

# **9A5N QRQ keyer**

## **User manual**

## Introduction

9A5N QRQ keyer is the second generation of keyers developed by the author based on microcontrollers. Keyer uses a 16-bit microcontroller with an operating frequency of 32MHz and with a large memory space for temporary and permanent storage of predefined telegraphic messages.

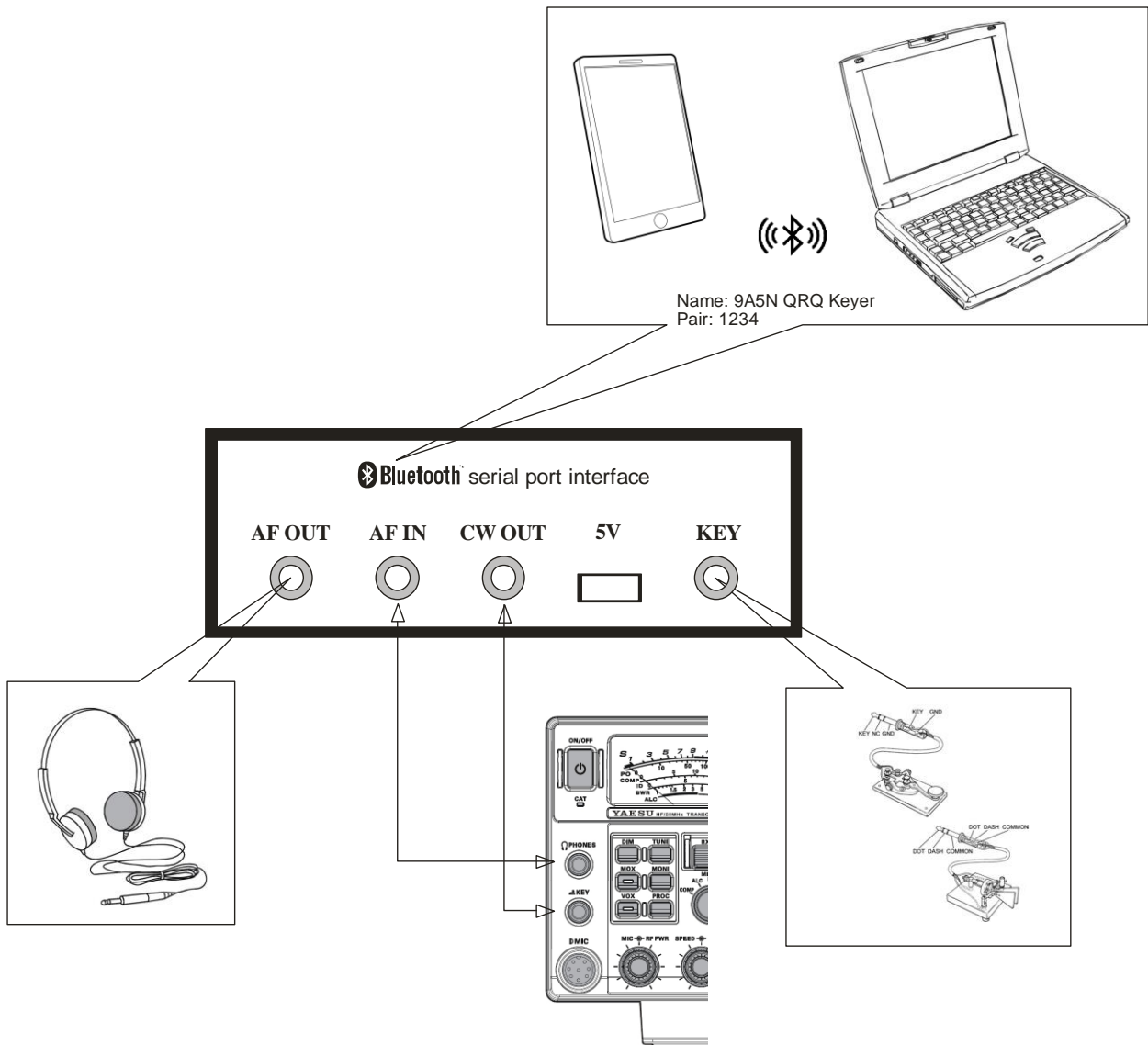
Keyer was primarily developed for learning and practicing telegraphy (high quality side tone and built-in AF amplifier) however, the ability to connect it wirelessly to a PC or smartphone via Bluetooth virtual RS-232 serial port (internal Bluetooth serial port module) give keyer wide application.

The aforementioned features make the 9A5N QRQ keyer completely different from other keyers on the global market that use a USB wired connection to a PC and have square waveform side tone that plays on the buzzer.

