

# WKUSB

*with*  
*WinKeyer3 IC*

## User Manual



**K1EL**  
HAM RADIO KITS

[www.k1el.com](http://www.k1el.com)

**For Rev C WKUSB**

Version 1.4

## Introduction

This document will describe WKUSB and its various interfaces. All the information you will need to connect WKUSB 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, 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 WKUSB, 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. WKUSB 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.

## How do I determine if my WKUSB is a Rev C ?

All Rev C WKUSBs have a black label on the bottom which identifies it:



There are two differences between Rev C and previous versions:

1. Rev C uses a CH340 USB interface (ref. Appendix B)
2. Improved ESD protection on the paddle and USB inputs.

In other respects, Rev C is the same due to a standard WK3 IC

## WKUSB Rev C Features

- USB 2.0 Interface with type B connector
- Iambic CW Paddle Interface
- Iambic A, B, Ultimatic & "Bug" paddle modes
- Two separate keying outputs
- Two separate PTT outputs
- Solid state relay outputs rated at 350V @120 mA
- 160 character input buffer
- Runs off USB and/or 3 internal AAA batteries
- ESD protection on USB and paddle inputs
- Firmware can be updated over USB
- Embedded commands
- Farnsworth spacing option
- Front Panel Speed Control
- Sidetone output with adjustable frequency
- Standalone Keyer Operation
- HSCW and QRSS Capability
- Standalone Practice Mode
- RTTY FSK transmit with WK3.1
- Works with any logging or contest application that supports WinKeyer !
- K1EL's latest WK3.1 IC
- Adjustable Speed 5-99 WPM
- Adjustable Weighting and dit/dah ratio
- Adjustable Keying Compensation
- Adjustable Letterspacing
- Adjustable PTT lead in and tail delays
- Optional autospacing
- Metal enclosure with RFI filtering
- Power supply monitoring
- 160 character input buffer
- Inexpensive cabling options
- Adjustable dit/dah ratio
- Adjustable speed pot range
- Paddle only sidetone
- Twelve stackable memory slots
- Single Op Two Radio (SO2R) Support
- Automatic beacon capability
- PIC CPU and CH340 USB Interface IC

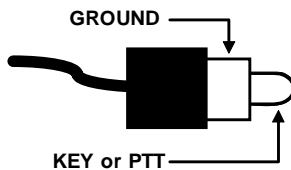
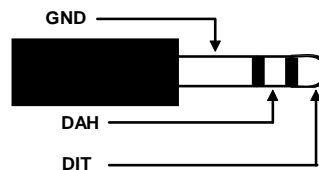
## Let's Get Started ! - USB drivers

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

WKUSB C uses a CH340N USB interface IC which is very reliable and easy to set up. Windows drivers for this device can be found on the WKUSB Rev C website. There are two versions, a self installing .exe and a zip file containing the individual drivers as used by Windows Update. Please read through Appendix B for our recommended install procedure.



The four central RCA jacks are keying outputs. The connector on the left is the USB connector and the connector on the right is an 1/8" stereo jack which accepts paddle set 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 RCA jacks provide two keying and two PTT outputs. The output selection is done by host software or by paddle command in standalone mode. PTT is enabled or disabled the same way. The outputs are isolated from WKUSB ground and all other WKUSB signals by optically coupled solid state relays. The outputs can switch voltages up to +/- 350 V at up to 120 mA and provide a 5000 volt isolation from WKUSB electrics. A universal keying cable can be purchased from K1EL.



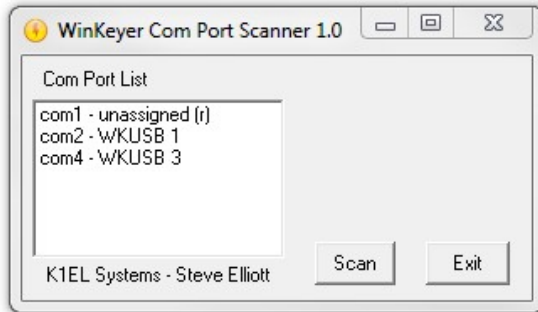
The USB connector is designed to mate with a Type A to B cable. This cable type is shown at left and is most commonly used with printers. The interface will run at USB 1.0 and 2.0 speeds and will also work with a USB 3.0 connector which is backward compatible with USB 2.0 interfaces by design.

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

## Let's actually run WKUSB !

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: <https://www.k1elsystems.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

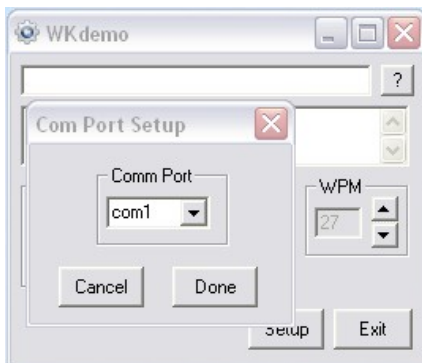


WK3DEMO is a general purpose test application that can exercise all of WKUSB's capabilities:

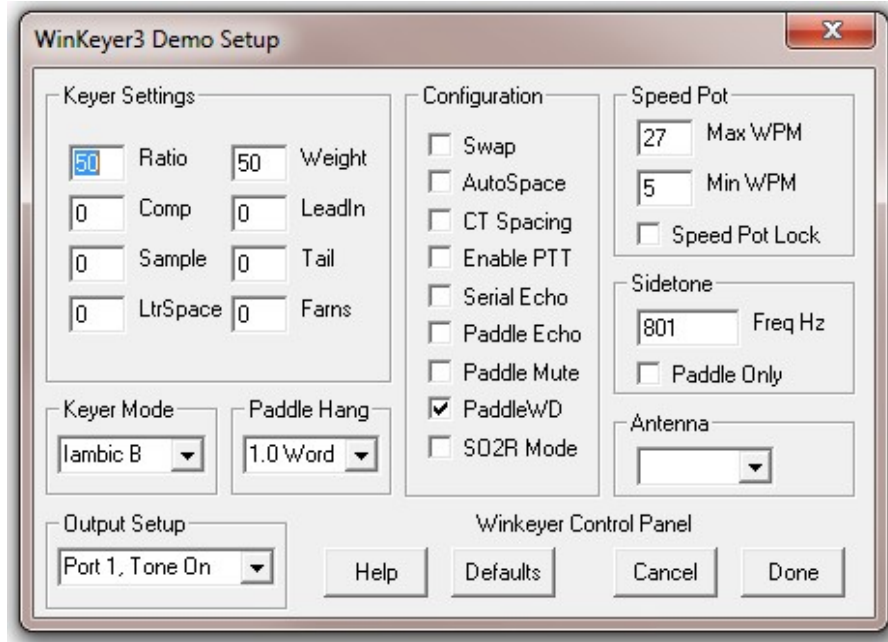
[https://www.k1elsystems.com/files/WK3demo\\_41\\_setup.exe](https://www.k1elsystems.com/files/WK3demo_41_setup.exe)

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 you 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 you will hear the text sent in Morse sidetone from the internal mini speaker. Enable Serial Echo in Setup and 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 be echoed in the upper text window. Use the Setup Dialog to try different keying options. You probably noticed 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 WKUSB.

The sidetone frequency can be changed as well as specifying whether it should be used for paddle entry only. An exact frequency selection is not always possible, it will round to the closest one it can do. There are several speed pot options, Speed Pot Lock forces WKUSB to only use the speed pot for speed control. When Speed Pot Lock is unchecked, the Up/Down arrows next to the WPM readout on the main dialog box can adjust 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.

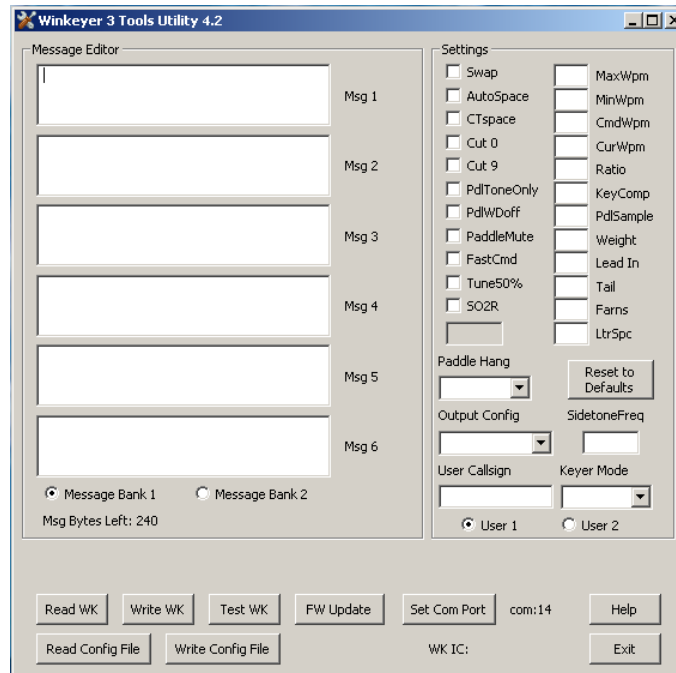
Although WKtest packs a lot of power in a small app, it is meant to be a hardware test tool. It gives you a good idea of what WKUSB is capable of and assures you that it is operating properly. The next logical step is to move on to a 'real' application, such as N1MM+. Before that we will look the WK3tools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

## WK3tools application

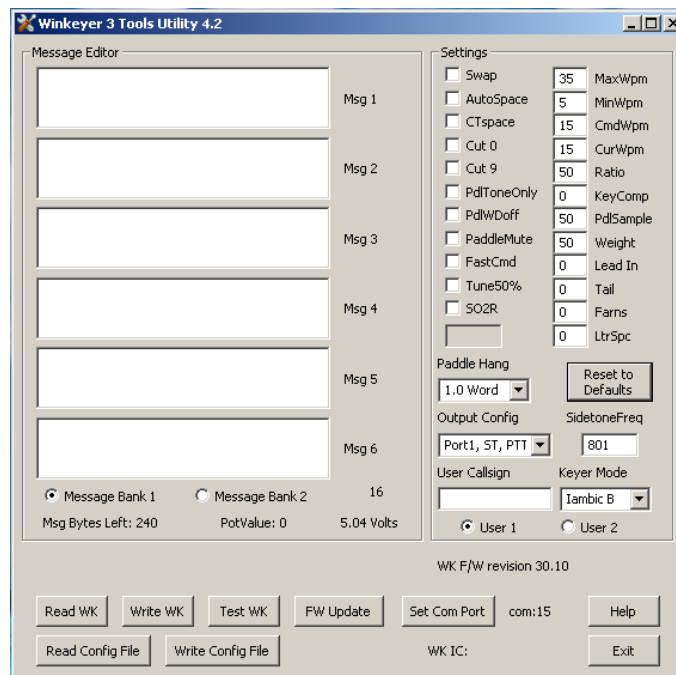
Normally when a WinKeyer device is not connected to a host application it will run in standalone mode. WKUSB can be powered over the USB port or by a set of internal batteries. WKUSB in standalone provides a complete Morse keyer with lots of capabilities. Originally, configurations changes could only be made using commands entered on the paddles. While it still has this capability, WK3tools allows you to make these changes from a PC application.

[https://www.k1elsystems.com/files/WK3tools\\_52\\_setup.exe](https://www.k1elsystems.com/files/WK3tools_52_setup.exe)

After installation, start the application and the following dialog box will appear:

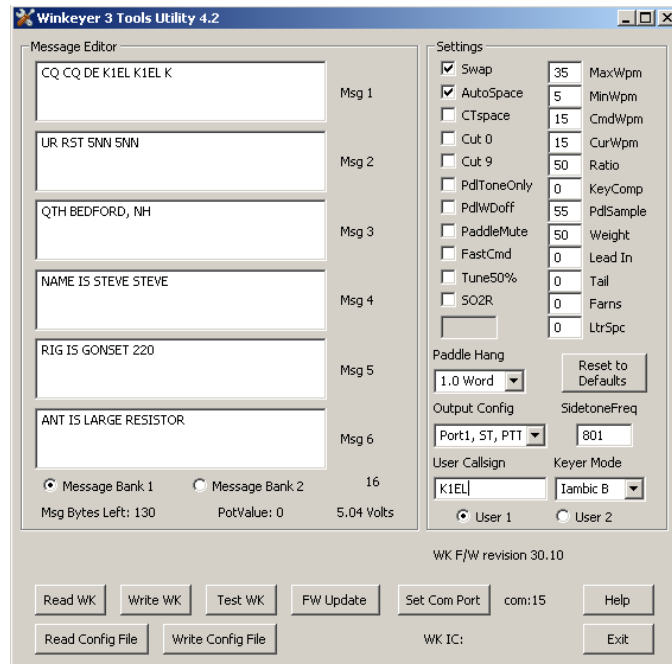


It will be empty since you haven't read in WK's current configuration settings. When WK is powered up for the very first time, all settings are loaded with defaults. To view these, connect WK to your PC, set the ComPort, and click on ReadWK. After a couple of seconds the dialog box will be updated to show you the current settings:



Now you can go in and tailor the standalone settings the way you want. The biggest timesaver is the ability to load messages. A tally of available message storage left is shown below the message text windows. Even though the WK box has only four message pushbuttons, it has the

capability to handle 12 messages in two banks. Don't forget that you have to write the messages back to WK with the WriteWK button. If you want to test them, press the buttons on top of WK.



Once you have saved messages and settings, they are ready for use. You can now disconnect WK from the PC. You don't need to use WK3tools again until you want to make changes. Don't forget you can still make changes to WK settings by using paddle commands. A detailed description of standalone paddle commands can be found in Appendix E.

### Restoring Factory Settings using WK3tools

This control is provided to restore original default settings. All of the control settings are initialized and messages are erased. Don't forget to write new settings back to WK when you are done.

### WK3tools Help File

A separate help document for WK3tools can be downloaded from the WK3tools webpage:

<https://www.k1elsystems.com/files/WK3ToolsOperationGuide.pdf>

### Upgrading WKUSB Firmware

WK3tools is used to install new firmware revisions in the field. A special webpage is provided for WK3 upgrade information:

[https://www.k1elsystems.com/WinKeyer\\_31.html](https://www.k1elsystems.com/WinKeyer_31.html)

**WKUSB FAQ #1: What is Standalone Mode and how is it different than Host Mode?**

The original Serial WinKeyer1 was designed to operate only while connected to a PC, under control of a PC based application, this is Host Mode. WinKeyer1 obtained power directly from the PC over unused RS232 control lines; if it was disconnected from a PC, it was inoperative. Soon hams started putting batteries in WinKeyers and using them standalone, in other words, not connected to a PC. This spawned many requests to add features to make a standalone WK more useful. When WKUSB was designed, we decided to include a complete 'second' keyer in the box and Standalone Mode was born. This second keyer shares the core 'K1EL Keyer Engine' with Host Mode but in other respects it is independent. It can operate on battery power or connected to a PC. It has its own settings memory and 12 message slots. Commands can be issued via paddle commands or through the use of WK3tools. Standalone settings do not influence Host Mode settings. In Host Mode, keyer settings are stored on the PC by the application and are loaded when you connect to WKUSB. In Standalone Mode the settings reside in EEPROM inside WKUSB. If you disconnect from an application, WKUSB will automatically go into Standalone mode and revert back to the saved settings from EEPROM.

**WKUSB FAQ #2: Are the internal batteries required if always connected to a PC?**

When WKUSB is plugged into a PC, power is automatically switched over to USB power by a diode switch.

In other words, WKUSB can operate with or without batteries. With batteries out, you can only use it when connected to a PC USB port. When first plugged in it will start up in standalone mode and then change over to host mode when you start up an application that uses WKUSB. In host mode, the PC app has control of WKUSB settings and messages. If you close the application down, WKUSB will remain powered but will revert back to standalone mode settings. If you start another host application up WKUSB will go back into host mode.

With batteries installed, the functionality is identical with the exception that you can unplug WKUSB from your PC and operate independently in standalone mode. Avoid disconnecting WKUSB while it actively controlled by a host application, this can cause the application to get confused and possibly hang. To be safe, always close the application before removing WKUSB.

**WKUSB FAQ #3: What about RTTY transmit, how does work ?**

Starting with WK version 3.1, we added RTTY transmit capability to WK. In addition to being able to run hardware FSK RTTY with N1MM+, FLdigi, and WK3demo, you can also run RTTY from a paddle in standalone mode. 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 21, 32, and 33. Paddle mode RTTY is covered on page 56.



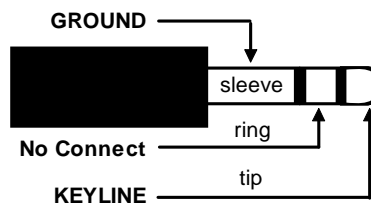
## Connecting WKUSB-SMT to your station

Minimally all you need to do is connect the USB cable from WKUSB to a USB port on your PC and connect the key output to the keying input on your rig. Since WKUSB 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.

www.k1el.com sells a WKUSB universal keying cable kit that covers 95% of keying requirements.

### Keying cable details

An RCA plug is required at one end and the other end will have a connector that will mate to your rig's key input. Usually this is a ¼ inch phone plug, we recommend using a stereo plug. A two wire cable is all that is required to key a typical transceiver. Plug the RCA side into either KEY1 or KEY2 and the other side directly into your transceiver's key input jack. On the transceiver end, the key line goes to tip and ground goes to the sleeve.



Transceiver/Transmitter end of cable

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, WKUSB will work fine because it has a solid state keying relay rated for  $\pm 350V$  switching at 120ma. This covers grid block keying but it probably will NOT be suitable for cathode keying. In any case always check the transmitter's manual for keying requirements before plugging WKUSB in.

WKUSB's RCA connectors are electrically isolated from WKUSB's enclosure and ground. This means that you can directly connect your transceiver to WKUSB with PC attached and not worry about ground loops or RF getting into your PC through WKUSB.

### 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, then WKUSB provides useful functionality. PTT is asserted before KEY and is held for the entire transmit interval. Both lead-in and tail delays are adjustable. The PTT connections are wired the same as the KEY, make sure that PTT is enabled in your configuration.

### Paddle Connection

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

WK3demo is useful for verifying that WKUSB 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 WKUSB jack. The tune function is useful for signal tracing.

### A cheap path to a keying cable

In today's world of phones and portable music players, RCA 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.



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.

