

# **serial\_txx**

**A Linux command-line tool for  
serial request & answer exchanges.**

## **USER MANUAL**

Revision: 1.0

Date: 18/01/2016

## **serial\_txx, A Linux command-line tool for serial request & answer exchanges.**

Copyright (c) Ramon CASTELLA MAYMO  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All product names referenced herein are trademarks of their respective companies.

## DOCUMENT REVISION HISTORY

REVISION	DATE	AUTHOR	DESCRIPTION
1.0	18/01/2016	RCM	First revision of the document. Document revision applicable to serial_txx 1.0.

## **TABLE OF CONTENTS**

1 INTRODUCTION.....	3
2 SETTING UP serial_tsr.....	3
2.1 DISTRIBUTION FORMAT.....	3
2.2 INSTALL PROCEDURE.....	3
2.3 BUILD PROCEDURE.....	4
3 USING serial_tsr.....	4
3.1 COMMAND-LINE FORMAT.....	4
3.2 RETURN VALUES.....	5
3.3 ERROR MESSAGES.....	5

# 1 INTRODUCTION

**serial\_txx** is a Linux command-line tool that implements a simple request/answer exchange between a host and a remote device using serial communications.

The goal of **serial\_txx** is to provide the means of sending a request to a remote device and receiving an answer from it within a certain time-out. The format and contents of the request is defined by the user. The contents of the answer is also user dependent and shall end with the character delimiter indicated as argument in the execution of the tool.

**serial\_txx** can be used for general serial communications or as a test tool for embedded software verification..

**serial\_txx** is implemented in C++ and uses a slightly modified version of the *SerialPort* class provided by the open source library **libserial** created by:

- Manish P. Pagey <[pagey@san.rr.com](mailto:pagey@san.rr.com)>
- Crayzee Wulf <[crayzeewulf@users.sourceforge.net](mailto:crayzeewulf@users.sourceforge.net)>
- Jay Sachdev <[jsachdev@users.sourceforge.net](mailto:jsachdev@users.sourceforge.net)>
- Jan Wedekind <[wedesoft@users.sourceforge.net](mailto:wedesoft@users.sourceforge.net)>

The **libserial** library can be downloaded from <https://github.com/crayzeewulf/libserial/>.

## 2 SETTING UP serial\_txx

### 2.1 DISTRIBUTION FORMAT

**serial\_txx** is distributed in a compressed tarball containing the following elements:

- the source and header files,
- the makefile and the installation script,
- the executable file,
- the user manual (this document).

The tarball filename has the following format:

***serial\_txx\_v<version>\_<build\_date>.tar.gz***

where <version> is the release version of the product and <build\_date> is the release build date.

### 2.2 INSTALL PROCEDURE

Execute the following procedure to install **serial\_txx** in your computer:

1. Uncompress and extract the contents of the tarball <tarball>:  
**\$ tar xvf <tarball>**

2. Enter the tarball directory:  
`$ cd <tarball>`
3. Execute the install script indicating the destination directory `<dst_dir>` (e.g. `/usr/local`):  
`$ sudo -E ./install.sh <dst_dir>`
4. Remove the tarball file and directory, if desired:  
`$ rm -rf <tarball>*`

The **serial\_txx** executable file will be installed in the `<dst_dir>/serial_txx/bin` directory and the `PATH` variable will be automatically updated in the `~/.bashrc` file.

The **serial\_txx** user manual will be installed in the `<dst_dir>/serial_txx/doc` directory.

## 2.3 BUILD PROCEDURE

The `serial_txx` tool is delivered already built. If, for any reason, you need to modify it, execute the following procedure to rebuild it:

1. Enter the tarball directory:  
`$ cd <tarball>`
2. Remove the existing binary files:  
`$ make clean`
3. Rebuild the tool:  
`$ make all`
4. If necessary, execute step #3 of the install procedure in order to replace the current installed version.

## 3 USING serial\_txx

### 3.1 COMMAND-LINE FORMAT

**serial\_txx** shall be executed, from a Linux shell command-line terminal or from a shell script, respecting the following format and argument order:

```
$ serial_txx <device> <baudrate> <request> <delimiter> <timeout>
```

where

- `<device>` is the name of the serial port device (e.g. `/dev/ttyS0`).
- `<baudrate>` is an integer value indicating the baudrate to be used by the serial port. The following values are accepted: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 and 460800. The character frame is fixed to 8 bits, no parity and 1 stop bit.
- `<request>` is a string containing the user defined request.
- `<delimiter>` is a character defining the end of answer delimiter.

- `<timeout>` is an integer value indicating the maximum number of milliseconds to wait for a valid answer after the request has been sent.

If a valid answer is received, it will be printed on the standard output stream. Otherwise, a time-out error message will be displayed on the standard error stream.

The following example shows the command-line necessary to send the *Hello?* request to a remote device connected to serial port `/dev/ttyS0`, using a baudrate of *9600*, a *newline* character delimiter for the answer and waiting *200* milliseconds:

```
$ serial_ttrx /dev/ttyS0 9600 'Hello?' $'\n' 200
```

## 3.2 RETURN VALUES

`serial_ttrx` will return an exit status equal to 0 at the end of a correct execution. Otherwise, it will return an exit status equal to 1.

## 3.3 ERROR MESSAGES

This chapter presents the error messages that can be generated by `serial_ttrx`. Each message error is identified, explained and some hints to avoid it are indicated.

<b>MESSAGE</b>	<b>Wrong number of arguments!</b>
MEANING	The tool has been executed with more or less arguments as expected.
HOW TO AVOID	Modify the tool invocation respecting the number of arguments, their value and the order defined in chapter §3.1.

<b>MESSAGE</b>	<b>Unsupported baud rate!</b>
MEANING	The baudrate value indicated as argument is not valid.
HOW TO AVOID	Modify the tool invocation respecting the allowed baudrate values defined in chapter §3.1.

<b>MESSAGE</b>	<b>Unable to open the serial port!</b>
MEANING	The tool can not open the indicated device for reading nor writing.
HOW TO AVOID	Verify that the indicated device exists and that the user has the right permissions to use it.

<b>MESSAGE</b>	<b>Input timeout!</b>
MEANING	The indicated time-out to wait for the answer has been exceeded.
HOW TO AVOID	Verify the following points: <ul style="list-style-type: none"> <li>• There is a connection between host and remote device.</li> <li>• The indicated serial device is correct.</li> <li>• The remote device is running and accepting requests.</li> <li>• The indicated time-out is enough.</li> </ul>