## Specification

- Side tone of high quality sine waveform
- Side tone frequency adjustable from 392 to 1397 Hz
- Built-in 1W Hi-Fi AF amplifier
- Side tone volume potentiometer
- AF output for connecting an external speaker or headphones
- AF input for connecting the AF output from the transceiver
- Iambic A/B, Ultimatic, Bug and Straight Key mode
- Adjustable keying speed from 5 to 99 WPM
- CW speed potentiometer with variable setting range
- Farnsworth adjustable CW speed for learning telegraphy
- Auto spacing function for the correct length of pauses between characters
- CW output with MOSFET transistor 60V/500mA
- Bluetooth serial port for wireless connection with a PC or smartphone
- Serial port input buffer with a capacity of 410 characters
- One temporary and 5 permanent message memories with a capacity of up to 1024 characters each
- CW decoder for correct keying control
- Possibility to calibrate the frequency of the microcontroller oscillator
- Compatibility with WinKeyer 3.1 protocol
- Mini USB power supply connector 5V/0.5A max
- Maximum power consumption 2W (depends on the side tone volume)

## Installation



## Front panel

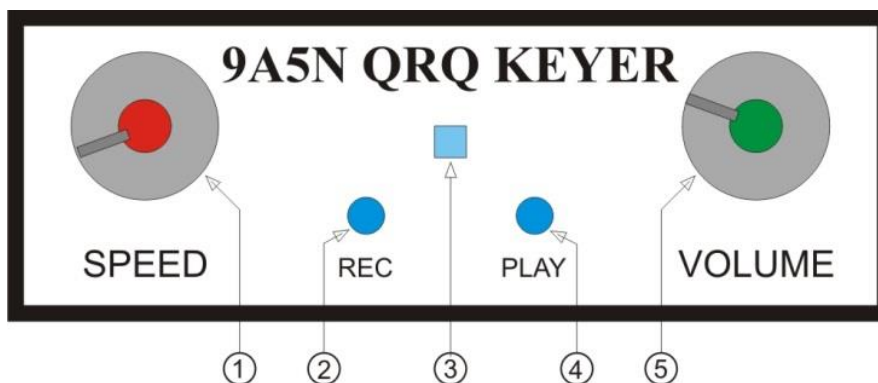


Figure 1 – Keyer front panel

### 1. SPEED – potentiometer

The potentiometer has two functions depending on the keyer operation mode.

Primarily, the potentiometer is used to adjust the CW speed. The basic speed setting range is 15 to 35 WPM but can vary from 5 to 99 WPM. Moving the potentiometer deletes the CW speed sent via the Bluetooth serial port and the new speed is defined by the position of the potentiometer.

The potentiometer is used to adjust the side tone frequency when the keyer is in Straight key mode and when the manual side tone frequency adjustment function is active.

### 2. REC - button

The button has several functions:

- Button pressed for 2 sec. - activation of the message recording functions into the RAM memory, keyer will key out the letter **S**. Keyer records everything that is keyed with the key and all the text received via the Bluetooth serial port. Recording ends with a short press on the button, keyer will key out the letter **R**.
- Simultaneous pressing of the REC and PLAY buttons - activation of the side tone frequency adjustment function. SPEED potentiometer is in use. The keyer on the AF output emits a continuous side tone. The setting is completed by briefly pressing the REC button, the keyer will key out the letter **R** and the selected frequency will be saved in the EEPROM.
- Simultaneous pressing of the REC button and the right lever / right side of the lever on the key - activation of key out information on the current CW speed. The keyer will start to key out the CW speed in the WPM. The REC button should be released. With the SPEED potentiometer, the CW speed can be changed and set to the desired value. Keyer will key out WPM as long as the lever on the key is pressed.
- Simultaneous pressing of the REC button and the left lever / left side of the lever on the key - activation of the keyer CW mode selection. Keyer will start key out the currently selected CW mode. The REC button and the key lever should be released. By briefly pressing the left lever / left lever on the key again, it will key out the next CW mode and so on in a circle. When the keyer will key out the desired CW mode, press the REC button. Keyer will key out the letter **R** and selected CW mode will be saved in EEPROM.

**3. Keyer On – indicator**

Blue LED diode lights up when the keyer has 5V supply.

**4. PLAY – button**

Push on the button and keyer will start to key out a message saved in the RAM memory.

The keying speed is defined by the SPEED potentiometer or by the speed command received via the Bluetooth serial port.

The keying out of the message can be interrupted at any time by brief pressing on the key lever/ levers.

**5. VOLUME – potentiometer**

The potentiometer for the side tone volume in the speaker/headphones.

