FUNTROICS K3NG Keyer

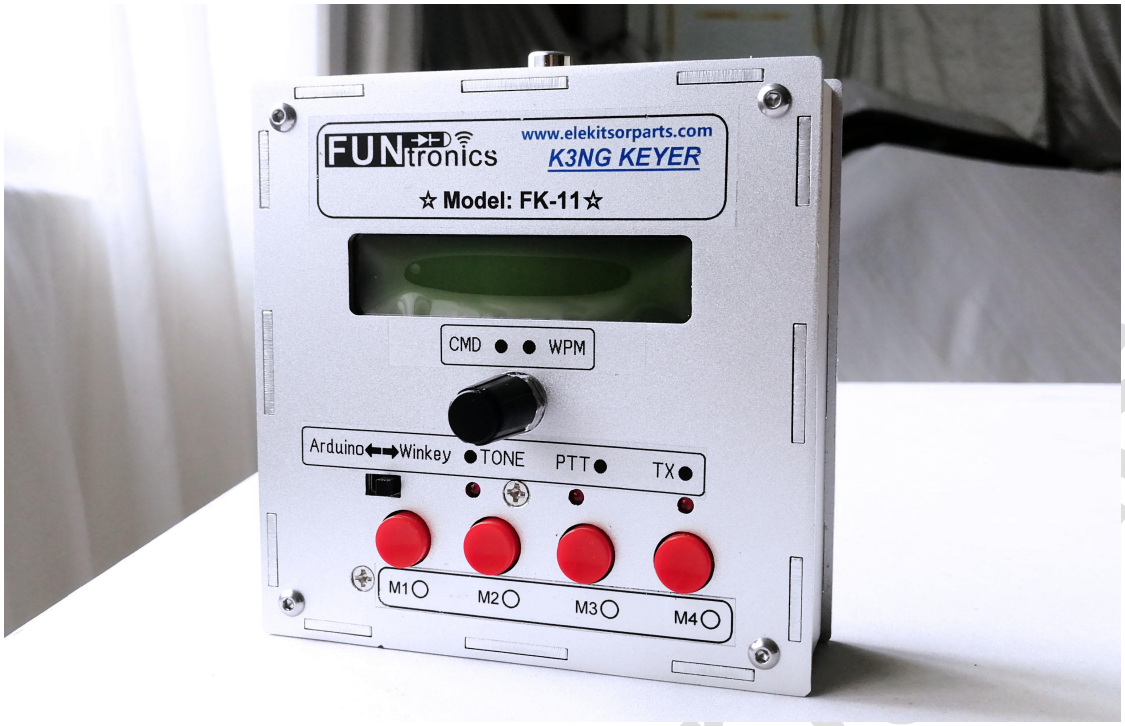
FK-11

User's Manual

Rev 1.0

Elekitsorparts Store

www.elekitsorparts.com



1. Description

This is an open source Arduino based CW (Morse Code) keyer with a lot of features and flexibility, rivaling other commercial keyers which often cost significantly more. The Arduino code comes from K3NG, and it's an open source code so you can fully customize it to fit your needs and also perhaps learn from it or find coding ideas for other projects. We redesigned the PCB, which could be fit into a compact aluminum enclosure, and trimmed the code so as to fit the features of AVR mega 2560. The Funtronics K3NG Keyer integrates nearly all of the features & functions of the K3NG's arduino code, such as LCD display, USB keyboard, rotary encoder and CW decoder etc. Moreover, there are some spare IOs left on the pcb, which could be left for the feature new features & functions.

2. Features

- CW speed adjustable from 1 to 999 WPM
- Programming and interfacing via USB port ("command line interface")
- USB Keyboard Interface for CW keyboard operation without a computer
- Logging and Contest Program Interfacing via K1EL Winkey 1.0 and 2.0 interface protocol emulation
- PTT outputs with configurable lead, tail, and hang times
- 1602 LCD Display
- 3 memories with macros
- Serial numbers
- CW keyboard (via a terminal server program like Putty or the Arduino Serial program)
- Speed potentiometer (optional – speed also adjustable with commands)
- QRSS and HSCW
- Beacon / Fox mode
- Iambic A and B
- Straight key support
- Ultimatic mode
- Bug mode
- CMOS Super Keyer Iambic B Timing

