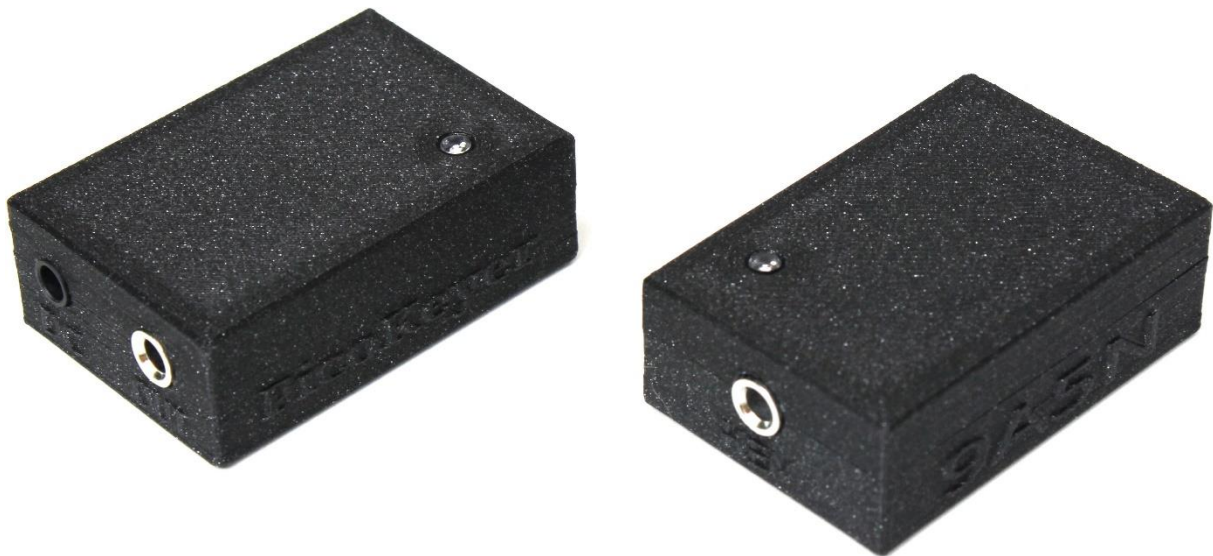


9A5N PicoKeyer & Pocket QSO Logger

User manual



Introduction

The 9A5N PicoKeyer is the "little brother" of the QRQ Keyer, which the author developed ten years ago for learning and practicing telegraphy. PicoKeyer also uses a 16-bit microcontroller with an operating frequency of 32MHz and a Bluetooth serial port module for wireless connection with PCs and mobile devices.

PicoKeyer is primarily developed for use in field CW operation, however, can be used at home also like small WinKeyer compatible CW interface.

PicoKeyer does not have its own sidetone and uses the sidetone of the connected radio station. All CW parameters are adjusted and changed with Bluetooth commands.

To easily set up and change PicoKeyer parameters, you can use a specially developed free application "9A5N Keyer BT Companion" for Android mobile devices available on Google Play (soon on App Store also).

Wirelessly connected via Bluetooth network with a smartphone/tablet on which the "Pocket QSO Logger" application is installed, enables field CW operation without the use of CW paddle.

Working without CW paddle makes it much easier for the operator to work CW in the field:

- No need to carry an expensive and sensitive paddle that can be large and heavy
- No need for a stable base and space for the paddle to stand on while keying
- The paddle is not exposed to outdoor conditions (temperature, dust, moisture)
- The paddle is not exposed to potential mechanical damage (drops, impacts)

Pocket QSO logger works like contest loggers (keying the entered call sign and report, macro messages and messages written with the keyboard) and the CW paddle is unnecessary.

Pocket QSO Logger, unlike other logging applications for smartphones, besides logging QSO's and exporting to ADI file, enriches Bluetooth connection with 9A5N keyers and YAESU radio stations FT-857 and FT-897 with built-in Bluetooth module for CAT. Any change in operating frequency and/or operating mode transfer to the log and no errors can occur during logging QSO's.

The possibility of wireless connection to a PC via a Bluetooth virtual RS-232 serial port (internal Bluetooth serial port module) allows real-time keying control as well as connecting the keyer to logging programs that support WinKeyer.

The previously mentioned features make the 9A5N PicoKeyer and Pocket QSO Logger completely different from keyers and mobile applications from other manufacturers on the global market.

