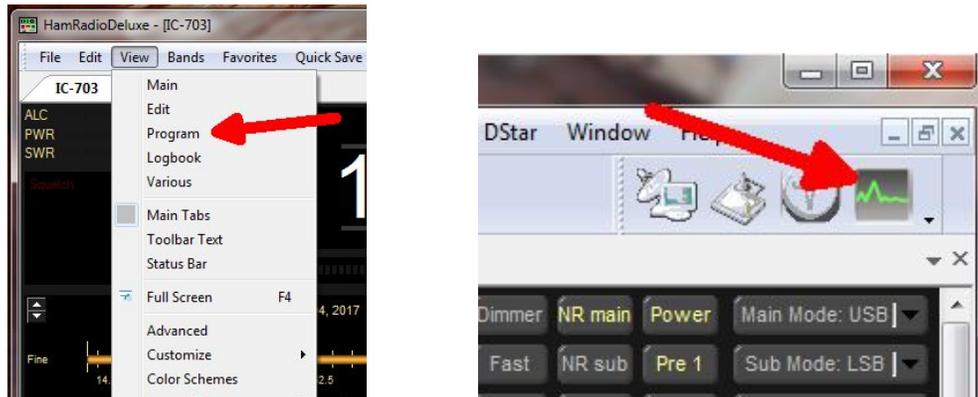


Now you are all set to run N1MM+ in RTTY mode with WKUSB. Entering an Fn macro key will trigger its message and you will see the RTTY string echoed in the Digital Interface Window as they are transmitted. You can enter free test in the Digital Interface Window by first clicking the Tx button, and then typing into the input edit window. When you are done typing, click Rx or enter] and N1MM+ will go back into listen mode. Clicking Esc:Stop in the main window or hitting the ESC key will halt transmission immediately.

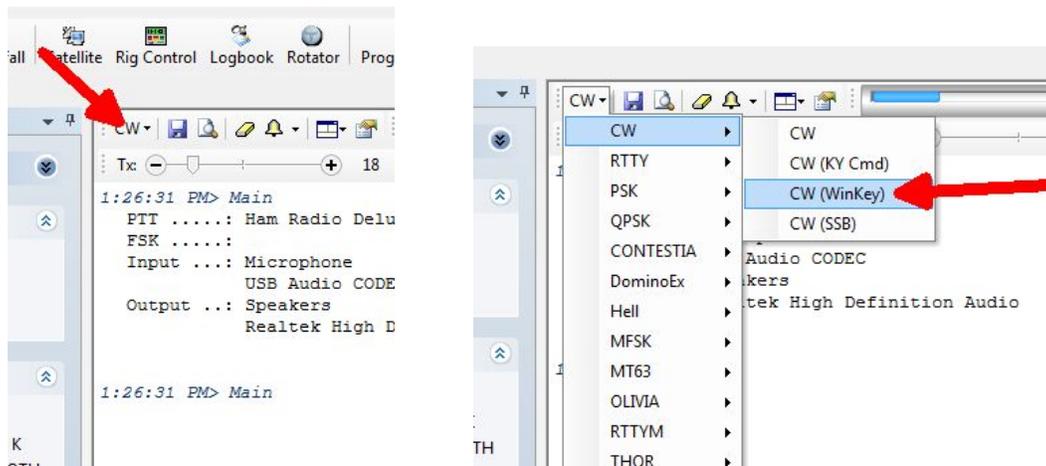
HRD Interface Guide

You will first need to know your WKmini's comport assignment, if you don't know it, follow the WKscan Tutorial on page 4.

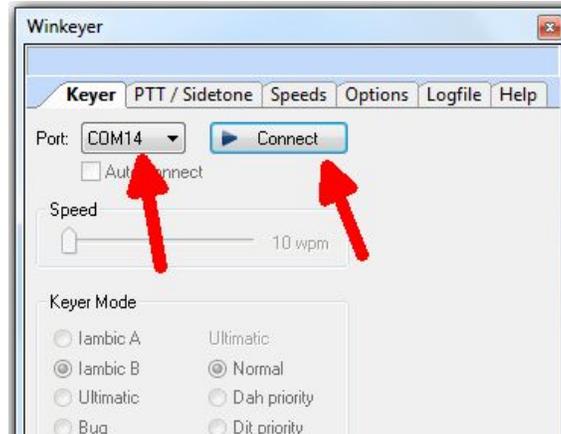
Start HRD and click on the DM780 icon in the upper left corner of the application. If it's not visible, you may have to enable program view as show on the left. (View->Program)



DM780 takes a few moments to start up, once it does, select CW mode from the mode list and then from the CW tab, select CW(Winkey):



This brings up the Winkeyer dialog box which has multiple tabs with a great deal of configuration choices. We will start with the Keyer Tab where we set WK's com port and then click Connect.



Now Connect changes to Connected and items in the dialog box are no longer grayed out. If you click Auto-connect, DM780 will connect to Winkeyer automatically on startup.

A speed slider allows you to change WinKeyer speed while you are in this dialog box for testing. Keyer mode displays a choice of paddle input options. Iambic B is the most often used so that is the default. If you select Ultimatic, three different variants for this mode are enabled

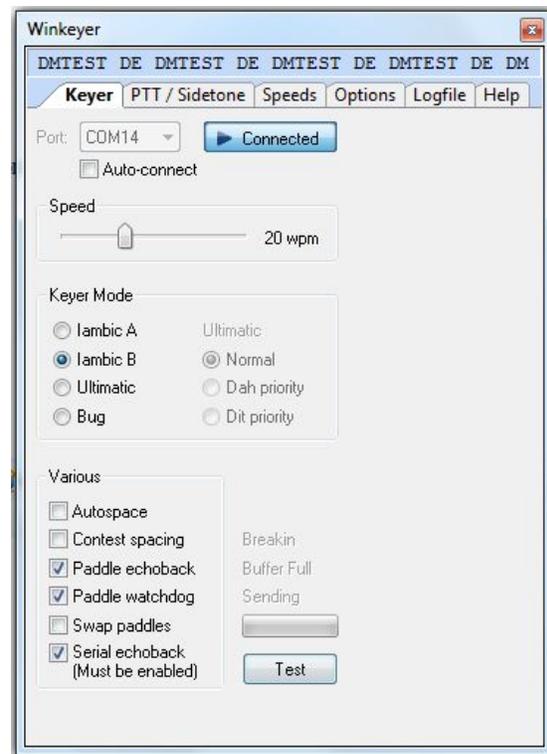
Autospace applies to paddle input only while contest spacing applies to message sending. When contest spacing is on, wordspaces are reduced to 6 dit times vs. standard 7 dit times.

Select paddle echoback and paddle entry will be displayed in with message sending. When paddle watchdog is on, transmit is disabled after 128 consecutive dits or dahs.

Paddle swap allows you to select dits on the right paddle instead of left.

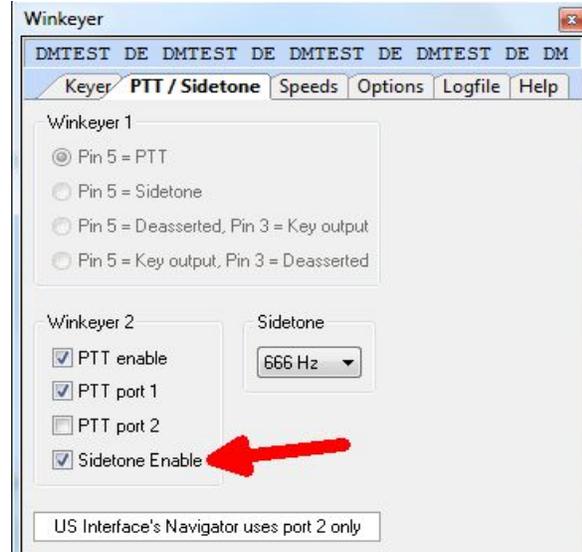
Serial loopback must be on in order for DM780 to keep track of outgoing Morse.

If you click on Test, a canned message will be sent with status displayed.