## Back panel

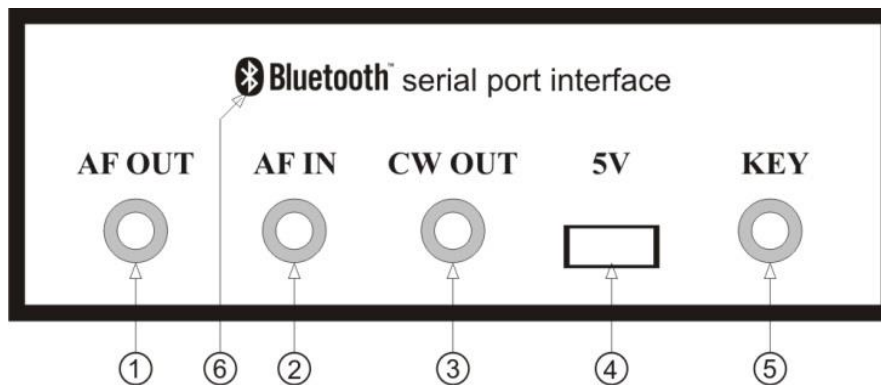


Figure 2 – Keyer back panel

### 1. AF OUT – socket

3-pole 3,5 mm socket for headphone or speaker. The AF output is MONO - the tip and ring are connected together. A 4 or 8  $\Omega$  impedance speaker should be connected between the tip or ring and the ground of the plug. By connecting the stereo headphones, the same sound will be heard on the left and right side.

### 2. AF IN - socket

3-pole 3,5 mm socket for the AF signal from the transceiver. The AF signal is taken from the headphone socket on the transceiver (PHONES). The AF signal should be connected between the tip and the ground of the plug. It is recommended to use a prefabricated audio stereo cable with 3.5 mm plugs as the connecting cable. The volume of the AF signal is regulated by a potentiometer on the transceiver.

This connection of the keyer and transceiver allows the receiver signal to be heard in the headphones / speaker connected to the keyer AF OUT during reception and side tones from the keyer during transmission. The internal side tone in the transceiver should be turned off.

The impedance of the AF IN input is 47 $\Omega$ .

### 3. CW OUT - socket

3-pole 3,5 mm socket for CW signal output. The output is of the "open collector" type connected between the tip and the ground of connector.

The CW OUT is connected to the key input on the transceiver (KEY). It is recommended to use a prefabricated audio stereo cable with 3.5 mm plugs as the connecting cable. The transceiver should be configured for the Straight key.

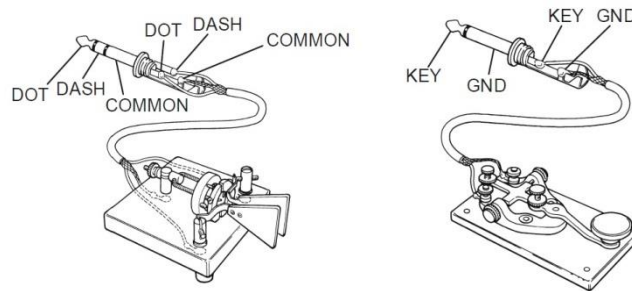
The transistor at the CW output can withstand a voltage of up to 60V and a current of up to 500mA.

### 4. 5V - socket

MINI USB socket for the 5V DC power supply. A standard USB cable with a MINI USB plug at one end is used to connect to the power supply. The power source can be a standard 5V charger for a mobile phones with a USB socket, a "Battery Pack" with a USB socket or a USB socket on a PC. You can use a car charger for mobile phones to power from a 13.8V source. The power supply should have a current capacity of 0.5A.

### 5. KEY - socket

3-pole 3,5 mm socket for telegraph key connection. All types of keys can be connected. The line for the dots connects to the tip, for the dashes to the ring and the common line to the ground of the plug. If the Straight key is used, one line is connected to the tip and the other to the ground of the plug.



### 6. Bluetooth serial interface – internal module

The keyer has a built-in Bluetooth serial port module that allows wireless connection of the keyer to a PC, tablet, smartphone, etc.

On the Bluetooth wireless network, the keyer appears with the name "9A5N QRQ keyer". "Pair code" is 1234. After pairing, a virtual RS-232 serial port for communication with the keyer will open on the connected device.

**9A5N QRQ Keyer commands**

All 9A5N QRQ keyer commands and settings are only available via the Bluetooth serial port. In order for keyer to distinguish a command from standard text sent via serial port, character # is added in front of the code for a particular command for which there is no CW code and keyer cannot key out it. Command codes can be written in upper or lower case, i.e. they are not case sensitive.

In parallel with 9A5N QRQ keyer commands, WinKeyer commands can be used also.

Some basic keyer commands and settings are available on the keyer itself using keyer buttons and connected telegraphy key.

