JS8Touch User Manual
Ken Thompson, M0GBP
Version 1.1.1b
15 June 2022

Table of Contents

Introduction

JS8Touch is a Python application that uses the JS8call API to provide a GUI that can be used with:

- The Raspberry Pi official touchscreen

  Touchscreen operation is aimed at portable operations where a keyboard or mouse is undesirable. It achieves chat operation by having programmable macro buttons, like FL-Digi, than can be used to transmit pre-defined messages.

- A Raspberry Pi or other Linux machine with a mouse, keyboard and HDMI monitor.

  If a keyboard and mouse is attached a full mouse/keyboard chat capability is possible for use in, for example, small portable laptops, or a conventional Raspberry Pi setup.

- JS8Touch can be used on the same machine as JS8Call or a different machine on the same network. This allows the GUI to be in a room other than the room housing the Transceiver.

JS8Touch is programmed using Python3 and Tkinter. As such it should run on any Linux machine. It would probably run on Windows.

I have never run portable, but I hope to one day soon. I would be interested to add functionality to JS8Touch for this type of operation and also functionality that suits small screen use. But JS8Touch is not a replacement for JS8Call's all encompassing GUI.

Ideas for extensions or bug reports are best provided in the Issues section of the Github repository.

Installing and Running JS8Touch

For installation and running see the file js8touch/Readme.md

After being started JS8Touch will wait for connection to JS8Call. This may take up to 15 seconds. Once connected the title bar will show the version number of JS8Touch and of JS8Call.

Quick Start Guide

- To call CQ

  Press the |CQ> Macro Button

- To reply to a CQ automatically with a preset message

  Click on an entry in the Band Activity Table

  Press the appropriate Macro Button e.g. |REPLY>

- To reply to a CQ by typing text

  Click on an entry in the Band Activity Table

  Press the Selected Frequency Button which now shows the callsign of the selected entry.

  Type your text into the Transmit Text area

  Press SEND

Band, Speed and Offset Control

Note: The SPEED button does not currently work due to a bug in JS8Call. Change Speed by using the JS8Call GUI.

The status area at the bottom of the screen has BAND and SPEED buttons. These buttons show the current band and speed. Pressing the button cycles around a set of bands and speeds defined in config/bands.txt and config/speeds.txt. These files can be edited to provide a set of bands and speeds of your choosing.

The OFFSET button allows the Current Offset to be selected. Since JS8Touch has no waterfall a random free frequency is selected by comparing up to 100 candidates against all the frequencies in the Band Activity Table

The UTC from the computer’s clock, and the selected dial frequency, speed, and audio offset, as obtained from the JS8Call, will be displayed in the status area. The offset is initialised to a value set in config/bands.txt when the band is changed.

The Band Activity Table

The left of the screen shows the messages received by JS8Call. This is not a copy of JS8Call’s area of the same name but is generated by JS8Touch from RX.ACTIVITY and RX.DIRECTED

messages. It will therefore differ from JS8Call's display.

The Table shows one line per frequency received; all signals within 10Hz or with the same source callsign appearing on the same line. So if two stations are conversing on the same frequency both station's messages will be shown. New frequencies are added to the top of the window. Click on an entry to select it.

The columns of the table are:

- CALL – The callsign of the station transmitting the current message.
- Age – The age, in minutes, of the last frame received on the frequency. The entry will be deleted after a number of seconds the value of which may be adjusted in config/config.txt. The selected entry is not deleted.
- Snr – The latest reported signal to noise ratio.
- Message – The text of the messages received on the frequency. Text is shifted right to left with a __ to show that the start of a message has rolled off the display. To display the complete QSO in the Received Text Area select the message.

The button and checkbox below the Table clears the Table and inhibits storage and display of Heartbeat messages and heartbeat replies. config/config.txt allows the initial state of the 'See HB' checkbox to be set.

Selection and the Received Text Area

Selecting a frequency in the Band Activity Table will show all the messages received at that frequency in the Received Text Area. The display will be updated as frames are received. Selecting a frequency will also display Callsign, Offset, and the Speed of the latest received message in the Selected Frequency Button below. This button can be used to initialise the sending of messages. The Received Text Area will also show transmitted text in red.

Pressing CLEAR clear's the Received Text Area and deselects the entry.

Sending Messages

Messages can be sent by a number of methods. Whatever the method, the message when sent will appear in the Transmitted Text Area in red and, when transmission is finished, will be transferred to the Received Text Area. During transmission the SEND button will turn to a green WAITING while waiting for a frame to transmit, and to a red TX while PTT is on.