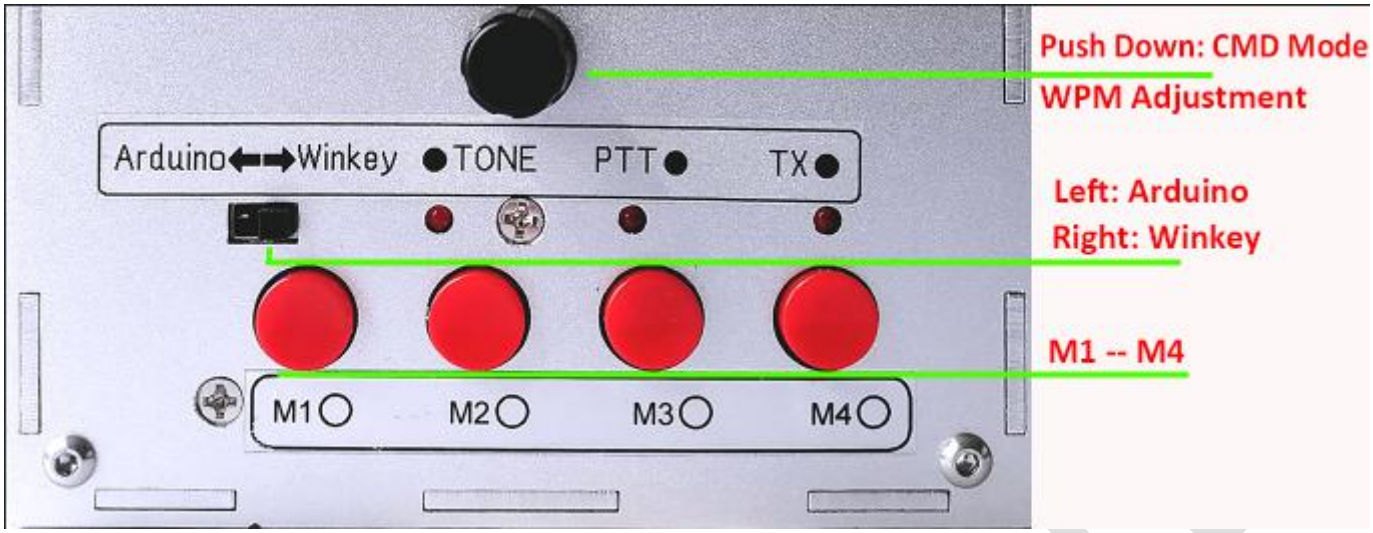
- Paddle reverse
- Hellschreiber mode (keyboard sending, memory macro, beacon)
- Farnsworth Timing
- Adjustable frequency sidetone
- Sidetone disable / sidetone high/low output for keying outboard audio oscillator
- Command mode for using the paddle to change settings, program memories, etc.
- Keying Compensation
- Dah to Dit Ratio adjustment
- Weighting
- Callsign receive practice
- Send practice
- Memory stacking
- “Dead Operator Watchdog”
- Autospace
- Wordspace Adjustment
- Pre-configured and Custom Prosigns
- Non-volatile storage of most settings
- Modular code design allowing selection of features and easy code modification
- Non-English Character Support
- CW Receive Decoder
- Rotary Encoder Speed Control
- Sleep Mode
- USB Mouse Support
- Alphabet Sending Practice
- QLF / “Messy” Straight Key Emulation
- USB Keyboard HID (Human Interface Device) Interface (Keyer = keyboard for your computer)
- Web Interface (UNDER DEVELOPMENT)

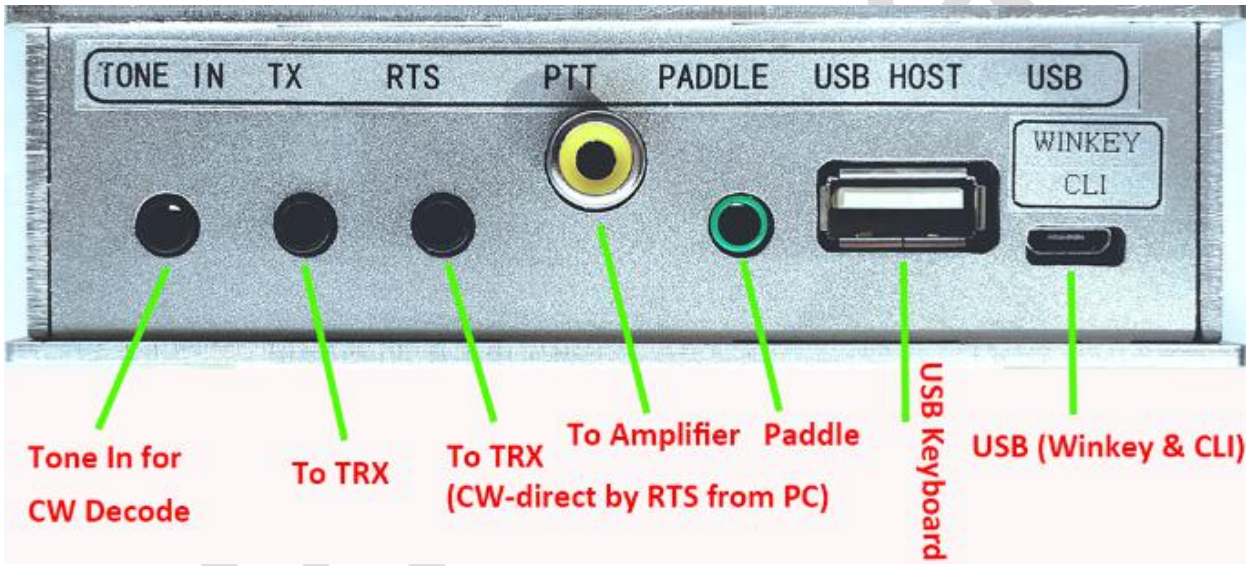
3. Power Supply

There is no DC port on the keyer, to power the keyer on, just plug in a micro-usb cable and connect it to your PC or a cell-phone power bank. The internal buzzer will beep “dit-dit-dit-dit, dit-dit” when the unit successfully starts up, and the “K3NG Keyer hi” will be displayed on the LCD.

There are 2 USB ports on the unit, either one is able to power the unit on.

4. Front and Rear Panels





Winkey & Arduino: Toggle the switch to left for Winkey emulation, and toggle the switch to right for arduino firmware uploading.

