NAME

picocom − minimal dumb-terminal emulation program

SYNOPSIS

picocom [*options* **]** *device*

DESCRIPTION

As its name suggests, **picocom** is a minimal dumb-terminal emulation program. It is, in principle, very much like **minicom(1)** , only it's "pico" instead of "mini"! It was designed to serve as a simple, manual, modem configuration, testing, and debugging tool. It has also served (quite well) as a low-tech "terminal-window" to allow operator intervention in PPP connection scripts (something like the ms-windows "open terminal window before / after dialing" feature). It could also prove useful in many other similar tasks.

When **picocom** starts it opens the terminal (serial device) given as its non-option argument. Unless the *--noinit* option is given, it configures the device to the settings specified by the option-arguments (or to some default settings), and sets it to "raw" mode. If *--noinit* is given, the initialization and configuration is skipped; the device is just opened. Following this, **picocom** sets the standard-input and standard-output to raw mode. Having done so, it goes in a loop where it listens for input from stdin, or from the serial port. Input from the serial port is copied to the standard output while input from the standard input is copied to the serial port. **picocom** also scans its input stream for a user-specified control character, called the "escape character" (being by default "C-a"). If the escape character is seen, then instead of sending it to the serial-device, the program enters "command mode" and waits for the next character (which is called the "function character"). Depending on the value of the function character, **picocom** performs one of the operations described in the "Commands" section below.

COMMANDS

Commands are given to **picocom** by first keying the "espace character" which by default is "C-a" (see "Options" below on how to change it), and then keying one for the function (command) characters shown here.

[escape character]
Send the escape character to the serial port and return to "transparent" mode. This means that if the escape character ("C-a", by default) is typed twice, the program sends the escape character to the serial port, and remains in transparent mode. This is a new behavior implemented in v1.4. Previously picocom used to ignore the escape-character when it was entered as a function character.

C-x
Exit the program: if the "--noreset" option was not given then the serial port is reset to its original settings before exiting; if it was given the serial port is not reset.

C-q
Quit the program *without* reseting the serial port, regardless of the "--noreset" option.

C-p
Pulse the DTR line. Lower it for 1 sec, and then raise it again.

C-t
Toggle the DTR line. If DTR is up, then lower it. If it is down, then raise it.

C-
Generate a break sequence on the serial line. A break sequence is usually generated by marking (driving to logical one) the serial Tx line for an amount of time coresponding to several character durations.

C-u
Baud up. Increase the baud-rate. The list of baud-rates stepped-through by this command is: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

C-d
Baud down. Decrease the baud-rate. The list of baud-rates stepped-through by this command is the same as for the "baud-up" command.

C-f
Cycle through flow-control settings (RTS/CTS, XON/XOFF, none).

C-y
Cycle through parity settings (even, odd, none).

C-b
Cycle through databits-number settings (5, 6, 7, 8).

C-v
Show program options (like baud rate, data bits, etc). Only the options that can be modified online (through commands) are shown, not those that can only be set at the command-line.

C-s
Send (upload) a file (see "Sending and Receiving Files" below)

C-r
Receive (download) a file (see "Sending and Receiving Files" below)

After performing one of the above operations the program leaves the command mode and enters transparent mode. Example: To increase the baud-rate by two steps, you have to type:

C-a, C-u, C-a, C-u

assuming of-course that "C-a" is the escape character.

SENDING AND RECEIVING FILES

picocom can send and receive files over the serial port using external programs that implement the respective protocols. In Linux typical programs for this purpose are:

— **rx(1)** - receive using the X-MODEM protocol

— **rb(1)** - receive using the Y-MODEM protocol

— **rz(1)** - receive using the Z-MODEM protocol

— **sx(1)** - send using the X-MODEM protocol

— **sb(1)** - send using the Y-MODEM protocol

— **sz(1)** - send using the Z-MODEM protocol

— **ascii-xfr(1)** - receive or transmit ASCII files

The name of, and the command-line options to, the program to be used for transmitting files are given by the "--send-cmd" option. Similarly the program to receive files, and its argumets, are given by the "--receive-cmd" option. For example, in order to start a **picocom** session that uses "sz" to transmit files, and "rz" to receive, you have to say something like this:

picocom --send-cmd "sz -vv" --receive-cmd "rz -vv"

During the picocom session, if you key the "send" or "receive" commands (e.g. by pressing C-a, C-s, or C-a, C-r) you will be prompted for a filename. At this prompt you can enter one or more file-names, and any additional arguments to the transmission or reception program. After that, picocom will start the the external program as specified by the "--send-cmd", or "--receive-cmd" option, and with any filenames and additional arguments you may have supplied. The standard input and output of the external program will be connected to the serial port. The standard error of the external program will be connected to the terminal which---while the program is running---will revert to canonical mode. Pressing 'C-c' while the external program is running will prematurely terminate it, and return control to **picocom**

OPTIONS

picocom accepts the following command-line options

--baud | -b
Defines the baud-rate to set the serial-port (terminal) to.

--flow | -f
Defines the flow-control mode to set the serial-port to. Must be one of:

— ´x’ for xon/xoff (software) mode

— ´h’ for hardware flow control (RTS/CTS)

— ´n’ for no flow control

(Default: ’n’)

--parity | -p
Defines the parity mode to set the serial-port to. Must be one of:

— ´o’ for odd parity mode.

— ´e’ for even parity mode.

— ´n’ for no parity, mode.

(Default: ’n’)

--databits | -d
Defines the number of data bits in every character. Must be one of: 5, 6, 7, 8

(Default: 8)

--esacpe | -e
Defines the character that will make picocom enter command-mode (see description above). If ’x’ is given, then C-x will make picocom enter command mode.

(Default: ’a’)

--noinit | -i
If given, **picocom** will not initialize, reset, or otherwise meddle with the serial port at start-up. It will just open it. This is useful, for example, for connecting **picocom** to already-connected modems, or already configured ports without terminating the connection, or altering the settings. If required serial port parameters can then be adjusted at run-time by commands.

--noreset | -r
If given, **picocom** will not *reset* the serial port when exiting. It will just close the filedes and do nothing more. This is useful, for example, for leaving modems connected when exiting **picocom picocom** using the "Quit" command (instead of "Exit"), which never resets the serial port. If "--noreset" is given then "Quit" and "Exit" behave essentially the same.

--nolock | -l
If given, **picocom** will *not* attempt to lock the serial port before opening it. Normally picocom attempts to get a UUCP-style lock-file (e.g. "/var/lock/LCK..ttyS0") before opening the port. Failing to do so, results in the program exiting after emitting an error-message. It is possible that your picocom binary is compiled without this option.

--send-cmd | -s
Specifies the external program (and any arguments to it) that will be used for transmitting files.

Default: "sz -vv"

--receive-cmd | -v
Specifies the external program (and any arguments to it) that will be used for receiving files.

(Default: "rz -vv")

--help | -h
Print a short help message describing the command-line options.

AUTHOR

picocom was written by Nick Patavalis (npat@efault.net)

AVAILABILITY

The latest version of "picocom" can be downloaded from: **http://efault.net/npat/hacks/picocom/**