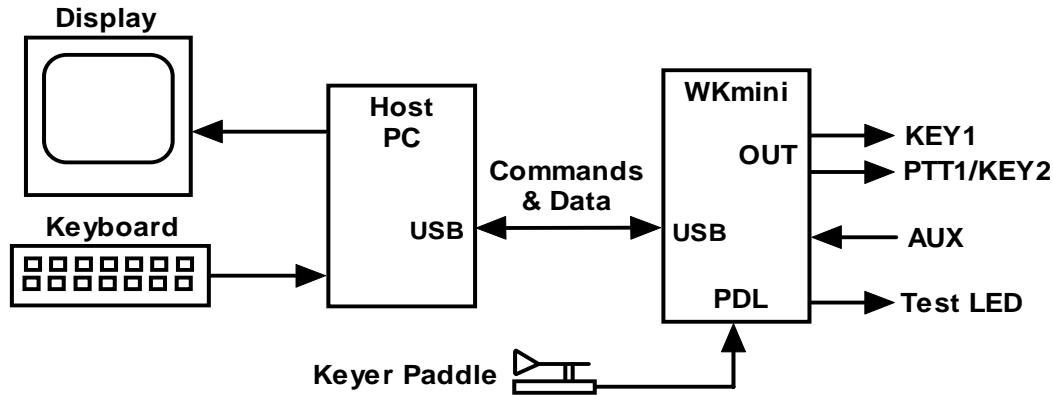
### Troubleshooting hints

- 1) *When I install batteries in WKUSB I don't get the initial R sent in sidetone.* This means that the WK3 IC is not starting up. Make sure the batteries are installed correctly by polarity. Look to be sure that both battery holder wires are connected to the PC board. If you don't get the R when plugging into a PC's USB connector, that signifies a fault in the unit. Please contact K1EL for further instructions.
- 2) *WKUSB 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 one of the KEY outputs. If you have it plugged into KEY2, try KEY1 or vice versa. In standalone mode, be sure you haven't accidentally issued a TX Mute command.
- 3) *I get dits or dahs only or CW letters are misformed.* This usually happens when you have not disabled your transceiver's internal keyer. You want to enable straight key mode. In most transceivers this function is controlled by menu selection, check your user manual.
- 4) *When I plug WKUSB in or install batteries, I get one R in sidetone but nothing else works after that.* This is a sign that one of the dit or dah paddles is stuck low. Remove the paddle connector and see if that clears the problem. It could also be caused by a paddle or paddle cable be a paddle cable problem.

- 5) *When I try to run WKUSB 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 WKUSB is connected to. Recheck your port selection using WKscan.
- 6) *WKUSB 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 WKUSB is not working. It could be a bad USB cable or bad connection to the PC. Note that WKUSB's 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 WKUSB or at the PC.
- 7) *I get an "Access to Port xx denied" message.* This probably means that one of the other K1EL programs or a non-K1EL program is open, perhaps in the background out of sight, and is still using the port, thereby locking out all other programs from accessing the port. To solve this issue, be certain that all other programs are closed and then try again.
- 8) *WKUSB is 'kicked off line' while you are transmitting.* There's a good chance RF is getting into the keyer or the PC disrupting USB communication between the PC and WKUSB. To solve this issue, first make sure the grounding system in your shack is properly designed and has the shortest possible path to a ground rod outside. If that doesn't help, you can use a 'bandaid' of installing snap-on ferrite "beads" on each end of the keyer to PC USB cable.
- 9) *I setup WKUSB and all was working until the PC went to sleep overnight, now the host app doesn't see WKUSB.* The solution is to close the application and restart. WKUSB is designed to be USB compliant and handle the condition where the host PC goes to sleep and then wakes up. Unfortunately, not all host applications are designed to preserve the connection to WKUSB. A working solution is to disable the PC from going into full sleep in the PC's Power Settings. Allowing a display and hard drive to sleep will save a great deal of power but keeping the CPU awake will preserve the USB connections.

## Theory of Operation

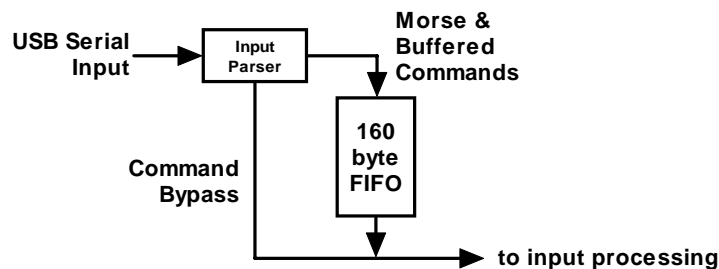
This section will briefly describe how WKUSB works. As shown in the block diagram below, the host PC is connected to WKUSB over a USB com port. WKUSB receives commands and Morse letters from the PC. Letters will be sent in accurately timed Morse.



PC to WKUSB Block Diagram

Commands modify WKUSB's operation in some way, for example changing operating speed, pausing transmission, or asking for status. Commands and Morse letters are processed differently in WKUSB. 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.



Data and Command Flow inside WKUSB

Since there are times when you don't want commands to take effect immediately, WKUSB 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.

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

- 1) Inform the host of a WKUSB 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

WKUSB accepts input from either host port or paddle. Paddle input will always take priority and will interrupt host data, automatically clearing WKUSB'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, WKUSB will pause for one word space time before it resumes serial data transmission.

## USB Sense

WKUSB is able to sense when the host PC it is connected to goes into standby and it will go into standby as well. It will stay in this state until the PC comes out of standby. It then wakes up and resumes. 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 the PC's power settings should be set to prevent sleep mode. Display and hard drive standby is not an issue since these don't affect USB operation.

## Serial Baud Rate

WK'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 WKUSB for use.

## Reset WK3/Restore Factory Defaults

There are cases when a host app misbehaves and can corrupt WKUSB's basic setup parameters. We know of no app that does this but when under heavy use with RF floating around unexpected things can happen, ESD is another issue that can cause this to happen. The easiest way to restore WKUSB's EEPROM settings is through WK3tools. Unplug WKUSB from the PC and then plug it back in after 10 seconds. 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.

## ASCII Code Assignments, Prosign Mapping

WKUSB maps several unused ASCII codes to standard prosigns as shown in the table below. Some abbreviations are not mapped (null), WKUSB will ignore these. 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.

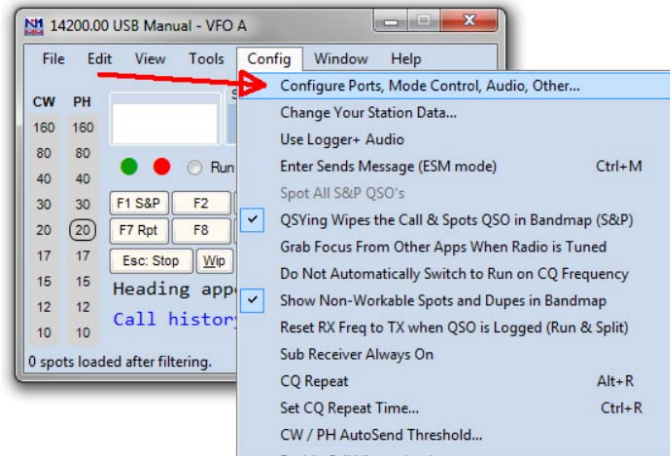
ASCII	Hex		Prosign		ASCII	Hex		Prosign
	0x20	<i>Is space</i>			.	0x2E	<i>Is period</i>	
!	0x21	<i>Is mapped to</i>	(null)		/	0x2F	<i>Is mapped to</i>	DN
"	0x22	<i>Is mapped to</i>	RR		:	0x3A	<i>Is mapped to</i>	KN
#	0x23	<i>Is mapped to</i>	(null)		;	0x3B	<i>Is mapped to</i>	AA
\$	0x24	<i>Is mapped to</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	Is mapped to	DU				
---	------	--------------	----	--	--	--	--

## N1MM Interfacing

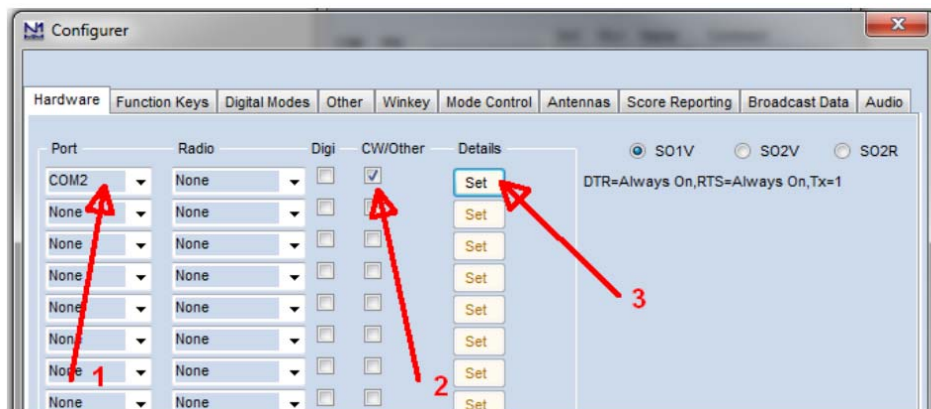
You will first need to know your WKUSB's comport assignment, if you don't know it, follow the WKscan Tutorial on page 4.

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

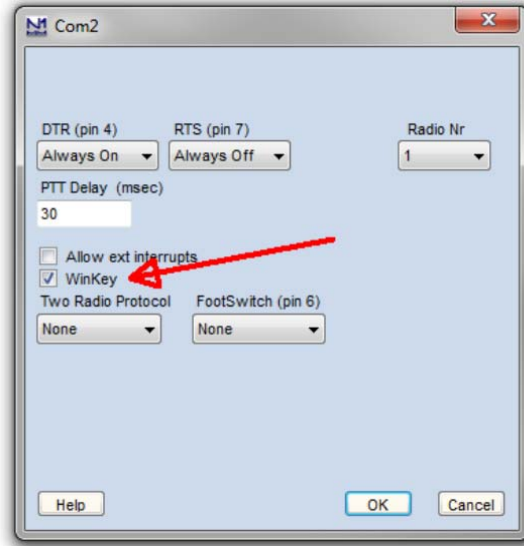


This brings up the Configurer dialog box. This is where devices attached to N1MM+ are set up. For this example we will set up a WinKeyer device on COM2.

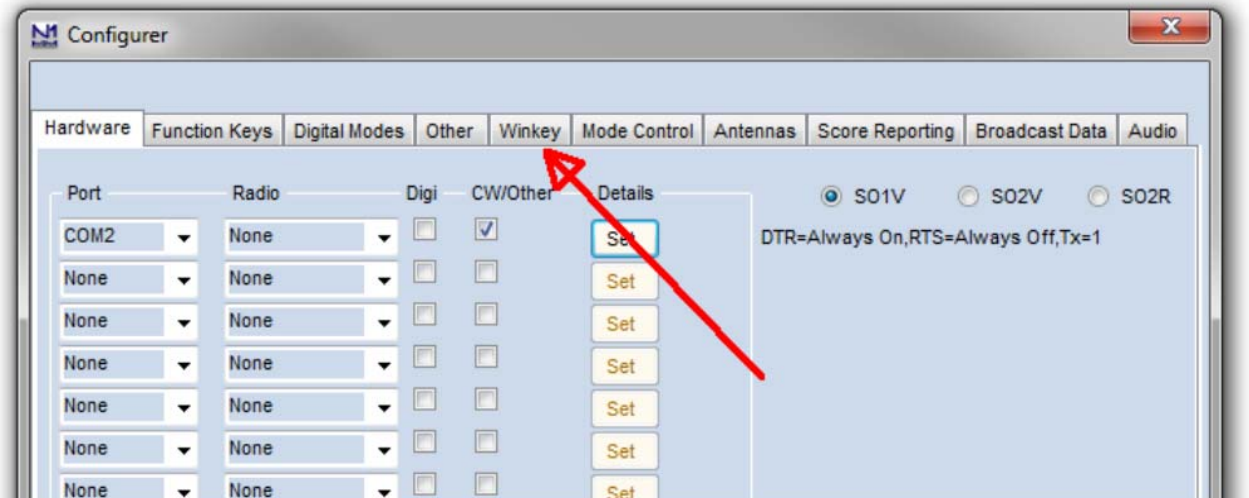
- 1) Select COM2 from the drop down box.
- 2) Specify that it is a CW device by clicking the CW/Other checkbox
- 3) Continue on to the Com Port setup dialog box by clicking Set



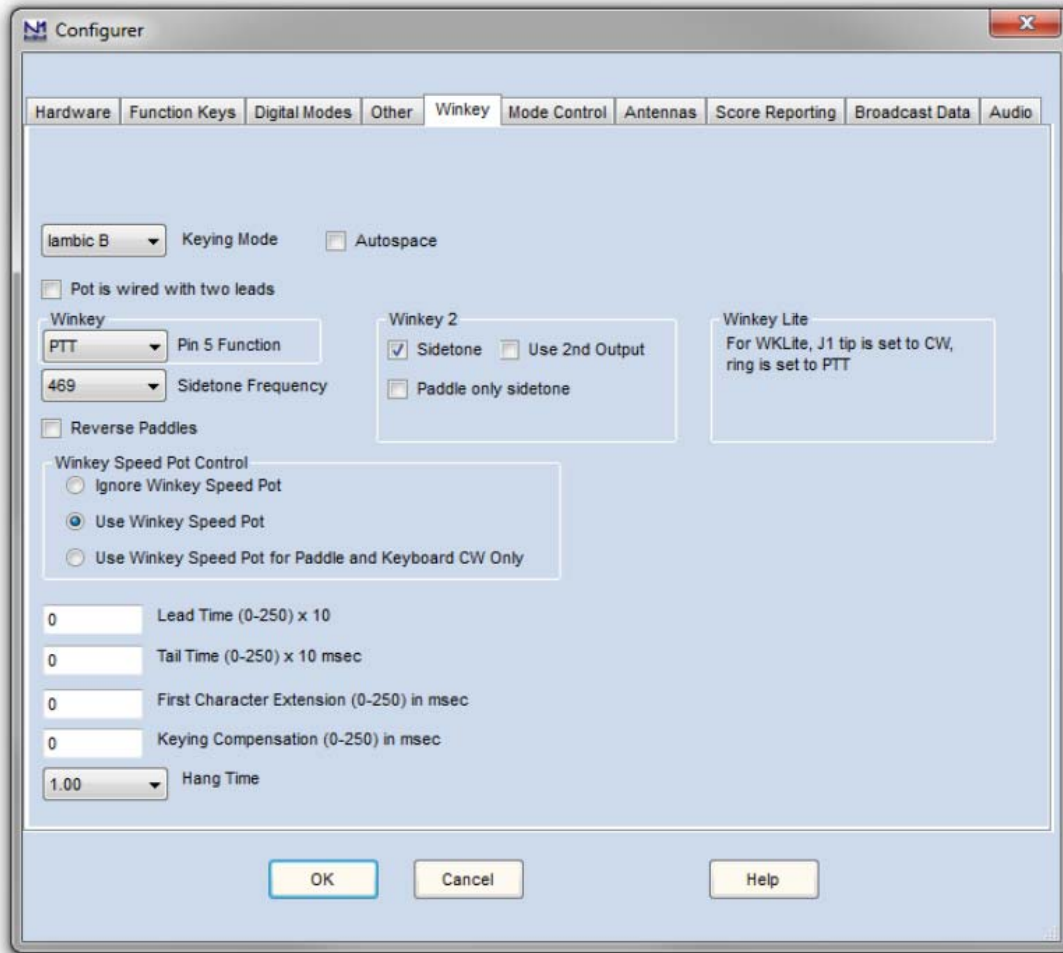
Clicking Set brings up the Com2 configuration dialog box. All we need to do here is click the WinKey checkbox, leave all the other settings as is.



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:

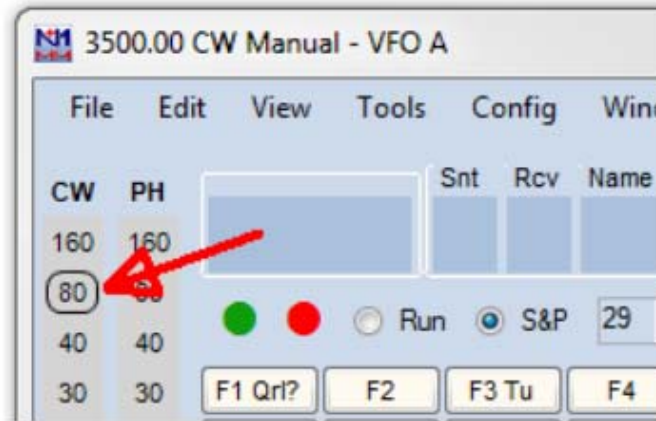
- 1) Ignore Speed Pot which only allows the host to control sending speed
- 2) Use Speed Pot which tells the host that the speed pot will set sending speed
- 3) Paddle & Keyboard where the host controls only the message speed

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

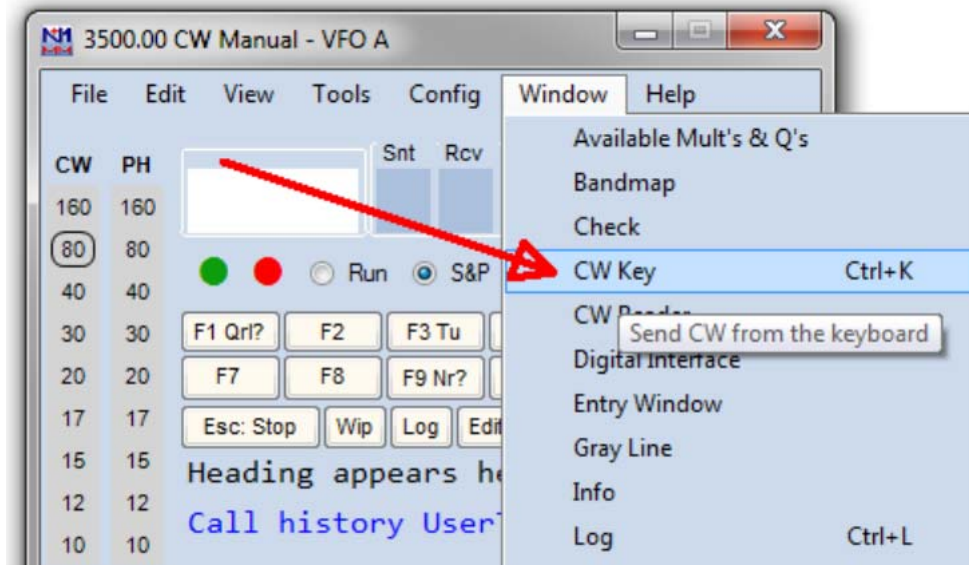
Hang Time is similar to PTT tail except that it is proportional to sending speed.