CMD & WPM: Rotate the encoder will adjust the WPM + or -, and push the encoder down will enter the CMD mode, there is

a beep from the buzzer while you pushing down the encoder.
TONE IN: This is for CW audio in for the internal cw decoder. The decoded codes will be displayed on the LCD. This is an experimental feature.
TX: To your transceiver's KEY IN port by a 3.5mm stereo cable.
RTS: To your transceiver's KEY IN port by a 3.5mm stereo cable. It's a new feature of this K3NG keyer, the CW signal is controlled by the USB-TTL's RTS pin, and you can define the RTS as a CW signal in the contesting softwares.
PTT: Connect this port to your amplifier by a RCA cable. It will activate your amplifier to transmit when it's in TX.
PADDLE: Connect your paddle to this port by a 3.5mm stereo cable.
USB HOST: When a standard USB keyboard is connected, you can send the CW code by the PC USB Keyboard.
USB: The unit is powered from the USB port. There are 2 USB ports on the unit, and either of them could power the unit on. However, the rear USB is the main USB, and the front USB is the sub. When doing the winkey emulation, the rear USB should be used. When working with the CLI command mode, the front USB is recommended. You can use the rear USB for the CLI also, but you need to activate the rear CLI function before doing it. To activate it, just hold down the CMD switch while resetting or powering the unit on, it will go into the CLI mode. ((If the default is K1EL Winkey emulation, it will go into CLI mode, and vice versa).

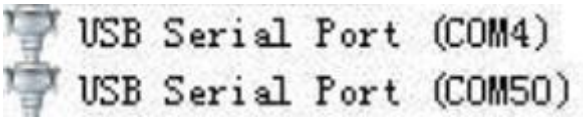
5. Winkey Emulation

The K3NG keyer supports winkey 1.0 & 2.0 emulation, and it is compatible with many contest & logging softwares. The following functions are currently implemented in the emulation:
Currently the following functions are implemented:

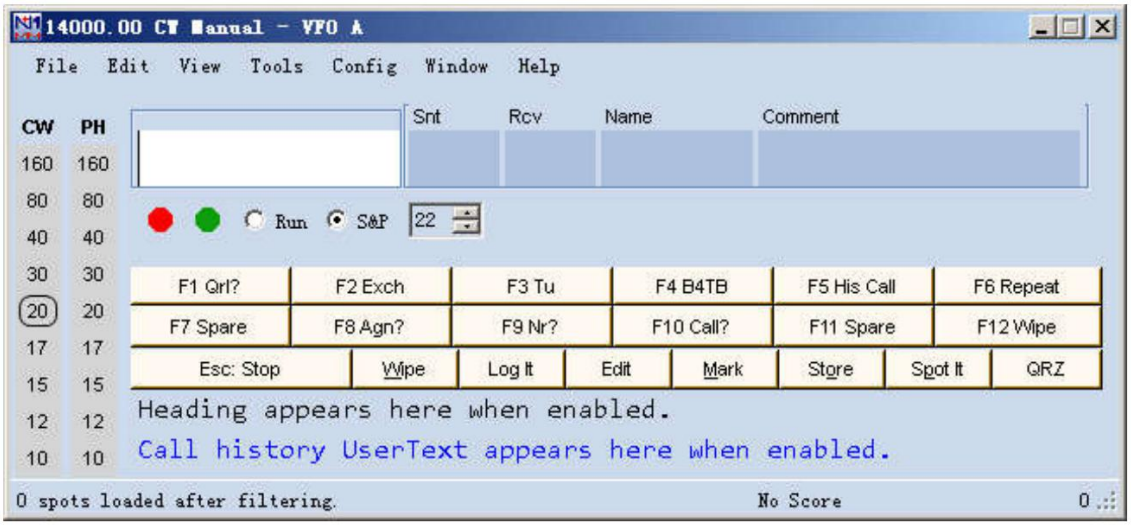
- CW Sending (of course)
- Pause
- Key Down
- Unbuffered and Buffer Speed Setting
- Iambic A & B / Bug Mode Settings
- Ultimatic in normal, dit priority, and dah priority modes
- Farnsworth
- Pointer Operations
- Backspace
- Sidetone Frequency Settings (Winkey 1, 2, and custom frequencies)
- Paddle Reverse
- Paddle Watchdog
- Keying Compensation
- Dit to Dah Ratio
- Contest Wordspace
- Autospace
- PTT Lead, Tail, and Hang Time
- Speed Pot Setup and Query
- First Extension
- Software Paddle
- Weighting
- HSCW
- Serial Echoback
- Prosigns

- Dual transmitter keying lines
- Paddle Only Sidetone
- Memory button reporting
- Standalone Message Sending

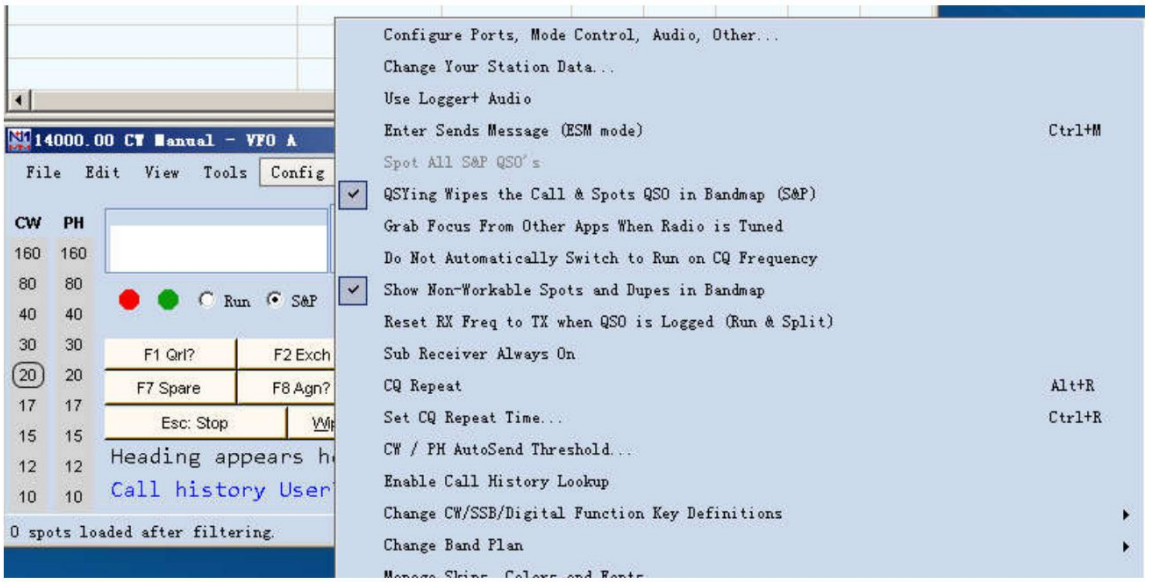
Connect the rear USB port to your PC and install the CH340 USB driver. Open your PC's device manager and note down the serial port number, such as COM4, COM6 etc. The settings for winkey could vary from different contest & logging softwares. Here we take the N1MM as an example. For other softwares, please consult the software developer or the online documents.



