

**NAME**

RA3701Control – a program to control the RACAL RA3701 receiver with look & feel similar to the receiver itself, with some enhancements.

**SYNOPSIS**

**RA3701Control** [ **--lang**=*<language code>* | **-l** *<language code>* ]

**DESCRIPTION**

A full manual has been not written yet. Briefly and with apologies for the bad English:

The "POWER" switch starts program execution and connection with the RA3701. Before starting the program, make sure to have selected the correct port and baud rate in the Options>Communication parameters... menu option. It is advisable to select the maximum possible speed, i.e. 9600 baud. Other parameters can be selected as desired. The chosen configuration will be saved automatically. For serial ports other than those listed, or USB, write the name of the port. The prefix */dev* under Linux and *\\* under Windows are optional. If not present, they are automatically added. Check also that your user has serial port access privileges; under Linux usually it must be a member of the "dialout" group.

In the current implementation, the receiver can be configured either with the handshake disabled (NONE protocol, designed to be used with multiple receivers connected via RS423A), in either CTR mode (CTR protocol) or RTS mode (RTS protocol). Both CRC and 1- or 2-byte addressing are supported, but LCC mode and 'diversity pair' configuration, which the author is not able to develop having only one receiver, are not. This is the configuration string returned by the author's receiver in RTS mode, parity even, address 01, and all currently supported modes enabled:

*RTS,9600,9600,EVEN,NOLCC,01,CRC,0,0*

Note that, at least on the author's RA3701 with firmware P87915/12, in RTS mode, activating the RTS line never activates the CTS line, contrary to what the manual seems to state. Therefore, the transmission code by default ignores the status of this line. If you want to check whether your receiver sets correctly that line or not, then check the "Wait CTS in RTS protocol" checkbox. After all, the manual says to use the CTR protocol when the receiver is computer-controlled, but without specifying why. Try and see for yourself.

When you start the program, the RX state (frequency, mode, filters, etc) will be read from RX itself. The state can also be saved to and restored from a file. Default file name is *<frequency>-<mode>.dat*, e.g. 10100.800-CW.dat.

Option "Enable S-meter" starts polling the received signal level (using the QRFL and QAFL commands) every 0.5s and display a bar-meter for the received signal strength. During the command duration, other keys are disabled, so selecting this option slows down somewhat operations.

Frequency can be changed by clicking on the RX display, with the mouse wheel, by writing the desired frequency in the spedit at lower right of the main window or by turning the tuning "knob", moving the mouse on it. To enable tuning (indicator TUNE lit), press the "TUNE+" button or click on the tuning knob. Clicking a second time disables tuning.

The tuning knob rate can be fixed (0.5, 1, 10 or 100 kHz per turn) or timed, as in the real receiver tuning knob: turning it slowly makes small frequency increments, down to 1 Hz, turning it faster makes bigger increments. Since it is difficult to maintain a well-defined tuning speed using the mouse, in the author's opinion it is wise to use one of the fixed tuning rates, so the initial default is fixed at 1kHz per turn. Try and see yourself.

Left-clicking on a figure in the RX frequency display decrements that figure by one, right-clicking increments that figure by one. This only happens if the final frequency is in the range of the RX. Clicking on the

decimal point the displayed frequency will be rewritten. Smallest increment/decrement available is 1 Hz.

If the mouse wheel is rotated with the pointer over one of the figures of the RX frequency, that figure is incremented or decremented. This now works also under Windows.

If the mouse wheel is rotated with the pointer over the frequency spindit, the frequency is incremented/decremented by 1 Hz.

Last, if the mouse wheel is rotated with the pointer anywhere else, RX frequency is incremented or decremented by the tuning step in effect.

The channels stored or recalled with the STORE, CHAN and CHAN SCAN keys can be local to the program or those stored in the receiver itself, depending on whether the menu "Use program channels" is checked or not.

It is also possible to synchronize the program channels and the receiver channels using the "Update channel list" menu item. Given the time needed to read all 100 channels, this function is not automatically invoked at program start like others (i. e. reading the available bandwidths) but only if explicitly requested.

In the STORE, CHAN e CHAN SCAN functions the channel number can be selected using the simulated tuning knob, as in the real receiver, but also by clicking with the left and right mouse buttons on the channel figures or by rotating the mouse wheel as above (increment/decrement that figure) or anywhere else (increment/decrement channel number by 1), or finally using the keyboard.

There are also some keyboard shortcuts:

- + : Tune up by the default tuning step in effect
- : Tune down by the default tuning step in effect
- / : Change tune rate (cycles between 1, 10, 100, 1000 Hz and LOCK)
- \* : Set receive frequency
- . : Enable/disable squelch
- 0..9 : Insert the chosen digit
- ENTER : same as ENTER button

allowing basic control of the receiver using only the numerical keypad.

These keyboard shortcuts are also accepted:

- U : Set USB mode
- L : Set LSB mode
- C : Set CW mode
- A : Set AM mode
- F : Set FM mode
- I : Set ISB mode (if ISB imodule fitted)

The above commands can be issued either upper or lower case. The frequency value for command \* can be entered either with the keyboard or with the on-screen numeric keys.

## OTHER COMMANDS & CONTROLS

The message display shows informations about the command being executed or status/info/error messages. Double-clicking on it clears display.

RX parameters (frequency, step, AGC, etc.) can be set also by selecting the desired value with the controls at the lower right of the main window.

The program tries to disable illegal keys for the actual state, e. g. pressing the "F" key, only the numeric keys are active and all other function keys are disabled. If this behaviour is not wanted, then check the "Option>Enable all controls" menu entry.

The message display shows informations about the command being executed or status/info/error messages. Double-clicking on it clears display.

#### **NOTE ON CONFIGURATION AND STATE FILES.**

All configuration files are kept under Linux in the .RA3701Control directory of your home directory, under Windows in the RA3701Control directory, which now can be put in any location and renamed as you like, as long as the executable file is contained in it and run from there. In case the program configuration has messed up, delete (or move) the "Config.xml" file and the program at next start will recreate it using the default configuration.

The channels stored with the STORE in the "Use program channel" mode are kept in the RXChan.dat file.

The AUX configuration is kept in the file AUXconf.dat. To change it, set the wanted configuration (mode, bandwidth, AGC and BFO if applicable, the other parameters are meaningless) and use the "File -> Save state" menu entry to save it with the name given above, possibly overwriting the existing file.

Files saved in the Channels subdirectory (using the "File>Save Channel" menu entry) will be read at program startup and shown in the Channels menu entry and can be recalled using only one mouse click. To allow to rename and delete the saved state files from within the program, a very minimal file manager is built in (menu entries "File>Manage states" and "File>Manage channels").

#### **TODO.**

- Complete the "MENU", "M1", "M2", "M3" and "M4" buttons implementation. At present only the "MENU" key is handled, not the submenus. Some commonly used functions have been duplicated in the "Other functions" menu item, but if in a more or less distant future that implementation will be completed, this will be no longer necessary.
- Implement remaining interface modes (LCC and diversity pair).
- Check the correctness of the state machine dealing with the memory channel functions against the RX behaviour.
- Write a full-blown manual.