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 WKUSB testing. First off select CW mode by clicking on a band in the CW column:

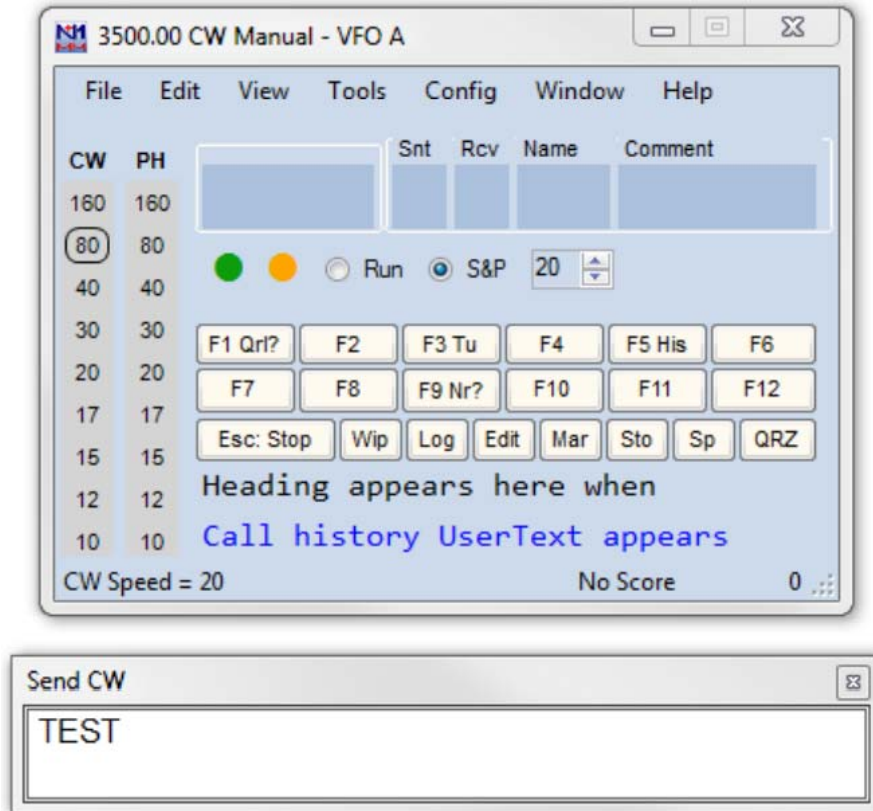




Now on the PC's keyboard enter CTL-T to put WKUSB into tune mode. Cancel tune mode by entering CTL-T a second time. With tune enabled you should hear sidetone coming from WKUSB. Now either enter CTL-K or select the CW Key Window as shown:



The CW keyboard window is now displayed and you can type into it to send letters to WKUSB. Note that when the Send CW window is up, the main N1MM+ dialog box is disabled. For a final test, turn the speed pot and you should see the CW Speed display in the main N1MM+ dialog box change. (This assumes that the speed pot has been enabled in the Winkey configuration dialog box)



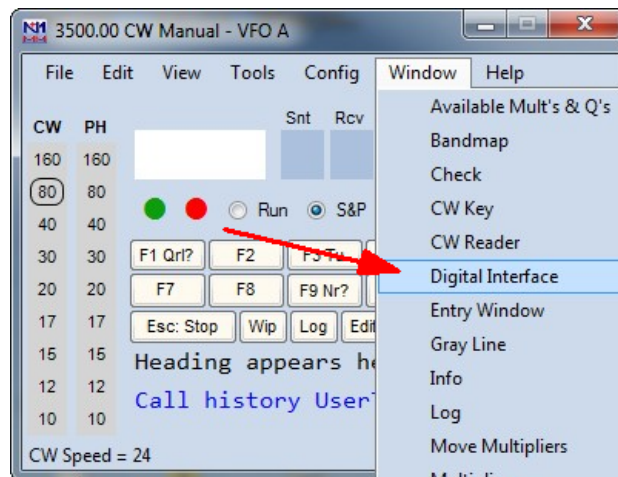
That's it for basic N1MM+ and WKUSB testing. Now it's time to learn all the amazing features in N1MM+ for setting up messages and getting up to speed with automatic CW contest exchanges.

## Running WK 3.1 with N1MM+ in RTTY mode

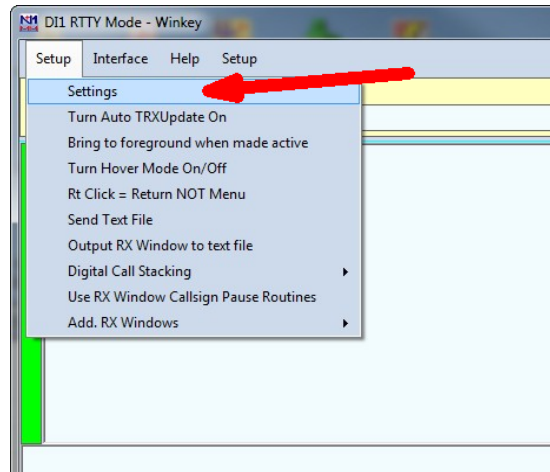
If you happen to have a WKUSB that uses the WK 3.1 controller IC, you can run it with N1MM+ in RTTY mode. Bring up the Configurer dialog for Winkeyer and you will see a new RTTY mode selection option:



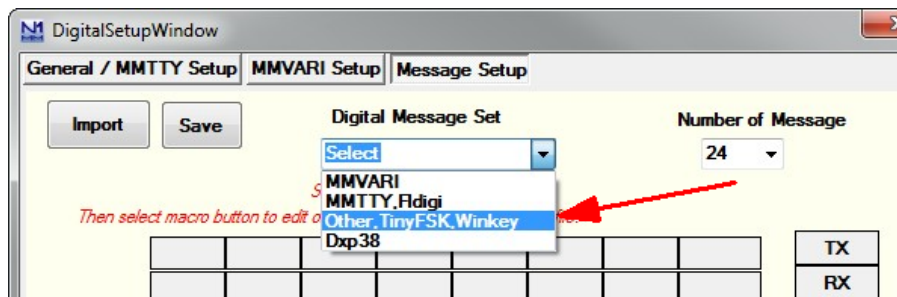
Click on Enable RTTY and then set a delay in both Lead Time and Tail Time; 40 mSec in each is a good starting point. Now click on the Configurer Digital Modes tab and in the left most drop down, select Winkey. Now close the Configurer and go back to the main N1MM+ window. From there we will open up the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



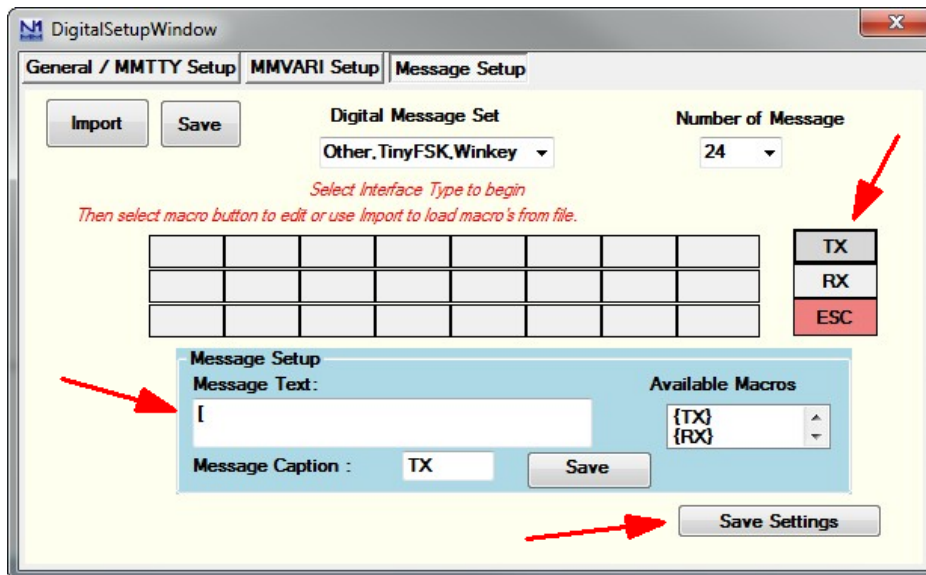
There are several things to set up in the digital interface window, first we will setup the macro keys for Tx, Rx, and ESC. Carefully follow the next instructions.



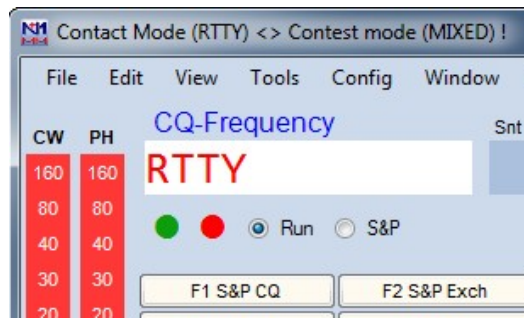
Click *Setup->Settings* to bring up the Digital Setup Window, click on the Message Setup tab and then select "Other, TinyFSK, Winkey" from the drop down menu.



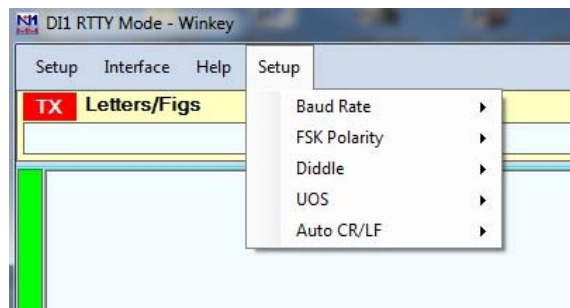
Using the picture below as a guide, click **Tx**, then enter [ in the message Text Window followed by **Save Settings**. Repeat this process by entering ] for **Rx** and then \ for **ESC**. Note that you have to re-select "Other, TinyFSK, Winkey" each time.



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:

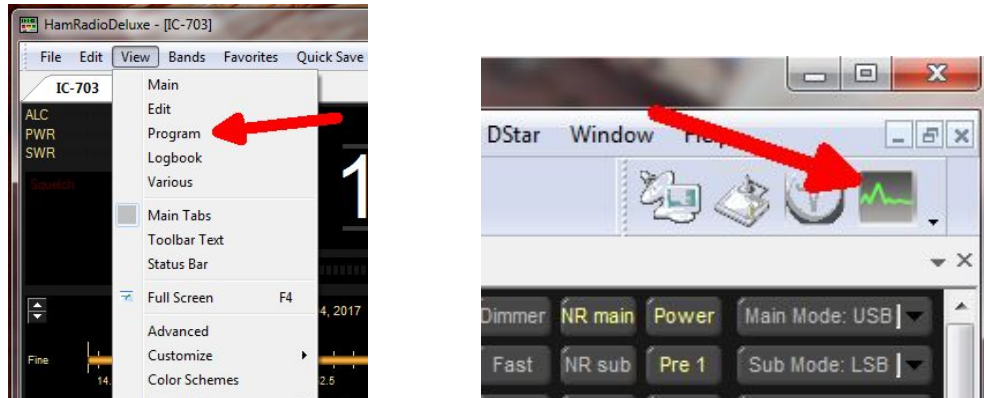


Now you are all set to run N1MM+ in RTTY mode with WKUSB. Entering an Fn macro key will trigger its message and you will see the RTTY string echoed in the Digital Interface Window as they are transmitted. You can enter free test in the Digital Interface Window by first clicking the Tx button, and then 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.

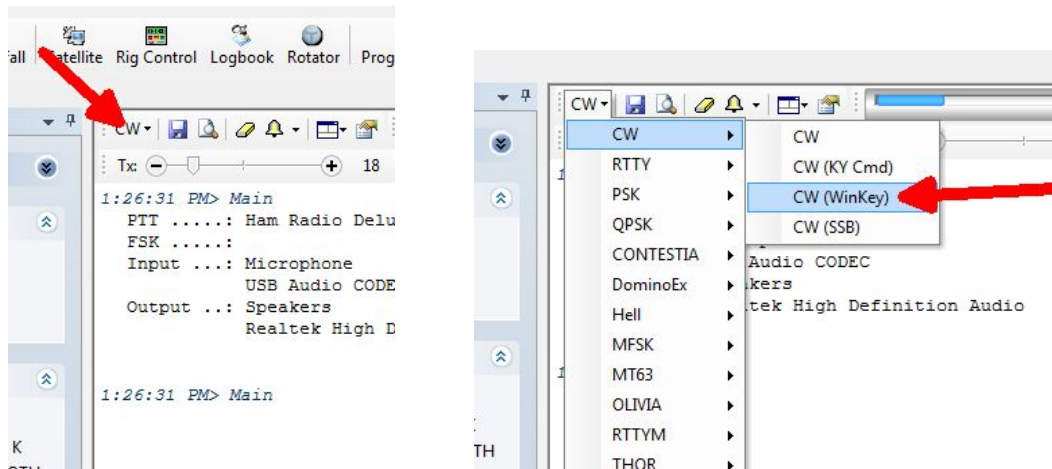
## HRD Interface Guide

You will first need to know your WKUSB's comport assignment, if you don't know it, follow the WKscan Tutorial on page 4. Be sure to close WKscan and keep WKUSB plugged in.

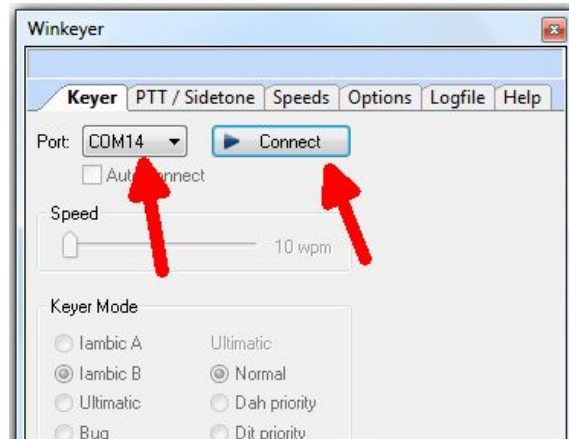
Start HRD and click on the DM780 icon in the upper left corner of the application. If it's not visible, you may have to enable program view as show on the left. (View->Program)



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 many 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 re-connect to Winkeyer automatically on startup.

A 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

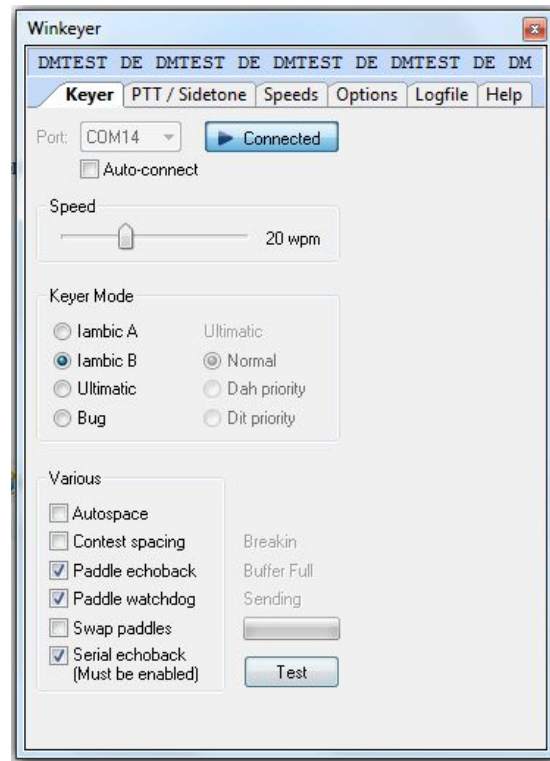
Autospace applies to paddle input only while contest spacing applies to message sending. When contest spacing is on, wordspaces are reduced to 6 dit times vs. standard 7 dit times.

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.

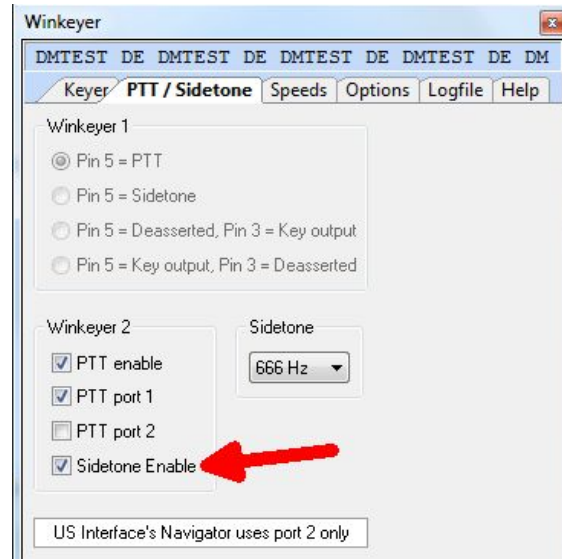
Serial loopback must be on in order for DM780 to keep track of outgoing Morse.

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 WKUSB 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.

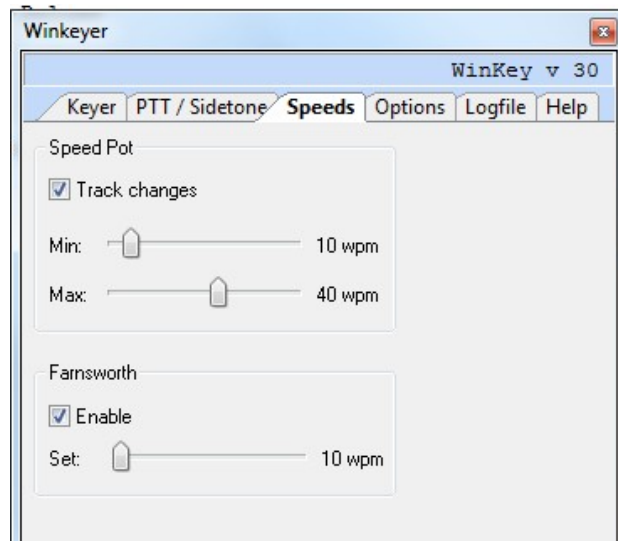
A range of sidetone frequencies are available. Note that sidetone enable can be unchecked if you would rather disable it and use your radio's sidetone instead.