PicoKeyer - specification

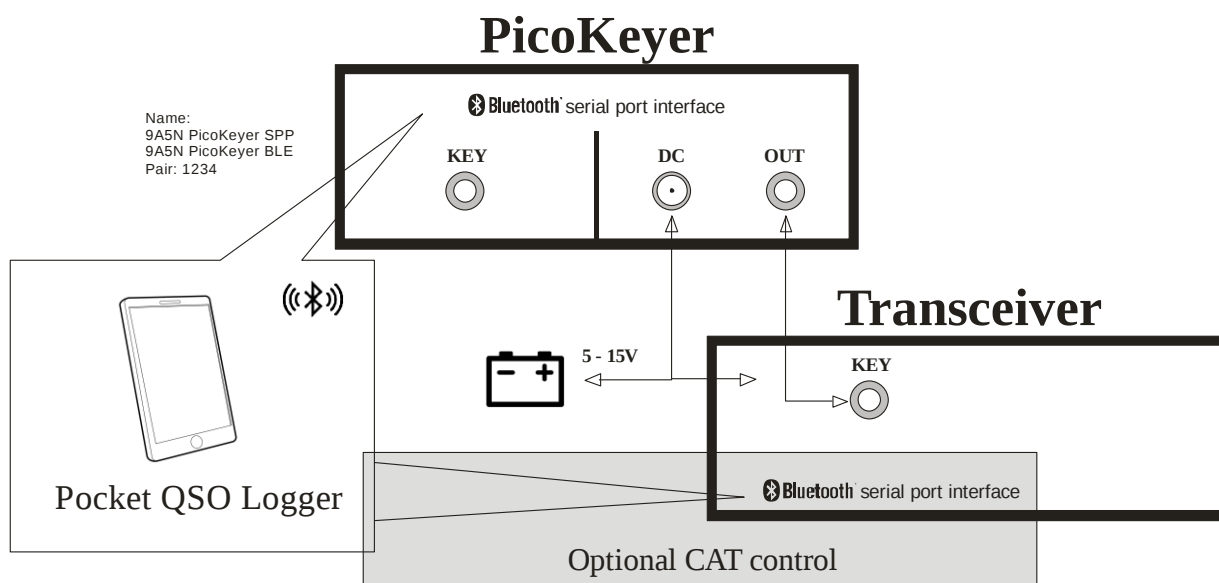
- Bluetooth module for wireless connection with a PC and mobile devices
- Iambic A & B, Ultimatic and "BUG" mode
- Adjustable keying speed from 5 to 99 WPM
- Auto spacing function
- Serial port input buffer with a capacity of 410 characters
- CW decoder for keying accuracy control
- Compatibility with WinKeyer 3.1 protocol
- Built-in magnets for easy attachment to the radio station steel housing
- 3D printed PLA housing in Galaxy black color
- CW output with MOSFET transistor 60V/500mA
- Power supply: 5 - 15V DC / 22 mA max.
- Dimensions: 46x31x16 mm
- Weight: 16 g

Pocket QSO Logger – specification

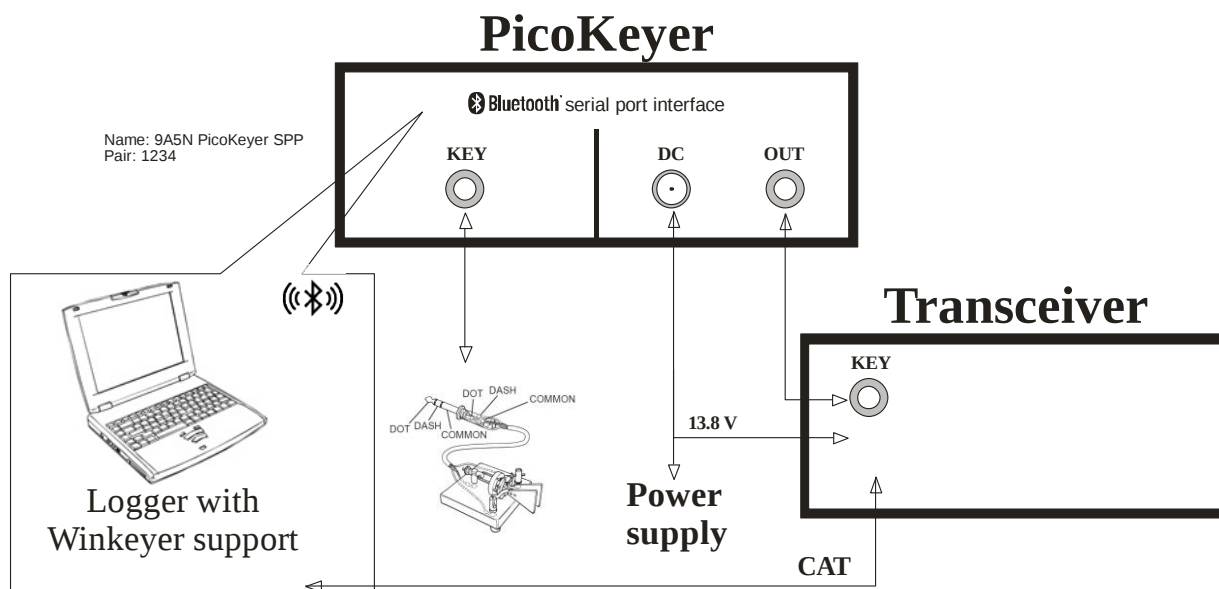
- Free application for Android available on Google Play (soon on App Store also)
- CW, SSB and FM mode of operation
- Bluetooth connection with 9A5N keyers (QRQ and Pico)
- Slider to adjust CW speed
- Six macros for CW mode
- Window for entering CW messages with the keyboard
- Bluetooth CAT control for YAESU FT-857 / 897
(If a Bluetooth serial port module is installed in the radio)
- Grid locator calculation based on the current GPS location
- Exporting the log to an ADI file and sending it by e-mail to an address of your choice
- Five predefined ADI file formats (POTA, WFF, IOCA, SOTA, Generic)
- Can be used for field QSO logging without any device connected

Detailed information and instructions for the Pocket QSO Logger application you can find at
pocketqsologger.com

Portable installation



HAM shack installation



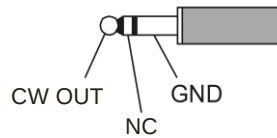
Scope of delivery

1. 9A5N PicoKeyer
2. Power supply cable 50 cm long with PP-002D connector on one side

The other required cable(s) should be provided by the customer.

Connectors on the 9A5N PicoKeyer

3. OUT – connector

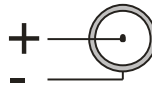


3-pole 3.5 mm socket for CW signal output. The output is of the "open collector" type, connected between the top and ground of the connector.

The OUT of the keyer connects to the key input on the transceiver (KEY). As a connecting cable, it is recommended to use a prefabricated audio stereo cable with 3.5 mm connectors. The transceiver built-in keyer should be Off as for the "Straight key".

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

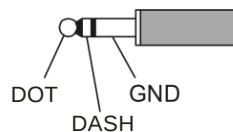
4. DC – connector



PJ-040 socket for connecting the 5-15V DC power supply. To connect to the power source, you need to make a cable with an appropriate connector or terminals on the power source side, depending on the connection method on the power source.

Any 5-15V DC power supply, the source from which the transceiver is powered (battery, 13.8 V supply) or one of the connectors on the transceiver where 5-15V voltage is available can be used. Powering from one of the connectors on the transceiver is also the most convenient because the keyer and transceiver will turn on and off simultaneously.

5. KEY – connector



3-pole 3.5 mm socket for connecting a telegraphy paddle. Only paddles that require the use of a keyer can be connected. "Straight key" cannot connect.

6. Bluetooth serial interface – internal module

The keyer has a built-in Bluetooth serial port module that enables wireless connection with a PC, tablet, smartphone etc.

On the Bluetooth wireless network, the keyer appears with the name "9A5N PicoKeyer SPP" and "9A5N PicoKeyer BLE". "Pair code", if required, is 1234. After pairing, a virtual serial port will open on the connected device for communication with the keyer.

9A5N PicoKeyer commands

All keyer commands and settings are available through the Bluetooth serial port only. In order for the keyer to distinguish the command from the standard text sent over the serial link, a # sign is added before the abbreviation for a specific command, for which there is no CW code and the keyer cannot key it. Command abbreviations are not “case sensitive” and can be written in upper- or lower-case letters.

WinKeyer commands can be used in parallel with the PicoKeyer commands if the PicoKeyer is in Host open mode.

List of commands:

#h	HELP	
#i	KEY INV/NOR	
#k	INFO	
#m_	MODE	_ = a/b/u/s/?
#p	AUTO SPACE ON/OFF	
#u__	SET MAX WPM	__ = 20-99/?
#v__	SET MIN WPM	__ = 05-70/?
#w__	SPEED IN WPM	__ = 05-99/?
#z	CWOUT ON/OFF	
[__	PROSIGN	__ = TWO CHAR.

WinKeyer supported 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
0B <00>	Key up immediate
0E <nn>	Mode bits
15	Request WinKeyer status
16 <nn>	Input buffer commands
1B [C][C]	Merge two letters into a Prosign
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 to emulate WinKeyer operation.

Description of the 9A5N PicoKeyer

Bluetooth serial port

The 9A5N PicoKeyer has a built-in Bluetooth module for the RS-232 serial port. The module allows wireless connection of the keyer with a PC, tablet or smartphone running the appropriate application for communication via the RS-232 serial port.

The module supports two operating modes, SPP and BLE and appears on the Bluetooth network with two names:

"9A5N PicoKeyer SPP"	Pair:1234
"9A5N PicoKeyer BLE"	Pair: Not required

Depending on the operating system of your device, you will connect to the module under the appropriate name:

- Windows	9A5N PicoKeyer SPP
- Android	9A5N PicoKeyer SPP or 9A5N PicoKeyer BLE
- IOS	9A5N PicoKeyer BLE

Communication in SPP mode is faster than communication in BLE mode and if possible, use SPP mode.

Once you have found the module on the network, Pairing should be performed in SPP mode. The Pair code is "1234".

In BLE mode, pairing is not required.

After pairing, the operating system will create a virtual RS-232 port. In Android and IOS systems, the COM port is defined by the name of the connected unit, while in Windows systems the assigned COM port number should be found in the Control Panel under Hardware > View devices and Printers > Properties > Services.

To communicate with the keyer, you can use the "HyperTerminal" program for the Windows operating system, or the "9A5N Keyer BT Companion" application for Android and IOS (soon).

Remark:

OS Windows 10/11 can sometimes change the numbers of virtual COM ports themselves. If communication with the keyer has stopped working, check the new COM port number assigned to the keyer.

CW speed

The keying speed is set by sending a command via the Bluetooth serial port.

The basic keying speed setting range is 15 to 35 WPM. The user can adjust the speed adjustment range to his needs using commands via the Bluetooth serial port:

- #u__ - maximum keying speed adjustable with a slider in WPM (20-99)
- #v__ - minimum keying speed adjustable with a slider in WPM (05-70)

Example:

#u90 – the maximum keying speed adjustable with the slider will be 90 WPM
 #v30 – the minimum keying speed adjustable with the slider will be 30 WPM

Regardless of the position of the speed slider, the keying speed can be changed at any time with the speed command via the Bluetooth serial port:

- #w__ - desired keying speed in WPM (05-99) or keying speed info ?

Example:

#w33 – keying speed is set to 33 WPM
 #w? – keying speed is 33 WPM

CW code generating modes

Keyer supports all CW code generation algorithms and semi-automatic key (Bug) operation simulation.

The table lists the codes for selecting a particular operating mode:

Code	Algorithm
a	Iambic A
b	Iambic B
u	Ultimatic
s	Bug simulation

The CW code generation mode is selected using commands via the Bluetooth serial port.

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

Combinations:

- #m_ - selects one of the CW code generation modes.

Instead of the _ character, the letters a,b,u and s can be used according to table 2.

- #m? - prints the current mode on the screen

Example:

#ma - the keyer generates CW code using the Iambic A algorithm

The CW code generation mode is saved in the EEPROM memory.

Automatic pause between characters

To improve the telegraph operator's sense of the exact length of pauses between characters, the keyer has an option to control the length of pauses. By activating automatic pauses between characters, the keyer will extend all pauses shorter than three dots to exactly three dots and "gluing" of characters will be avoided.

Automatic pause generation is activated/deactivated by commands via the Bluetooth serial port. The command acts as a toggle switch. If automatic pause is not active, sending the command #p will activate it and vice versa.

The currently selected state of the automatic pause can be checked in any moment with the command #k (#K) – keyer info.

Example:

First command #p – auto pause activated

Second command #p – auto pause deactivated

Status of automatic pause is saved in the EEPROM memory.

Swapping inputs for dot/dash

Most telegraph operators use a paddle so that dashes are on the right side and the dots on the left side of the paddle. It is a normal dot/dash layout. However, there are telegraph operators who using the opposite arrangement or are left-handed for whom the opposite arrangement is more natural. It is a reversed dot/dash layout. To cover both cases without the need for electrical switching of the lines on the paddle, the normal/reverse arrangement can be changed programmatically.

Input switching for dots/dashes can be done via the Bluetooth serial port with the command `#i` (`#I`). The command acts like a toggle switch. If the dot/dash inputs are normal, sending the `#i` command will reverse the dot/dash inputs and vice versa.

The currently selected dot/dash inputs state can be checked in any moment with the command `#k (#K) – key info`.

Example:

First command #i – dot/dash reverse

Second command #i – dot/dash normal

Status of dot/dash inputs swapping is saved in EEPROM memory.

"Key down" function for adjusting the linear amplifier

A constant signal from the transmitter is required to adjust the output stage of the linear amplifier.

The Keyer will close/open the CWOUT output by sending the Bluetooth command #Z (#Z).

Example:

#z – CWOUT closed #z – CWOUT open

„Prosign“ function

The Prosign function allows you to create CW codes for letters of national alphabets by combining the CW codes of two letters of the English alphabet.

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

Example:

We want the keyer key out „sretan rođendan“. CW code of letter “đ” is -..-- and we can get it with combining CW codes of letters D (-..) and M (--).

We are entering: sretan ro**dm**endan

Informative and auxiliary commands

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

- #h – help - print a list of all keyer commands on the screen
- #k – keyer info - print current keyer settings on the screen