If you click on the PTT/Sidetone tab a list of hardware options are displayed. Winkeyer1 settings are historical and can be ignored. Even though WKmini uses a WinKeyer 3 IC, the Winkeyer2 settings are the ones we are concerned with. This is where you setup the output port operation. If you want a single key port with PTT, click PTT enable and PTT Port 1. If you want two separate output ports don't click any of these options.

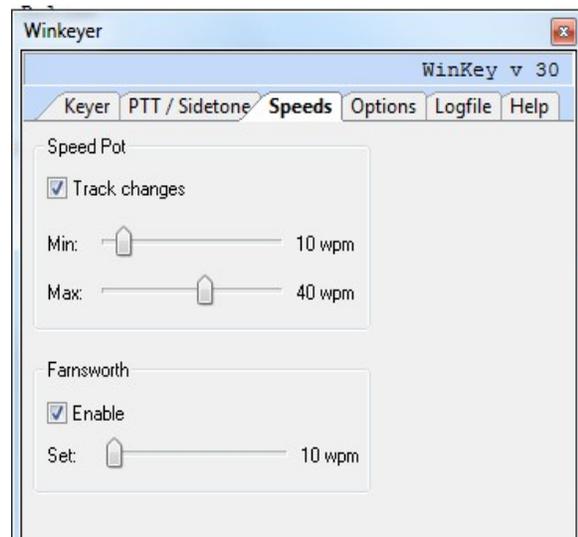
Sidetone frequency is irrelevant for WKmini since it does not have built in sidetone capability.



The Speeds Tab is mostly concerned with speed pot settings. Since WKmini does not have a speed pot, we will unclick track changes.

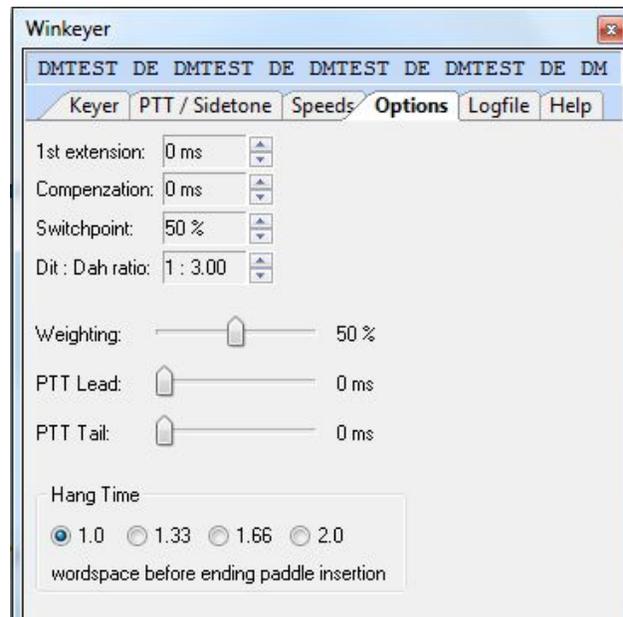
Farnsworth can be enabled if desired but for practical purposes, it is used mostly for code practice and doesn't apply to normal amateur radio exchanges. But it's there if you want it.

Note that the WinKeyer IC version is displayed at the top of this dialog box.



The Options tab contains settings to adjust the timing of the Morse sent by WinKeyer. These settings are discussed in detail in Appendix D of this document.

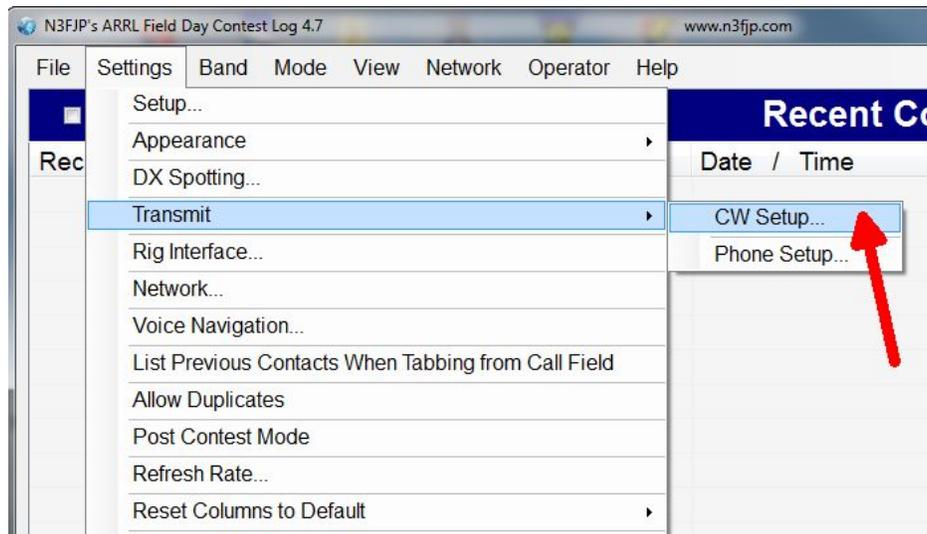
Also listed are settings for PTT timing. PTT turns on before sending starts and shuts off after sending is complete. Lead specifies how long PTT will turn on before keying starts. The Tail delay specifies how long PTT will stay on after keying ends. There is a separate tail delay setting for paddle sending called hang delay. Most operators want PTT to not drop out between letters but also drop out very soon after they stop sending. Hang time sets the delay in wordspace times making it proportional to sending speed.



Now WKmini is ready to work with HRD/DM780. It is a very complex and multi-featured application and I am only able to cover just the WKmini interface portion. It will take some time to get familiar with the application.

N3FJP AC Log Interface Guide

Start AC Log. From the main application page, select the Settings->Transmit->CW Setup tab.



That will bring up the CW Setup Form shown below. A list of the active com ports are shown, click on the one WKmini is assigned to. Next select Winkeyer and then click Configure Winkeyer.

CW Setup Form (Ctrl W for this form or Ctrl > Shift W for the CW Mini form)

Com Port:

- COM1
- COM2
- COM32

Keying Options

None
 RTS
 Winkeyer

DTR

Timing Options

Sleep
 Timer
 Loop

Changing speeds will disable Winkeyer Speed Pot Lock

| | | | |
|--------|------|----------|-------------------------------------|
| WPM | Word | Characte | |
| Faster | More | More | Send F9 Characters: |
| 18 | 0 | 0 | <input type="button" value="Test"/> |
| Slower | Less | Less | |

Loop Sec

| | | | | | | | | |
|----|-----------|-----|----|--|----|------------|-----|-----------------------------------------|
| F1 | TEST K1EL | 2.1 | F4 | | F7 | | F10 | |
| F2 | | | F5 | | F8 | | F11 | |
| F3 | | | F6 | | F9 | VV DE K1EL | F12 | <input type="button" value="Keyboard"/> |

Faster Escape Check
 Show Mini on Startup

[Click for more info on rig to PC CW interfaces.](#)

Description:

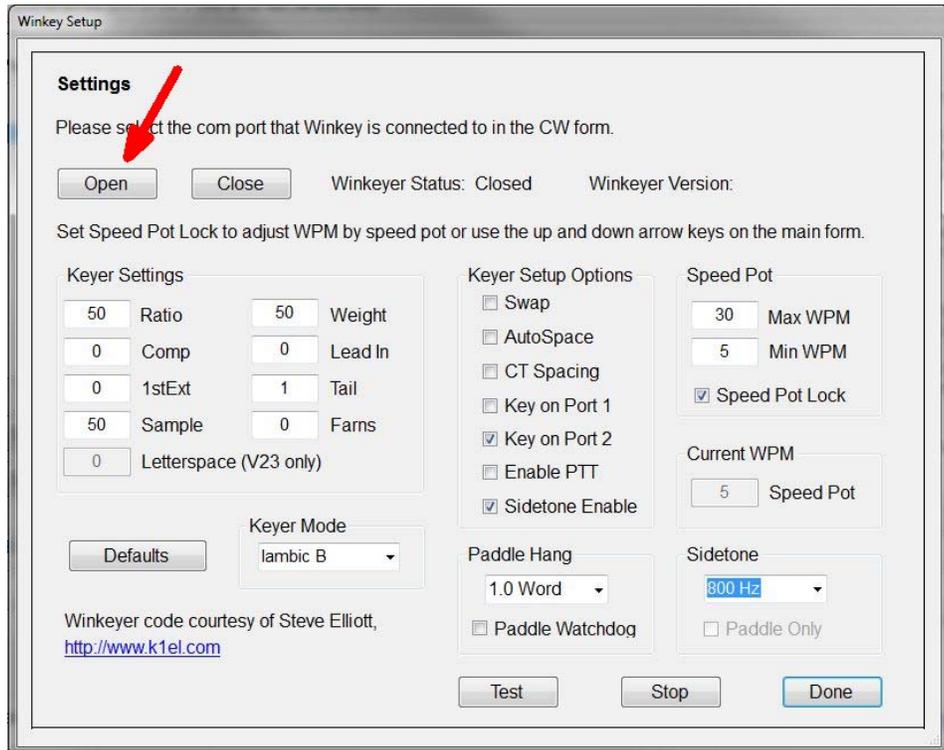
To use computer generated CW, first, you will have to build or purchase an interface to connect from your computer's serial port to your transmitter. Schematics and purchase options are available on my website (www.n3fjp.com).

Most folks only have USB ports now, but in most cases a USB to serial adapter will work fine. You'll find them on line or at your local Best Buy.

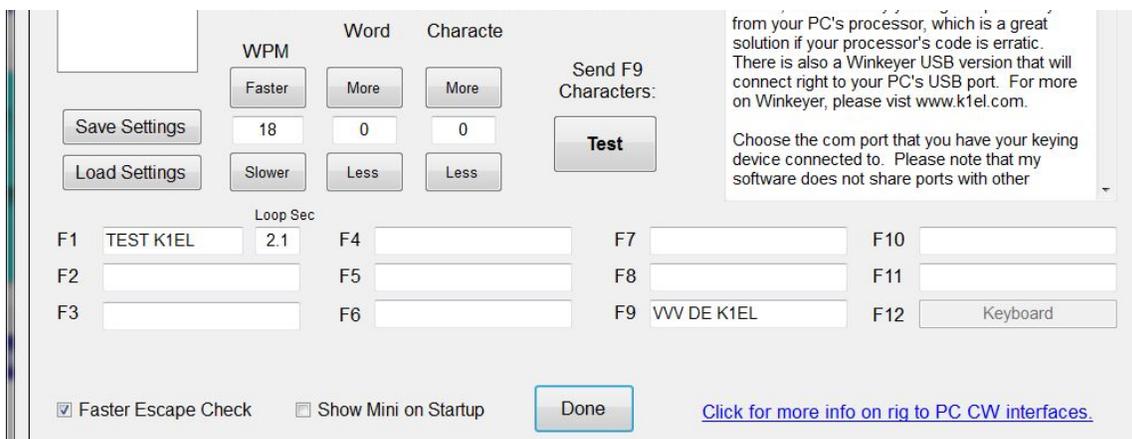
Alternatively, you can select the Winkeyer device, which will key your rig independently from your PC's processor, which is a great solution if your processor's code is erratic. There is also a Winkeyer USB version that will connect right to your PC's USB port. For more on Winkeyer, please visit www.k1el.com.

Choose the com port that you have your keying device connected to. Please note that my software does not share ports with other

This brings up a form that presents the Winkeyer options that can be modified. First thing to do is click Open. That connects AC Log to WKmini, the status will change to Open and WKmini's firmware revision will be displayed. Most of the options should be familiar; any that aren't can be referenced in Appendix D. The only thing to be aware of is that WKmini does not have a sidetone speaker or a speed pot, so those options will not have any effect on WKmini's operation. Sidetone enable has to be turned on in order for WKmini's debug LED to work.



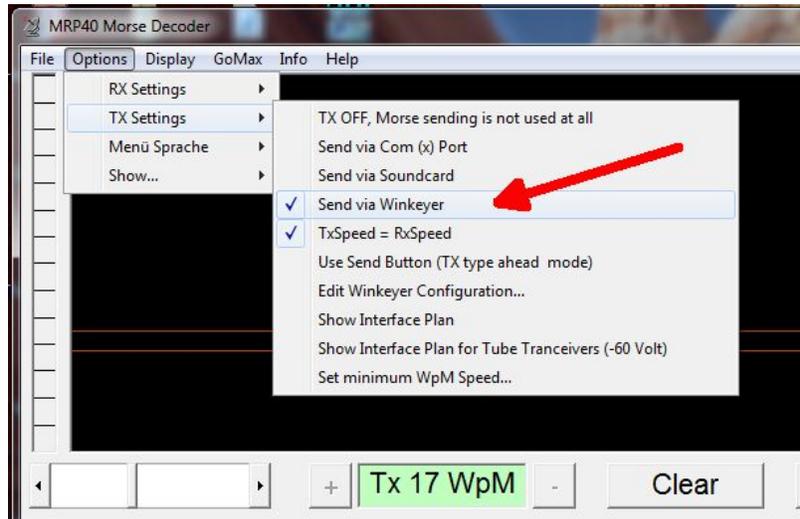
Click Done and you will return to the CW Setup form, you can enter messages in the F1 through F11 slots. F1 is set up as a repeating message with adjustable loop delay. When you click Test, the message in slot F9 will be sent. Click Save Settings to save you configuration and then click Done to return to AC Log where you can start using WKmini on air.



MRP40 Interface Guide

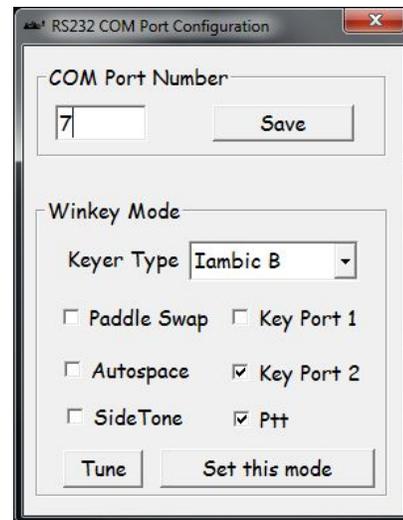
This is one of my favorite applications to use with WKmini. It's not a logging program but instead it's a conversational, Morse terminal. It has an excellent CW reader that uses your PC's sound card to perform DSP tone detection. CW reception is displayed in a multi-channel, sideways, waterfall display. The best part is that it includes a WinKeyer interface allowing you can send as well as receive. It's very easy to hook up WKmini. From the main page, click:

Options->Tx Settings->Send via Winkeyer



A simple WinKeyer configuration dialog box is displayed. Enter the com port number as reported by WKscan and click Save. WKmini is now active and you can set up configuration options. Select paddle Keyer Type, Iambic B is most commonly used. Then if you want to use a single key port with PTT select Key Port 1 and Ptt, else select either Key Port 1 or Key Port 2 leaving Ptt off. Enable sidetone so that WKmini's debug LED will work.

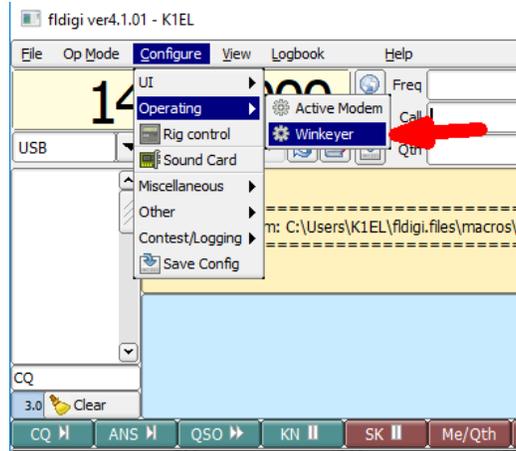
Click and hold Tune and WKmini will key it's output until you release. To get back to the main application, click on the X in the upper left corner of the dialog box.



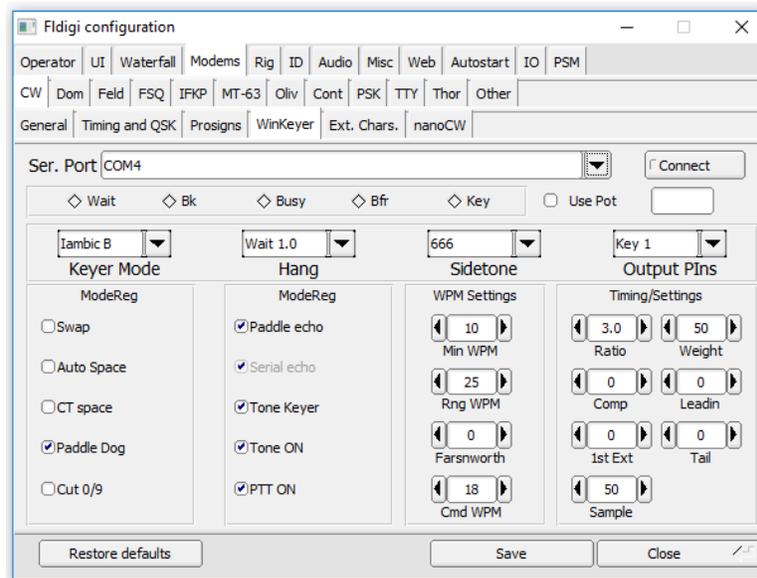
Now simply type in the lower text window and Morse will be sent by WKmini. I recommend setting TxSpeed=RxSpeed in TX Settings, that way if you copy someone you want to work, your transmit speed will be set to match theirs automatically. I also like to adjust the text size to display more lines. That option is under Display.

FLdigi Interface Guide

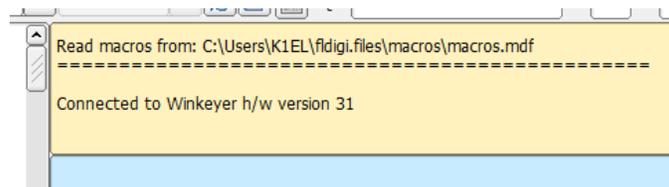
This is another very powerful amateur radio application. It's quite easy to set FLdigi up to use Winkeyer in either CW or RTTY mode. If you don't know the comport that your WKUSB is connected to, use our WKscan app. To start, we need to setup WinKeyer configuration:



Selecting WinKeyer will bring up the following configuration dialog box,



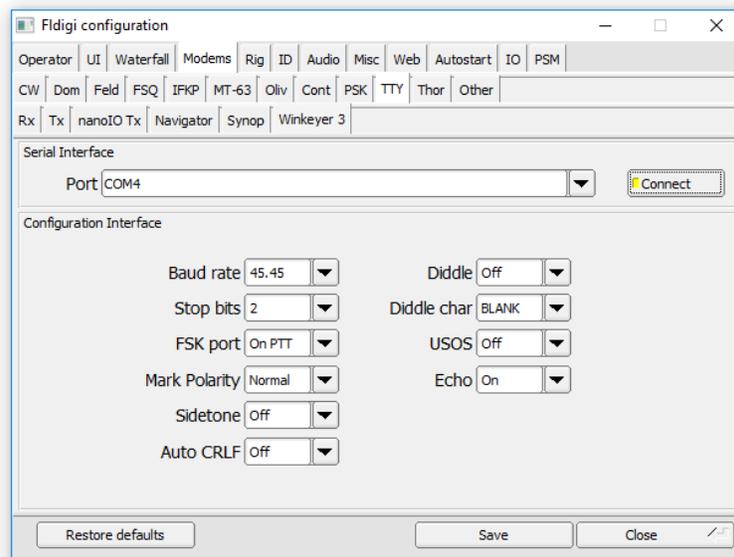
Enter WKUSB's com port and the review the default Winkeyer settings. For the most part, the defaults will suffice but you may want to change paddle swap or use a different keyer mode. Click on Connect and FLdigi will open up the WK interface. The following message will displayed on the main application window to indicate that the interface is open and ready to operate:



Once Winkeyer has been selected and configured, FLdigi will open it automatically when it starts next time so you won't have to go through this setup procedure every time.

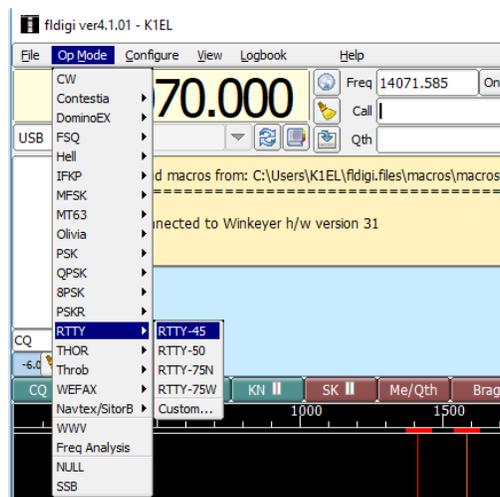
Running WK3 with FLdigi in RTTY mode

If you happen to have a WKUSB that uses a WK 3.1 controller IC, you can run it with FLdigi in RTTY mode. Bring up the configuration dialog and select the TTY tab and the Winkeyer 3 tab:



All of WK's RTTY settings are presented here. Select the correct com port and adjust the defaults as required then click connect.

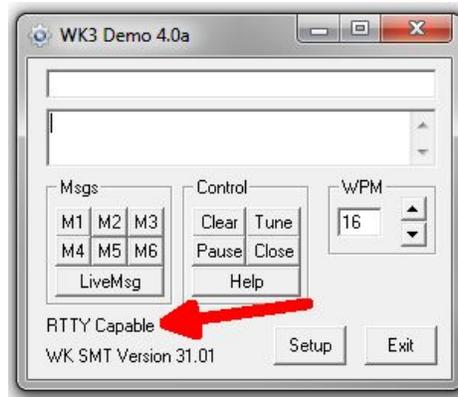
To set RTTY mode in the main operating window, simply choose Op Mode followed by RTTY at the speed you want to run at. WK RTTY operation, like WK CW, is well integrated into FLdigi and once set up you should be good to go. FLdigi offers excellent documentation on how to run their application and that is the best place to go if you need help getting started.



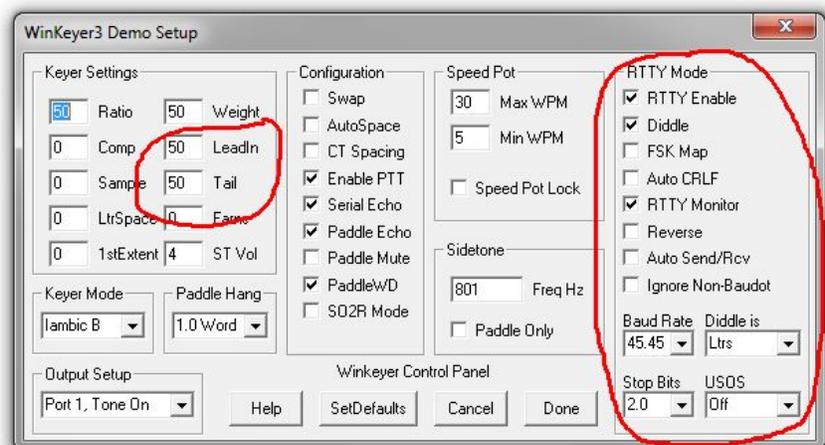
Running WK3demo Version 4.0 in RTTY mode

An easy way to test out WKUSB 3.1 in RTTY mode interface is to run it with WK3demo 4.0. Since WKdemo only does RTTY transmit, you will need a means to read and decode received RTTY signals. You could use the popular apps MTTY, FLdigi, or some other PC based RTTY decoder. Many amateur radio transceivers have RTTY decode built in which would make this very easy.