**Bluetooth serial port commands:**

#_	KEY OUT MEM MSG	_ = 0-5
#a_	PRINT MEM MSG	_ = 0-5
#b	BEACON ON/OFF	
#c	CLR WINDOW	
#f__	TONE FREQ	__ = 00-22/s/?
#h	HELP	
#i	KEY INV/NOR	
#k	INFO	
#l__	FARNS SPEED	__ = 15-99/c
#m_	MODE	_ = a/b/u/s/m/?
#o__	OSC TUNE	__ = +/-s/?,1-9
#p	AUTO SPACE ON/OFF	
#r_	EEPROM -> RAM	_ = 1-5/?
#s_	RAM -> EEPROM	_ = 1-5/?
#t___	BEACON TIME	___ = 000-999/?
#u__	POT MAX WPM	__ = 20-99/?
#v__	POT MIN WPM	__ = 05-70/?
#w__	SPEED IN WPM	__ = 05-99/?
#x	RESET KEYER	
#y	TEXT -> RAM ON/OFF	
#z	CWOUT ON/OFF	
[__	PROSIGN	__ = TWO CHAR.

**Manual commands:**

REC 2sec.	TEXT -> RAM ON/OFF
REC + PLAY	TONE FREQ. ADJUSTMENT
REC + right key lever	SPEED IN WPM
REC + left key lever	CW MODE SELECTION
PLAY	KEY OUT RAM MSG



**WinKeyer commands**

00 <02>	Host open
00 <03>	Host close
02 <nn>	Set CW speed in WPM immediate
07	Get current CW speed pot setting
0A	Reset input buffer and cancel Key down
0B <01>	Key down immediate
15	Request WinKeyer status
16 <nn>	Input buffer commands
1B [C][C]	Merge two letters into a Prosigne
1C <nn>	Set CW speed in WPM buffered

**Remark:**

All other WinKeyer commands will be ignored or the keyer will respond to the command with a predefined message in order to emulate the WinKeyer operation.

## Description of the 9A5N QRQ Keyer operation

### Bluetooth serial port

The 9A5N QRQ keyer has a built-in Bluetooth interface for the RS-232 serial port. The interface enables wireless connection of the keyer to a PC, tablet or smartphone on which there is a suitable application for communication via the RS-232 serial port.

The interface appears on the Bluetooth network as "9A5N QRQ keyer". Once you have found the interface on the network, Pairing should be performed. The pair code is "1234". After pairing, the operating system will create a virtual RS-232 port. For Android and iPhone, the COM port is defined by the name of the connected unit, while for Windows the number of the assigned COM port should be found in the Control Panel under Hardware > View devices and Printers > Properties > Services.

You can use Hyper Terminal for Windows OS or Serial Bluetooth Terminal for Android to communicate with keyer.

#### Remark:

Windows 10 can sometimes change the virtual COM port numbers themselves. If your communication with the keyer has stopped working, check the COM port number assigned to the keyer. In some cases you will need to delete keyer from Bluetooth devices and reconnect it again.

### Side tone

When creating software, special attention was dedicated to the quality and waveform of the side tone. Side tone quality is very important for error-free CW keying, especially at higher CW speeds. A quality side tone does not tire the telegrapher and allows him to achieve a higher keying speed with fewer errors.

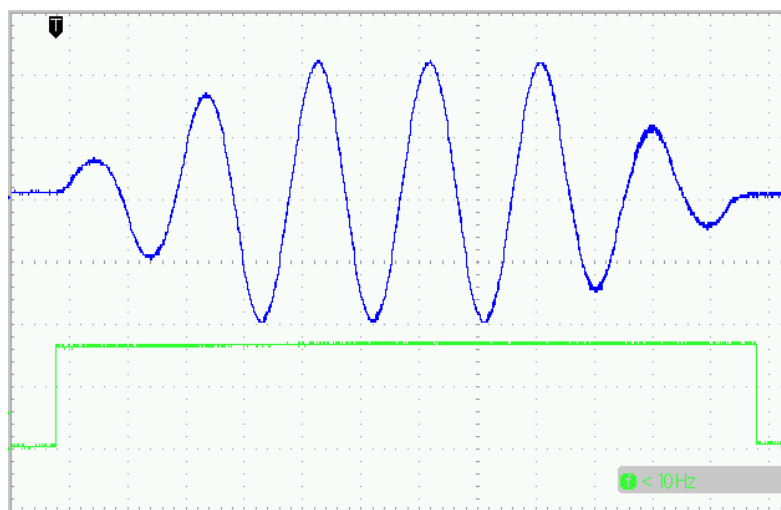


Figure 3

Figure 3 shows the waveform of the side tone in correlation with the CW OUT signal.