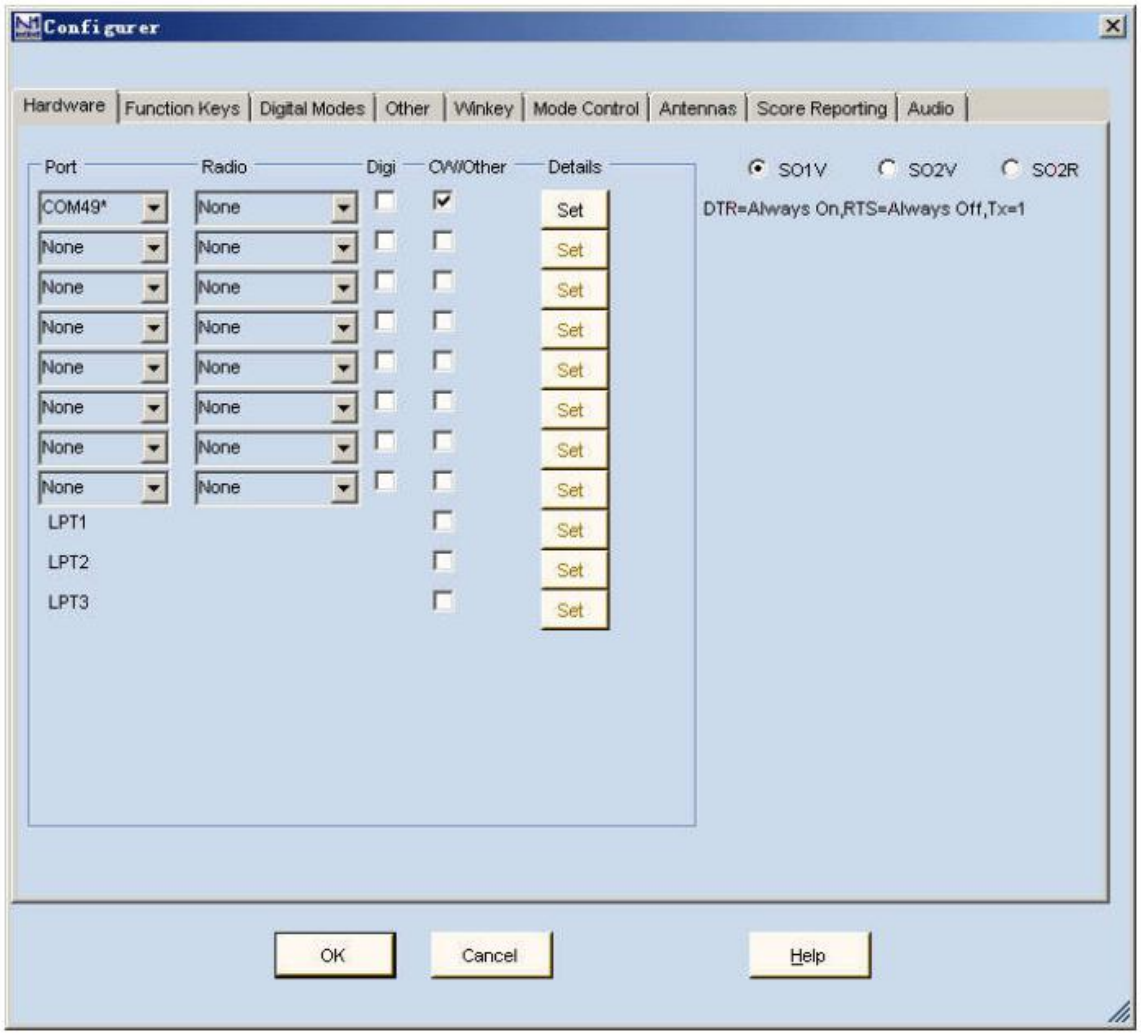
1. Open and Load the N1MM.