WK3demo 4.0 has a new RTTY menu that is enabled when you connect with a keyer that has a version 3.1 WK IC. This is what you should see when you open WKUSB and it has WK3.1:



Now if you click on setup you will be able to access the RTTY controls:

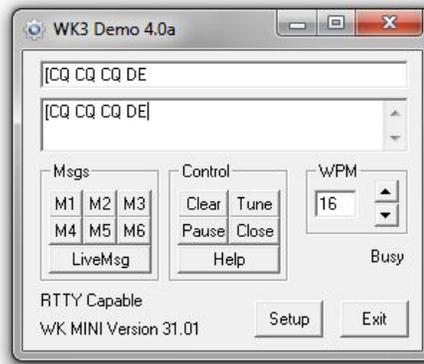


These are the recommended settings that will satisfy most setups. First of all, select RTTY Enable, Diddle, and RTTY monitor. The only other settings that you may have to adjust in this group are FSK Map and Reverse. FSKMAP will swap the PTT and KEY Outputs. Normally WK's KEY out goes to your transceiver's KEY input and PTT goes to the transceiver's FSK input. FSKMAP will swap these two outputs. We will run in direct FSK mode, not AFSK so we connect to the digital FSK input not a mic input. Selecting RTTY monitor will echo letters to the transmit window as they are being sent.

Be sure to enter values for PTT Lead In and Tail. This will insure that the beginning and end of your RTTY transmissions do not get chopped.

Click on Done to get back to the main menu. Once RTTY enable is set, WK will act quite differently. It will not send CW, only RTTY. Let's send a test CQ, first we enter a [in the entry

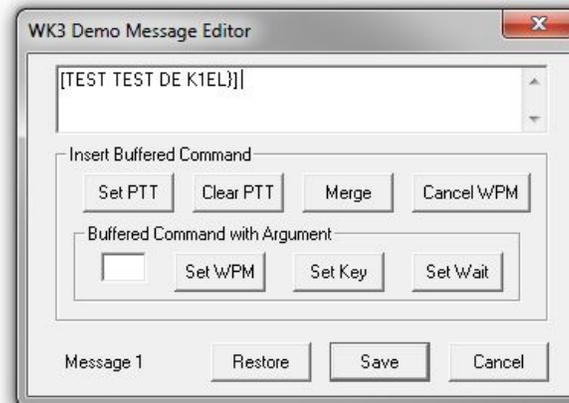
window. This will key your transceiver and assert the FSK output to the mark state. Since we selected Diddle on, it will start sending blank characters until we actually enter some text.



As we enter text it will be echoed to the upper window after it is sent. When you are finished, enter a trailing] which will unkey the transceiver so you can listen for a reply. It may be helpful to monitor your transmission to be sure you are operating correctly. If KEY and PTT are swapped you will hear the transmission clipped on and off since it is being keyed by FSK. When it's set up correctly the transmit output will be on all the time and be frequency shift keyed by FSK. Note that it might be necessary to toggle Reverse. The best way to tell is to listen to other stations and set Reverse to match their shift polarity.

If you want to insert a CRLF sequence enter the } character. Two other actions you may need are Pause and Clear, both of these work by clicking the appropriate button. Tune will key the transceiver with no FSK.

Now let's enter an RTTY message. On the WK3demo main dialog box, move the cursor over a message button and right click on it. The following dialog box will be presented:



We start an RTTY message with [and end it with]. Note that a CRLF token } is included at the end just before the] so that a CRLF will be sent at the end of the message. Save the message and now when you left click the message button your message will be transmitted.

Interfacing WKmini to other applications

The basic procedure will be the same for other apps that use Winkeyer. Some apps first require a port mapping procedure like N1MM or HRD while others take you directly to the Winkeyer interface dialog box where you select the port and set up the configuration. If you run into any problems, first contact the application developer since they will be able to offer the best help. If that is not possible post an inquiry on the K1EL Yahoo Group. Odds are that there is someone else in the group who has used the app and they can give you first hand advice. If no one picks up the trail, we will jump in and do what we can. Note that we have a large list of apps that support WinKeyer on our website.

Revision History

| | |
|----------|------------------------------------------|
| 12-20-17 | - Rev 1.0 Initial Release |
| 02-03-18 | - Rev 1.1 Minor error corrections |
| 07-10-18 | - Rev 1.2 Minor error corrections |
| 03-26-19 | - Rev 1.3 Added WM 3.1 changes/additions |

Appendix A - WK Version 3.1 Firmware

Version 3.1 is a major upgrade in functionality for WinKeyer based CW keyers. Besides many bug fixes, several new features have been added. The most notable is the ability to generate an accurately timed RTTY serial transmit stream. WKmini RTTY is generated in host mode with an application running on a PC.

RTTY Feature Set

Winkeyer supports standard 5 level Baudot transmission. Four operating speeds are supported, 45.45, 50, 75, and 100 baud with either 1.5 or 2 stop bits. Optional diddle generation, automatic end of line carriage return/line feed and UOS transmit are supported.

RTTY Host Mode

The RTTY API is described in the latest WK3 Datasheet. It is fairly easy for an application developer to add RTTY capability to an existing application that already supports WK since all of the basic interface procedures are the same. That includes buffer full handshaking, pause, immediate abort, and transmission of ASCII characters in Baudot (instead of Morse) with automatic FIGS/LTRS generation.

RTTY Standalone Mode

Unlike WKUSB, WKmini is not able to run in standalone mode so RTTY standalone is not supported.

Using WK3demo in RTTY mode

This is covered in detail starting on Page 31.

Other Changes for Version 3.1

K1EL WK Applications

There are two new app releases to support WK 3.1.

WK3tools 5.0 is required for upgrading existing WK3 keyers to version 3.1.

WK3demo 4.0 has upgrades to support RTTY mode. You can enter messages and configure host mode RTTY transmit operation.

Sidetone Volume Command

A somewhat limited volume control has been added to WK3. As it turns out that only when the level is set to either 1 or 2 that the level actually decreases noticeably.

First Extent Command

This command was removed from Version 3.0 but has been reinstated for Version 3.1

WK API changes

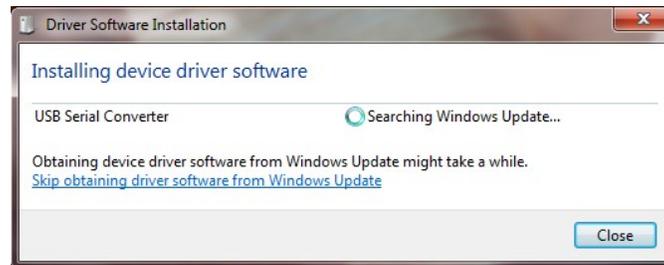
Besides the new RTTY commands several new commands were added to load and unload the RTTY, Volume and X2MODE values.

Appendix B - Installing the FTDI USB Driver for WKmini

As mentioned, don't plug WKmini in yet, we have to deal with the USB driver first.

For Windows XP or Vista users, we have a special procedure for you on page XX.

For Windows 7 and newer, the procedure is usually easy. Just plug *WKmini* into your PC's USB port and it will be detected. Then a dialog similar to the one below will be displayed as Windows tries to locate a driver.



If you are connected to the Internet, the search will extend to the web and it can take a long time to complete, a 15 minute wait is not unheard of. If you don't want to wait, or are not connected to the Internet, preloading the FTDI driver is recommended.

Preloading the FTDI USB Driver

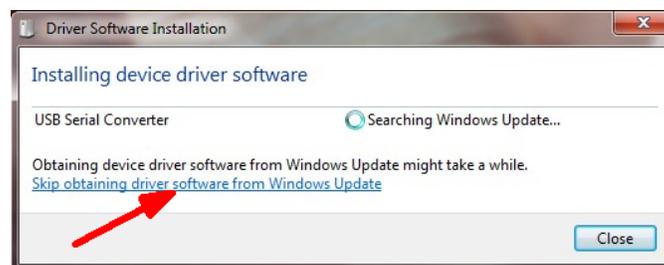
Follow this link to navigate to the FTDI Driver webpage: <http://www.ftdichip.com/Drivers/VCP.htm> and download the setup executable zip file:

Currently Supported VCP Drivers:

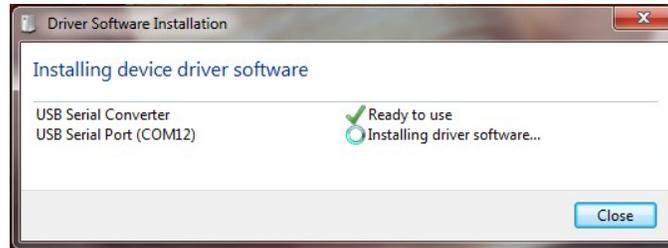
| Operating System | Release Date | Processor Architecture | | | | | | | Comments |
|------------------|--------------|------------------------|--------------|-----|-----|--------|--------|-----|-------------------------------------------------------------------------------------------------------------------------------------|
| | | x86 (32-bit) | x64 (64-bit) | PPC | ARM | MIPSII | MIPSIV | SH4 | |
| Windows* | 2017-08-30 | 2.12.28 | 2.12.28 | - | - | - | - | - | WHQL Certified. Includes VCP and D2XX. Available as a setup executable Please read the Release Notes and Installation Guides. |

Click here

Create a directory on your PC and extract the setup zip file into it. Then run the setup.exe file and the latest driver will be installed on your PC. Now plug *WKmini* in and this time you can cancel the driver search:



It might take a few minutes for the Windows Update Search to end. Windows will then look on your local hard drive for the driver and will install it automatically. Usually two separate drivers are installed, one for a USB serial converter and one for a USB com port.



Sometimes the com port assignment will be displayed so watch for that. If you miss it, no big deal, we can determine it later using the WKscan app. At this point you can return to page 2 and start testing WKmini.

USB Driver Installation for Windows XP or Vista

FTDI no longer officially supports the Windows XP operating system. Luckily, they still offer a driver that works with XP, posted on their website: <http://www.ftdichip.com/Drivers/VCP.htm>

No Longer Supported:

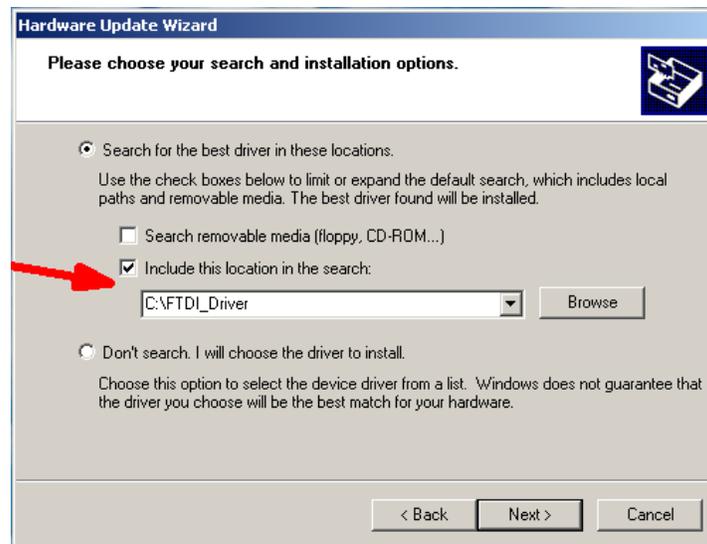
| Operating System | Release Date | Processor Architecture | | | | |
|---------------------------------------------|--------------|------------------------|--------------|-----|-----|------|
| | | x86 (32-bit) | x64 (64-bit) | PPC | ARM | MIPS |
| Previous Windows Release | 2017-03-10 | 2.12.26 | 2.12.26 | | | |
| Windows XP, Vista, Server 2003, Server 2008 | 2012-04-13 | 2.08.24 | 2.08.24 | | - | - |

The table shown is located on the lower part of the page. Pick the appropriate version for your OS, 32 or 64 bit, and download the driver zip file.

Now make an empty directory on your PC and extract the driver into that directory. I usually use a directory called FTDIdriver at the root of the C drive (C:\FTDIdriver) Now plug in WKmini and you should get this dialog:



Select 'Install from a list or specific location' and click Next



On the next dialog box, enter the path to the directory where you extracted the driver files. In my example it's C:\FTDIdriver. Now click next.

It can take a few minutes to install the drivers and it may be divided into two separate processes, one for the USB Serial Converter and one for the USB Serial Port.

Hopefully the install will finish without errors and the drivers will be ready to use. At this point you can return to page 2 and continue testing *WKmini*.

FTDI provides detailed installation guides for download. Links can be found on the driver webpage. FTDI's installation guides are excellent and covers most all issues you may run into if the drivers don't install.

Manual Driver Installation Win 8 and newer operating systems

In most cases, current version PCs ship with the FTDI USB driver pre-installed. This means you do not need to manually install a USB driver. Alternatively, if you are connected to the internet when you plug WKmini in, Windows will go out and find the driver on its own. However this can take a very long time. If you want to skip that, you can manually install the driver as follows.

Use the very latest driver version (it may be even newer than the one shown below):

Currently Supported VCP Drivers:

| Operating System | Release Date | Processor Architecture | | | | | |
|------------------|--------------|------------------------|--------------|-----|-----|--------|-------|
| | | x86 (32-bit) | x64 (64-bit) | PPC | ARM | MIPSII | MIPSI |
| Windows* | 2017-08-30 | 2.12.28 | 2.12.28 | - | - | - | - |
| | | | | | | | |

Pick the appropriate version for your OS, 32 or 64 bit, and download the driver zip file.

Appendix C - Determine WKmini's COM port when you don't have WKscan:

Left click on Start and then right click on *MyComputer*:



Click on *Properties*:



Click on *Hardware*:



Click on *Device Manager*:



Then Double Click on *Ports (COM & LPT)* to expand the port list:

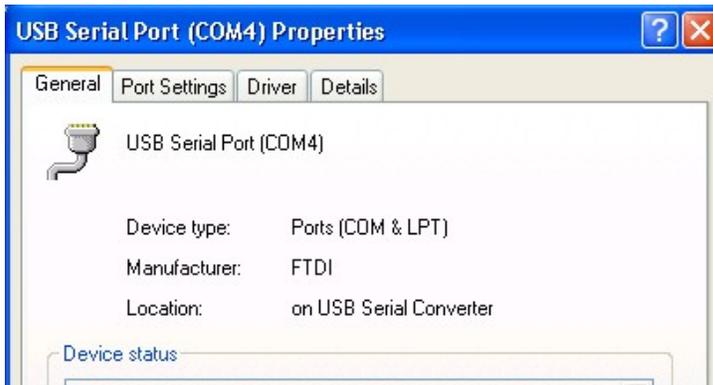


The port labeled *USB Serial Port (COMx)* is the WKmini port:

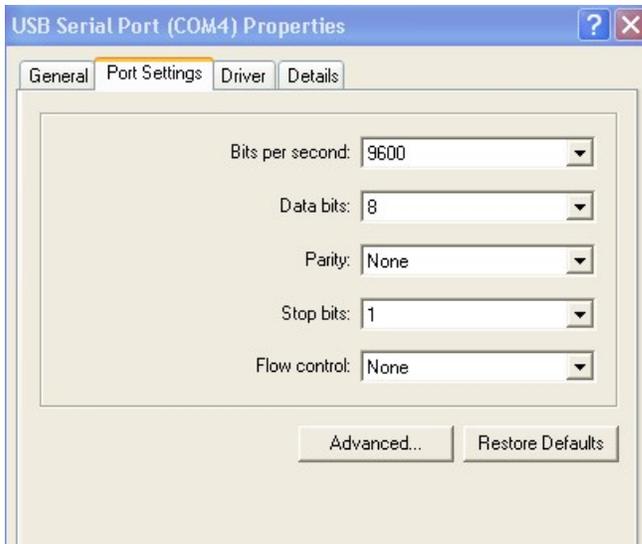


If you want to change the port assignment, double click on USB Serial Port and

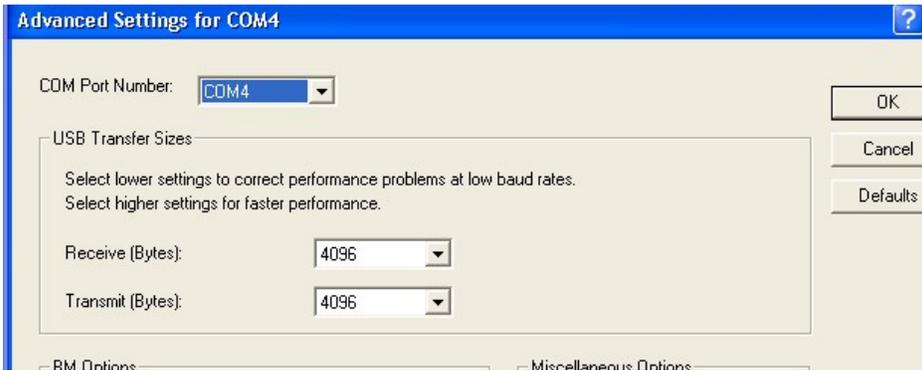
Then Click on *Port Settings*:



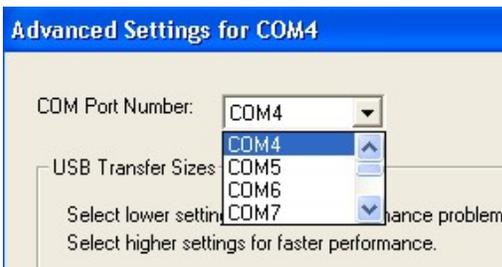
Click on *Advanced*:



Finally you will see the port selection control:



By clicking on the down arrow to the right of the COM Port Number List Box you will get a list of available com ports. Click on the one you want and you are finished.

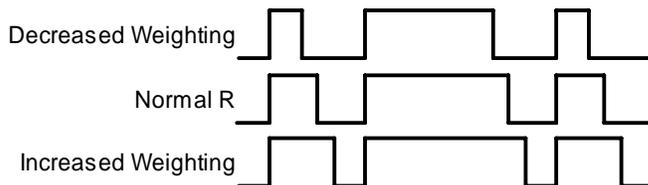


In the event you installed a new USB com port or reassigned a different port the com port you last used may not be available. In that case you will need to map WKmini to a new com port.

Appendix D – WinKeyer Settings Defined

Weighting

A proportional amount is either added or subtracted from the length of all transmitted dits and dahs. A value of 50 selects no weighting adjustment. Values less than 50 reduce weighting and values greater than 50 increase weighting. Note that weighting does not affect sending speed because any increase in keyed time is subtracted from spacing time. Reduction in weighting results in a thinner sounding keying while increased weighting results in a heavier sound. Since weighting tracks speed, a given weighting will sound the same at all speeds.



| |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>WEIGHTED DIT TIME = $(((\text{WEIGHT} \times 50) / 50) \times \text{DIT TIME})$</p> <p>WEIGHTED DAH TIME = $(((\text{WEIGHT} \times 50) / 50) \times \text{DAH TIME})$</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

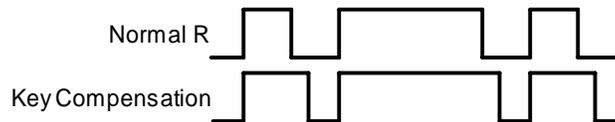
Weighting Example

Keying Compensation

A fixed amount is added to the length of all dits and dahs. QSK keying on modern transceivers can cause shortening of dit and dah elements which is especially noticeable at high speeds. WK3 can compensate for the shortening by adding a uniform length to each dit and dah element. The

adjustments are made in one-millisecond steps and the maximum adjustment allowed is 250 mSecs. Key compensation is very similar to Weighting in that an adjustment added to a dit or dah is then subtracted from the spacing between them so the overall speed is not changed. The difference between weighting and compensation is that compensation is independent of speed, so if 10 msec of key compensation is selected, 10 msec will be always be added regardless of speed. Be aware that at high speeds, large values of key compensation can reduce inter-element space to zero.

In the following figure, an R is represented without key compensation (nn = 00) and with 12 mSecs of key compensation (nn = 12). Note that each dit or dah starts at the same place even with compensation on.



Keying Compensation Example

Letterspace

Specifies an additional letterspace, in percentage of a wordspace, to be applied between all letters. Minimum adjustment is 0% while maximum addition is 62%.

Paddle Sample Delay

This setting controls when WK3 will start looking for a new paddle press after sensing the current one. If there is not enough delay, the keyer could send unwanted dits or dahs. If there is too much delay, the operator is held back because they can't paddle ahead. The default value is one dit time (50) and is adjustable in percent of a dit time. A setting of 55 – 65 may seem familiar to Accukeyer users. If the paddle sensitivity is set to zero, dit and dah paddle memory is disabled.

The delay is calculated with this formula:

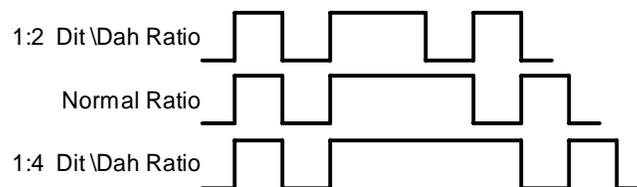
$$\text{DELAY_TIME} = (\text{SWITCHPOINT} \times \text{DIT_TIME}) / 50 \text{ where Switchpoint is a value between 10 and 90.}$$

Dit/Dah Ratio

The ratio between the length of a dit vs. the length of a dah is controlled by this setting. The agreed upon standard s ratio is 1:3 (dit:dah), a dah is three times longer than a dit.

The formula to determine dah/dit ratio is: **DAH/DIT = 3*(nn/50)**

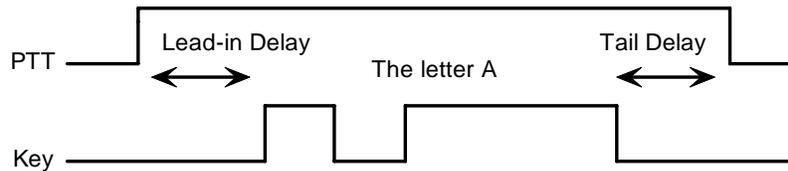
A value of 50 selects 1:3, a value of 33 would select 1:2, and a value of 66 would select 1:4. Any value other than 50 causes distortion of the Morse waveform. Some ops use this option to make their CW sound less "machine like" but a little goes a long way !



Three ratio settings for the letter R

PTT Lead and Tail Times

WK3's PTT output is used to control an accessory device in addition to normal CW transmitter keying. In most cases this device is a linear amplifier but it could be an antenna changeover relay or PTT input of a transmitter. In all of these cases there are delay requirements that must be met to insure that the accessory device is switched on before transmission begins and is held on until transmission completes. This prevents damage to the accessory device due to hot switching. WK3 provides three independent PTT delays to meet this requirement.



PTT Lead-in and Tail Example

The first of these is Lead-In delay which initiates a keying event. When transmit starts, PTT will be asserted first and then, after the Lead-In delay expires, the key output will be asserted. Lead-In can be set to a value from 0 to 250 milliseconds in 10 millisecond increments.

Tail Delay specifies the amount of time PTT will be released after Key is released. Like Lead-In delay, it is adjustable from 0 to 250 milliseconds in 10 millisecond increments.

In setting tail delay there are two considerations, the first is to prevent hot switching and the second is to add delay to fill in between letters sent by paddle. Most ops don't want their amplifier to be switched in and out between letters. An issue arises in that Tail delay is not solely proportional to sending speed. This is problematic when someone sets a comfortable tail delay at a slow speed and then increases the speed to a much faster rate. At the faster rate, the tail delay will hold too long after keying stops. Alternatively, if a comfortable delay is set at a faster WPM rate, PTT will then drop out between letters at a slower speed. To solve this problem, paddle tail delay is governed by a separate parameter called hang time.