Side tone is a sine waveform with a gradual increase in amplitude at the beginning and a gradual decrease in amplitude at the end. The resulting sound signal is soft and clean, it does not tire the telegrapher hearing aid and he can fully concentrate on the telegraphy.

The sine waveform is obtained by direct digital synthesis (DDS). The sinusoid is defined in 255 points and is generated by pulse width modulation (PWM) with a resolution of 10 bits at an operating frequency of 62.5 kHz. A large number of points and high PWM resolution ensures the generation of a sinusoid of various frequencies with minimal distortion and minimal contribution of higher harmonics. The high PWM frequency allows easy filtering from the generated AF sine wave signal.

Side tone can be one of 23 frequencies ranging from 392 to 1396.9 Hz. The frequencies of individual tones are defined according to the chromatic scale. The codes of individual frequencies and associated notes are listed in Table 1.

Code	Frequency	Note	Code	Frequency	Note	Code	Frequency	Note
00	392	g4	08	622,3	d#5	16	987,8	b5
01	415,3	g#4	09	659,3	e5	17	1046,5	c6
02	440	a4	10	698,5	f5	18	1108,7	c#6
03	466,2	a#4	11	740	f#5	19	1174,7	d6
04	493,9	b4	12	784	g5	20	1244,5	d#6
05	523,3	c5	13	830,6	g#5	21	1318,5	e6
06	554,4	c#5	14	880	a5	22	1396,9	f6
07	587,3	d5	15	932,3	a#5			

Table 1 – side tone frequencies

The side tone frequency can be changed manually with a potentiometer on the keyer or with commands via the Bluetooth serial port.

- Manual selection of side tone frequency

By pressing the REC button and then PLAY at the same time, the keyer switches to the side tone frequency setting mode and a constant tone will be heard from the speakers / headphones. Release both keys. You changing the frequency with the SPEED potentiometer. When you have selected the frequency that is most suitable for you, press the REC button briefly. Keyer will save the code of the selected frequency in the EEPROM memory and key out **R**.

- Selection of side tone frequency via the Bluetooth serial port

The #f (#F) command selects, saves, and prints information about the current side tone frequency. What the command will do depends on the character or characters behind the letter f.

Possible combinations:

- # f00 to # f22 - selection of side tone frequency code according to Table 1
- #fs - save the code of the selected frequency in the EEPROM memory
- #f? - print the currently selected side tone frequency on the screen of the connected device

Example:

# f13 - the side tone frequency of code number 13 is selected

#fs - the code of the selected side tone frequency is stored in the EEPROM

#f? - Information on the currently selected side tone frequency is displayed on the screen

Advice:

Use the # f00 through # f22 commands to change the side tone frequency until you find the frequency you are most comfortable listening to as you keying. When you have found the code with the most comfortable frequency, use the #fs command to save it permanently to the EEPROM memory.

### CW speed

The CW speed is set with the SPEED potentiometer or by sending a commands via the Bluetooth serial port.

The basic range for adjusting the CW speed with the potentiometer is 15 to 35 WPM. The user can adjust the CW speed setting area to their needs using commands via the Bluetooth serial port:

- #u\_\_ - maximal CW speed adjustable with potentiometer in WPM (20-99)

- #v\_\_ - minimal CW speed adjustable with potentiometer in WPM (05-70)

Example:

#u60 - the maximal CW speed adjustable by the potentiometer is 60 WPM

#v05 - the minimal CW speed adjustable by the potentiometer is 5 WPM

Regardless of the position of the SPEED potentiometer, the CW speed can be changed at any time with the command via the Bluetooth serial port:

- #w\_\_ - desired keying speed in WPM (05-99) or keying speed information (?)

Example:

#w33 - The keying speed is set to 33 WPM

#w? - keying speed is 33 WPM

If you want to return the CW speed control to the SPEED potentiometer, it is enough to move the potentiometer a little bit. The CW speed set via serial port will be deleted and will be redefined by the position of the SPEED potentiometer.

Information on the current keying speed can be obtained at any time by simultaneously pressing the REC button and the right key lever or with #w? command

## CW code generation modes

Keyer supports all CW code generation algorithms, simulation of the semi-automatic key (Bug) and monitor mode for operation with the Straight key.

The table lists the codes for selecting each operating mode:

Code	Algorithm
a	Iambic A
b	Iambic B
u	Ultimatic
s	Bug simulation
m	Monitor ( side tone only )

Table 2

The CW code generation mode can be selected manually using the buttons on the keyer and the connected key or with commands via the Bluetooth serial port.