2. On the menu bar, click the "Config", and then click the "Configure Ports, Mode Control, Audio, Other…"

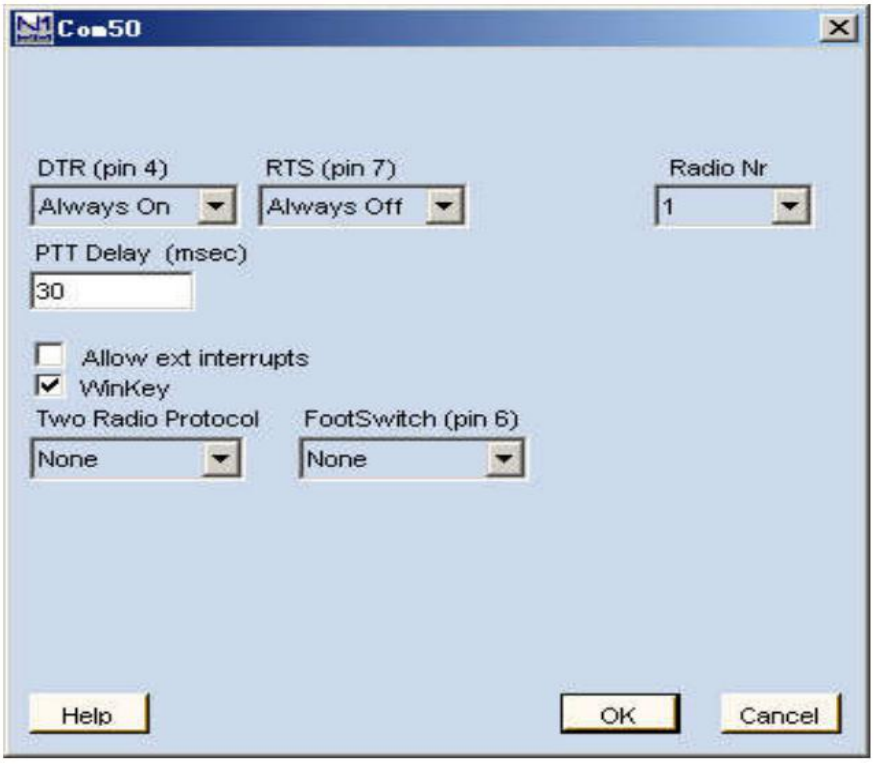


3. Here we see a new window with many tabs for configuration.

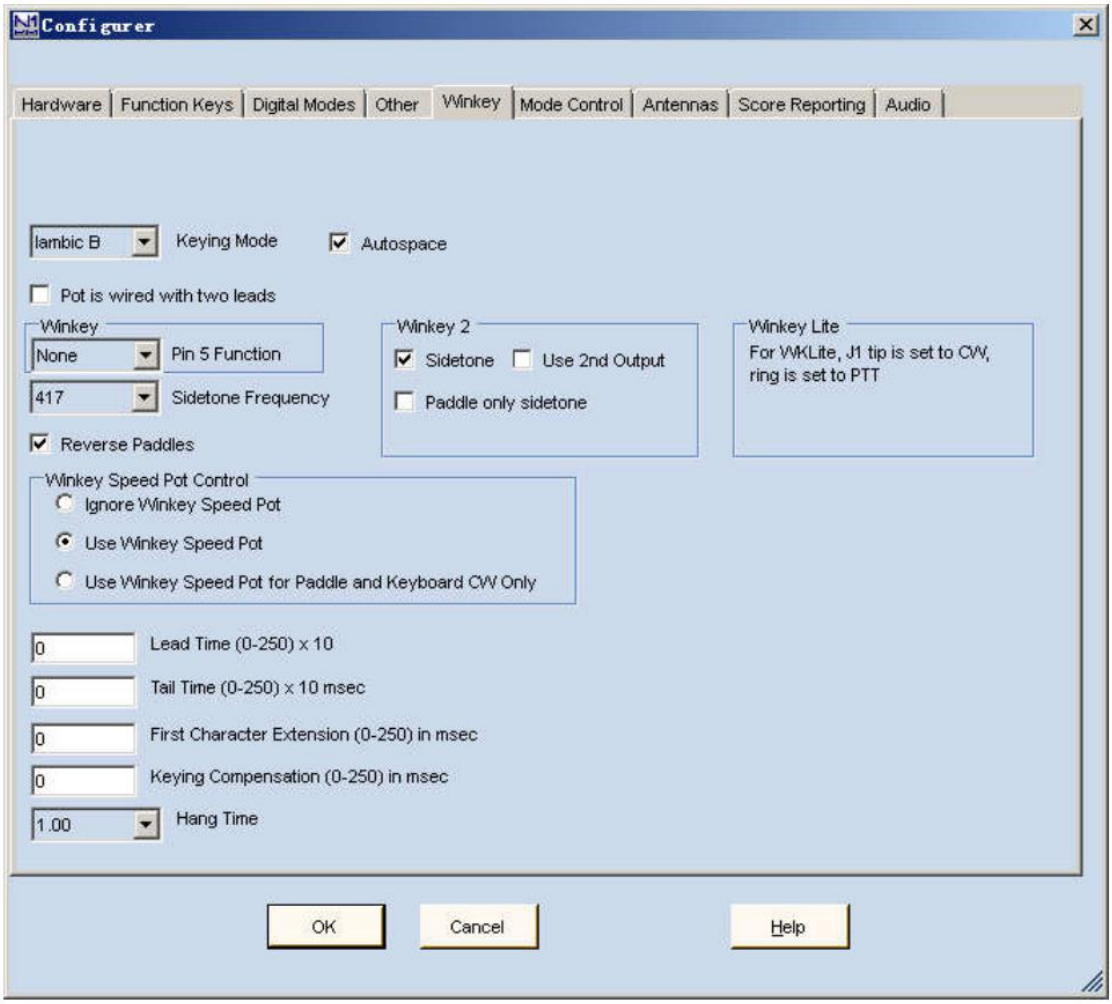


4. Choose the “Hardware” tab, set the port to your USB-serial com port number, which was noted down in the previous steps, for example COM50. Set the “Radio” column to “NONE”. Un-tick the “Digi” while ticking the “CW/Other”, and the final settings should be like this:

5. Click “Set” in the “Details” column, and here comes a new prompting window. Tick the “Winkey” option, and click OK to return to the main “Configurer” window.

