

**NAME**

**SkantiControl** – a program to control the Skanti TRP8000 series transceivers with look & feel similar to the CU8000 control unit.

**SYNOPSIS**

**SkantiControl** [ **--lang**=<language code> | **-l** <language code> ]

**DESCRIPTION****CAVEAT EMPTOR**

*This is only the third version of a full-blown manual. If you find any errors in it, please report. I'm not fluent in English, so apologies for the bad syntax. Any corrections to the worst sentences are welcome.*

*This program has been extensively tested only on the author's own TRP8255. The Skanti "TRP8000 REMOTE CONTROL COMMUNICATION PROTOCOL" manual seems to be valid for all models, but if on some of them the program behaviour is not correct (e.g. in a TRP875x the power level setting) please report. That manual (or at least the issue 2A) is indeed far from being complete.*

**BEWARE OF**

*When using the program pay attention to the following details:*

- The CU8000 to which the microphone is connected **LOSES PRIORITY** and gets disconnected from the TU when a remote command is sent to it and reacquires it only following the **RELEASE PRIORITY** command or after a 5s timeout without further commands. During this time the microphone, AUX and PTT inputs connected to the CU8000 are deactivated. Since in order to read the status (S-Meter, PWR-Meter, tune status) in the current implementation it is necessary to continuously send commands to the CU, the activation of this function prevents putting the RTX into transmission in the normal way. To do this, one could use the corresponding remote command (the [TRANSMIT] button or the equivalent hamlib command), but then the audio signal would necessarily have to come from the TELEX input. In short: if you intend to operate in voice or digital modes, disable the *Read Status* option. Operating in CW instead this is not necessary.
- To operate in digital modes (FT8, PSK31, etc.) with the RTX set in USB mode, connect the audio signal and the PTT to the AUX input (it is advisable to use the "LINE TRANSFORMER BOARD", if available) and disable the option mentioned above. If you intend to work with WSJT-X using the integrated hamlib-compatible server for frequency control (in this case it is recommended to use the "RIG" mode) you must also set the "Transceiver" option to "Disabled". For the same reason explained above it is also advisable not to use the PTT command via hamlib, but a dedicated serial or parallel port. At least this is what the author did to work with his own TRP8255 in conjunction with WSJT-X. If anyone wanted, they could check if it is possible to avoid all this, and with what consequences, by using the TELEX input.

**QUICK START GUIDE**

Before executing the program for the first time, install in your system the fonts provided in the *Fonts* subdirectory. Under Linux, if the installation is done using one of the provided Debian packages this is automatically done; under Windows they must be manually installed, or at least must be installed the Digital Dream Fat and the DejaVu Sans Mono fonts, since chances are that the Arial fonts are already installed in the system.

Under Linux, the program executable and files are automatically installed in the proper locations by the provided Debian packages.

Under Windows, the program folder now can be put in any location, but the program executable must be in that folder, since its location is used to discover the location of other files. You can create a launch icon on the desktop.

At first start of the program, go to the *Options* menu and configure the *Serial port* submenu item. If the serial port where the CU8000 is connected is not one of the first two (ttyS0 and ttyS1 for

Linux, COM1 and COM2 for Windows and ttyAMA0 and ttyUSB0 on Linux on a Raspberry PI), or it is a USB one, select the *Custom* submenu and write the name of the wanted port, e.g. /dev/ttyS4 or /dev/ttyUSB0 for Linux or \\.\COM11 for Windows. The /dev/ prefix under Linux and the \\.\ under Windows are optional. If omitted, they will be automatically added.

The serial speed defaults to *2400 baud* since this is the default speed of the CU8000, so it is not required to go to the *RS232 speed* submenu of the *Options* menu to set it, at least if your CU8000 is in the standard configuration. Otherwise, see The *RS232 speed* submenu in the *Options* menu.

The configuration is automatically saved every time an option is modified. An exception to this are the starting size and position of the main window, which are saved only at program end, read and restored at program start. See also the *BUGS, LIMITATIONS AND REMANING THINGS TO DO* section.

*NOTE. If you use a USB serial port under Linux and have more than one of them, it is wise not to use the /dev/ttyUSBx names, since the enumeration of the USB bus may assign different names from one boot to another (what now is /dev/ttyUSB0 at the next boot may become /dev/ttyUSB1 and so on). Use instead the names found in the /dev/serial/by-path/ folder, that are unique and persistent across reboots. The names found in the /dev/serial/by-id folder also are persistent, but if there are two or more equal USB serial ports, these names may be not unique.*

Then click on the [Supply ON/OFF] button in the lower right corner of the main window. This button only starts and stops the program's connection with the CU8000, does not switch ON and OFF the transceiver. The program should connect to the Skanti CU8000 and begin to work.

If not, double-check the configured port, the serial connections and make sure that in your CU8000 is installed the remote control version of the firmware and that the remote control feature is enabled in the configuration PROM. Under Linux also check that the user running the program is member of the group to which the serial port belongs, in Debian and Ubuntu the *dialout* group. If not, in a terminal as superuser do *adduser user dialout* where *user* is the name of the user that will use the program, then reboot the system. This is not mandatory, but it is the easiest way to be sure that the change goes in effect.