### - Manual mode selection

By simultaneously pressing on the REC button and then the left lever on the key, keyer switches to the CW mode selection and will key out code of the currently selected CW mode. Release the REC button and lever on the key. Pressing the left lever on the key again will key out the next CW mode code. Press the left lever repeatedly to select the desired CW mode. When keyer will key out the desired CW mode code, press the REC key briefly. Keyer will save the code of the selected mode in the EEPROM memory and key out **R**.

The described procedure can be used to select all modes except Monitor mode.

Since keyer will be used as a monitor when a straight key or Bug is connected to keyer, the procedure for selecting Monitor mode is as follows:

- Switch On keyer
- After 2 seconds, when the keyer starts to key out 9A5N, press the straight key/Bug
- When the keyer finishes keying of 9A5N, release the key
- Keyer is in Monitor mode
- The next time you press the key, a side tone will sound and the CW OUT will close

### Remark:

As mechanical keys often have problems with closing the contacts (oxidation, dust, bouncing, etc.), the function of the signal filtering of the key closed contact is active in Monitor mode. All closed contact signals shorter than 0.56 ms will be ignored. Signals longer than 0.56 ms but shorter than 20 ms will be extend to 20 ms. In this way, a much cleaner side tone and CW Out signal was obtained, without interruption by the key contact bouncing.

### - Mode selection via Bluetooth serial port

The #m (#M) command selects and prints information about the current CW code generation mode. What the command will execute depends on the character behind the letter **m**.

Possible combinations:

- #m\_ - select one of the CW code generation modes. Instead of character \_, the letters a, b, u, s and m can be used according to Table 2.
- #m? - the current mode will be printed on the screen

Example:

#ma - keyer generates CW code according to the Iambic A algorithm

Remark:

When you change the operating mode from Monitor mode to another mode, the keyer will reset!

Monitor mode is not saved in EEPROM memory. When switched on again, the keyer will operate in one of the previously selected CW modes.

You can exit Monitor mode manually by pressing the Play button.

### **Auto space between characters**

In order for the telegrapher to improve his sense of the exact pause length between characters, keyer has the option to control the length of pauses. By activating the auto space function, keyer will extend all pauses shorter than three dots to exactly three dots.

Auto space function can be only activated / deactivated by commands via the Bluetooth serial port. The command acts as a toggle switch. If auto space is not active, sending the #p command will activate it and vice versa.

Each change of auto space function state will be saved in the EEPROM memory. The currently selected auto space state can always be checked with the #k (#K) - keyer info command.

Example:

First command #p – auto space On

Second command #p – auto space Off

### **Dot/ dash input swapping**

Most telegraphists use the key so that the dashes are on the right and the dots are on the left key lever / lever side. This is a normal dot/dash layout. However, there are telegraphists who are accustomed to the opposite layout or are left-handed to whom the opposite layout is more natural. This is an inverted dot/dash layout. To cover both cases without the need to electrically switch the lines on the key, the normal / inverted layout can be changed in the software.

Dot/dash input swapping can only be done via the Bluetooth serial port with the #i (#I) command. The command acts as a toggle switch. If the dot/dash inputs are normal, sending the #i command dot/dash inputs will be inverted and vice versa.

Each change of dot/dash inputs will be saved to EEPROM memory.

The currently selected dot/dash input state can always be checked with the command #k (#K) - keyer info.

Example:

First command #i - dot/dash layout inverted  
 Second command #i - dot/dash layout normal

### **Farnsworth CW speed for learning telegraphy**

Farnsworth's idea of CW speed when learning telegraphy is to avoid memorizing the character CW code in the form of a number of dots and dashes, but to memorize the melody of each character. To avoid counting of CW elements, the keying speed should be greater than 15 WPM. For students, the pauses between characters at that speed are too short, so with Farnsworth keying, these pauses can be extended and give students more time to "decode" the keyed character.

Farnsworth keying can only be activated / deactivated by commands via the Bluetooth serial port.

At Farnsworth keying we have two speeds:

- Character CW speed
- Pause between characters of lower CW speed

Farnsworth keying is activated by the command #l\_\_ (#L\_\_) where the characters \_\_ represent the character CW speed which can be from 15 to 99 WPM.

The CW speed for pauses is adjusted with a SPEED potentiometer from 5 WPM up to the Farnsworth character CW speed.

Example:

# 125 - a character keying speed is 25 WPM  
 SPEED potentiometer at minimum - pauses are 5 WPM in length

Farnsworth keying is deactivated with the command #lc (#LC).

### **Beacon mod**

In beacon mode, the keyer will automatically key out a message stored in RAM at selected intervals. During intervals between beacon messages, all keyer functions are available and keyer can be used normally.

The last selected beacon mode status is stored in the EEPROM memory.

When the keyer is turned on again, if the beacon mode was previously turned on, the keyer will continue to work as a beacon.