Hang Time is adjustable in four steps and is measured only in wordspace and dit times rather than fixed milliseconds. This means that the delay will accurately track sending speed. In other words it will decrease automatically as sending speed increases and vice versa. The four settings for Hang Time are:

- HangTime = 0: wait 1 wordspace + 1 dit before ending paddle insertion
- HangTime = 1: wait 1 wordspace + 2 dits before ending paddle insertion
- HangTime = 2: wait 1 wordspace + 4 dits before ending paddle insertion
- HangTime = 3: wait 1 wordspace + 8 dits before ending paddle insertion

To sum up, Tail delay should be set to provide just enough delay to prevent hot switching for machine sent CW while hang delay is designed to hold PTT between letters independent of sending speed. PTT will automatically be held between machine sent letters and words but will drop out quickly after a message has been sent as long as there is no trailing wordspace.

Paddle Watchdog

A paddle watchdog counter disables keying output after 128 consecutive dits or dahs. WK3 assumes that this is an unintended condition. This watchdog can be disabled if desired.

Paddle Swap

This bit swaps the operation of the dit and dah paddles. It's used mostly to accommodate left handed ops or to correct for a miswire in a keying cable.

Paddle Only Sidetone

Local sidetone is generated only for paddle generated Morse, sidetone is inactive when text is sent from the host. Since WKmini does not have local sidetone this command is ignored.

Paddle Mute

This locks out the paddle so that it will not generate Morse when pressed.

Paddle Echo back

When enabled, all characters entered on the paddles will be echoed back to the host.

Serial Echoback

Serial Echo Back tells WK3 to echo each Morse letter that originated at the host. It can be used to allow a host application to stay exactly in sync with Morse letters as they are sent. Each letter is sent to the host after it has been sent in Morse. This permits the host to track WK3's progress in real time. Note that buffered commands and their parameters are not echoed back to the host.

Autospace

Autospace works in the following way: If you pause for more than one dit time between a dit or dah, WK3 will interpret this as a letter-space and will not send the next dit or dah until the full letter-space time has been met. The normal letter-space is 3 dit spaces. WK3 has a paddle event memory so that you can enter dits or dahs during the inter-letter space and WK3 will send them as they were entered. With a little practice, autospace can help you to send better.

Contest Spacing

When contest spacing is enabled, wordspace time is reduced by one dit. Instead of seven dits per wordspace, contest spacing selects six dits per wordspace. The idea is that exchanges can be sent faster with reduced space between words.

Paddle Keyer Modes

There are six different keying modes supported by WK3: Iambic mode A, Iambic mode B, Straight Key/Bug, Ultimatic, Dit priority mode, and Dah priority mode. In either Iambic mode, alternating dits and dahs are sent while both paddles are held closed. In mode B an extra alternate dit or dah is sent after both paddles are released. In straight key/bug mode a dah paddle press will key the transmitter for as long as the paddle is pressed and dits will be generated automatically when the dit paddle is pressed. In Ultimatic mode, when both paddles are pressed, the keyer will send a continuous stream of whichever paddle was last pressed. Hold dah then press dit->sends dits. Hold dit then press dah -> dahs are sent. Ultimatic dit and dah priority mode will generate dits and dahs automatically in response to single paddle presses, but when both paddles are pressed either dit or dah has priority.

Speed Pot Lock (not applicable to WKmini)

This tells WK to ignore host speed control and to instead lock the current sending speed to the speed pot.

Speed Pot Minimum and Maximum

The limits for the speed pot can be specified. The minimum must be greater or equal to 5 while the maximum must be less than or equal to 99 WPM. This allows the speed pot to be tailored to the user's needs.

Sidetone Enable

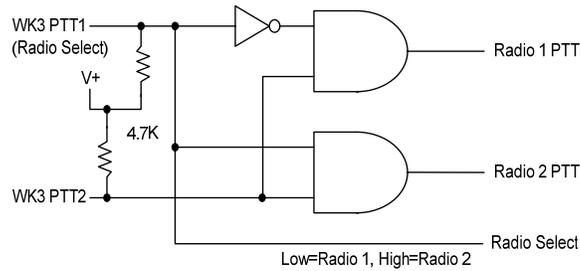
Sidetone can be turned on or off. Local sidetone is not needed if the user is going to use the sidetone generated by the transceiver they are using. This is the case in the majority of setups. That's why WKmini does not have a built in sidetone speaker. The enable control is used instead to enable or disable the debug keying LED.

Sidetone Frequency (not applicable to WKmini)

WK3's local sidetone frequency is adjustable in fine increments from about 500Hz to 4000Hz

SO2R Enable (not applicable to WKmini)

This is a special mode that is only applicable to WK keyers that have two keying outputs, each with a dedicated PTT output. WK3 provides a basic SO2R capability. When SO2R mode is enabled, PTT operation is modified. PTT1 becomes a radio select output while PTT2 becomes a common PTT output for both ports. To utilize PTT for two radio setups requires additional gating as illustrated below:



The PTT outputs on WKMINI are open collector so pull up resistors are required to generate logic high and low conditions. PTT1 will be logic zero when radio 1 is selected and logic one when radio 2 is selected.

Topical Index

| | |
|------------------------------------------|----|
| Introduction | 2 |
| Introduction | 2 |
| What is WinKeyer and why do I need it ? | 2 |
| WKmini Features | 2 |
| Product Warranty, Support, and Liability | 2 |
| Let's Get Started ! | 3 |
| USB drivers | 3 |
| WKmini Connections | 3 |
| Let's actually run WKmini in CW mode ! | 5 |
| WKscan Tutorial | 5 |
| WKdemo tutorial | 5 |
| WK3tools application | 7 |
| Connecting WKmini to your station | 7 |
| Keying cable details (also see page XX) | 7 |
| A cheap path to a keying cable | 8 |
| Troubleshooting hints | 9 |
| Theory of Operation | 10 |
| Paddle Input Priority | 11 |
| USB Sense | 11 |
| SO2R Mode | 11 |
| Serial Baud Rate | 11 |
| Reset WKmini/Restore Factory Defaults | 11 |
| ASCII Code Assignments, Prosign Mapping | 13 |
| N1MM Interfacing | 14 |
| Running WK 3.1 with N1MM+ in RTTY mode | 19 |
| HRD Interface Guide | 22 |
| N3FJP AC Log Interface Guide | 26 |
| MRP40 Interface Guide | 28 |
| FLdigi Interface Guide | 29 |
| Running WK3 with FLdigi in RTTY mode | 30 |
| Running WK3demo Version 4.0 in RTTY mode | 31 |
| Interfacing WKmini to other applications | 33 |
| Revision History | 33 |
| Appendix A - WK Version 3.1 Firmware | 34 |
| RTTY Feature Set | 34 |
| RTTY Host Mode | 34 |
| RTTY Standalone Mode | 34 |

| | |
|-----------------------------------------------------------------------------|-----------|
| Using WK3demo in RTTY mode | 34 |
| Other Changes for Version 3.1 | 34 |
| K1EL WK Applications | 34 |
| Appendix B - Installing the FTDI USB Driver for WKmini | 35 |
| Preloading the FTDI USB Driver | 35 |
| USB Driver Installation for Windows XP or Vista | 36 |
| Manual Driver Installation Win 8 and newer operating systems | 38 |
| Appendix C - Determine WKmini's COM port when you don't have WKscan: | 39 |
| Appendix D – WinKeyer Settings Defined | 42 |
| Weighting | 42 |
| Keying Compensation | 42 |
| Letterspace | 43 |
| Paddle Sample Delay | 43 |
| Dit/Dah Ratio | 43 |
| PTT Lead and Tail Times | 44 |
| Paddle Watchdog | 44 |
| Paddle Swap | 44 |
| Paddle Only Sidetone | 44 |
| Paddle Mute | 45 |
| Paddle Echo back | 45 |
| Serial Echoback | 45 |
| Autospace | 45 |
| Contest Spacing | 45 |
| Paddle Keyer Modes | 45 |
| Speed Pot Lock (not applicable to WKmini) | 45 |
| Speed Pot Minimum and Maximum | 45 |
| Sidetone Enable | 45 |
| Sidetone Frequency (not applicable to WKmini) | 45 |
| SO2R Enable (not applicable to WKmini) | 46 |
| Topical Index | 47 |