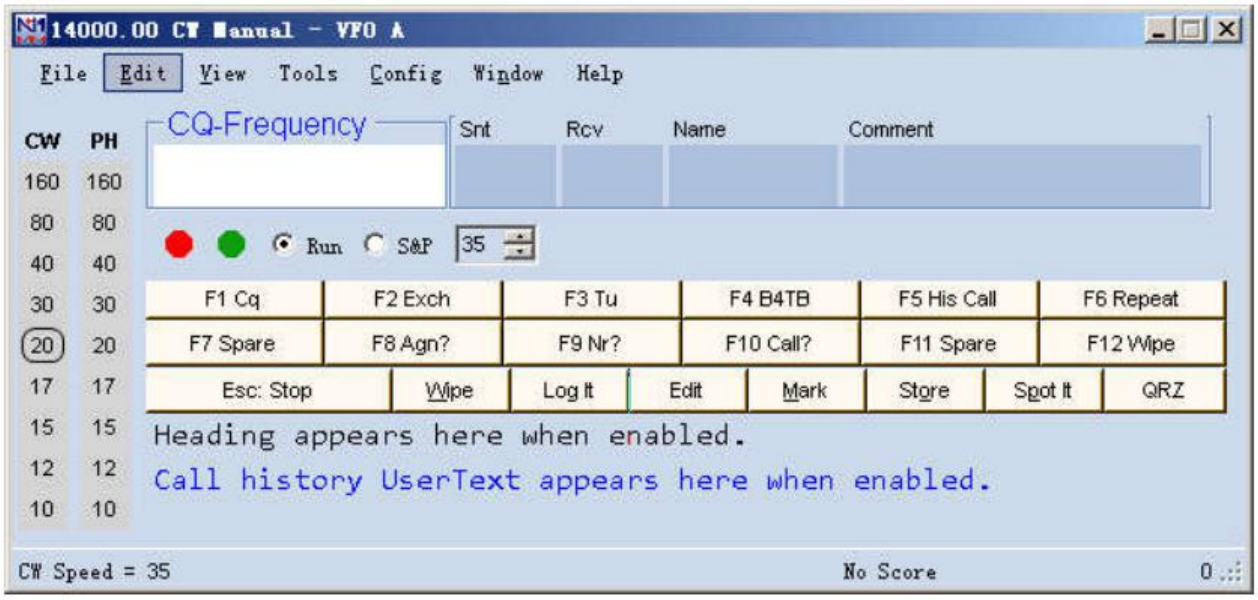


6. Select the “Winkey” tab to enter the winkey setup, and configure all of the settings as the following:



- **Keying Mode:** lambic A / lambic B /Utimatic /Semi-Automatic, any one is OK, but it should be coincide with your K3NG keyer's keying mode. If you are not sure of your keyer's keying mode, please use CLI command to check or configure it.
- **Autospace:** Recommend ticking it up, especially useful when working with some old radios which do not have the T/R delay settings.
- **Pot is wired with two leads:** Un-tick it. The early winkeys used a pot with 2 wires to make the WPM adjustment.
- **Pin 5 Function:** NONE
- **Sidetone Frequency:** Sidetone frequency settings, and lower frequencies are recommended for protecting your hearing, especially in a long time contest.
- **Reverse Paddles:** Paddle reverse setting.
- **Sidetone:** Ticking this option will enable the sidetone output, vice versa.
- **Use 2nd Output:** Un-tick it, there is no second out in this keyer.
- **Paddle only Sidetone:** Ticking it will disable all other sidetones output except the paddle.
- **Ignore Winkey Speed Pot:**: To disable the winkey speed pot for wpm adjustment.
- **Use Winkey Speed Pot:** To enable the winkey speed pot for wpm adjustment.
- **Use Winkey Speed Pot for Paddle and Keyboard CW Only:** Ticking this option will make the winkey speed pot only effective to the paddle and keyboard CW.

7. When finishing all of the settings, click OK to exit to N1MM's main operation window. The keyer will communicate with N1MM in 1200bps, and you are going to hear a beeping in 5 seconds if it connects with N1MM successfully.



6. Keyer Configuration by CMD

To enter command mode, press the CMD (rotary switch) button, and you will hear a "boop beep", after which you can enter various commands by sending character using the paddle. (Note that if you're in bug or straight key mode, you will temporarily be switched to iambic in command mode.)