There are two ways of sending messages:

- Using the macro buttons
- Using a keyboard to send free text

To send a message JS8Call requires the callsign, audio offset, and message text. Own station callsign followed by the : character, that is required at the start of a message, is appended by JS8Call.

Using Macro Buttons

Ten macro buttons fit vertically on the right of the screen. The title and effect of the macro buttons is defined in the file config/macros.txt.

The NAME field contains the text displayed on the screen. I have used | to indicate that the Transmit Text area is cleared first and > to indicate that the message is automatically transmitted.

The OFFSET field controls the audio offset of the transmission.

- CALL - transmit on the offset of the selected activity
- CHAT - transmit on a random free offset in the chat region (1000 to max offset)
- HB - transmit on a random free offset in the heartbeat region then return the frequency to its current offset. If SEE HB is Off then Heartbeat messages are not stored so the algorithm cannot test for used frequencies.
- CURRENT - use the Current Offset

The TEXT field contains plain text, the characters | and > and macro variables

If the first character is | the Transmitted Text screen will be cleared. If the last character is > the message will be sent automatically.

Macro variables are enclosed in []. The tag in these variables are replaced when the message is sent

- [CALL] – The callsign of the selected frequency as obtained from the Selected Frequency Button.
- [NAME] – your name from config/config.txt
- [QTH] – your QTH from config/config.txt
- [SNR] - SNR as obtained from the Selected Frequency
- [GRID],[GRID4],[GRID6] – Maidenhead grid as obtained from JS8Call

- [MYCALL] – My callsign as obtained from JS8Call
- [SPACE] - insert a space character. Useful as the format of macros.txt does not allow trailing spaces.

I have constructed a set of useful macros of this type CQ, REPLY, SNR, ME, BRAG, BYE, HB . The other macros are used to assist keyboard operation.

Using the Keyboard

Free text transmission requires that the audio offset and, for some messages, the callsign of the contacted station be set. There are number of ways to do this:

- Use the Selected Frequency Button

  Pressing this button will place the callsign into the Transmitted Text Area and will set the offset for transmission to the offset defined in config/config.txt which is one of call, chat, or current. See the Macro Button Section for the definition of these. The Area is now ready for text to be typed and SEND to be pressed to transmit it.

- Typing Text

  Type the callsign, if required, and the message to be sent into the Transmitted Text Area. Then press the SEND button to transmit it. The offset is set to to the offset defined in config/config.txt which is one of call, chat, or current. See the Macro Button Section for the definition of these.

- Use a Macro Button

  Macros such a kb-call, kb-chat, and kb-curr are programmed to set the audio offset, and to place the callsign of the Selected Frequency into the Transmitted Text Area. The Area is now ready for text to be typed and for SEND to be pressed to transmit it.

Using the Raspberry Pi Touchscreen

NOTE: I could not make the touchscreen work with Bullseye unless I changed the Pi's /boot/config.txt to use the fake kms (dtoverlay=vc4-fkms-v3d) instead of the real one (dtoverlay=vc4-kms-v3d) .

To use the Raspberry Pi official touchscreen just connect it to the Pi. If the touchscreen alone is used the GUI will fit the screen and touches will register correctly. If an additional HDMI monitor is used the Pi's Screen Configuration Utility might be required to arrange the two screens. Touch will not register correctly unless the Touchscreen is the leftmost top monitor.

Out of the box config/config.txt is set to disable backlight control. If backlight control is required rpi-backlight must be installed and permissions set first. The full documentation is here https://pypi.org/project/rpi-backlight/ but the installation instructions are:

- install from pip3

  pip3 install rpi-backlight

- set permissions

  ```
  echo 'SUBSYSTEM=="backlight",RUN+="/bin/chmod 666
  /sys/class/backlight/%k/brightness /sys/class/backlight/%k/bl_power"' |
  sudo tee -a /etc/udev/rules.d/backlight-permissions.rules
  ```

- reboot

  sudo reboot

- enable in config/config.txt

  enable_backlight = yes

Effect of Using AUTO in JS8Call

My little experience with using AUTO in JS8Call, including heartbeat ACK's, is that it does not interfere with messages sent by JS8Touch. However the messages transmitted will not be seen by JS8Touch. The PTT label in the status area will change to red to indicate transmission.

INBOX Messages

There is no specific function to send MSG type messages however macros can be programmed to assist this.

Logging

It would be nice if logging could be achieved from JS8Touch but the API does not include this.

Debugging

A log of the inputs and outputs of JS8Touch and of key events is displayed in the terminal window and written to the file debug/ log.txt. Some messages to/from JS8Call are not logged.