Remark:

To ensure the correct operation of the beacon, the beacon message must be saved in EEPROM memory 1. When the keyer is turned on, the message stored in EEPROM memory 1 is copied to RAM automatically.

Beacon mode is turned on and off with the #b (#B) command. The command acts as a toggle switch.

Example:

#b - beacon mod On ..... #b - beacon mod Off

The time interval between two beacon messages is defined by the #t (#T) command.

- #t\_\_\_ - time interval in seconds (000-999) between the end of keying the message and the resumption of keying the next message.

The time is always entered with three digits.

Example:

#t090 - keyer will wait 90 seconds after keying the message finish and then will start keying again

### "Key down" function

A constant signal from the transmitter is required for the valve linear amplifier output stage adjustment.

Keyer will close/open the CWOUT by sending the Bluetooth command #z (#Z). The command acts as a toggle switch.

Example:

#z – CWOUT closed ..... #z – CWOUT open

### "Prosign" function

The Prosign function allows you to create CW codes of national alphabet letters by merging the CW code of two English alphabet letters.

When entering text that contains letters that are not in the English alphabet, in these places in the text you should enter [- an open square bracket and then two letters whose merged CW code gives the required combination.

Example:

We want keyer will key out "SRETAN ROĐENDAN". CW code of letter Đ is -.-  
- and can be get it with merging of letters D (-..) and M (--).

We type: sretan ro[dmendan



## RAM and EEPROM memory for storing messages

Keyer has the ability to store 6 predefined messages, one of which is a temporarily stored message in RAM and 5 permanently stored messages in EEPROM memory.

Each message has 4096 bits of memory space that can store 510 characters (102 words PARIS) in average. The exact number of possible stored characters depends on the number of CW elements in each character. Each CW element occupies 2 bits and it is easy to calculate how much memory a single letter will occupy. The letter E (dot + pause) occupies 4 bits while the numbers (5 lines + pause) occupies 12 bits. A simple calculation shows that 1024 letters E or T (4096/4) or 341 numbers (4096/12) can be stored in each memory.

Each message that you want to save permanently in one of the 5 places in the EEPROM memory is first saved in RAM as a temporary message and then copied to the desired EEPROM memory.

Recording of the message in RAM starts and stops:

- manually using the REC button
- with the #y command via the Bluetooth serial port

The message can be written to RAM:

- by keying with key
- with keyboard or by sending text file via Bluetooth serial port

Press the REC button and wait 2 seconds for the keyer to key out **S**. Previously saved message in RAM will delete. Release the button and you can start keying the desired message with the key or start writing it with the PC / smartphone keyboard. When you have finished keying or writing the message, press the REC button again. Keyer will finish recording and key out **R**.

Also, message recording can be started by sending the #y command. When you finish keying or writing a message, recording ends by sending the #y command again.

With Hyper Terminal you can send already written text file however, text in file must start and end with #y. Text written with Notepad in form of one raw will ensure correct keying without extra pauses and false WinKeyer commands caused with LF and CR saved in multiple raw text.

Example of saving a message with commands #y:

#yHELLO#y – keyer will save message HELLO into the RAM memory

During message recording, keyer does not perform any CW code control and saving CW elements as you keying them. For example, if you will key 10 dots and 10 dashes without pause, keyer will save them in RAM.

If you make a keying or writing error while recording a message, stop the recording and restart it. The previously recorded message with error will be deleted and you must rekey or rewrite the entire message again.

A message saved in RAM can be keyed out:

- by pressing the PLAY button
- with command #0 via Bluetooth serial port

Messages stored in the EEPROM memory can only be keyed out with the #\_ command where the character \_ is replaced by the number of the desired EEPROM memory (1-5).

Example:

#2 – keyer will key out message in the EEPROM memory 2

A message stored in RAM memory is lost by switching Off the keyer. When the keyer is switched On again, the message stored in EEPROM memory 1 is automatically copied to RAM memory. If EEPROM memory is empty, RAM memory will remain empty.

If you want to save a message permanently, the message stored in RAM should be copied to one of the 5 EEPROM memories. Copying is done with the command #s\_ (#S\_). Instead of the character \_, enter the number of the desired EEPROM memory (1-5).

Example:

#s2 – message in the RAM memory is copied to the EEPROM memory 2

If you want to copy a message stored in the EEPROM memory back to the RAM memory, you can do so with the command #r\_ (#R\_). Instead of the character \_, enter the number of the desired EEPROM memory (1-5).

Example:

#r3 – message stored in the EEPROM memory 3 is copied to the RAM memory

If you have forgotten the texts of saved messages or want to check them, you can use the #a\_ (#A\_) command. The keyer will print the text of any message on the screen of the connected device without key out the message. Instead of the character \_, enter the number of the desired message memory (1-5).

Example:

#a4 – the text of the message stored in the EEPROM memory 4 will be printed

You can get an overview of the current state of all memories (empty / message) with the commands #r? or #s?.

Example:

#s? or #r? – keyer will print on the screen

#### EEPROM MEMORY

1 – MSG

2 – MSG

3 – NO MSG

4 – NO MSG

5 – NO MSG

RAM MEMORY – MSG

If you want to delete any of the messages stored EEPROM memory, you can do so as follows:

1. Delete a message in RAM, if any, by saving an empty message by pressing REC button or sending the commands #y#y
2. Save empty RAM message to the EEPROM memory you want to delete

Example:

#y#y - delete a message in RAM memory  
 #s1 - saving empty RAM to EEPROM memory 1

### **Informative and support commands**

In addition to the previously described commands that can be sent via the Bluetooth serial port, here are some other useful commands:

- #c - delete printout in Hyper Terminal window
- #h - help, keyer will print a list of all available commands on the screen
- #k - keyer info, keyer will print current keyer settings on the screen
- #x - reset keyer

When resetting the keyer with command #x (#X), the parameters stored in the EEPROM memory will be loaded and the message stored in the EEPROM memory 1, if any, will be copied to the RAM memory. Keyer behaves in the same way as after switching Off and On. However, the Bluetooth serial port connection will not be disconnected and a help list of all keyer commands will be printed on the screen of the connected device.

### **WinKeyer compatibility**

Most logging programs have support for the WinKeyer serial port, which allows you to manage external keyers compatible with the WinKeyer protocol.

9A5N QRQ Keyer supports WinKeyer 3.1 protocol.

Keyer supports all WinKeyer commands needed to make CW contacts using a PC and an installed logging program. WinKeyer setup commands are not supported and has no effect on the settings of the 9A5N QRQ Keyer.

Unlike WinKeyer, the 9A5N QRQ Keyer does not have a PTT output. Lead and Tail times are set on the transceiver as in the case of using a key directly connected to it.

Keyer compatibility with the WinKeyer protocol has been tested with the most popular programs used in contests such as Dxlog, N1MM and MixW which does not mean that keyer will not work with other programs that support WinKeyer.

## Oscillator frequency adjustment

The accuracy of the microcontroller oscillator frequencies affects the accuracy of the side tone frequency, the Beacon time and the accuracy of the CW speed.

The microcontroller oscillator frequency is factory set with an accuracy of  $\pm 1\%$ . If you consider it important, the oscillator frequency can be adjusted to an accuracy of  $\pm 0.1\%$ .

It is enough to make the adjustment once.

Oscillator frequency measurement can be performed in three ways with:

- the frequency meter connected to the measuring point TP1
- the HF transceiver
- the stopwatch measuring Beacon time

Oscillator frequency correction can only be performed via the Bluetooth serial port with the command #o\_\_ (#O\_\_). The characters \_\_ are replaced by +n, -n, s or ?, where n is a number from 1 to 9.

The simplest and fastest way to adjust the oscillator frequency is by using a frequency meter connected to TP1 where a 16 MHz ( $f_{osc}/2$ ) signal is present.

If the measured frequency is  $< 16$  MHz, increase the frequency of the oscillator with the command #o+n. Select the number n depending on the frequency deviation from 16 MHz. If in the first step the oscillator frequency did not reach 16 MHz, repeat the command #o+n as many times as necessary until you measure the frequency closest to 16 MHz.

If the measured frequency is  $> 16$  MHz, use the #o-n command to decrease the oscillator frequency until you measure the frequency closest to 16 MHz.

When you have completed the oscillator frequency adjustment and reached the value closest to 16 MHz, save the correction factor permanently in the EEPROM with the #os command.

You can read the current absolute value of the correction factor at any time with the command #o?. The correction factor absolute value is limited to +31 and -32.

The receiver of the HF transceiver can also be used as a frequency meter. Connect a piece of wire to the TP1 as an antenna to get the stronger signal on the receiver. Look for an oscillator signal around 16 MHz. Depending on the frequency, correct the oscillator frequency as described above and save the correction factor in the EEPROM.

Ultimately, you can adjust the oscillator frequency by accurate measurement of the Beacon time. Record a single letter E message in RAM memory and set the Beacon time to 900s (15 min). Activate the beacon and use a stopwatch to measure the time between two keyed out letters E. If the measured time is longer than 900s, the oscillator frequency should be increased or decreased if the time is shorter than 900s. As previously described, perform the oscillator frequency correction until the Beacon time is as close to 900s as possible and save the correction factor in the EEPROM.