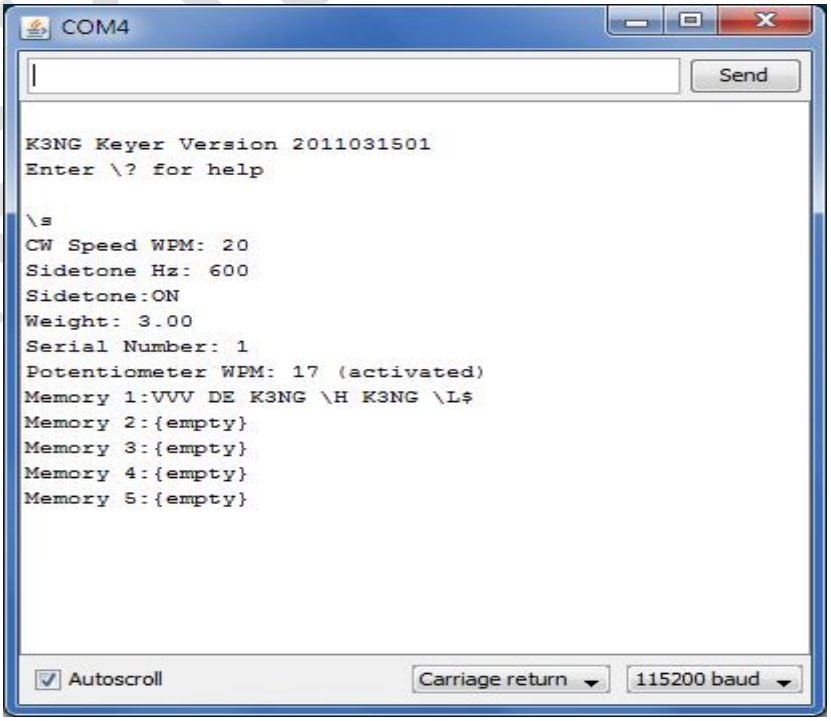
If you enter a bogus command or the keyer didn't recognize the character you sent, it will send a question mark, upon which you can retry your command.

To exit command mode, send X in CW using the paddles or just press the CMD again upon which you will hear "beep boop" and you'll be back in regular sending mode.

```
A - Switch to Iambic A mode
B - Switch to Iambic B mode
D - Switch to Ultimatic mode
E - Announce the speed in WPM
F - Adjust sidetone frequency
G - Switch to bug mode
I - TX enable / disable
J - Dah to dit ratio adjust
N - Toggle paddle reverse
O - Toggle sidetone on / off
P# - Program a memory
S - Alphabet Send Practice
T - Tune mode
V - Toggle potentiometer active / inactive
W - Change speed
X - Exit command mode (you can also press the command button (button0) to exit)
Z - Autospace On/Off
# - Play a memory without transmitting
```

7. Keyer Configuration by Serial Command Line Interface (CLI)

The keyer has a serial Command Line Interface (CLI) using the built in Arduino USB port or AVR serial port. Please note that in this Funtroncs K3NG Keyer, the front USB is used for CLI only. Simply connect it to your computer and use a terminal program such as the Arduino serial port program or Putty. If you use the Arduino program, it's recommended that you set it for carriage return (lower right). To use the CW keyer functionality, simply type in what you want to send. In the Arduino serial interface you will need to hit Enter to send the data to the keyer for it to start sending. Programs like Putty will immediately send the characters and the keyer will send the code immediately as well.



Commands are preceded with a backslash (" \ "), the key above your Enter key (at least on US PC keyboards). To see a help screen, enter backslash question mark " \? " (no quotes). The status command (\s) is a useful command for viewing various settings and seeing the contents of the memories. If you enter a double backslash ("\\"), all sending buffers will be cleared and any memory sending will stop (this includes sending invoked by the PS2 keyboard or Winkey interface protocol emulation features).

CLI commands:

- \? Help
- \# Play memory #
- \a Iambic A mode
- \b Iambic B mode
- \c Switch to CW (from Hell)
- \d Ultimatic mode
- \e#### Set serial number to ####
- \f#### Set sidetone frequency to #### hertz
- \g Bug mode
- \h Switch to Hell sending
- \i Transmit enable/disable
- \j### Dah to dit ratio (300 = 3.00)
- \k Callsign receive practice
- \l## Set weighting (50 = normal)
- \m### Set Farnsworth speed
- \n Toggle paddle reverse
- \o Toggle sidetone on/off
- \p# Program memory #
- \q## Switch to QRSS mode, dit length ## seconds
- \r Switch to regular speed mode
- \s Status
- \t Tune mode
- \u Manual PTT toggle
- \v Toggle potentiometer active / inactive
- \w### Set speed in WPM
- \x# Switch to transmitter #
- \y# Change wordspace to # elements (# = 1 to 9)
- \z Autospace on/off
- \+ Create prosign
- \!## Repeat play memory
- \|#### Set memory repeat (milliseconds)
- * Toggle paddle echo
- \^ Toggle wait for carriage return to send CW / send CW immediately
- \~ Reset unit
- \& Toggle CMOS Super Keyer Timing on/off
- \%## Set CMOS Super Keyer Timing %
- \. Toggle dit buffer on/off
- \- Toggle dah buffer on/off

- \: CW send echo inhibit toggle
- \{ QLF mode on/off

8. Keyer Configuration by PC Keyboard

A standard PC usb keyboard is supported by Funtronics K3NG Keyer, to do it, just simply connect it to the keyer's USB HOST port on the rear panel. You can use the Keyboard to send the CW code instead of a paddle, and nevertheless configuration of the keyer could also be done by the PC Keyboard.