The Speeds Tab is mostly concerned with speed pot settings. You can change speed pot swing by adjusting its minimum and maximum values.

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.

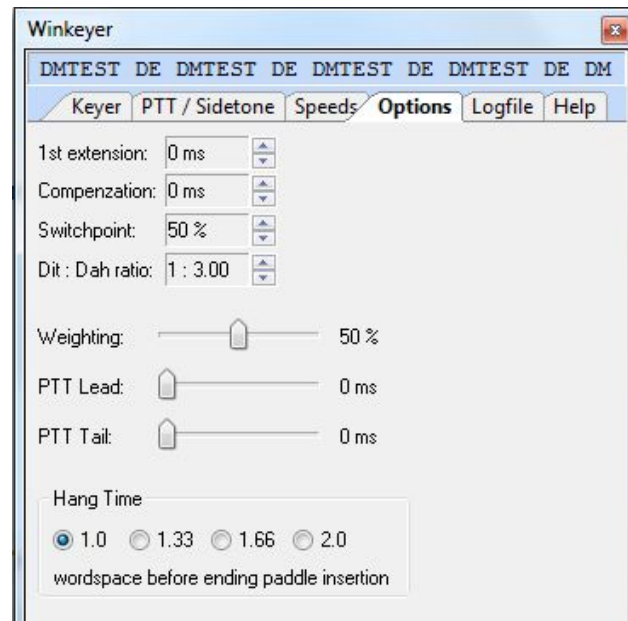
Note that the WinKeyer IC version is displayed at the top of this dialog box.





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.

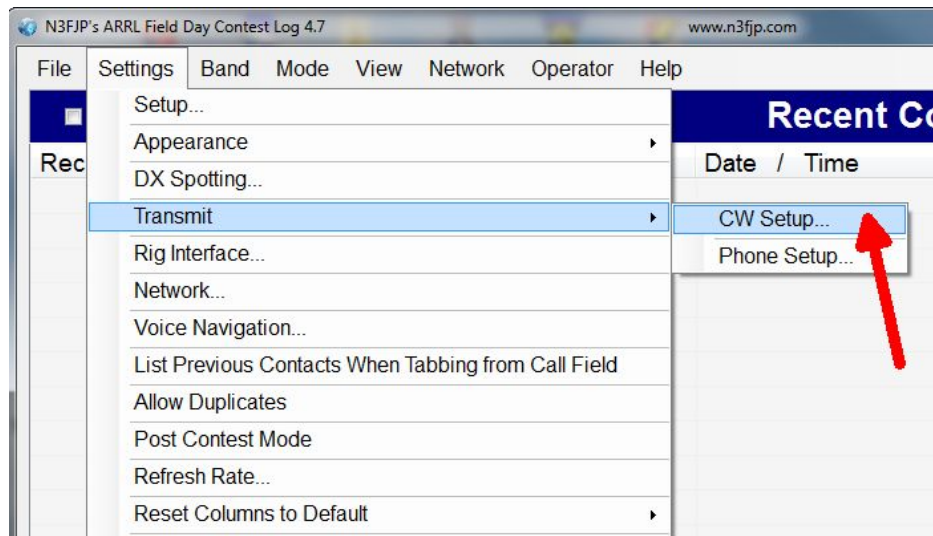
Also listed are settings for PTT timing. PTT turns on before sending starts and shuts off after sending is complete. Lead specifies how long PTT will turn on before keying starts. The Tail delay specifies how long PTT will stay on after keying ends. There is a separate tail delay setting for paddle sending 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 WKUSB 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 WKUSB 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 WKUSB 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**

  
 18    0    0

**Loop Sec**

F1 TEST K1EL    2.1   
 F4   
 F7   
 F10   
 F2   
 F5   
 F8   
 F11   
 F3   
 F6   
 F9 VV DE K1EL   
 F12 Keyboard

Faster Escape Check   
  Show Mini on Startup   
    
[Click for more info on rig to PC CW interfaces.](#)

**Description:**

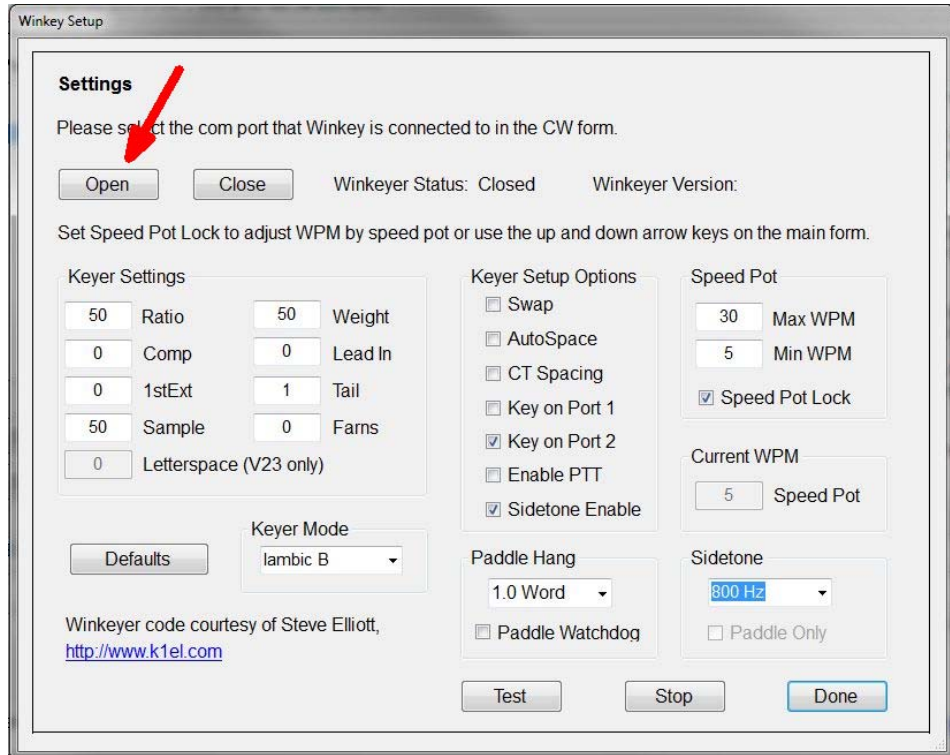
To use computer generated CW, first, you will have to build or purchase an interface to connect from your computer's serial port to your transmitter. Schematics and purchase options are available on my website (www.n3fjp.com).

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.

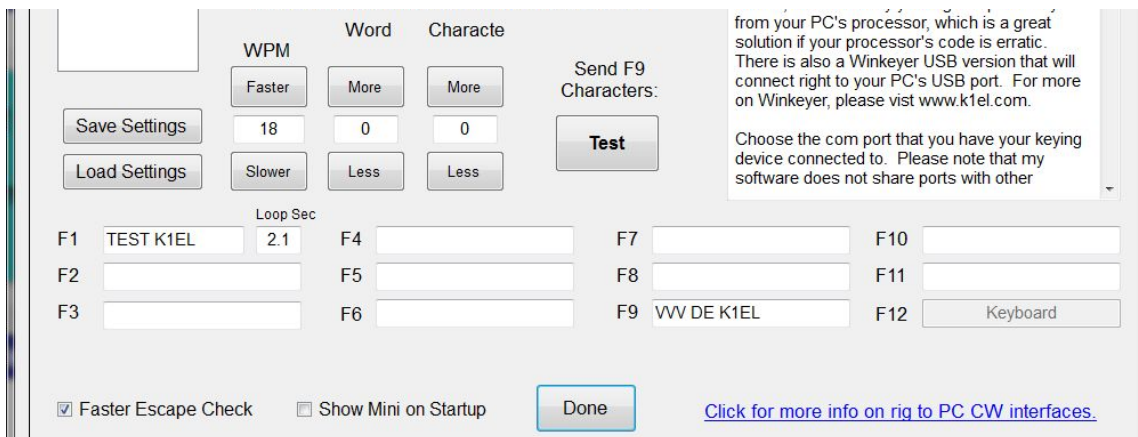
Alternatively, you can select the Winkeyer device, which will key your rig independently from your PC's processor, which is a great solution if your processor's code is erratic. There is also a Winkeyer USB version that will connect right to your PC's USB port. For more on Winkeyer, please visit www.k1el.com.

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 WKUSB, the status will change to Open and WKUSB's firmware revision will be displayed. Most of the options should be familiar, any that aren't can be looked up in Appendix E. Sidetone can be turned off by unchecking Sidetone Enable if you would rather use your radio's sidetone.



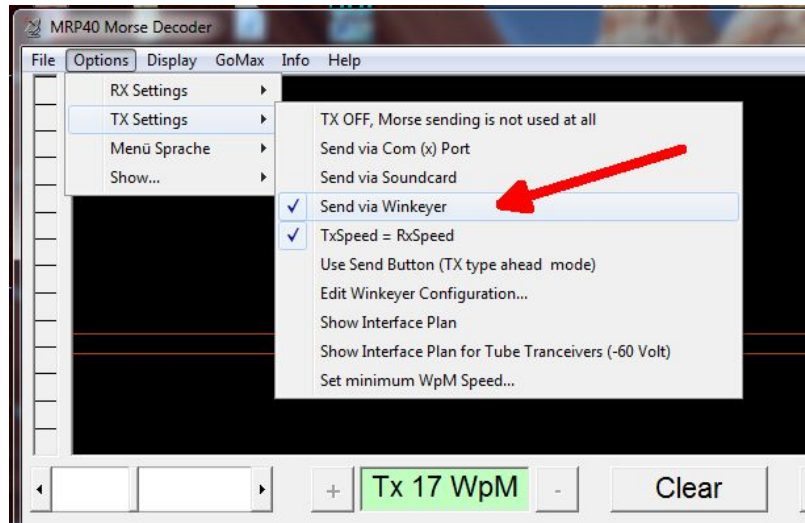
Click Done and you will return to the CW Setup form, you can enter messages in the F1 through F11 slots. F1 is set up as a repeating message with adjustable loop delay. When you click Test, the message in slot F9 will be sent. Click Save Settings to save you configuration and then click Done to return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



## MRP40 Interface Guide

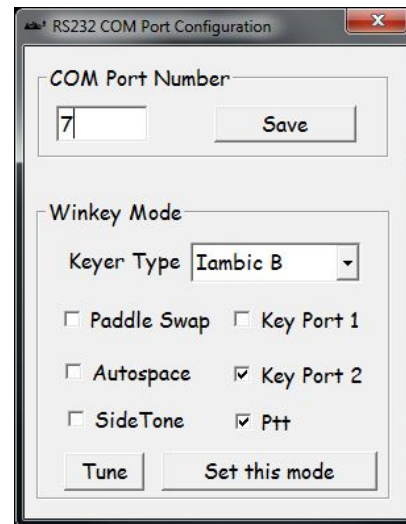
This is one of my favorite applications to use with WKUSB. It's not a logging program but instead it's a conversational, Morse terminal. It has an excellent CW reader that uses your PC's sound card to perform DSP tone detection. CW reception is displayed in a multi-channel, sideways, waterfall display. 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 WKUSB. 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. WKUSB 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.

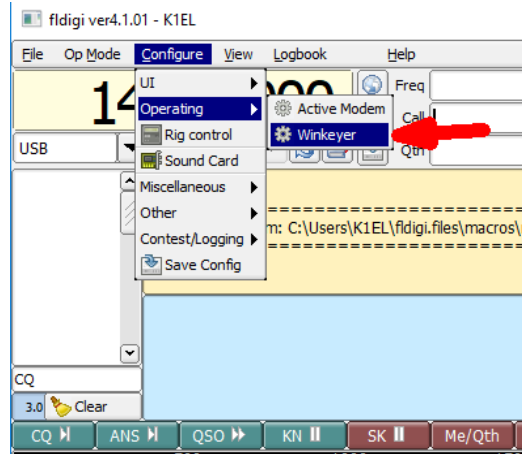
Click and hold Tune and WKUSB will key its output until you release. To get back to the main application click on the X in the upper left corner of the dialog box.



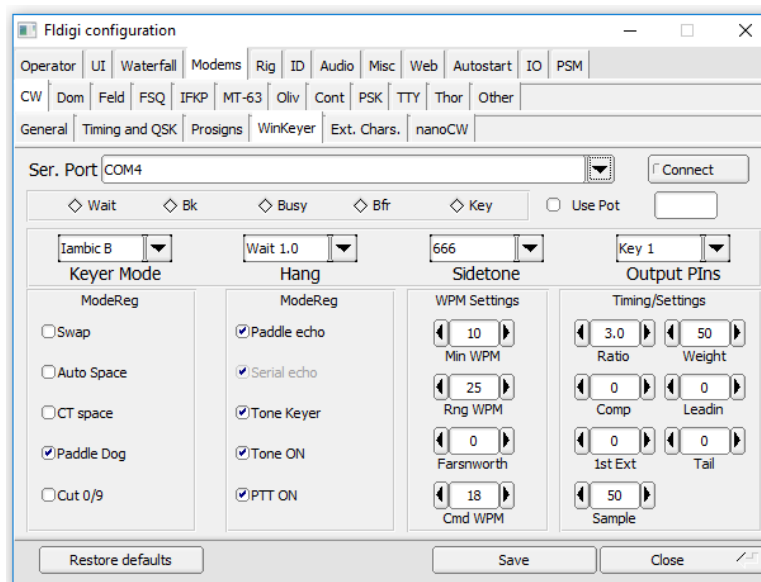
Now simply type in the lower text window and Morse will be sent by WKUSB. I recommend setting TxSpeed=RxSpeed in TX Settings, that way if you copy someone you want to work, your transmit speed will be set to match theirs automatically. I also like to adjust the text size to display more lines. That option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

## FLdigi Interface Guide

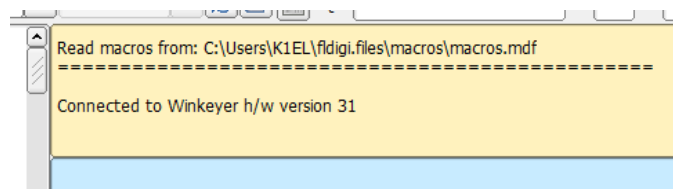
This is another very powerful amateur radio application. It's quite easy to set FLdigi up to use Winkeyer in either CW or RTTY mode. If you don't know the comport that your WKUSB is connected to, use our WKscan app. To start, we need to setup WinKeyer configuration:



Selecting WinKeyer will bring up the following configuration dialog box,



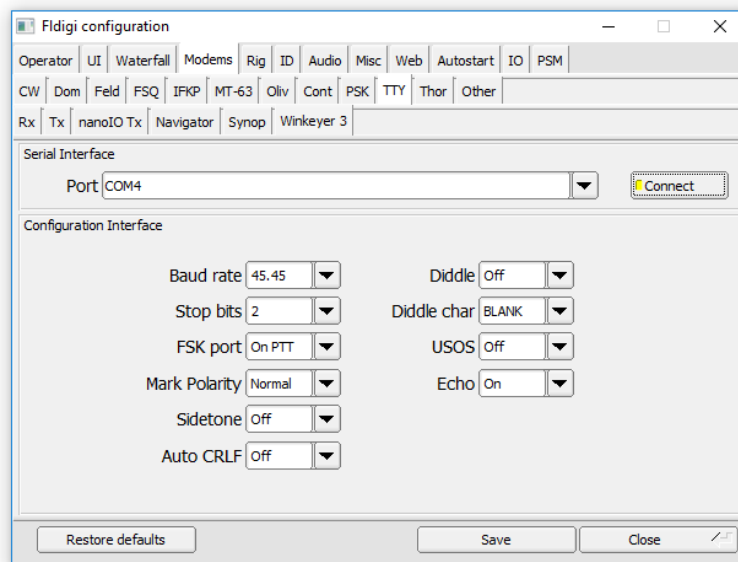
Enter WKUSB's com port and the review the default Winkeyer settings. For the most part, the defaults will suffice but you may want to change paddle swap or use a different keyer mode. Click on Connect and FLdigi will open up the WK interface. The following message will displayed on the main application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

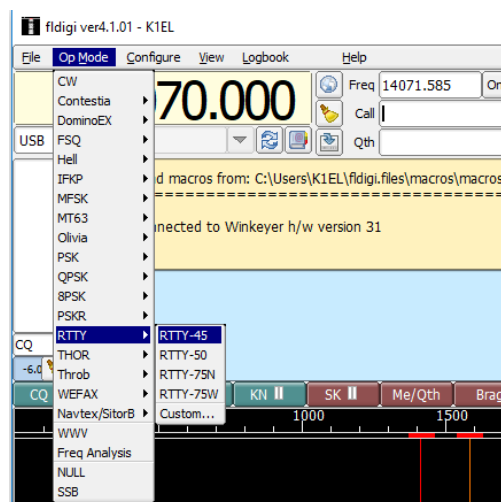
## Running WK3 with FLdigi in RTTY mode

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.

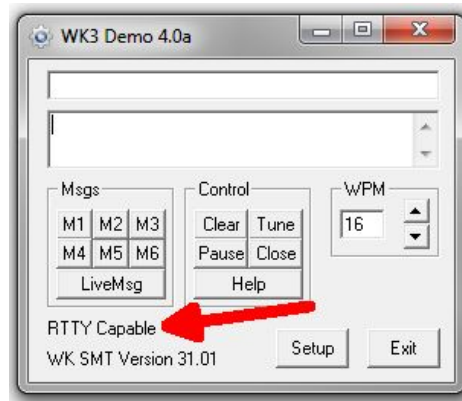
To set RTTY mode in the main operating window, simply choose Op Mode followed by RTTY at the speed you want to run at. WK RTTY operation, like WK CW, is well integrated into FLdigi and once set up you should be good to go. FLdigi offers excellent documentation on how to run their application and that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



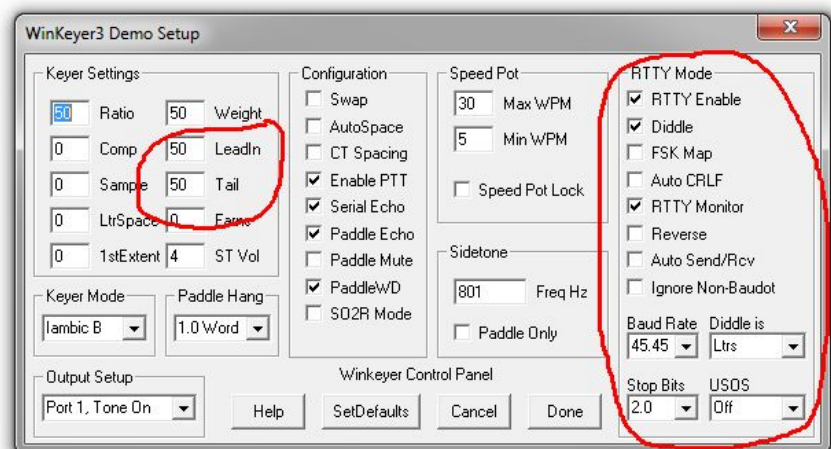
## Running WK3demo Version 4.0 in RTTY mode

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:



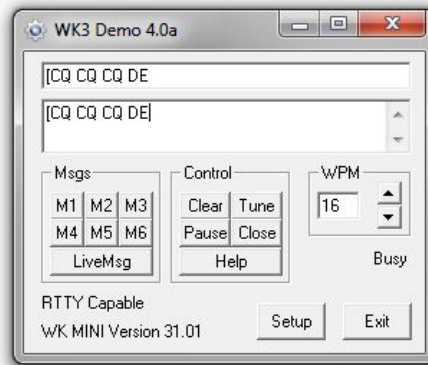
Now if you click on setup you will be able to access the RTTY controls:



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. When Reverse is unchecked, FSK=1 (HiZ) on mark. When reverse is checked FSK=0 (0v) when checked. Note that the RCA outputs employ open drain outputs which act as switches, which works fine when driving a logic input on a transceiver.

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.

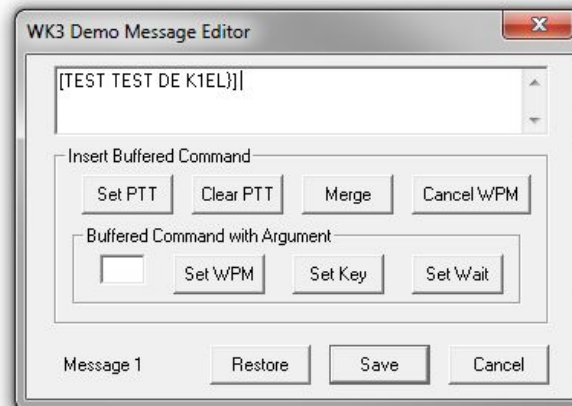
Click on Done to get back to the main menu. Once RTTY enable is set, WK will act quite differently. It will not send CW, only RTTY. Let's send a test CQ, first we enter a [ in the entry window. This will key your transceiver and assert the FSK output to the mark state. Since we selected Diddle on, it 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 WKUSB 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.

## Support and Warranty

WKUSB 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 WKUSB for repair and we will issue an RMA. Please submit questions by e-mail to [k1el.kitsinfo@gmail.com](mailto:k1el.kitsinfo@gmail.com)

Watch the K1EL website for updates and new products: <https://www.k1elsystems.com>

**While every effort has been made to insure that the WKUSB designs is as complete and accurate as possible it is still possible to cause equipment damage or incur personal injury if:**

**WKUSB is not used as intended.**

**WKUSB is connected incorrectly.**

**Safety guidelines outlined in this document are not followed.**

**WKUSB is modified in any way.**

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

## Revision History

12-11-17	-	Revision 1.0
04-01-18	-	Revision 1.1
10-01-18	-	Revision 1.2
03-19-19	-	Revision 1.3
02-20-22	-	Revision 1.4 added Rev C info

## 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. RTTY can be generated in host mode and also standalone mode. This is especially useful for dealing with transceivers that only provide RTTY receive capability. Using WK 3.1, you can generate RTTY without 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

The user issues special paddle commands to initiate RTTY mode and also to control Rx/Tx change over. Since WK can keep track of line length and automatically word wrap at the end of a line, it makes it easy to send Baudot strings by paddle. There are special Morse commands to start and stop an RTTY transmission, as well as insert a CRLF or send an RY test string. In addition, all of WK's standard message slots are available for RTTY as well as the User callsign slot. Since WK has two separate user setups, one setup could be used for CW and the other for RTTY. That allows you to switch back and forth very easily.

### WK3 RTTY standalone commands

**ADMIN Commands:** Press CMD button, wait for R, then enter X followed by:

**Y** – RTTY Mode toggle, WK responds with A when RTTY is enabled and an N when turned off.

**K** – Reverse toggle, this reverses the sense of mark and space. WK responds with an A when reverse is turned on and an N when turned off

**L** – Diddle toggle, WK responds with an A when enabled and an N when off.

**Z** – FSK MAP toggle. WK uses PTT and KEY to transmit RTTY. When in transmit mode, PTT is used to key the transceiver and KEY is the FSK output. When FSKMAP is enabled the function of KEY and PTT is reversed; KEY keys the transceiver and PTT is the FSK output.

**Procedural commands:** when WK is in RTTY mode these commands can be entered at any time, a pushbutton press is not required.

**AA** (di-dah-di-dah) toggles WK3 in and out of RTTY transmit mode. When transmit mode is enabled the transceiver is keyed and FSK is asserted as specified by the Reverse and FSKMAP settings. If Diddle is enabled it will start and stay on as long as RTTY is enabled.

**MM** (dah-dah-dah-dah) Tells WK to generate a CRLF sequence.

**MN** (dah-dah-dah-dit) Tells WK to generate an RY string for testing. A CRLF is generated followed by a line full of repeating RYRYRY patterns.

WK will send messages in Baudot format when RTTY is enabled. WK will automatically go into RTTY transmit mode before the message is sent and leave RTTY mode after the message is complete. Special RTTY codes, MM and MN, can be inserted into messages if desired. AA can't be entered since it is used to end message entry. A message is aborted by pressing the paddles.

Note that the RTTY settings are preserved by using the save command.

### **Using WK3demo in RTTY mode**

This is covered in detail starting on Page 33.

## **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. ALSO, You will find that RTTY setup has been added to the WK3tools standalone utility.

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

#### **Serial Number Changes**

The serial number can be preserved in Version 3.1. The current value will be written to EEPROM when a save command is issued. Serial numbers up to 50,000 are now allowed.

#### **Report Revision Standalone Command**

A new extended standalone command has been added that reports the keyer major and minor version. Enter **X Q** and WK will report the major and minor revisions in Morse.

#### **Many Bug fixes**

A large number of small bugs were fixed in version 3.1. For example, all illegal paddle input is now flagged and reported. In version 3.0 there were cases where a bad command entry could cause a watchdog timeout.

#### **WK API changes**

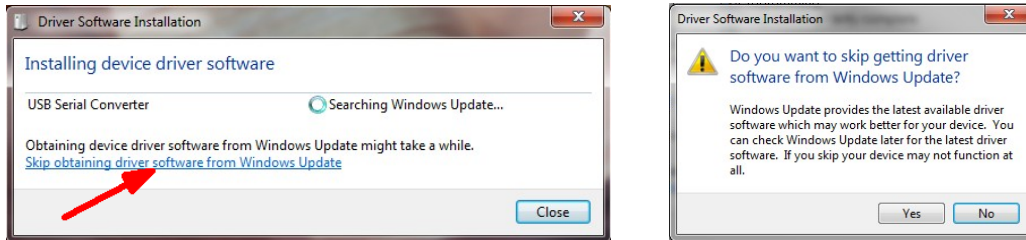
Besides the new RTTY commands several new commands were added to load and unload the RTTY, Volume and X2MODE values.

## Appendix B - Installing the CH340 USB Driver for WKmini

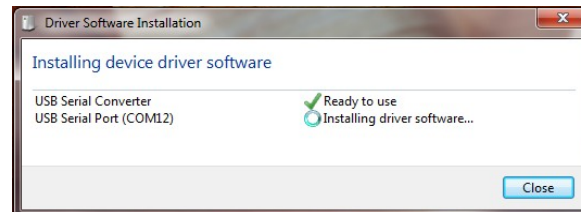
*As mentioned, don't plug WKUSB in yet, we have to deal with the USB driver first.*

For Windows, first thing to do is create a folder on your PC called CH340 or something similar, the name doesn't really matter. Then download CH341SER.exe and Windows-CH340-Driver.zip from <https://www.k1elsystems.com/WKminiRevB.html> and copy them into the new folder. Unzip the Windows-CH340-Driver.zip file.

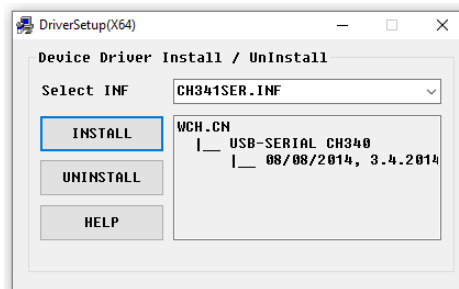
Now plug WKmini into your PC and wait to see if Windows Update will automatically search for a USB driver. This usually doesn't happen, but if it does click Skip obtaining driver....



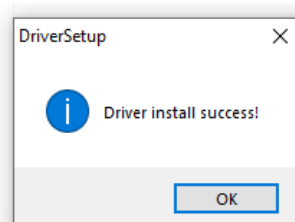
Windows will then look on your local hard drive for the driver and will install it automatically.



If Windows Update did not start, simply run CH341SER.exe and this dialog box will be displayed:



Click Install and after a few minutes, the following message box will display:



Now we can return to page 3 to continue WKmini hardware setup and testing.

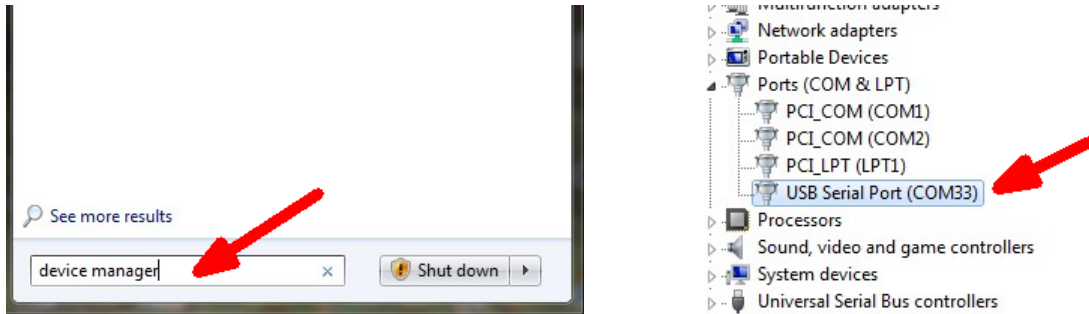
## Appendix C – WKUSB USB COM port assignment information

### Win 7 and newer Com Port Viewing and Reassignment

First of all we need to get a device list. The easiest way to do this is to click the Start button in the lower left which will bring up a search edit box.

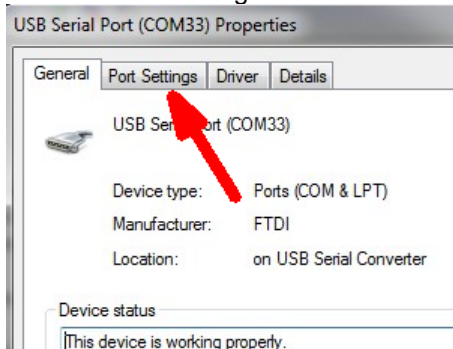
Enter Device Manager and hit return.

That will bring up a list of active devices

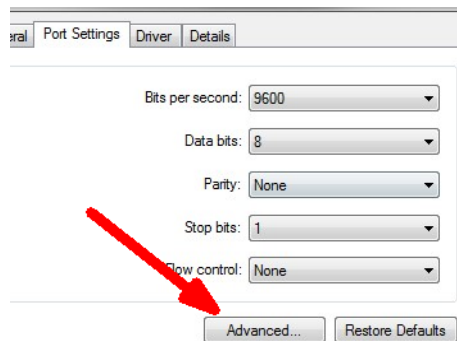


WK's port will be listed under Ports (com & LPT). If more than one USB Serial Port is listed disconnect WKUSB's USB cable and see which device entry is removed. Now you know WKUSB's com port assignment. In this example it's 33 which may not work with all WK apps. To change it, double click WKUSB's USB Serial port entry to view the port properties dialog box.

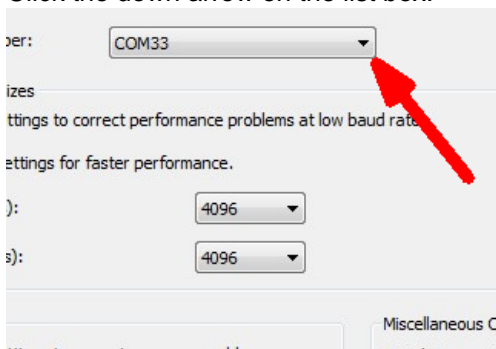
Click on Port Settings Tab:



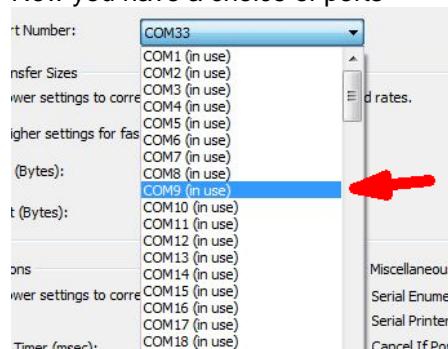
Then click Advanced:



Click the down arrow on the list box:

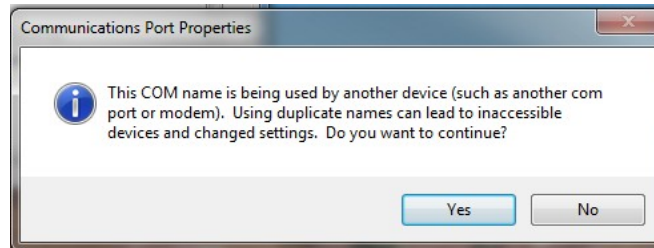


Now you have a choice of ports



Since Windows does not do a great job of keeping track of USB ports when a device is removed, there are usually lots of stale port assignments that are still tagged as in use. It's usually pretty easy to determine which ports are not in use and reassign one of those to WKUSB. If you choose something under 15 you are probably not going to have port issues with any WK application.

You will get this pop up when you pick an 'in use' port. If you are sure the port is not assigned to any other serial com device, click Yes and then continue on to close all remaining dialog boxes.



### WinXP Com Port Viewing and Reassignment

To start, move cursor to lower left corner of the screen and left click on Start. Then...

- 1) Right click on *MyComputer* and 2) Right click on Properties:



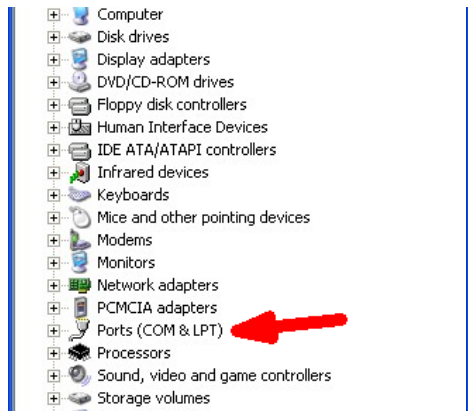
Right click on *Hardware*:



Right click on *Device Manager*:



Expand (COM & LPT)



Look for *USB Serial Port (COMx)* x is the port

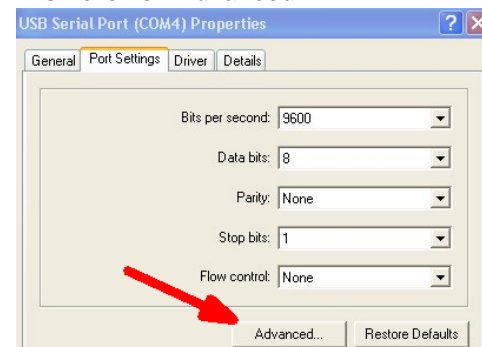


In the example above, the USB com port assignment is 4. If you have multiple USB Serial Ports listed and you are not sure which one is WK, disconnect WK from the computer and see which one is removed from the list. Then just plug back in and make a note of the assignment. Sometimes the port assignment may be in the 30s, 40s, or even higher. Windows does not do a great job managing port address when a USB com port is unplugged. Instead of deleting unused ports it just keeps assigning higher and higher port values. Some applications don't allow com ports higher than 10 so you might want to change the port assignment. Go back to the device list as shown above and double click on the WK's USB Serial Port entry. That will bring up the USB Serial Port Properties dialog box.

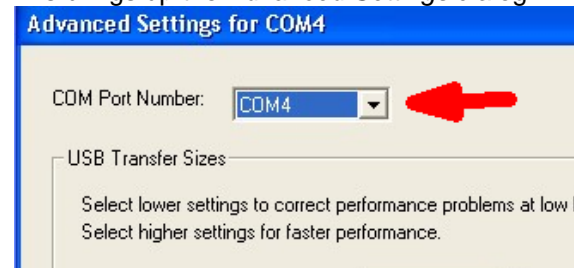
Click on *Port Settings*:



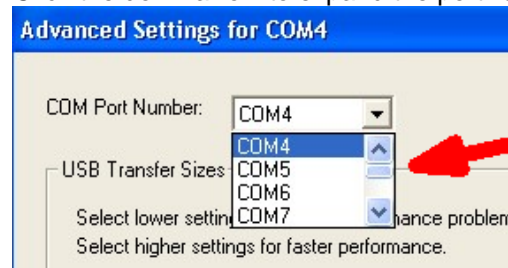
Then click on *Advanced*:



This brings up the *Advanced Settings* dialog:



Click the down arrow to expand the port list:

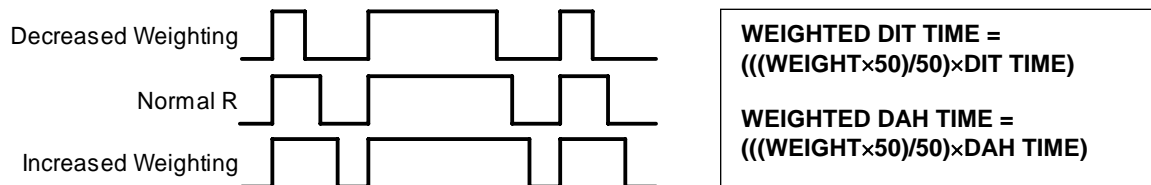


A list of available com ports is shown, select the desired port and click OK. Windows will warn you that the port is in use but as long as you are sure the port is free then click OK. Now close all remaining com port related dialog boxes and you are finished.

## 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.

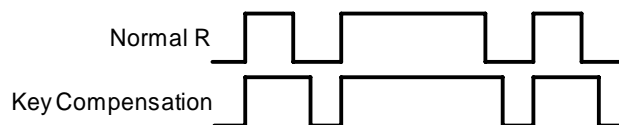


Weighting Example

### 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 WKUSB 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:



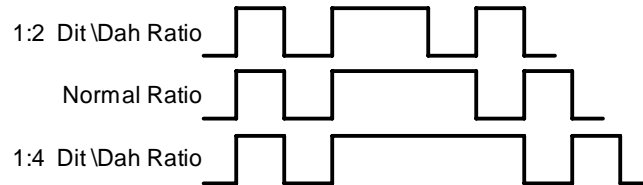
$DELAY\_TIME = (SWITCHPOINT \times DIT\_TIME) / 50$  where Switchpoint is a value between 10 and 90.

### 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 ratio is 1:3 (dit:dah), a dah is three times longer than a dit.

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

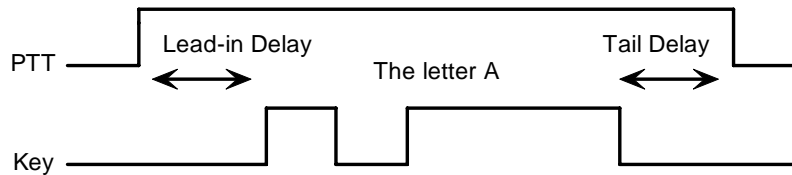
A value of 50 selects 1:3, a value of 33 would select 1:2, and a value of 66 would select 1:4. Any value other than 50 causes distortion of the Morse waveform. Some ops use this option to make their CW sound less “machine like” but a little goes a long way !



Three ratio settings for the letter R

### PTT Lead and Tail Times

WKUSB'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. WKUSB 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. WKUSB 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.

### **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 WKUSB 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 WKUSB'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, WKUSB 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. WKUSB has a paddle event memory so that you can enter dits or dahs during the inter-letter space and WKUSB 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 Keyed Modes**

There are six different keying modes supported by WKUSB: 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

This tells WKUSB 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

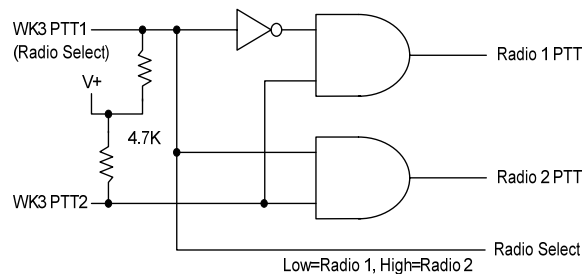
Sidetone can be turned on or off. Local sidetone is not needed if the user is going to use the sidetone generated by the transceiver they are using. This is the case in the majority of setups. That's why WKUSB does not have a built in sidetone speaker. The enable control is used instead to enable or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

### Sidetone Frequency

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

### SO2R Enable

This is a special mode that is only applicable to WK keyers that have two keying outputs, each with a dedicated PTT output. WKUSB 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 WKUSB 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.

## Appendix E WKUSB (WK3) Standalone Mode

### Standalone Command Mode

If the command push-button is pressed and held, WKUSB (WK) will respond after about two seconds with the letter **R** in sidetone only. This means WK is ready to accept a command, you simply enter the command letter in Morse on the paddles and the command will be executed. Some commands require additional information which WK will prompt you for by outputting the letter **E** (for enter). All commands provide some sort of feedback to tell you if the command was understood and executed properly, in most cases an **R**. If an illegal command is entered WK will respond with a question mark.

**Important Note !** When in command mode, transmitter keying is disabled and replies are sent in sidetone only. Thus in order to use command mode you must have a sidetone speaker of some sort. If sidetone had been disabled with the **A** command it will be re-enabled automatically when entering command mode.

### Standalone Command List

<b>A</b> - Select sidetone on or off	<b>N</b> - Load 4 Digit Serial Number
<b>B</b> - Start Easy Beacon	<b>O</b> - Select output key port
<b>C</b> - Set Command Speed in WPM	<b>P</b> - Start Practice Mode
<b>D</b> - Decrement Serial Number	<b>Q</b> - Query current settings
<b>E</b> - Swap message banks	<b>R</b> - Review message without transmitting
<b>F</b> - Set Farnsworth Speed	<b>S</b> - Set preferred WPM speed
<b>G</b> - Set 50% Duty Cycle Tune	<b>T</b> - Load PTT Tail Time
<b>H</b> - Set Paddle Hang Time	<b>U</b> - Select Autospaceing on/off
<b>I</b> - Set Letterspace Adjust	<b>V</b> - Set Keying compensation in mSec
<b>J</b> - Set Paddle Sensitivity	<b>W</b> - Set Key Weight
<b>K</b> - Select Keyer Mode	<b>X</b> - Select Extended Command Mode
<b>L</b> - Set PTT Lead-In Time	<b>Y</b> - Set Dit/Dah Ratio
<b>M</b> - Mute Transmit (CPO mode)	<b>Z</b> - Select sidetone frequency

In the command descriptions below, the **[n]** or **[nn]** notation means that additional parameters must be entered on the paddles after the command. A letter displayed in **BOLD** is something you enter, **BOLD ITALIC** is what WK responds with. A **[pb]** means that WK will wait for you to press one of the message pushbuttons.

**A - Sidetone enable** is toggled when this command is entered. Toggle means if the sidetone was on when this command was issued it will be turned off and vice versa. WK will acknowledge this command by responding with an **R**. Note: If sidetone is disabled it will be re-enabled temporarily for command mode.

**B [m,dd] - Easy Beacon:** Any message slot in the current bank can be repeated at a specified rate. After entering **B**, the user is prompted to enter a single digit message number (1-6) and then prompted to enter a two digit beacon cycle time (1-99) in seconds. A beacon is interrupted by hitting either paddle. The beacon is timed from start of beacon to start of next beacon. For example if you set a delay time of 10 seconds, the beacon will start every 10 seconds regardless how long the message is. If the message is longer than delay time, then there will be no gap between messages.

**C [nn] – Command WPM:** WK uses different speeds for command transactions and keyed transmit. Changes in transmit speed will not affect command speed. After the **C** command is issued enter the speed in WPM. If the speed is valid, WK responds with an **R** otherwise a **?** See the **S** command for details on setting transmit Morse speed.

**D - Decrement Serial Number by 1:** WK responds with a **D** after the decrement.

**E - Swap message banks:** WK has two separate message banks, six messages each. In response to this command a single dit is echoed when bank one is selected and two dits when bank two 2 is selected.

**F [nn] - Set Farnsworth Speed:** This is used primarily for code practice. Letters are sent at the Farnsworth speed while maintaining the default code speed. For example, if Farnsworth is set to 25 WPM and the operating speed is set to 7 WPM, individual letters will be sent at 25 WPM while spacing between letters is at a 7 WPM rate. To disable Farnsworth mode, set it to zero.

**G – Set tune duty cycle default:** This sets the default tune duty cycle which can be either 50% or 100%. Some folks prefer a 50% tuning duty cycle to reduce power dissipation in transmitter finals and/or linear amplifier. An **A** is echoed when 50% is enabled, an **N** is echoed when set to 100%. Note that once tune starts, the duty cycle can be changed by hitting the paddles.

**H [n] - Set Transmit PTT Hang Delay Time:** Sets a PTT delay that is proportional to sending speed. The delay begins after paddle sending stops. You can select one of four delays:

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

After entering the command letter, you will be prompted with an **E** to enter the desired hang time as a number 0 to 3 as indicated in the table. Hang delay is different than PTT tail delay in that it is proportional to code speed while PTT delay is a primarily a fixed delay time. This means you don't have to change the PTT delay every time you change sending speed.

**I [nn] - Set Letterspace Adjustment:** nn is a value 0 to 31, specifying an additional letterspace to be applied between letters. Multiply nn by two to arrive at the actual adjustment percentage. For example a value of 7 applies 14% additional letterspace between letters. The maximum adjustment is 62%.

**J [nn] - Paddle Sample Point** controls when WK will start looking for a new paddle press after sensing the current one. If there is not enough delay the keyer will send unwanted dits or dahs, if there is too much delay it may slow you down because you can't send ahead of the keyer. The default value is one dit time (50) and is adjustable in a fraction of a dit time. Faster operators report a setting somewhat less than default is more pleasing. **If the paddle sensitivity is set to zero, both dit and dah paddle memories are disabled.** The delay is calculated with this formula:

$DELAY\_TIME = (nn \times DIT\_TIME) / 50$  where switch point is a value between 01 and 99.

**K - Set Keying Mode:** There are six different keying modes supported by WK: Iambic mode A, Iambic mode B, Straight Key/Bug, Ultimatic, Dit priority mode, and Dah priority mode.

After the **K** command is issued the current mode is set by entering a single letter:

Iambic B: Enter B  
 Iambic A: Enter A  
 Ultimatic: Enter U  
 Straight Key: Enter S (This is also the Bug setting)  
 Dit Priority: Enter E (Ultimatic: dits take priority when both paddles are pressed)  
 Dah Priority: Enter T (Ultimatic: dahs take priority when both paddles are pressed)

**L [nn] - PTT Lead In Time** can be set to a value between 0 and 99 which is a subset of the entire possible range of 0 to 255. See the Set PTT Tail description on page 42 for more information.

**M - Toggle transmit mute:** Mute transmit when you want to use WK as a Code Practice Oscillator (CPO) When muted, WK will send CW in sidetone only. In response to this command, the WK will echo an **A** when mute is turned on and an **N** when mute is turned off.

**N [nnnn] - Load 4 Digit Serial Number** All four digits must be entered including leading zeroes. The serial number is played by inserting a play message token /N into a message. The serial number is automatically incremented after playing. See *Embedded Command* section for more details.

**O - Swap Key Output Port:** Each time the **O** command is issued, the key port is toggled back and forth between key port 1 and key port 2. When port 1 is selected, WK responds with a single dit and two dits are echoed when port 2 is selected. If you have two transceivers, this feature frees you from moving cables around when you want to switch radios.

**P [m,d] - Start Practice Mode:** A dual mode multi-level code practice program is built into WK. There are two styles of practice, receive only and echo (receive/respond) practice. There are four levels of practice organized by easy to difficult letter groups. The four levels are:

Level 1: E T A N I M W S G D U K O R  
 Level 2: C Q P J F B V Y H X Z L including level 1  
 Level 3: 1 2 3 4 5 6 7 8 9 0 including level 1 & 2  
 Level 4: ? / , . AR SK BT AS including level 1 & 2 & 3

The syntax for entering a practice command is:

**P m d** where m is R for receive or E for echo practice and d is a single digit 1 to 4 to select difficulty level.

**P R 2** selects Level 2 receive practice

**P E 4** selects Level 4 echo practice

Receive Practice Description: Random characters from the selected level are sent in groups of five. Practice will continue until the command pushbutton is pressed.

Echo Receive/Transmit Practice Description: WK will send a random character from the selected level and you must respond by echoing the character back on the paddles. If you get it right WK will repeat the first character followed by a new character. Now you must echo back both characters. WK will continue to add characters until a set of five characters has been completed successfully. After that it will start over with a new set. If you miss a character WK will respond with an **X** and start over with a new sequence of characters. When you want to end practice, respond with di-dah-di-dah or press the command pushbutton. By default practice will be sent on the active output key port, enable transmit mute (M command) to inhibit this.

A pseudo random letter generator is used to generate the sequence of letters. Normally the seed for the generator is a random number but sometimes it is useful to generate the same sequence over and over during training sessions. Use the P command with an S to set a fixed seed. The format for the command then is P S d where d is a seed value between 1 and 99. A seed value of 0 will return to fully random operation.

**Q - Query WK Current Settings:** WK will respond with current settings sent in the following format:

**WPM** is sent first  
**N** followed by Serial Number  
**M** followed by free msg memory space in letters available  
**C** followed by command WPM  
**W** followed by weight  
**L** followed by lead time  
**T** followed by tail time  
**V** followed by key compensation  
**F** followed by Farnsworth WPM  
**I** followed by Letterspace adjustment  
**J** followed by Paddle Sample Adjust  
**Y** followed by dit/dah ratio  
**B** followed by speed pot min WPM  
**T** followed by speed pot max WPM  
**V** followed by major firmware revision  
**M** followed by minor firmware revision  
 Example: **V31 M12** for Revision 3.1.12

**R [pb] - Review a message without transmitting:** After **R** is entered, WK will respond with an **E**. Press the message button of the message you wish to play. The message will be sent in sidetone only. If you try to play an empty slot, WK will respond with **MT**. Embedded commands will be sent as is without expansion. In other words /S10TEST will be sent as: **DAH-DI-DI-DAH-DIT 10TEST**

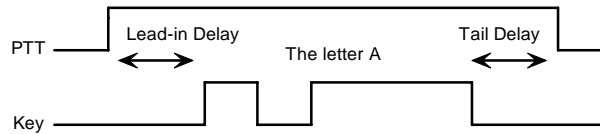
**S [nn] - Set Favorite Speed in WPM:** One value is entered between 5 and 99. This speed is activated when the speed pot is turned fully counterclockwise. The speed pot will act normally above this setting.

The minimum WPM is 5 WPM, the maximum speed is 99 WPM. If the speed is valid, WK responds with an **R**, otherwise a ?

**T [nn] - Set PTT Tail Time:** WK provides a transmitter PTT output that can be used to switch a transmitter or linear amplifier over to transmit mode in advance of actual CW keying. You have control over the time delay between when PTT is asserted and when CW keying will start, this is lead-in. You also have control over how long the transmitter will stay in transmit after keying has stopped; this is tail delay. The tail delay is handled differently for CW sent by paddle and CW sent by message. Paddle delay is controlled by the Hang Time setting while message PTT delay is controlled by the Tail setting. The formula to calculate tail time is:  
 Tail Delay = Three Dit Times + (Tail Setting *times* 10 milliseconds)

Examples:

At 20 WPM, Tail set to 7, Tail Delay =  $(3 \times 60) + (7 \times 10) = 250$  mSec  
 At 40 WPM, Tail set to 7, Tail Delay =  $(3 \times 30) + (7 \times 10) = 160$  mSec  
 At 20 WPM, Tail set to 0, Tail Delay =  $(3 \times 60) + (0 \times 10) = 180$  mSec  
 At 15 WPM, Tail set to 55, Tail Delay =  $(3 \times 80) + (55 \times 10) = 790$  mSec



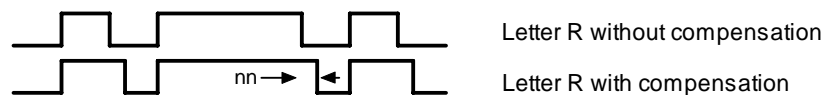
PTT Lead-in and Tail Timing Example

In general we want a very short tail time when sending messages and we want PTT to stay asserted between letters while sending with a paddle set. That's the reason the delay is dictated by two mechanisms.

**U - Turn Autospace Mode Off and On** When autospace is enabled WK will automatically insert proper inter-letter space between letters. Each time the **U** command is issued WK will toggle between modes responding with an **A** for autospace enabled an **N** for autospace disabled.

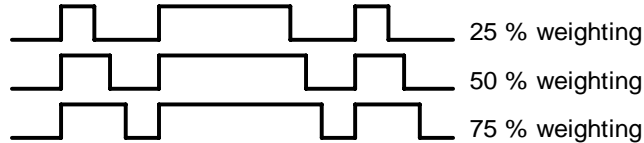
Here is how autospace works: If you pause for more than one dit time between a dit or dah WK will interpret this as a letter-space and will not send the next dit or dah until the letter-space time has been met. The normal letter-space is 3 dit spaces but this can be increased by using the **I** command. WK has a paddle event memory so that you can enter dits or dahs during the inter-letter space and WK will send them as they were entered. With a little practice, autospace will help you to send near perfect Morse. If autospace seems to be less forgiving than you like, you can increase the letterspace setting to relax the timing.

**V [nn] - Keying Compensation** allows a fixed amount of time to be added to the length of all dits and dahs. QSK keying on modern transceivers can cause shortening of these elements which is especially noticeable at high speeds. WK allows the length of the elements to be increased uniformly to compensate for this. The adjustments can be made in one-millisecond steps. The maximum adjustment is 31 mSecs. Key compensation is very similar to Weighting in that any adjustment added to the dits and dahs is subtracted from the spacing so the resulting speed is not changed. The difference between weighting and keying 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. So be careful at high speeds with large values of keying compensation, dits and dahs may run together with no spacing at all.



Key Compensation Timing

**W [nn] - Keying Weight** can be adjusted in percentage from 25% to 75%. When set to 50 % the dit time is equal to the inter-element time, which is normal. Values less than 50 reduce weighting while 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 sound while increased weighting results in a heavier sound. Since weighting tracks speed, a given weighting will sound the same at all speeds.



Three Key Weighting Examples

### X - Enter Extended Command Mode

An additional set of commands are located in a sub menu. These are commands that are not entered very often so an additional paddle action is not a significant issue. Extended commands require two entries, an **X** followed by a sub command. Here is the procedure in detail:

Press and hold the command pushbutton and WK will respond with an **R**  
 Enter an **X** and WK will respond with an **E** (command request)  
 Enter desired Extended Command with additional parameters, if required.

A list of extended commands is provided in the section titled *Extended Command List* found after this section.

**Y [nn] - Set Dit/Dah Ratio:** nn ranges from 33 to 66. Entering **Y 50** sets the standard 1:3 ratio. For example a value of **33** selects a dit/dah ratio of 1:2 while a value of **66** selects 1:4. The ratio formula is:

$$\text{Ratio of 1:N where } N = (nn * 3) / 50 \quad \text{example } nn = 40 \text{ give a ratio of } 1:((40*3)/50) = 1:2.4$$

**Z - Change Sidetone Frequency:** After this command is entered the sidetone oscillator will be keyed at a steady rate with transmit muted. Pressing the paddles will raise or lower the frequency. The range varies nearly continuously from 300 Hz at the low end to 2000Hz on the high end. It takes a while to sweep through the whole range. Pressing the command pushbutton will end this command and store the new sidetone frequency. Like all settings, use the Preserve Settings extended command to save the new sidetone setting in EEPROM.

### Extended Command List

**A – Pushbutton Input Diagnostic:** is used to verify the correct operation of the switch network connected to WK input pin. This design allows four switches and a potentiometer to share a single pin on WK. The state of the network will produce a unique numeric value which WK uses to identify a pushbutton press.

After the Extended **A** command is issued, the numeric network state will be sent in Morse sidetone. For proper pushbutton operation, the input values must fall within the following ranges:

Command PB .....	0 to 40	PB 3 .....	101 to 130
PB 2 .....	41 to 100	PB 4 .....	131 to 160

To stop this diagnostic state, press and hold either paddle until the readout stops.

**C - Toggle Contest Spacing:** When contest spacing is enabled, word space is set to 6 dit times as opposed to the normal 7 dit times. An **A** is sent when CT space is enabled, an **N** is sent when it is turned off.

**E[nn] - First Element Extension:** Forces the first dit or dah of a transmission to be elongated to allow for receive to transmit delay. Enter a value between 0 and 99 mSecs for nn.

**F - Toggle Fast Command Response Time:** Normally WK will enter command mode when the command pushbutton is pressed for 2 seconds. This may be too long for some operators. When fast response is enabled, the delay time is reduced to 1.3 seconds. An **A** is sent when fast is enabled, an **N** is sent when it is turned off.



**G - Report Speed Pot Value:** When this command is issued, WK will respond with the current speed pot value send in Morse sidetone.

**K - Reverse Toggle:** When in RTTY mode, this will toggle the mark and space states. Normally this can be done at the transmitter but this allows an easy way to do it. An **A** is sent when reverse is enabled, an **N** is sent when it is turned off.

**L - Diddle Toggle:** When in RTTY mode, this will toggle the automatic generation of diddle which fills idel time during RTTY transmit. It is particularly effective during paddle RTTY. An **A** is sent when diddle is on, an **N** is sent when diddle is turned off.

**M - Load Callsign:** Use this command to load a stored callsign. It works just like a message entry and the destination is the currently enabled user slot. Each user slot has one unique callsign. Since the callsign is treated like any other message, embedded commands can be included if desired. The only way to play back a callsign is by embedding the **/M** command in a message.

**N - Select Cut Number for number 9:** When enabled, an **N** will be substituted for the number nine when sending a serial number. Command response: an **A** is sent when enabled, an **N** is sent when disabled.

**O- Toggle SO2R Mode:** When SO2R is enabled normal PTT is turned off and instead PTT1 is used to signal which output port is active. When port 1 is active, PTT1 is asserted (radio 1 selected) when port 2 is active, PTT1 is not asserted (radio 2 selected) Note that the PTT1 output is open collector and is on when asserted. Command response: an A is sent when enabled, an N is sent when disabled.

**P - Toggle PTT Mode:** The user can choose whether to use automatic PTT while keying. This is a toggle command and after it is entered, WK will respond with an **A** for PTT enabled, or **N** for PTT disabled. Automatic PTT is disabled if you wanted to control PTT manually by embedded command.

**Q - Report Major and Minor Firmware Revisions:** Upon receipt of the is command WK will send the two revision numbers, Major followed by Minor in Morse Sidetone.

**R [nn nn] - Set Speed Pot minimum and maximum:** After the command is entered WK will prompt for two values, the minimum WPM and the maximum WPM. An error will be flagged if the minimum value entered is greater than the maximum, if the minimum value is less than 5 WPM. Maximum WPM allowed is 99. If the speeds are valid, WK responds with an **R**, otherwise a ?

**S - Preserve Settings in EEPROM:** will respond with a letter **R** to signify that settings were saved successfully.

**T - Select Cut Number for Zero:** When enabled, a **T** will be substituted for the number zero when sending a serial number. An **A** response means this cut is enabled, an **N** response means it is disabled.

**U - Swap Users:** WK provides two complete user configurations, each with a unique callsign. Both message banks are available to both users. This allows a universal set of messages to be created that will work with either user. For example a message like **CQ CQ CQ DE /M /M K** will send the selected user's callsign. In response to this command a single dit is echoed when user 1 is selected and two dits when user 2 is selected.

**V - Voltage Readout:** WK will measure the current supply voltage and send it in Morse sidetone. For example a voltage of 4.52 will be sent as 4r52 with the r indicating the decimal point.

**W - Modify Sidetone Volume:** After this command is entered the sidetone oscillator will be keyed at a steady rate with transmit muted. Pressing the paddles will raise or lower the volume. It really only has two levels. Pressing the command pushbutton will end this command and store the new sidetone frequency. Like all settings, use the Preserve Settings extended command to save the new volume setting in EEPROM..

**X - Toggle paddle swap (Exchange dit/dah inputs):** WK will respond with a letter **A** to signify when paddle swap is enabled and an **N** when turned off.

**Y - RTTY Enable Toggle:** This command will turn RTTY transmit mode off and on. Once RTTY has been enabled, WK will respond to the special RTTY control codes to control RTTY transmitting. See page ?? An **A** is sent when RTTY mode is on and an **N** is sent when RTTY is turned off. When RTTY is enabled, it is not possible to send CW by paddle or message. This setting is ignored when WK enters host mode.

**Z - FSKMAP Toggle:** When in RTTY mode, this will toggle the KEY and PTT assignments to allow flexible FSK and KEY assignments. An **A** is sent when FSKMAP is on, an **N** is sent when FSKMAP is turned off.

## Standalone Message Functionality

Messages are loaded by pressing and holding the command button until WK responds with an **R**. Then press the message pushbutton of the memory slot you wish to enter. When WK is ready to accept a new message it will respond with an **E**. If you wait too long, WK will respond with a **?** and you have to start over. In some keyers, such as WK, only four pushbuttons are provided, messages 5 and 6 are accessed by pressing two pushbuttons in the following sequence after the **R** is echoed:

Press and hold either message button 2 or 3. (this specifies message 5 or 6 respectively)  
Press the command pushbutton.  
Release both push buttons.

(Message 5 and 6 can be played in a similar manner by first pressing and holding pushbutton 2 or 3, then pressing the command pushbutton, and finally releasing both)

After WK responds with an **E**, enter the message directly on the paddles at a steady rate, making sure to leave proper space between letters. To insert a word space simply pause for longer than a word space and WK will respond with an **E** to signify a word space insertion. You can force a wordspace insertion by entering di-di-dah-dah (**IM**). This allows you to put a wordspace at the beginning of a message or insert more than one wordspace in a row. A ½ letterspace pad character can be inserted by entering di-di-dah-dah-dit (**IG**).

When the message has been completely entered, press the command push-button or enter di-dah-di-dah (**AA**) and WK will respond with an **R** to signify that the message was accepted and stored. If you make a mistake while entering a message, press and hold the command pushbutton and WK will backspace through the letters that have been entered. When you reach the position you want, release the button and new letters can then be added after that position. If the message memory becomes full while entering a message, WK will stop further loading, respond with an **F**, and return WK back to non-command mode. There are 240 letters in message memory that can be distributed in any way between twelve message slots. The length of the individual message slots is not fixed. This means, for example, you could have one message of 80 characters, one message with 5 characters, and a third with 10 characters and still have 141 locations left to split among the remaining three slots. Keep in mind that each word spaces, as well as embedded commands, occupy memory locations.

When you stop at the end of a message, WK will usually enter a wordspace before you have a chance to press the command pushbutton. There are cases when you do not want a wordspace at the end, especially if you are using PTT to key an amplifier. This is because WK will hold PTT during that added wordspace which will delay an exchange turnaround. The best way around this is to end a message with the di-dah-di-dah (**AA**) character inserted immediately after the last letter in the message. This ends message entry mode immediately.

If you are having problems loading messages into WK, make sure you leave adequate space between letters and are not sending much faster or slower than current command speed. If, for example, you enter an **A** followed by a **T** and end up with a **N**, you are not allowing enough space between letters. It's a fine line though because if you allow too much space WK will interpret that as a word space. Temporarily lowering the command speed (see command **C**) can help while you learn how the process works.

To play a message back, simply press the desired message button and the message will be sent. There are two ways to abort a message: 1) Press both command and PB4 or 2) Press and hold either paddle.

If you want to review the message without keying the transmitter, use the **R** (review) command. Note that review will expand the play ½ letterspace pad character as di-di-dah-dah-dit and expand embedded commands.

## Standalone 'Two Press' Message Button Functionality

You can trigger message 5 and 6 directly with the following sequence:

- 1) Press and hold PB2 for message 5 or PB3 for message 6
- 2) Press the command PB (you now have two PBs pressed)
- 3) Now release both pushbuttons and selected message will play.

### Standalone Quick Tune Command

This follows a Two Press sequence:

- 1) Press and hold PB4
- 2) Press the command PB (you now have two PBs pressed)
- 3) Now release both pushbuttons and the key and PTT outputs will be enabled.

Pressing the command pushbutton will end tune. You can toggle between 50% and 100% tune duty cycle by hitting the paddles during tune. Tune will be automatically turned off after 99 seconds.

### Standalone Quick Serial Number Decrement

Sometimes during contest operation, a serial number has to be reissued. Since the serial number is automatically incremented when it's played, we need a way to decrement the serial number. This can be done one of two ways, using the **D** paddle command or by pressing and holding the CMD PB and then quickly pressing either paddle. In either case, after the serial number is decremented, WK echoes a single dit.

There is a third way to decrement a serial number, that is with a **/D** command embedded in a message. A message can be built that pre-decrements the serial number before sending it. For example the following two messages can be created:

**Message 1:** 5NN /N QSL ?

**Message 2:** /D/N QSL ?

Message 1 is played as the initial exchange and if the serial number needs to be resent, message 2 would be used. The second message pre-decrements the serial number before sending it. Both of these messages leave the serial number incremented after it is sent.

### Embedded Message Command List

It is an easy procedure to embed commands in a message. The format is the fraction bar **DN** (D and N sent together as one letter) followed by the desired command letter. If you want to insert the DN prosign into a message but don't want it to be interpreted as a command simply enter DN twice.

Example: **K1EL/1** would be entered as **K1EL//1**

### Embedded message command table

<b>/Bnn</b>	Set a beacon cycle time of nn seconds (nn=00 to 99). Put this at the beginning of a message to set the beacon period.
<b>/Cn</b>	Call message in slot n (1-6), return and resume current message.
<b>/D</b>	Decrement serial number.
<b>/E</b>	Toggle message banks
<b>/Hn</b>	Set HSCW speed. See table below for determining n.
<b>/Inn</b>	Increase letterspacing within a message, nn is a value from 0 to 31 percent times 2.
<b>/Knn</b>	Key transmitter for nn seconds. (nn=00 to 99)
<b>/M</b>	Play user callsign.
<b>/N</b>	Play Serial Number with auto post increment.
<b>/On</b>	Select key output port; 1 for port 1 or 2 for port 2.
<b>/P</b>	Pause and wait for paddle entry and then continue after one word space time. The pause is ended three ways 1) paddle entry 2) Press a msg PB (2-6) or 3) Press the cmd PB to cancel.
<b>/Q</b>	Set QRSS speed. See table below for determining n.
<b>/Snn</b>	Set a speed change within a message. (nn=WPM, 5 to 99)
<b>/Un</b>	Turn PTT on or off. PTT is turned on when n = 1 and off when n=0.
<b>/V</b>	Send the current supply voltage in Morse, useful for beacons.
<b>/Wnn</b>	Wait for nn seconds. (nn=00 to 99)
<b>/Xn</b>	Cancel speed override, for example cancel HSCW, QRSS, or /Snn speed.
<b>/Yn</b>	Force a relative speed change up. Add n to the current WPM. n=(0-9)

<b>/Zn</b>	Force a relative speed change down. Subtract <b>n</b> from the current WPM. n=(0-9)		
<b>/1</b>	Jump to message 1	<b>/2</b>	Jump to message 2
<b>/3</b>	Jump to message 3	<b>/4</b>	Jump to message 4
<b>/5</b>	Jump to message 5	<b>/6</b>	Jump to message 6
<b>AA</b>	End message load immediately		<b>(DI-DAH-DI-DAH)</b>
<b>IG</b>	Insert ½ letterspace pad		<b>(DI-DI-DAH-DAH-DIT)</b>
<b>IM</b>	Insert wordspace pad		<b>(DI-DI-DAH-DAH)</b>

## Rate Table

n	HSCW Rate	QRSS Rate
0	1000 lpm (200 wpm)	3 sec dit
1	1500 lpm (300 wpm)	6 sec dit
2	2000 lpm (400 wpm)	10 sec dit
3	3000 lpm (600 wpm)	12 sec dit
4	4000 lpm (800 wpm)	30 sec dit
5	6000 lpm (1200 wpm)	60 sec dit

## Embedded Command Examples

**/B60BCON DE K1EL BEDFORD NH** send BCON DE K1EL BEDFORD NH every 60 seconds

**UR RST IS /P QSL** pause to allow the user to enter the RST then resume automatically

**/K05 /W10 VVV DE K1EL** key down for 5 secs, wait 10 secs, and then send VVV DE K1EL

**CQ CQ CQ DE /M /M /M** send a 3x3 CQ using the user callsign

**/H2CQ CQ DE K1EL K1EL K1EL/S15 DE K1EL** send 1<sup>st</sup> part at 1500 lpm and the 2<sup>nd</sup> at 15 WPM

**CQ CQ CQ DE K1/I10TMT/I00 K** send message with 20% extra space in TMT. A more manageable way of doing this would be to load **K1/I10TMT/I00** in the callsign slot to get this spacing any time /M is used.

**CQ CQ CQ DE /Z4K1EL K1EL K1EL/Y4 K** send the callsigns 4 WPM slower then return to normal WPM.

**QTH IS /E/C1/E NAME IS STEVE** play the QTH from message bank 1 and then return for the name.

**/Q2EL /1** continuously send EL at QRSS10 speed (this message is in slot 1). Avoid inserting a space between the QRSS command and the start of text: /Q2 EL unless you want a long delay at start of message.

**/B10K1EL BCON/W2/VVOLTS** send K1EL BCON, wait 2 secs, send XrXX VOLTS repeat every 10 secs

**SOM/E/C3/E/U1/W5/U0/S50K1EL//1/XEOM** send SOM, swap to msg bank 2, call msg 3, swap back to msg bank 1, turn on PTT for 5 seconds. Change speed to 50 WPM send K1EL/1, cancel 50WPM send EOM.

## Standalone Beacon Operation

There are several ways to run a beacon from WK in standalone mode. The first way is to load a message followed by a delay and jump. This would look like this, loaded into slot one:

**K1EL BCON /W12 /J1**

K1EL BCON will be sent followed by a 12 second delay after which there is a jump back to the beginning.

The second way is to use the embedded beacon command, which is easier:

**/B12K1EL BCON**

Again K1EL BCON will be repeated every 12 seconds, but now the time between beacons will be exactly 12 seconds. In the previous example, the time between beacons would be 12 seconds plus the time it takes to send the message. This is more in line with what most users are looking for.

Finally, the third method is an 'on the fly' beacon. In this mode, any message slot can be sent as a beacon, for example if K1EL BCON is loaded into slot three, the user would press the command button, wait for the R, and then enter **B 3 12**. This will beacon message 3 every 12 seconds. The advantage is that the message does not specifically have to be formatted as a beacon. The disadvantage is you have to enter the beacon paddle command every time you want to send it.

One application of a beacon is to set up a CQ loop that will send CQ, pause for some length of time, and then send it again until someone answers. If the beacon is in the pause state, you can simply answer and this will end the beacon. If the beacon is in the middle of the message, pressing the paddle once will halt the message allowing you to listen and then break in to answer. If you are waiting in the pause state and you want to halt the beacon but not send anything yet, multi-press PB4 and the CMD PB.

## Standalone QRSS/HSCW Operation

The /H and /Q buffered commands allow HSCW or QRSS strings to be sent in standalone mode. (Note that QRSS speeds are not supported in Host Mode) HSCW and QRSS strings can be aborted with either a paddle press or a Command+PB4 pushbutton press. Upon abort normal keying speed is resumed.

## PTT Functionality

PTT in standalone mode works the same way as it does in host mode. Note that in standalone mode the maximum allowable value for paddle entry is 99. This means the largest lead-in or tail delay that can be entered by paddle command is 990 mSec. If you need to enter a longer delay in standalone mode, use the WK3tools application which allows these values to be entered from the host over the USB port.

When entering a message by paddle and minimal tail time is essential, be sure to end all messages with the di-dah-di-dah (**AA**) character to prevent WK from automatically appending a wordspace. If there is a wordspace at the end of a message WK will hold PTT till the end of the wordspace.

## Preserve Settings

WK setting changes made by paddle commands are not automatically copied into permanent storage. That means the settings will be lost if power is cycled. To save the settings permanently, use extended **S** command. Press and hold the command PB until the **R** and enter an **X** followed by a **S**. This will save current settings in nonvolatile memory so that all settings will be retained on power cycling. Note that messages are saved directly into nonvolatile memory so the **S** command is not required to preserve messages.

## Restore User Defaults (Warm Restart)

It's possible that a paddle command could be entered incorrectly which puts WK into an undesired state. Rather than try to figure out what went wrong, an easier way to handle this is restore settings from EEPROM. You can do this by either cycling power or by forcing a warm restart. To do the later, first press and hold the command pushbutton. An **R** will be sent followed by a **MN** (dah-dah-dah-dit) 5 seconds later which lets you know user defaults have been reloaded.

## Restore Factory Defaults

Refer to Page 13 for a description of how to reset factory settings.

## Keyer Lock

A lock feature is provided to disable paddle input and message pushbuttons. This is useful when you want to pack up a battery powered keyer and insure that it stays off with batteries still connected. It is also handy to lock the keyer paddles to keep little hands from sending "messages". While the keyer is locked it is held in low power shutdown mode and draws about 10 uA.. To lock the keyer, press the command pushbutton, wait for the R, and then enter a period (di-dah-di-dah-di-dah). WK will respond with an L to let you know it is now in a locked state. To unlock the keyer, press and hold the both paddles closed for about 8 seconds and WK will wake up and send an R. This is a very reliable lock mechanism because once the paddles are removed there is no possible way to accidentally unlock WK.

## Shortcuts and Speed Ups

Fast Command responses – WK uses **A** for positive responses instead of an **R** or **Y**

Cut Numbers on Command Entry – When entering numeric values use cut numbers to save time. For example when changing the command speed to 19 WPM, use **S U N** instead of **S 2 9**

These are the command cuts used in WK: T=0, A=1, U=2, V=3, 4=4, 5=5, 6=6, B=7, D=8, N=9

Single digit number entry – If you are entering a parameter that usually requires two digits but you only need to enter one, just enter the single digit and WK will figure out that there is only one. For example instead of entering **T 0 7** you can simply enter **T 7**.

WK will respond with an **E** or **I** when swapping users, message banks, or output port. Think of it as a single dit for 1 and a double dit for 2. This is much faster than responding with a Morse 1 or 2.

Fast Message Entry – Just press the command button till you get an **R** then simply press the message button you want to load. To stop load mode, either press the command button or enter di-dah-di-dah.

Fast Tune Mode – Press and hold pushbutton 4 and then press the command button to start tune. Press the command button to end tune.

Fast Serial Number decrement – Press the command button and then press either paddle to decrement the serial number. WK echoes a single dit to signify that the command was accepted.

Fast Command Response – If you think WK takes too long to respond to the command button, enable fast command reply mode which halves the delay time. This is the extended command **F**.

Command Prompts – WK will respond to commands that require additional input with an **E**. The E stands for “Entry Required” and is fast and efficient. For example if you enter the Weight command **W**, WK will respond with an **E** to let you know it is waiting for the value.

## WKUSB Standalone Tutorial

On 1<sup>st</sup> time power up, or by the restore factory defaults command, WK will be reset to these settings:

Operating WPM:15	Command WPM:15	Sidetone: 800 Hz	Weight:50%
KeyComp:0	Letterspace Adjust:0	SampleAdjust:50%	KeyMode:Iambic B
Sidetone:On	Autospace:Off	Key/PTT: Port 1	Serial Number:0001
Min Speed Pot: 5 WPM	Max Speed Pot: 35 WPM	SO2R Mode: Off	No 0 or 9 Cuts
User: 1	Message Bank: 1	Tune Duty Cycle: 100%	Farnsworth: Disabled

After WK has been powered up, pressing the paddle keys will generate dits and dahs both in sidetone and keyed output. Let's enter a simple command to change the sidetone frequency. Press and hold the command pushbutton (CMD PB) until WK answers with an **R**. Then, without hesitation, enter a **Z** on the paddles. If it did not understand the command or you are late entering the command, WK will respond with a question mark. After a successful command entry, WK will respond by turning on a constant sidetone with keyed output disabled. By pressing the paddles right or left you can raise or lower the sidetone frequency. When you are satisfied with the pitch, end the command by pressing the command PB. WKUSB uses a small speaker and you'll find that higher frequencies produce the loudest sidetone.

Now we will enter an extended command, one that will swap the paddles. Press and hold the command pushbutton (CMD PB) until WK answers with an **R**. and then, without hesitation enter an **X** on the paddles. WK will answer with an **E** to let you know it is waiting for an extended command entry. In this case, enter another **X** which is the swap command. WK will reply with an **A** letting you know that paddle swap is now en**A**bled and the dit and dah paddles have been reversed. To swap the paddles back, repeat the command and this time WK will respond with an **N** for **N**ot enabled.

Now try changing the keying mode. Enter the **K** command followed by a letter signifying a keying mode (see page 21). Here are three examples, **K B** will set iambic mode B, **K A** will set iambic mode A, or **K U** will set Ultimate mode. You might want to experiment with a couple of the modes just to try them. The sample adjust command allows you to tweak the paddle sample point; the default is a value of 50. Setting the sample point to zero will disable both dit and dah paddle memories.

Let's learn about the speed pot. If you turn the speed pot while sending you will notice that the sending speed will change. It will go faster clockwise and slower counterclockwise. When the pot is turned all the way counterclockwise, favorite speed is selected. Favorite speed is set by the **S** command. To change this, press the CMD PB, wait for the **R**, and then enter **S**. WK will respond with an **E** telling you it's waiting for you to enter the favorite WPM. Enter a **1** followed by a **0** to set 10 WPM. Now you will get 10 WPM fully counterclockwise.

The default range for the speed pot is 5 to 35 WPM. Let's change the range using the extended **R** command (see page 40). Let's set a 10 WPM minimum and a 25 WPM maximum. Press the CMD PB, wait for the **R**, and then enter **X**, wait for the **E** and then enter **R** for pot range. WK will respond with an **E** telling you it's waiting for you to enter the minimum WPM value. Enter a **1** followed by a **0**. WK will then send another **E** while it waits for the maximum WPM value, enter a **2** followed by a **5**. If you did it right, WK will answer with an **R** and now the speed range will be 10 WPM to 25 WPM. Repeat the command and use the **T** shortcut for the zero, in other words enter a **1** followed by a **T**.

The WPM rate used while entering commands can be changed using the **C** command.

The Weight, Keying Compensation, Letterspace, and Dit/Dah Ratio commands adjust the way Morse is generated. Read each command descriptions on pages 21, 23, and 24 to understand how they work.

You can save your settings in WK's internal EEPROM with the extended command **S** for **Save** settings.

You can change the keyer's output port with the **O** command. This allows you to key one of two radios from the same keyer. This saves swapping cables around when you want to move from one radio to another. It does require a separate keying cable from WKUSB to each keyer.

Now try entering some messages. Review the procedure for message loading on page 25. WK has two great features associated with messages. The first is backspace, if you make a mistake while entering a message you can backspace to fix the error. The second is that the size of the message slots is not fixed. If you only use two bytes in slot one, only two bytes of message memory are used up, not an entire slot. Once you have mastered message loading you can tackle some embedded commands. An easy command to start with is the speed change command. In slot one enter: **/S10SLOW /S25FAST**. Note that the DN prosign (/) is used as the command identifier. This message will play at two different speeds. Note that after playing this message the operating speed will be returned to the original speed, in other words the speed change is not permanent. If you need to insert / into a message simply enter it twice, this tells WK it is not a command: **K1EL//1**

Another set of speed related embedded commands are **/Y** and **/Z**. These invoke a relative increase or decrease in sending rate. They are different than the **/S** command since they add or subtract from the current speed. That means you can move you speed pot around to different speeds but the relationship between the current speed and the **/Y** and **/Z** commands remains constant. For example, look at the message:

**/Y5CQ TEST CQ TEST DE /Z5 WB6JJB.**

The CQ TEST CQ TEST portion will be sent 5 WPM faster than the current WPM rate and then slowed back down for the callsign portion. If the current WPM rate was 10 WPM the accelerated rate would be 10+5=15 WPM. If the current WPM rate was 23 WPM, the accelerated rate would be 28 WPM. The message would have the same affect in both cases; to send the CQ portion of the message 5 WPM faster than the callsign portion. This feature can be helpful to emphasize a portion of a message by slowing it down.

Now let's compose a beacon message. In message slot 2 enter: **/B60/K05 BCON DE K1EL NH** When this message is played, the keyer will key down for 5 seconds and then send BCON DE K1EL NH. The **/B60** command tells the keyer to repeat the message every 60 seconds no matter how long the message itself is. To cancel a beacon, press the CMD PB or the paddle. WK will stop the loop and respond with an **I** to let you know something was cancelled.

Serial numbering is easy to use. First enter a starting serial number with the **N** command. You need to enter all four digits including leading zeroes. Next select the way you want WK to send 0s and 9s in a serial number. Use the extended **T** and **N** commands for this. To play a serial number simply insert a **/N** command into a message. It's tricky though since the serial number is automatically incremented every time it is played. If you want to send a serial number twice in the same message you have to decrement it after you send it like this: **UR NR /N/D /N QSL ?**

Let's look at the embedded Pause command. The **/P** command can be thought of as a three way branch. When it is encountered, WK will wait for one of three things to occur:

- 1st branch: User paddles something and the message will continue
- 2nd branch: User presses a message button to play a message
- 3rd branch: User dual-presses CMD+PB4 to cancel the message entirely.



Here's a set of two messages that will illustrate one way the **/P** pause command can be used.

In slot 2 enter: **CQ TEST DE K1EL K /P UR NR/N QSL?/P**

In slot 3 enter: **UR NR /D/N QSL?**

When you press PB 2, the message in slot 2 will send CQ and then pause to let you listen for a reply. If there is no reply, hit message PB 2 again to repeat the CQ. If there is a reply, enter the station's callsign and WK will automatically continue, send the serial number, and then pause again. If the station needs a repeat of the serial number press PB 3 to play the message in slot 3. Since the serial number is incremented after the **/N** command in message 2, you need to pre-decrement it in order to send the correct number. Message 3 can be repeated until the station acknowledges the exchange. At that time press PB2 to start the whole process again.

WK supports two alternate sending formats. They are selected by embedded commands in a message. QRSS is extremely slow CW for VLF work, and HSCW is extremely fast CW typically used for meteor scatter QSOs.

Here are examples of each:

QRSS: **/K10/ Q2EL/2** Key down for 10 seconds followed by EL at QRSS6 rate then repeat.

HSCW: **/H3K1EL K1EL K1EL K1EL K1EL K1EL K1EL /1** Callsign in slot 1 is repeated at 3000 lpm

## WK Standalone RTTY Tutorial

RTTY capability was introduced in WK version 3.1. Besides the ability to send RTTY in host mode it can also operate RTTY in standalone mode with input from a paddleset or message pushbuttons.

To get started in RTTY mode, enter a set of useful messages into WK's message slots. No special characters are required for a RTTY message. Next, put WK into RTTY mode with the extended command **Y**. Depending on your setup requirements, you may need to issue a mark/space reverse toggle command (extended command **K**). In addition, you may need to issue the FSKMAP toggle command (extended command **Z**) to get your KEY and PTT outputs configured correctly. Usually, WK's KEY output is fed to your transceiver's KEY input to key the transmitter and PTT is fed to the transceiver's FSK input to generate FSK RTTY. The function of KEY and PTT can be reversed by FSKMAP toggle if need be. It's very important to set an appropriate PTT lead in and tail delay. This prevents FSK from being chopped at the start and end of transmission. A good starting value is 50 for both lead in and tail. Once the transceiver is configured to operate in RTTY mode you are ready to go. The only other configuration decision is whether diddle should be on or off. The recommendation is to use diddle to fill in the empty space between paddle input letters.

To start a RTTY transmission, enter the Morse code di-dah-di-dah. This will key your transceiver and start FSK diddle. You can then enter letters on the paddle and they will be sent in FSK Baudot. WK's Morse sidetone will be on so you can hear what you enter but you won't be able to hear the RTTY transmission unless you can monitor it through you transceiver. It can be confusing but some ops like to hear both. It is possible to overrun the RTTY input buffer if you send Morse very fast, about 20 - 25 WPM works well. To insert a CRLF, enter the Morse code DAH-DAH-DAH-DAH. For tuning or testing, an RYRYRY string can be generated by entering the Morse code DAH-DAH-DAH-DIT. When you are done sending enter another di-dah-di-dah will unkey the transceiver and return you to receive mode.

If you want to send one of your messages, simply press the message button and the transmit process will proceed automatically. You do not need to insert the di-dah-di-dah codes into a message WK will automatically do that for you with message sending. Remember that you have two banks of six messages to work with.

Some ops like to set up one of WK's users for RTTY and the other for regular CW. That way you can switch back and forth with a simple user toggle command. If you use some for thought you can load messages that will work for both modes.

While paddle RTTY is not as easy as keyboard, it does provide a way of operating RTTY with minimal equipment, especially if you have a transceiver that will decode and display received RTTY.

## WK3 Standalone Commands

A	- Select sidetone on or off	N	- Load 4 digit serial number
B	- Start Easy Beacon	O	- Select output key port
C	- Set command speed in WPM	P	- Start Practice Mode
D	- Decrement serial number	Q	- Query current settings
E	- Swap active message bank	R	- Review message without transmitting
F	- Set Farnsworth Speed	S	- Set favorite WPM
G	- Toggle 50% tune duty cycle	T	- Load PTT Tail Time
H	- Set Paddle Hang Time	U	- Select Autospacing on/off
I	- Set Letterspace Adjustment	V	- Set Keying compensation in mSec
J	- Set Paddle sensitivity	W	- Set Key Weight
K	- Select keyer mode	X	- Enter an Extended Command
L	- Set PTT Lead In Time	Y	- Set Dit/Dah Ratio
M	- Mute Transmit, Tx sidetone only	Z	- Set sidetone frequency

## Extended Command Table

A	- Push button diagnostic	P	- PTT enable on/off
C	- Contest spacing on/off	Q	- Report Firmware revisions
D	- Favorite Speed mode on/off	R	- Set speed pot range (Min/Max)
E	- Set first extent	S	- Save settings in EEPROM
F	- Fast command response on/off	T	- Cut 0 on/off
G	- Report Speed Pot Value	U	- Select user 1 or 2
K	- Toggle RTTY Reverse Enable	V	- Report power supply voltage
L	- Toggle RTTY Diddle Enable	W	- Modify Sidetone volume
M	- Load user callsign	X	- Swap paddles
N	- Cut 9 on/off	Y	- Toggle RTTY Mode Enable
O	- SO2R Enable on/off	Z	- Toggle RTTY FSKMAP

## Standalone Embedded Command Table

/Bnn	Set a beacon cycle time of nn seconds (nn=00 to 99).		
/Cn	Call message and then return		
/D	Decrement serial number.		
/E	Swap message banks		
/Hn	Set HSCW speed. See table below for determining n.		
/Inn	Set Letterspace increase (nn=00 to 31)		
/Knn	Key transmitter for nn seconds (nn=00 to 99)		
/M	Play User Callsign		
/N	Play Serial Number with auto increment.		
/O	Keying Port Toggle command		
/P	Pause and wait for paddle entry and then continue after one word space time. The pause can be ended in three ways: 1) Paddle Press, 2) Press a message PB 3) Dual-press CMD+PB4 to cancel.		
/Qn	Set QRSS speed. See table below for determining n		
/R	Merge next two letters into a prosign		
/Snn	Set a new sending speed (nn=WPM, 5 to 59)		
/Un	PTT Off/On command (n=0 or 1)		
/V	Play supply voltage		
/Wnn	Wait for nn seconds (nn=00 to 99)		
/X	Cancel /S, /H, or /Q command		
/Yn	Relative WPM Change Up (n=0 to 9) current WPM changed to WPM+n		
/Zn	Relative WPM Change Down (n=0 to 9) current WPM changed to WPM-n		
/1	Jump to message 1	/2	Jump to message 2
/3	Jump to message 3	/4	Jump to message 4
/5	Jump to message 5	/6	Jump to message 6

n	HSCW Rate	QRSS Rate
0	1000 lpm (200 wpm)	3 sec dit
1	1500 lpm (300 wpm)	6 sec dit
2	2000 lpm (400 wpm)	10 sec dit
3	3000 lpm (600 wpm)	12 sec dit
4	4000 lpm (800 wpm)	30 sec dit
5	6000 lpm (1200 wpm)	60 sec dit

Rate Table for /Hn or /Qn commands

## Topical Index

Introduction	2
What is WinKeyer and why do I need it ?	2
How do I determine if my WKUSB is a Rev C ?	2
WKUSB Rev C Features	2
Let's Get Started ! - USB drivers	3
Let's actually run WKUSB !	4
WKscan Tutorial	4
WKdemo tutorial	4
WK3tools application	5
Upgrading WKUSB Firmware	7
Connecting WKUSB-SMT to your station	9
A cheap path to a keying cable	10
Troubleshooting hints	10
Theory of Operation	12
Paddle Input Priority	13
USB Sense	13
Serial Baud Rate	13
Reset WK3/Restore Factory Defaults	13
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 WKUSB to other applications	33
Support and Warranty	33
Revision History	33
Appendix A - WK Version 3.1 Firmware	34
RTTY Feature Set	34
RTTY Host Mode	34
RTTY Standalone Mode	34
WK3 RTTY standalone commands	34
Using WK3demo in RTTY mode	35
Other Changes for Version 3.1	35

<b>K1EL WK Applications</b>	<b>35</b>
<b>Appendix B - Installing the CH340 USB Driver for WKmini</b>	<b>36</b>
<b>Appendix C – WKUSB USB COM port assignment information</b>	<b>37</b>
Win 7 and newer Com Port Viewing and Reassignment.....	37
WinXP Com Port Viewing and Reassignment.....	38
<b>Appendix D – WinKeyer Settings Defined</b>	<b>40</b>
Weighting.....	40
Keying Compensation.....	40
Letterspace.....	40
Paddle Sample Delay.....	40
Dit/Dah Ratio.....	41
PTT Lead and Tail Times.....	41
Paddle Watchdog.....	42
Paddle Swap.....	42
Paddle Only Sidetone.....	42
Paddle Mute.....	42
Paddle Echo back.....	42
Serial Echoback.....	42
Autospace.....	42
Contest Spacing.....	42
Paddle Keyed Modes.....	42
Speed Pot Lock.....	43
Speed Pot Minimum and Maximum.....	43
Sidetone Enable.....	43
Sidetone Frequency.....	43
SO2R Enable.....	43
<b>Appendix E WKUSB (WK3) Standalone Mode</b>	<b>44</b>
Standalone Command Mode.....	44
Standalone Command List.....	44
Extended Command List.....	48
Standalone Message Functionality.....	50
Standalone 'Two Press' Message Button Functionality.....	50
Standalone Quick Tune Command.....	51
Standalone Quick Serial Number Decrement.....	51
Embedded Message Command List.....	51
Embedded message command table.....	51
Embedded Command Examples.....	53
Standalone Beacon Operation.....	53
Standalone QRSS/HSCW Operation.....	54

---

PTT Functionality .....	54
Preserve Settings.....	54
Restore User Defaults (Warm Restart) .....	54
Restore Factory Defaults .....	54
Keyer Lock .....	54
Shortcuts and Speed Ups .....	54
WKUSB Standalone Tutorial.....	55
WK Standalone RTTY Tutorial.....	57
<b>WK3 Standalone Commands</b> .....	<b>58</b>
<b>Extended Command Table</b> .....	<b>58</b>