If all goes as it should, then you can configure the other options in the *Options* menu as desired. See below The *Options* menu for more informations.

From version 0.99 there is also a TCP server compatible with hamlib net rigctl, listening by default at the address 127.0.0.1:4538, whose implementation is currently only partial and approximate, a little more than the bare minimum needed to make it work with WSJTX 2.7.0, FLDIGI 4.2.06, GRIG 0.9.0, XDX 2.92, QSSTV 9.5.11 and XLOG 2.0.25. Tests has been made mostly under Linux. When another application connects to the SkantiControl server, the NET "LED" at the top left of the main window lights up, and then switches off when all clients have disconnected.

*NOTE. Some of the commands and/or buttons may do nothing on your TRP8000 if that function is disabled in the configuration PROM (e.g. the [VERY NARROW] button) or the relevant option is not installed (e.g. the [SQUELCH] button). This manual assumes a fully open configuration PROM and all options/features available.*

## USING THE PROGRAM AND FUNCTIONS NOT AVAILABLE IN THE REAL CU8000

Most of the buttons should behave as those in the real CU8000, e.g. pressing [RX], [TX], [ENTER] the receive frequency is copied in the transmit frequency. Some buttons have also autorepeat, as in the real CU8000. If you find differences with respect to the behaviour of the real CU8000, please report. A few commands however do not have been yet written/completed, see the *BUGS, LIMITATIONS AND REMAINING THINGS TO DO* section for more informations.

Some buttons and controls however have different and/or additional capabilities or only exists in the program and not in the real CU8000:

- The tuning can be done not only with the usual [TUNE ↓] and [TUNE ↑] buttons but also with the mouse buttons and the mouse wheel. Pressing the left mouse button over one of the RX and TX frequency digit decrements that digit, pressing the right mouse button increments that digit, in the Skanti CU8000 spirit (decrements at left, increments at right). Rotating the wheel with the cursor over one of the abovementioned digits has the same effect. Rotating the mouse wheel anywhere else increments or decrements the frequency by the tune rate in effect. This is the fastest tuning method, since a single character must be sent to the CU8000. Since the TX frequency can be set only in 100Hz steps, when the RX frequency is incremented or decremented in 10Hz steps and the Transceiver submenu is set to *Deferred update* or *Immediate update*, or the [TX=RX] button is pressed, the TX frequency is set to the nearest 100Hz value.

Instead of the mouse wheel you can use also the sdrgadgets VFO tuner V2, which has been tested and found to work directly out of the box. The button integrated in the knob, at least in the default configuration, has been found to type a space with autorepeat (which starts too soon, in the author's humble opinion), so in this program in all modes except CW that button makes a RX/TX switching as the keyboard SPACE BAR key. Or at least makes that if you can press the button fast enough to not start the autorepeat feature.

- The [TX=RX] and the [RX=TX] buttons copy the RX frequency to the TX frequency ([TX=RX]) or the TX frequency to RX frequency ([RX=TX]) allowing a quick return from split frequency mode when the *Transceiver* submenu is set to *Disabled*. If the RX frequency is set in 10Hz increments and the [TX=RX] button is pressed, the TX frequency is set to the nearest 100Hz value.
- The tune rate can be selected either by pressing the [TUNE RATE] button or by clicking on the corresponding "LED". The behaviour is however different: when using the [TUNE RATE] button, the behaviour is the same of the CU8000, so the figures lower than the chosen tuning step are left active and displayed. By clicking on the corresponding "LED", these figures are reset to zero and the 10 Hz one, if present, not displayed anymore.

Example: being at 10Hz step, frequency 12345.67 kHz:

- clicking the [STEP] button: step becomes 100Hz, the frequency stays at 12345.67 kHz and clicking on the [TUNE ↑] button the frequency becomes 12345.77 kHz;
- clicking on the "LED" under the 100Hz figure: step becomes 100 Hz, the frequency is reset to 12345.70 and displayed as 12345.7 (no more 10Hz figure) and clicking, as above, on the [TUNE ↑] button, the frequency becomes 12345.8 kHz.
- Pressing once the [SET TIME] button starts, as usual, the time input on the TX frequency display. Pressing it one more time sets the TX frequency display to the current PC time, possibly converted to UTC time if the *Use UTC time* option is checked. [ENTER] then writes the new time to the Skanti CU8000. The state machine of the [SET TIME] button has been revised to closely follow the behaviour of the real CU8000, so pressing [RX] after [SET TIME] the wakeup time can be entered and pressing [RETURN] after [SET TIME] the "Enter dormant state" command is issued. Pressing [RCL] after [SET TIME] issues the "Read wakeup time" command, but the time is only shown in the CU TX display, so a warning message about that is displayed in the message display and after a 5s delay the normal operation is resumed.
- The message display shows informations and/or hints about the command being executed or status/info messages (e. g. tuning and reduced power status) or information messages (e.g. those coming from the *Read XX8000 version* and *Read TU8000 configuration* submenus). Double-clicking on it clears display.
- The [TRANSMIT] button allows to remotely put the transceiver in transmit mode when the priority is assigned to the program and not to the CU8000, e.g. when the *Status reading* submenu is enabled. See also the *Release priority* submenu in the *Other commands* menu. The button caption becomes RECEIVE when in transmit mode and returns to TRASMIT when the RTX is put back in receive mode.

- Some commands can be issued also using the keyboard. See the **KEYBOARD COMMANDS** section for more informations.
- To allow entering a negative BFO offset using the on-screen buttons, when the *Set BFO frequency* submenu is selected, a small [+/-] button is shown at the right of the [ENTER] button. To store in the CU8000 the current BFO frequency use the *Store BFO frequency* submenu item in the *Other commands* menu. The current BFO frequency can be also modified with the [BFO ↓] and [BFO ↑] buttons. The current BFO frequency is also stored in the state files saved and restored when a state file is reloaded.
- If you want to use the Skanti transceiver in digital modes, e.g. FT8, it is advisable to connect the audio/PTT coming from the PC to the AUX audio/PTT input and disable the *Enable status* submenu. When the *Enable status* submenu is checked, the priority is never released to the CU8000, so the PTT must be operated with the remote command, i.e. the [TRANSMIT] button and the TX audio must come from the TELEX input. The same holds for phone operations. In CW mode the key is instead active whether the *Enable status* submenu is checked or not.

## THE MENU TREE

The *File* menu.

This menu contains the submenus dealing with the file management and the *Exit* submenu.

The *Load state* submenu.

This submenu is active only after the program has successfully connected to the CU8000. It allows to load in the program and in the CU8000 a previously saved state (frequency, mode, etc.).

The *Save state* submenu.

This submenu is active only after the program has successfully connected to the CU8000. It allows to save the current state in a file. That state can then be reloaded using the *Load state* submenu. If the saved state file name is "F500-XXXXXX.dat" or "F2182-YYYYYY.dat", then the buttons [2182] and [500], of little or no use in an amateur environment, are renamed [XXXXXX] and/or [YYYYYY] and these two states can be reloaded by clicking on the corresponding button. XXXXXX and YYYYYY must contain only legal characters in a file name and can be of any length, but only the first six characters are shown in the button caption. If XXXXXX and/or YYYYYY are omitted (i.e. the file name is "F2182-.dat" or "F500-.dat") then the buttons are by default renamed [CH1] and/or [CH2].

The *Save channel* submenu.

This submenu is active only after the program has successfully connected to the CU8000. It allows to save the current state in a file, but in the "Channels" subdirectory. The state files saved in that subdirectory are read at program start and used to dynamically build the *Channels* menu, allowing to restore a saved state with a few mouse clicks.

The *Manage states* submenu.

This submenu is always active. It shows a very minimal file manager that allows to rename, copy, move to the *Channels* directory and delete the stored state files from within the program.

The *Manage channels* submenu.

This submenu is always active. It shows a very minimal file manager that allows to rename, copy, move to the *States* directory and delete the stored channel files from within the program.

The *Reload channel directory* submenu.

This submenu is always active. It allows to rebuild the *Channel* menu item in case that is deemed required. The channel directory should be reloaded and the relevant menu items rebuild every time that directory is modified, adding, removing or renaming a channel. This submenu has been added as a security in case that would not happen or the channel subdirectory is modified by another program.

The *Exit* submenu.

This submenu is always active and allows to close the program. The program can also be closed

with the window decoration button. Before the closing is actually done, a confirmation window pops up. The connection to the CU8000, if already established, is automatically terminated, so there is no need to manually terminate it with the [SUPPLY ON/OFF] button before exiting.

The *Other functions* menu.

This menu contains the less used CU8000 functions and is active only after the program has successfully connected to the CU8000.

The *Read TU8000 configuration* submenu.

This submenu reads the TU8000 configuration and displays it in the message display. An example, taken from the author's own TRP8255, may be "X1A345SF". The meaning of these codes is explained in the section 4.5.1 of the Skanti "TRP8000 REMOTE CONTROL COMMUNICATION PROTOCOL" manual.

The *Read CU8000 version* submenu.

This submenu reads the CU8000 firmware release date and version and shows it in the message display. An example, taken from the author's own TRP8255, may be "910910-920" (CU8000 firmware version 92.0 dated 10 september 1991).

The *Read TU8000 version* submenu.

This submenu reads the TU8000 firmware release date and version. An example, taken from the author's own TRP8255, may be "860708-310" (TU8000 firmware version 31.0 dated 8 july 1986).

The *Switch antenna OFF* submenu.

This submenu issues the "Switch antenna off" command. The antenna gets disconnected and the TX disabled. Both the CU8000 and the program display, instead of the TX frequency, the current time. Don't know what is the difference, if any, from the [TX ON/OFF] button.

The *Program receiver tune step* submenu.

This submenu allows to define the programmable tune step to use, in the range 0.1 ... 99.9 kHz. To actually use it, remember to check the *Enable prog tune step* submenu in the *Options* menu (see below).

The *Read BFO frequency* submenu.

This submenu allows to read the BFO frequency stored in the CU8000. The value is shown in the message display.

The *Set BFO frequency* submenu.

This submenu allows to define a new BFO frequency. To store it in the CU8000 use the *Store BFO frequency* submenu (see below). The BFO frequency can also be modified with the [BFO ↓] and [BFO ↑] buttons.

The *Store BFO frequency* submenu.

This submenu allows to store in the CU8000 the current BFO frequency.

The *Test alarm* submenu.

This submenu allows to test the alarm signal generator. The alarm signal is only sent to the BF amplifier and can be heard in the loudspeaker. To stop it, use the *Stop alarm* submenu (see below). This command has a 7s timeout, so an attempt has been made to get around this feature (maybe a bug?) by resending the command every 3 seconds. Since during the test the CU is not responsive to commands apart the "Stop alarm" and the "Status reading" ones, the program buttons and controls are disabled and reenabled only when the "Stop alarm" command is issued with the *Stop alarm* submenu item.

The *Stop alarm* submenu.

This submenu stops the generation and, if applicable, the transmission of the alarm signal and reenables the program buttons and controls.

The *Send alarm* submenu.

This submenu allows to send the alarm signal. **USE IT ONLY ON A DUMMY LOAD!** Before the alarm signal is actually sent, a confirmation windows pops up. The alarm signal is sent also to the BF amplifier and can be heard in the loudspeaker. To stop it, use the *Stop alarm* submenu (see

above). This command has a 7s timeout, so an attempt has been made to get around this feature (maybe a bug?) by resending the command every 3 seconds. Since during the alarm signal sending the CU is not responsive to commands apart the "Stop alarm" and "Status reading" ones, the program buttons and controls are disabled and reenabled only when the "Stop alarm" command is issued with the *Stop alarm* submenu item.

The *Set option register* submenu.

This submenu allows to define and store the 8 bits contained in the Option register. The value must be entered in binary. For the meaning of these bits see the TRP8000 manual.

The *Set preset register* submenu.

This submenu allows to define and store the 8 bits contained in the Preset register. The value must be entered in binary. For the meaning of these bits see the TRP8000 manual.

The *Set guard register* submenu.

This submenu allows to define and store the 8 bits contained in the Guard register. The value must be entered in binary. For the meaning of these bits see the TRP8000 manual.

The *Fill TU8000 scan buffer* submenu.

This submenu should allow to enter a number of frequency pairs in the scan buffer. The scan buffer functions are not too well understood by the author, so unexpected behaviours may happen. This command also has the 7s timeout whis is not too simple to get around. The only workaround found so far is to enable the status reading feature using the *Status reading* submenu. When a timeout expires and in the subsequent command a DLE is received, the program refuses the command and tries to reconnect the CU.

The *Advance scan buffer pointer* submenu.

This submenu should allow to advance the scan buffer pointer. The scan buffer functions are not too well understood by the author, so unexpected behaviours may happen. This command also has the 7s timeout whis is not too simple to get around. The only workaround found so far is to enable the status reading feature using the *Status reading* submenu. When a timeout expires and in the subsequent command a DLE is received, the program refuses the command and tries to reconnect the CU.

The *Reset scan buffer pointer* submenu.

This submenu should allow to reset the scan buffer pointer. The scan buffer functions are not too well understood by the author, so unexpected behaviours may happen. This command also has the 7s timeout whis is not too simple to get around. The only workaround found so far is to enable the status reading feature using the *Status reading* submenu. When a timeout expires and in the subsequent command a DLE is received, the program refuses the command and tries to reconnect the CU.

The *Self test (auto)* submenu.

This submenu allows to start the automatic self test. These tests do not check the transmit operations, so it is safe to start them at all times. For more informations on the performed tests see the TRP8000 manual.

The *Self test (manual)* submenu.

This submenu allows to start the manually-advanced self test. **WARNING! SOME OF THESE TESTS KEY THE TX ON FREQUENCIES NOT ALLOWED TO AMATEUR SERVICE, SO USE IT WITH CARE AND ONLY ON A DUMMY LOAD!** To switch to the next/previous test use the [DIMMER ↓] and [DIMMER ↑] buttons. These buttons are disabled when a test starts and reenabled when the test finishes. To end the test sequence, press the [ENTER] button. After a delay, hopefully long enough to allow the completion of the reset procedure that follows, the program will try to reconnect to the TU8000. For more informations on the performed tests see the TRP8000 manual.

The *Self test from... (auto)* submenu.

This submenu allows to start the automatic self test starting with the selected test number. These tests do not check the transmit operations, so it is safe to start them at all times. For more

informations on the performed tests see the TRP8000 manual.

The *Self test from... (manual)* submenu.

This submenu allows to start the manually-advanced self test starting with the selected test number. **WARNING! SOME OF THESE TESTS KEY THE TX ON FREQUENCIES NOT ALLOWED TO AMATEUR SERVICE, SO USE IT WITH CARE AND ONLY ON A DUMMY LOAD!** To switch to the next or previous test use the [DIMMER ↑] and [DIMMER ↓] buttons. These buttons are disabled when a test starts and reenabled when the test finishes. To end the test sequence, press the [ENTER] button. After a delay, hopefully long enough to allow the completion of the reset procedure that follows, the program will try to reconnect to the TU8000. For more informations on the performed tests see the TRP8000 manual.

The *Re-read configuration from CU8000* submenu.

This submenu is available only when the *CU8000 firmware 92.0* option is enabled. It allows to re-read the CU8000 status (frequency, mode, etc.) in case that is deemed necessary and set the program accordingly.

The *Release priority* submenu.

This submenu allows to release immediately the priority to the CU8000 without the need to wait 5s for the internal priority timeout to expire. **WARNING: the priority CANNOT BE RELEASED when the Status reading submenu is checked**, since commands are continuously sent to the CU8000 and the released priority is immediately again grabbed. If this command is issued with the *Status reading* submenu enabled, a warning message will be shown in the message display.

The *Reset RTX* submenu.

This submenu allows to reset the RTX. After a delay, hopefully long enough to allow for the completion of the reset procedure, the program tries to reconnect to the CU8000.

The *Restore current state* submenu.

This submenu allow to restore in the program and in the CU8000 the current state (frequency, mode, etc.) in case something has messed up. Hopefully the use of this submenu will never be required.

The *Exit Chan/Scan modes* submenu.

This submenu allow to exit the channel scan/scan buffer state (and, if the firmware v80 is used, the channel mode) and restore the normal operations. The same effect is obtained by loading a new channel (see the *Channels* submenu).

The *Reconnect to CU* submenu.

This submenu allows to terminate the current connection and try to reconnect the program to the CU and is useful whenever a timeout expires (some remote commands have a 7s timeout) and the CU resets. Hopefully the use of this submenu will never be required, since when a DLE is received from the CU the program tries to reconnect to the CU.

The *Band scan window* submenu.

This submenu shows a window allowing to scan a frequency interval with programmable start, end, step, mode and bandwidth, making a graph of the results. Since to speed up as possible the operations the CU programmable step feature has been used, the frequency step can be set only in 100 Hz intervals from 100Hz to 99.9 kHz. Beware that scanning a wide frequency range in small steps may require a very long time. Note that the higher scan speed is obtained using the 100Hz and 1kHz steps, which do not require the use of the slower programmable step feature. The scan data can be saved in CSV format for importing in a spreadsheet or suchlike. The saved CSV files can be also reloaded in the program window, but beware that this function is NOT a general CSV file reading utility, it can read only the files saved by the program itself. In the saved CSV files the first line contains a string describing the parameters used for that scan. The signal strenght data are written in the CU original format, i.e. as an integer number from 0 to 20. The frequency data instead are written as a real number down to 1Hz.

The *Options* menu.

This menu contains the program configurations and options.

#### The RS232 Port submenu.

This submenu allows to select the serial port to which the CU8000 is connected. If that serial port is not one of the first two (ttyS0 and ttyS1 for Linux, ttyAMA0 and ttyUSB0 for Linux on a Raspberry PI and COM1 and COM2 for Windows) or is a USB one, select the "Custom" submenu and write the name of the wanted port, e.g. /dev/ttyS4 for Linux or COM11 for Windows. The /dev/ prefix for Linux is optional. If omitted, it will be automatically added.

*NOTE. If you use a USB serial port under Linux and have more than one of them, it is wise not to use the /dev/ttyUSBx names, since the enumeration of the USB bus may assign different names from one boot to another (what now is /dev/ttyUSB0 at the next boot may become /dev/ttyUSB1 and so on). Use instead the names found in the /dev/serial/by-path/ folder; that are unique and persistent across reboots. The names found in the /dev/serial/by-id folder also are persistent, but if there are two or more equal USB serial ports, these names may be not unique.*

#### The RS232 Speed submenu.

This submenu allows to select the baud rate used to connect to the CU8000. Only two baud rates are possible on the PC-CU8000 connection: the default 2400 baud and 300 baud which, besides being too slow, requires a compatible CU8000 and the change of a jumper in it. It is recommended to use the 2400 baud speed.

#### The Inter-character delay submenu.

This submenu allows to select the delay inserted after every character sent to the CU8000. On the author's CU8000, with the 92.0 firmware version, no delay is required, so the default value for that configuration is 0. If errors in sending the commands to the CU8000 show up, try to increase that value. The delay value previously used was 10ms.

#### The TCP Server address:port submenu.

This submenu allows you to set the IP address and port on which the hamlib net rigctld compatible TCP server listens. By default the server listens on port 4538 of the localhost address, so the value found in this entry will be 127.0.0.1:4538. Note that there are no authentication or encryption of any kind as this server is designed only to connect other programs to SkantiControl, e.g. WSJTX, FLDIGI and XLOG, on the same machine. Therefore, pay a lot of attention if you want to use this feature for what it was not designed for, that is to control a remote TRP8000 via the internet.

*NOTE. The implementation so far made in this version of the hamlib rigctld commands is far from complete and has only been tested with WSJTX 2.7.0, FLDIGI 4.2.06, GRIG 0.9.0, XDX 2.92, QSSTV 9.5.11 and XLOG 2.0.25. For WSJTX, if you want to use one of the split modes, which is highly recommended, it is advisable to use the RIG one because in the FAKE IT one the transmission frequency is rewritten at each reception-transmission switchover, consequently forcing an automatic TUNE which, in addition to slowing down switching, causes unnecessary wear to the transceiver and remote tuner, if any. Remember that the transceivers of the TRP8000 series automatically recall the antenna tuning at the next transmission after each modification of the transmission frequency.*

#### The TUNE when remote changes frequency submenu.

This submenu allows you to choose whether when another application, connected to the SkantiControl TCP server, modifies the transmission frequency, the antenna tuning (TUNE) is called automatically or not by the program. This can be useful or detrimental, depending on the program. For programs like WSJTX, where the frequency, once chosen, remains fixed (in the NONE and RIG split modes, at least), this is useful, otherwise it would be detrimental, as for any modification of the transmission frequency would be automatically recalled by the program the antenna tuning. For the same reason in WSJTX it is advisable to use the RIG split mode and not the FAKE IT one (see above).

#### The CU8000 firmware 92.0 submenu.

This submenu is enabled only after the program has successfully connected to the CU8000. If selected, after verifying that the CU8000 firmware is really version 92.0, the use of the command to read the status of the CU8000 available in that version is enabled. Otherwise, the last program



status is automatically saved on disk and reread at the next start. When this option is selected, the file of the last status is still read at the start of the program, but not reloaded into the CU8000, since not all the parameters are returned by the "Read CU8000 status" command (e.g. the stored frequency of the BFO is not) and others are not readable at all (e.g. the value of the programmable tune step).

#### The *Transceiver* submenu.

This submenu is enabled only after the program has successfully connected to the CU8000. It has three sub-submenus, *Disabled*, *Deferred update* and *Immediate update*. If set to *Disabled* the RX and TX frequencies are independent as on the real CU8000 and split operation is possible. If set to *Deferred update* after about 1s has elapsed after the last change in the RX (or TX) frequency the RX (or TX) frequency is copied to the TX (or RX) frequency. If set to *Immediate update* the RX (or TX) frequency is copied in the TX (or RX) frequency every time the RX (or TX) frequency changes.

*NOTE. If the Transceiver operation is required, it is suggested to use the Deferred update mode, since the Immediate update slows down somewhat the operations: at every change in RX frequency, the TX frequency also must be rewritten and keep in mind that the maximum serial speed is only 2400 baud. When split operation, either local or remote, is wanted, select the Disabled option. This is required by some program, e.g. WSJTX.*

#### The *Status reading* submenu.

This submenu, if checked, starts polling the CU8000 status (received and transmitted signal level, antenna tuning status, etc) every 500ms and displays two bar-meters, one for the received signal strength and one for the transmitted signal power. In the received signal strength meter an approximate translation of the signal level in S units and in the transmitted signal meter a percentual indication of the transmitted power are also displayed. Although beautiful, this slows down the program and prevents to release the priority to the Skanti CU8000, so the microphone/PTT connected to the CU8000 remains disabled, TX audio must come from the TELEX audio input (for TELEX mode) or AUX audio input (for other modes except CW) and the PTT must be operated using the remote command, i. e. the [TRANSMIT] button.

#### The *Use UTC time* submenu.

This submenu, if checked, makes the program use UTC time when programming the CU8000 internal real-time clock or displaying the time in the frequency display when the CU8000 and so the program are in the "TX OFF" state, doing the required conversion from the PC local time. If unchecked, the time programmed and displayed will be the PC time as-is.

#### The *Enable prog tune step* submenu.

This, if checked, enables the programmable tune step feature. Doing that, a fourth state is added in the [TUNE RATE] button behaviour, which becomes 10→100→1000→programmable step. That state is shown in the program as well in the CU8000 as all step "LEDS" off. By default, the programmable tune step value is set to 10 kHz. To define your own, see the *Program receiver tune step* submenu in the *Other functions* menu. To enable in the CU8000 the programmable tune step the bit 4 of the Option register must be set to 1, i.e. 00010000 binary or 16 decimal. That is done automatically by the program when this submenu is checked.

*NOTE. Using the programmable tuning step the time required to change frequency slows down considerably, probably because this is an internal feature of the CU8000 and it is actually necessary to rewrite the entire frequency in the TU8000.*

#### The *Always start in LP mode* submenu.

This submenu, when checked, precisely does what it says: regardless of the power level set in the saved state, or read from the CU8000 itself, at program start the TX power is set to LOW. This feature is intended for those who have a non-Skanti automatic remote antenna tuner which does not like to tune at full power.

#### The *LED color* submenu.

This submenu has three sub-submenus, *Red*, *Green* and *Yellow* and allows to choose the wanted

color for the simulated "LEDS" in the program. The default color is *Yellow* , as on the real CU8000.

The *Font magnification %* submenu.

This submenu allows to define a magnification factor for the fonts used in the main and help windows of the program for buttons, labels, etc. The main menu font is not affected. The need of that arises from the wide variations in font rendering size from one operating system to another. While in Linux the design font size seems adequate, in Windows it is too small, at least in the author humble opinion. So this submenu allows to select the magnification factor that better suit your taste. Six magnification factors are available: 100% (i.e. no magnification), 110%, 125%, 150%, 175% and 200%.

The *Enable all controls* submenu.

The program by default tries to disable non-active buttons for the chosen mode and/or state (e.g. the AGC SLOW and AGC OFF in TELEX mode). If you want to leave all keys enabled regardless of they are active or not, then check this submenu.

The *Show commands* submenu.

This submenu is useful only to debug an incorrect command sent to the CU8000 or to examine the syntax of the CU8000 remote commands. If checked, all commands sent to the CU8000 are also displayed in the message window.

The *Channels* menu.

This menu contains the channels defined with the *Save channel* submenu of the *File* menu.

The submenus of this menu are dynamically built from the channel files saved (see the *Save channel* menu). Choosing one on the submenus loads in the program and in the CU8000 the corresponding state. If no channel file is saved, this menu contains only the *No channels defined* item, which is disabled.

The *Help* menu.

This menu contains the submenus dealing with the program informations, i.e. show the copyright notice, the built-in manual and the program license terms.

The *About* submenu.

This submenu pops up a window showing the program author, copyright, version and the author of the translation, if any. Press on the [OK] button to dismiss it.

The *Manual* submenu.

This submenu pops up a window showing this manual, which is available also as text and PDF file and as a manpage under Linux. Press on the "OK" button to dismiss it. The PDF, html and text versions, including the text version displayed in the program, are machine-generated from the man page using man, col, ps2pdf and sed, so may contain some little formatting issues. However, any errors are guaranteed to be original by the author.

The *GNU License* submenu.

This submenu pops up a window showing the license under which this program is distributed, i.e. the GNU GENERAL PUBLIC LICENSE version 3. You may apply to this program, at your option, any later version of that license. The sources can be downloaded from the same page where You downloaded this program.

*NOTE. The GNU General Public License is always shown in English, since this is the only version with legal validity. The translations in other languages may help non-english speaking people to understand what that license means, but have NO LEGAL VALIDITY. The same holds for the COPYRIGHT section below.*

## KEYBOARD COMMANDS

As said earlier in the *QUICK START GUIDE* there are also some keyboard commands. These commands may be issued either in upper case or in lower case. The available commands are:

+ or ↑	Tune up (+ also sign for BFO frequency offset)
– or ↓	Tune down (– also sign for BFO frequency offset)
/	Change tune rate
*	Invoke TX tune
R	Set receive frequency
T	Set transmit frequency
U	Mode USB
L	Mode LSB
A	Mode AM
C	Mode CW
M	Mode MCW
E	Mode R3E
X	Mode TELEX
.	Toggle between LOW, MEDIUM and FULL power
ENTER	Same as [ENTER] button
SPACE BAR	Receive/Transmit toggle

The frequency value for commands R and T (or [RX] and [TX] buttons) can be entered either with the keyboard or with the on-screen numeric buttons. The same holds for any numerical parameter, e.g. the BFO frequency in the *Set BFO frequency* submenu of the *Other functions* menu.

The menu items are also keyboard-selectable in the usual manner (ALT+the underlined letter/figure).

## CONFIGURATION, STATE AND DOCUMENTATION FILES

All configuration files are kept in the *.SkantiControl* folder of your home directory (Linux) or *SkantiControl* (Windows) directory, which now can be put in any location. See also the *Quick start guide* . The state files are kept in the *States* subdirectory, the channel files in the *Channels* subdirectory. State and channel files, including the *LastState.dat* file, are binary files which share the same format, the configuration file instead is a text file that can be modified using any text editor. The configuration file *config.dat* and the last state file *LastState.dat* are kept instead in the *.SkantiControl* (linux) or *SkantiControl* (Windows) directory. In case the program configuration has messed up, delete (or move, or rename) the *Config.dat* and *LastState.dat* files and the program at next start will recreate them using the default configuration.

However, the program tries to recognize the old versions of the configuration file, keeping the parameters present, assigning the default value to the missing ones and saving the updated version. Only if the file is really messed up gives up trying to understand something and sets and saves the default configuration. In both cases a window is shown explaining what is happening.

The documentation files, including the text, PDF and html versions of the manual and the GNU license v3.0 under Linux are kept in the directory */usr/share/doc/skanticontrol* .

Under Windows all these files are kept instead in the directory *SkantiControl* , along with the other files and directories needed by the program.

The text version of this manual and the GNU license v3.0, always shown in the authoritative english language version, are also readable from the program itself. See the submenus *Manual* and *GNU License* in the *Help* menu. Last, the manual pages under Linux are kept in the */usr/share/man/man1* (english version) and */usr/share/man/it/man1* (italian version). These two are the only man pages available so far.

*NOTE. The manual base versions are the man pages (SkantiControl-en.1 and SkantiControl-it.1). The other formats are machine-generated from these. The most accurate versions are the PDF one and the text one; the html one leaves something to be desired in accuracy. So, if you cannot use the*

*man page, it is recommended to use the PDF version or the text version, which is readable also from within the program.*

## INTERNATIONALIZATION (AKA I18N)

In the 0.98 version the internationalization of the program has been implemented, using the Lazarus built-in i18n feature. The program internationalization is enabled by default using the language configured in the system. If a translation in that language does not exist, then the fallback macaronic english version is used.

If, for testing purposes or for any other reason it is wanted to start the program in a language other than that configured in the system, use the switch `--lang=<language code>` or `-l <language code>` where `<language code>` is the two-characters wanted language code, e.g. (for italian) execute the program as `SkantiControl --lang=it` or `SkantiControl -l it`.

If someone wants to contribute a translation in their own (or another) language (now only the italian translation and the macaronic english default are available) is welcome. In the author's web page a package for internationalization is available. Install the `poedit` (or similar) program, copy the file `SkantiControl.po` to `SkantiControl.lang.po`, where `lang` is the code of the new language (e.g. `SkantiControl.fr.po` for French or `SkantiControl.es.po` for Spanish), open it with `poedit` (or whatever) and then start to do the translation.

It is also possible to edit the `.po` file with any text editor, but the resulting file is not directly usable, since it must be first compiled into the `.mo` binary format, which `poedit` does automatically when saving the `.po` file and a generic text editor probably not (unless it's EMACS), so it is required to manually compile such file using the `msgfmt` program. For an example, see the `SkantiControl.it.po` file. If you modify the `.po` file with a text editor and cannot compile it yourself, if you send me that `.po` file, i will be glad to do the compile and send back to you the `.mo` compiled file so you can test your translation. To use the resulting `.mo` file, under Linux rename it to `SkantiControl.mo` and put the renamed file in the `/usr/share/locale/lang/LC_MESSAGES` folder, where `lang` is the abovementioned language two-letter code. Under Windows put it in the `Skanticontrol/locale` folder without renaming it.

Regarding the manual, if you also want to translate it, the starting point is one of the Linux manual pages, `SkantiControl-en.1` or `SkantiControl-it.1` which are enclosed in the i18n package mentioned above. The manual pages are text files you can edit with any text editor. To see the result, however, a man page viewer is required. Under Linux there are many, under Windows if you do not want to install an entire set of Unix utilities (e.g. UnxUtils), you can download and try the `mandoc` program.

In the i18n package the scripts used to generate the text, PDF and html versions of the manual pages are also enclosed, as shell scripts for the BASH shell, although these can be used only under Linux, except having an installation of the Unix system utilities (or at least an installation of those used, i.e. `bash`, `man`, `ps2pdf` and `sed`) under Windows. See the **CONFIGURATION, STATE AND DOCUMENTATION FILES** section for more informations on the name and location of the manual page files.

*NOTE. The captions of the main window buttons and labels are intentionally excluded from the translation, since the program aims to reproduce the look and feel of the Skanti CU8000 control unit.*

## BUGS, LIMITATIONS AND REMAINING THINGS TO DO

The main bug that the author was aware of was that on some systems the position and size of the main window were not restored correctly at the beginning, even though in the configuration file the parameters width, height, X and Y appeared to be correct. Thanks to Mark G4FPH for reporting this issue and for the tests done. This bug may have finally been tracked down (for the curious, it was a race condition caused by having mistakenly put some of the width and height restore code in the `FormCreate` procedure instead of in the `FormShow` procedure), but a thorough testing is needed. If it is seen again, please report.

There are some limitations and/or things not yet implemented or implemented but not tested due to various reasons: lack of understanding of the documentation, lack of time, lack of options in the author's own TRP8255, laziness of the program author.

- From the 0.98 build 400 version onward the SCAN command syntax has been completed and a try to circumvent the 7s timeout present has been made. A thorough check is however required.

- Check the command DUPLEX, feature which is not available on the author's own TRP8255.
- Study and understand better the SCAN BUFFER commands and implement them correctly.
- Thoroughly check correctness of the state machine against the CU8000 behaviour.
- Study and understand one of the free Windows installers (e.g. Inno Setup or NSIS) and provide Windows 32 and 64 bit installation files.

There are also some problems attributable at least to the incompleteness of the documentation, if not to a real v92 CU firmware bug: some commands, e.g. the "Test alarm" one, work but after some 7s without other commands issued cause a RTX reset not documented anywhere.

The road to version 1.00 is still very long...

## SOURCES, NEW VERSIONS AND CONTACT INFO.

The sources (one single archive for all supported architectures and operating systems, available in *.tar.gz* and *.ZIP* formats) and new binary installation versions of this program can be found at the author website:

<http://www.i1epj.ham-radio-op.net/software.html#SkantiControl>

To modify and/or recompile the sources the Lazarus IDE for FreePascal and the jujiboutils and INET packages are required. It is recommended to install them using the *Online Package Manager* provided in Lazarus.

The SkantiControl author can be reached by email at

[i1epj@aricasale.it](mailto:i1epj@aricasale.it) (<mailto:i1epj@aricasale.it>)

## ACKNOWLEDGEMENTS

Dave, G0WBX/G8KBV is the original author of the Skanti TRP8000 series remote commands list used in this program.

The idea of optionally substitute a user-configurable channel to the 500 and 2182 buttons came from Mark, G4FPH.

The main window objects scale code is inspired by an example by Ari Hirviniemi for Delphi.

## COPYRIGHT

SkantiControl - a program to control the Skanti TRP8000 series transceivers.

Copyright 2012-2025 (C) *Giuseppe Perotti, I1EPJ, i1epj@aricasale.it*

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

## THAT'S ALL, FOLKS!

At least until the next version...