USB Special Key Assignments:

- F1 through F12 – play memories 1 through 12
- Up Arrow – Increase CW Speed 1 WPM
- Down Arrow – Decrease CW Speed 1 WPM
- Page Up – Increase sidetone frequency
- Page Down – Decrease sidetone frequency
- Right Arrow – Dah to Dit Ratio increase
- Left Arrow – Dah to Dit Ratio decrease
- Home – reset Dah to Dit Ratio to default
- Tab – pause sending
- Delete – delete the last character in the buffer
- Esc – stop sending and clear the buffer
- Scroll Lock – Merge the next two characters to form a prosign
- Shift – Scroll Lock – toggle PTT line
- CTRL-A – Iambic A Mode
- CTRL-B – Iambic B Mode
- CTRL-D – Ultimatic Mode
- CTRL-E – Set Serial Number
- CTRL-G – Bug Mode
- CTRL-H – Hellschreiber Mode (requires FEATURE_HELL)
- CTRL-I – TX Line Disable/Enable
- CTRL-M – Set Farnsworth Speed (requires FEATURE_FARNSWORTH)
- CTRL-N – Paddle Revers
- CTRL-O – Sidetone On/Off
- CTRL-T – Tune
- CTRL-U – PTT Manual On/Off
- CTRL-W – Set WPM
- CTRL-Z – Autospace On/Off
- SHIFT-F1, F2, F3… – Program memory 1, 2, 3…
- ALT-F1, F2, F3… – Repeat memory 1, 2, 3…
- CTRL-F1, F2, F3… – Switch to transmitter 1, 2, 3…
- Keypad / – Dit Paddle
- Keypad * – Dah Paddle
- Keypad ENTER – Tune / Straightkey

There may have some issues using just the Arduino USB client port +5V to power the Arduino and PS2 keyboard, with

operation being erratic or the keyboard just not functioning at all. This is due to computer USB ports not being able to supply enough current. The solution is simple: power the Arduino board directly using the cellphone power bank.

9. CW Speed Adjustment

The CW sending speed can be adjusted several ways:

- The W command in command mode
- The command line interface \w command
- The memory macro \w, or \y and \z for incremental increases or decreases
- Holding down the CMD switch and pressing the left and right paddles
- Rotary Encoder Speed Control

10. Iambic Modes

To switch to Iambic A mode, use the A command in command mode or \a in the command line interface.
To switch to Iambic B mode, use the B command in command mode or \b in the command line interface.

11. Straight Key Support

Go into straight key mode by holding down the right paddle when powering up or power resetting. This places the keyer exclusively in straight key mode with very limited functionality.

12. Bug Mode

To go into bug mode, use the command mode G command or the command line \g command.

13. Utimatic Mode

To go into Ultimatic mode, use the command mode D command or the command line \d command.

14. Memory Operation and Memory Macros

Memories can be manually played using buttons M1 through M3, or using the \# command in the CLI (for example, \1 plays memory 1). In command mode the memories can be sent without transmitting by entering the number of the memory.
Memories are programmed using the CLI \p# command or in command mode using the P command.
To program memory 1 with CQ CQ CQ DE K3NG, the command would be \p1CQ CQ CQ DE K3NG.
To program memory 1 using command mode, enter command mode by pressing CMD switch and sending the P command. After hearing a beep, send the CW code to be stored and when finished, hit the CMD to exit programming. The keyer will then play back the memory. If the keyer doesn't recognize a character you sent it will send a question mark in its place.
Macros can be placed in memories to do cool things. Some macros include:

- \# Jump to memory # (1 through 9)
- \c Play serial number with cut numbers
- \d### Delay for ### seconds
- \e Play serial number, then increment
- \f#### Set sidetone to #### hertz
- \h Switch to Hell sending
- \i# Insert memory # (This is different from \#, the jump macro. The insert memory macro plays another memory, then comes back to the memory the keyer was originally playing. The jump command jumps to the other memory and doesn't come back.)

- \l Switch to CW (from Hell mode)
- \n Decrement serial number, do not send
- \q## Switch to QRSS mode, dit length ## seconds
- \s Insert space
- \r Switch to regular speed mode
- \t### Transmit for ### seconds
- \u Activate PTT
- \v Deactivate PTT
- \w### Set regular mode speed to ### WPM
- \x# Switch to transmitter # (1, 2, 3, 4, 5, or 6)
- \y# Increase speed # WPM
- \z# Decrease speed # WPM
- \+ Prosign the next two characters

Please note that both command line commands and CW memories are case insensitive. Up to 12 memories can be configured, with some caveats. Nine memories are supported in the CLI and in memory macros, and the full 12 are supported with the PS2/USB keyboard.

15. CW Dah to Dit Ratio Adjust

The CW dash length to dot length ratio can be adjusted using the J command in command mode. Upon entering the J command you will hear a repeating dit dah. Use the left and right paddles to shorten or lengthen the dah. Squeeze both paddles to exit the weight adjust command. After that you can enter X or press the command button to exit command mode.
The ratio can also be adjusted in the command line interface using the \j command. \j300 sets the keyer for a normal 3:1 ratio, \j250 would set it for a 2.5:1 ratio, for example.

16. Autospace

The autospace feature can be toggled on and off with the Z command in command mode, the \z command in the command line interface, and CTRL-Z using the PS2 keyboard. This feature “cleans up” manual sending a bit by automatically inserting a wordspace delay if the operator waits more than one dit after sending a dit or dah to paddle either paddle.

17. Others

There are also some other settings that are less important than the listed ones above. For those other settings, please visit https://blog.radioartisan.com/arduino-cw-keyer/ for more detailed information.

18. Arduino Firmware Loading

As the K3NG keyer is a open source project, the Funtronics K3NG Keyer supports the firmware updating by compiling and uploading a Arduino sketch. To do this, you need to download the Arduino IDE and the K3NG code with libraries from our web store. If you want to customize the arduino code, you need some basic programming knowledge on Arduino. Read the documents on Arduino.cc before compiling and coding.

To download the Arduino IDE, click here: https://www.arduino.cc/en/Main/Software
For the arduino sketches and libraries, please download them from our web store. You may also download them from K3NG web, but you have to modify the pin definitions part in the sketch as we have different pin definitions from the original K3NG arduino code.

19. Support

For any questions and problems, contact support@elekitsorparts.com for help.

NOTES:

NOTES: