

# Rust 'serialport-rs' *Transmit\_Receive\_Timing* Utility's Arguments, Logging and and Test Operation

## Table of Contents

<b>1. Serial Port Test and Characterization Application Utility 'receive_timing_info'</b>	<b>3</b>
<b>2. Test Logic Flow-Chart for 'receive_timing_info' Utility</b>	<b>7</b>
<b>3. V4.2.1 serialport-rs Behavior for 'read()' and 'set_timeout()' Methods on Linux and MS-Windows 10/11 Platforms</b>	<b>8</b>
a. Linux specific read() behavior =>	8
b. Windows specific read() behavior =>	8
<b>4. Utility 'receive_timing_info' test execution data log snippets for Insightful Scenarios that Induce Read Timeouts.</b>	<b>10</b>
<b>5. Patch from Github PR #79 for v4.2.1 'serialport-rs' Crate's 'set_timeout()' Method</b>	<b>17</b>

## 1. Serial Port Test and Characterization Application Utility 'receive\_timing\_info'

This document describes the **receive\_timing\_info** utility application, which is a timing characterization program for testing and supporting the serialport-rs crate. It builds with and uses the serialport-rs crate, and was created in response to the serialport-rs Github repo issue #106. This document (in its revised V1.3 edition) tests the PR #79 functional patch for the serialport-rs crate's Windows platform set\_timeout() method. I've tested this patch, using the 'receive\_timing\_info' test utility to verify Lar's patch functions as needed. The patch eliminates an issue under Windows in which a zero (0) read timeout setting currently causes the serial port's read() method to block indefinitely (an infinite timeout) if all requested data isn't received. Specifically, the read() operation returns immediately with whatever data is available on entry to read() – even if no data is available. In addition, it allows the read() with a non-zero timeout setting to also behave identically to the Linux platform's read operation. Specifically, the read() operation returns as soon as any data arrives, or times out if the timeout period expires with no data yet arrived. with whatever data is available on entry to read() – even if no data is available.

Note that without patch PR #79 the current Windows read() exhibits entirely different (let's call them opposite) behaviors when compared with the Linux platform – not a desirable situation. I do not have a Mac OS development system, and therefore haven't generated any timing data for it.

The *receive\_timing\_info* utility application source is packaged in a single file named `receive_timing_info.rs`, which can be built using cargo by copying it into the serialport-rs project's *examples* sub-folder. The application has been tested on both Windows 10 and Ubuntu 22.04 LTS (both x64 Intel hardware). In theory it should run on any platform supported by the standard rust compiler, as well as the **serialport-rs** crate itself. Currently this test application uses three utility crates beyond those that serialport-rs crate itself presently uses – which are identified in the following paragraph.

To build `receive_timing_info`, follow these steps:

- a. Using 'git', clone the crate project source for serialport-rs from the Github repo at <https://github.com/serialport/serialport-rs>.
- b. Copy the 'receive\_timing\_info.rs' source file into the cloned crate project's *examples* sub-folder.
- c. This new test application currently requires a few additional dependencies beyond those which serialport-rs crate presently requires. The additional dependency crates are **'log'**, **'fast-log'**, and **'spin-sleep'**. One may add these additional dependencies to serialport-rs crates's 'Cargo.toml' file by submitting the following three commands from the serialport-rs crate project's top-most folder while in a terminal command window =>
  - i. **'cargo add log'**
  - ii. **'cargo add fast\_log'**
  - iii. **'cargo add spin\_sleep'**
- d. Then build the `receive_timing_info` test application by entering the following commands, also while in a terminal command window with your working directory set to the top-most folder of the serialport-rs project.

'cargo clean'

'cargo build --example receive\_timing\_info' or 'cargo build --release --example receive\_timing\_info'

- e. The resulting executable file will be located in the project's 'target\debug\examples\' or 'target\release\examples\' sub-folder respectively.

Next is an example MS-Windows terminal window command line that launches the new test application =>

```
>receive_timing_info.exe --txport=COM5 --rxport=COM6 --baud=115200 --log=D:\filename.log --rxtmo=20 --posttxdelayms=0 -xfrstalledtmo=12000
--txlen=10 --rxlen=20 --repeat=10 --fuldbg=Y
```

The following table shows the available command line arguments that the receive\_timing\_info utility supports as command line arguments at launch. Note that the test currently defaults to using the crate's 'None' flow control setting and therefore doesn't test any platform or hardware specific serial port's flow control functionality. If desired this could be added.

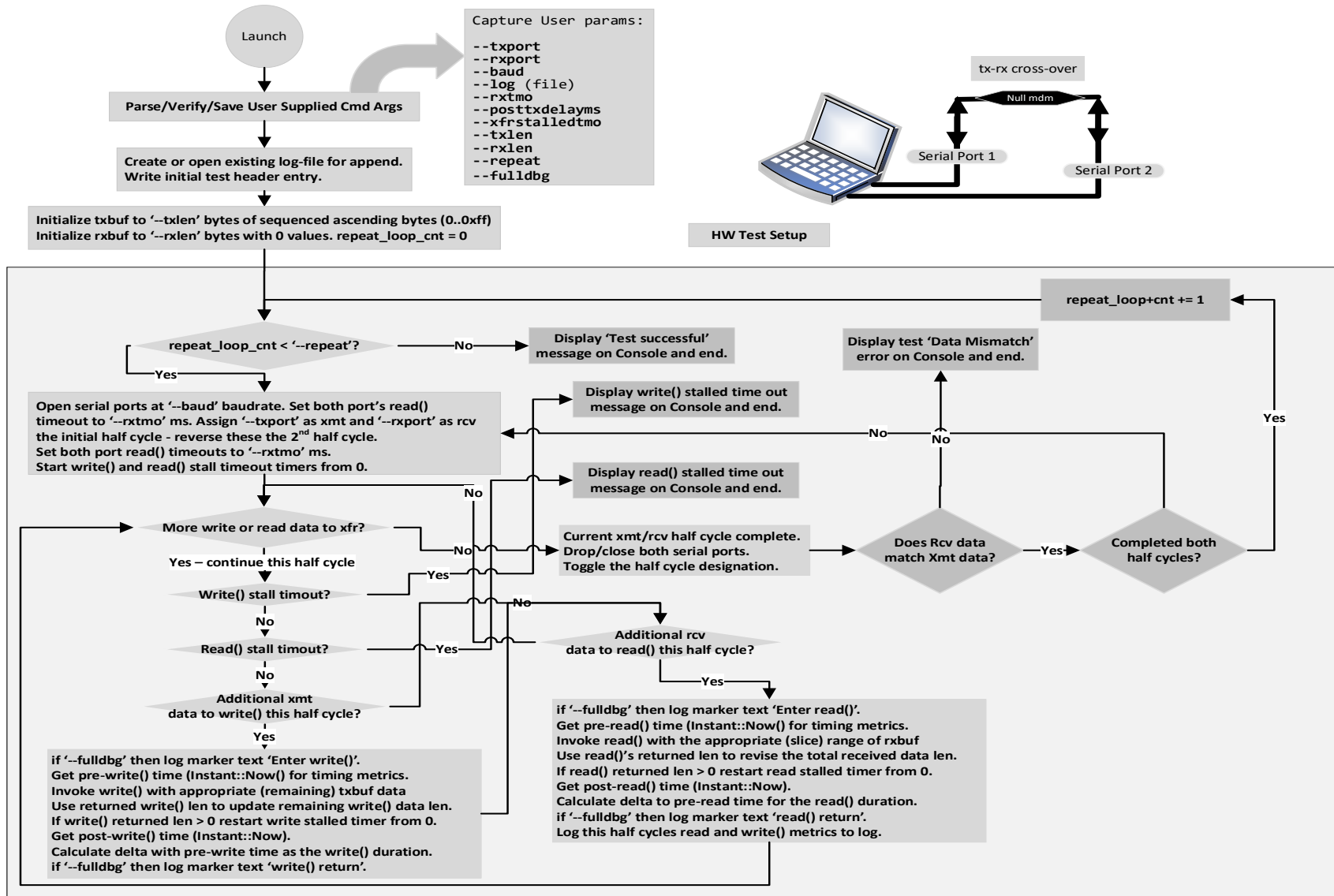
Syntax	Switch Purpose	Required or Optional
<code>--txport=port-name</code>	Platform specific port name (i.e. <i>COM4</i> )	Required
<code>--rxport=port-name</code>	Platform specific port name (i.e. <i>COM5</i> )	Required
<code>--log=file-path</code>	Platform specific log-file path	Required
<code>--baud=bbbb</code>	Baud rate (integer) – typical values are <i>9600, 115200, 200000</i> etc...	Required
<code>--rxtmo=tttt</code>	<b>read()</b> timeout in <b>ms</b> (integer)	Required
<code>--posttxdelayms=tttt</code>	The delay in <b>ms</b> to wait after <b>write()</b> before invoking the corresponding <b>read()</b>	Optional (default=0)
<code>--xfrstalledtmo=tttt</code>	The time-period before aborting when an in-progress test transfer sequence of consecutive <b>write()</b> or <b>read()</b> invocations time-out repeatedly, when additional transfer data is expected. Separate timers are internally maintained for <b>write()</b> and <b>read()</b> . The respective transfer stalled timer is reset when any <b>write()</b> or <b>read()</b> in the sequence returns success and a positive 'len' value. This parameter's used value will automatically be set to the greater of 10000 (the default when not specified), or ' <b>rxtmo</b> x 4' when ' <b>rxtmo</b> x 4' is greater than both the default (10000) and the specified value. Note that the value actually used is indicated in the launch-time initial displayed informational text line.	
<code>--txlen=nnnn</code>	Specifies the total number of bytes to <b>write()</b> each half cycle. This byte sequence is an auto-generated [u8] array buffer with an ascending order byte pattern sequencing from <b>0 .. 0xff</b> , repeating until exactly ' <b>txlen</b> ' bytes are generated. This buffer is then supplied to the <b>write()</b> invocation.	Required

<p><b>--rxlen=nnnn</b></p>	<p>Specifies the number of bytes requested when initially invoking <b>read()</b> following its associated <b>write()</b> operation in a given half cycle. While outstanding transmitted data has not yet been received, <b>read()</b> will be invoked repeated until the lesser of '<b>rxlen</b>' and '<b>txlen</b>' total bytes of accumulated data have been read, or until an '<b>--xfrstalledtmo</b>' timeout occurs. Subsequent <b>read()</b> invocations (following the initial <b>read()</b> in the same half-cycle use a request length that is reduced in magnitude downward from '<b>rxlen</b>' by the count of bytes already received.</p> <p>'<b>txlen</b>' and '<b>rxlen</b>' are commonly specified with the same positive value when primarily testing for transfer throughput, data transfer integrity and timing metrics. Note that these two parameters may be specified with different values, which has benefits for certain testing purposes. To test the <b>read()</b> timeout behavior and its platform specific timing characteristics, specifying values for which '<b>rxlen</b>' &gt; '<b>txlen</b>' will (or should!) result in <b>read()</b> timeout errors, before an eventual '<b>xfrstalledtmo</b>' timeout eventually halts the test. One caveat for the current Window's crate is that this test scenario blocks the test application indefinitely when the read timeout parameter is set to 0 ('<b>--rxtmo=0</b>'). Specifying with '<b>rxlen</b>' &lt; '<b>txlen</b>' parameter values should return '<b>rxlen</b>' bytes successfully, while leaving '<b>txlen</b>' – '<b>rxlen</b>' bytes unread in the read serial port's input buffer (which is not a problem).</p> <p>Finally note that specifying a '<b>txlen</b>' value larger than ~7500 on Windows will likely result in lost or corrupted transfer data. This is due to data buffer overruns in the platform OS layer, and is a by-product of the current test application's simple implementation/design. Similar data overruns may also happen on Linux, but at a larger '<b>txlen</b>' value: I haven't yet characterized this under Linux. To elaborate, currently the test's blocking <b>write()</b> invocation executes prior to the initial invocation of the blocking <b>read()</b>, with both occurring inline in the test application's main() thread. This current simple design does not support separately threaded and overlapped <b>write()</b> and <b>read()</b> invocations. This can be added in a subsequent version, or possibly in a separate new test program.</p>	<p>Required</p>
<p><b>--repeat=nnnn</b></p>	<p>Total number of read/write full cycles. Note that each full cycle consists of two (2) half-cycles, where each half-cycle writes '<b>txlen</b>' bytes to one of the two serial ports and (attempts to) read '<b>rxlen</b>' bytes from the other, while the second half-cycle reverses the direction of transmission with the two serial ports. Note that both ports are closed and reopened at each half-cycle transition. The closing and reopening of the ports at each cycle (or half cycle) may be eliminated or altered in a future release, or in a new test app.</p>	<p>Optional (default=1)</p>

<b>--fulldb=Y (or N)</b>	<p>'Y' enables additional debug log messages to be generated during the test. This flag should normally be 'N' (disabled), since enabling it results in undesirably lengthening certain timing metrics. 'N' is its default setting.</p> <p>Enabling the flag is useful when a specific test execution hangs indefinitely (which is abnormal), and with it enabled the run-time test logic inserts additional log-file 'entering' and 'returning' log marker text lines before and after each run-time invocation of <b>'write()'</b> and <b>'read()'</b>. Then one can examine the resulting log-file after the hang up occurs, and its clear which specific invocation is the cause of the hang-up. This is mainly useful in theorizing whether the hangup is due to faulty hardware, a bug in target specific crate code, electrical or cabling issues, or a previously undiscovered bug in the test software.</p> <p><i>Note that 'read()' or 'write()' blocking indefinitely could be detected with a separate monitoring thread, and an error message displayed for the test operator at execution time. This requires re-implementing the test with a multithreaded design (or in a separate new test application).</i></p>	Optional (default=N)
--------------------------	---	----------------------

<This space left intentionally blank>

## 2. Test Logic Flow-Chart for 'receive\_timing\_info' Utility



### 3. V4.2.1 serialport-rs Behavior for 'read()' and 'set\_timeout()' Methods on Linux and MS-Windows 10/11 Platforms

As previously mentioned I couldn't find online documentation which details the serialport-rs crate's read() and write() methods expected behavior for various calling scenarios of baud rate settings, the actual time of arrival of arriving serial line data in relationship to the point in time at which the read() method is invoked, and the active setting of the read() timeout value for the port. Therefore, I ran experiments after examining the crate's (platform specific) source code from the v4.2.1 serialport-rs Github repo master branch at <https://github.com/serialport/serialport-rs>. Next I describe the behaviors I identified for both Windows 10 and (Ubuntu) Linux 22.04 LTS. My assumption at this time, although I haven't confirmed it, is that Windows 11 should produce the same behavior (with possibly slightly different timing metrics) as Windows 10 – which I used during my testing.

#### a. Linux specific read() behavior =>

For *Linux*, a positive ( $> 0$ ) read timeout value is the period of time that read() blocks when no incoming data has arrived. If at least one (1) byte of data is received and available in the read's internal buffer, whether before initial entry to read() or any time thereafter prior to the timeout period expiring, read() returns immediately (at that instance) with the available data. If no received data is available at the conclusion of the timeout period, then read() returns at that time with a timeout error (and no data). The Linux read() returns the available data as soon as its available (and at initial entry when received data's already available), and its returned data len is only limited in size by the read() invocation's input buffer parameter length. With a read timeout setting of 0, the Linux read() returns with whatever data (if any) is available on initial entry, or immediately with no data, but always return immediately with or without data. When read() returns with no data (because none is available), it returns a timeout error indication. Please note also that the Linux read() non-zero timeout setting does not result in a read() internally extending its blocked time period waiting for *additional* data to arrive in order to more fully satisfy the requested data length, but rather only extends its wait when no data at all has been received.

#### b. Windows specific read() behavior =>

The Window's read() for v4.2.1 behaves differently in various respects to the Linux implementation.

On Windows when the timeout value is non-zero ( $> 0$ ), then the read() behaves only roughly similar to Linux, and varies in a couple of measurable aspects.

- i. The Windows read() waits for the entire timeout period while the entire requested read byte count (buffer size) is not yet available, and returns only at the conclusion of the read() timeout period with the lesser data amount that is available at conclusion of the timeout period - but no timeout occurs if at least one byte of data is returned. The Window's read() returns sooner than the timeout



period *only if* the read's entire requested data count (the supplied buffer's size) is received prior to the conclusion of the read timeout period.

- ii. The second difference from the Linux `read()` is more consequential to applications. When the Windows read timeout set-point value is 0, the Windows `read()` blocks until ALL requested bytes (per the supplied buffer's length) become available. Essentially a 0 timeout set-point in Windows operates as an infinite timeout period in order to satisfy the `read()`'s full requested length. To be clear, in this scenario a 0 timeout setpoint and having less than the read buffer's length of data arriving, the `read()` invocation blocks (forever) and never times-out.

As a practitioner's note regarding the Windows `read()` operation, with smaller read timeout settings in the range from 1 and 15 ms, a `read()` invocation can be randomly extended in time by up to ~15 to 20 ms prior to returning with already received data, or a timeout indication if no data has arrived within the timeout period. While this is certainly not ideal, by all accounts is something Microsoft is aware of and offers no solution to. Their documentation indicates this relates to the Windows OS API specification that Windows sleep timers have a published resolution of no less than ~16 milliseconds. This random delay clearly occurs with its native API `Readfile()` method, which is internally used by the crate's `read()` trait method, but is only a significant timing hindrance if one is attempting to write applications which must deterministically (hard real-time) respond to incoming serial data with repeatable timing accuracies in the sub (<) 100 ms range. These types of application timing requirements are generally not suitable for Windows as a target computer, for this and several other reasons. It is well known that Windows is *not* a real-time operating system!

#### 4. Utility 'receive\_timing\_info' test execution data log snippets for Insightful Scenarios that Induce Read Timeouts.

Next are three (3) generated Windows test logs with parameters which intentionally induce read timeouts. Note that each log file's second text line entry indicates the run-time supplied command line argument values which were provided. These are generated using the current release 4.2.1 SerialPort-rs crate.

Note that only the initial (first) half cycle of each of these test runs was undertaken, since the induced read timeout (induced by specifying an `-rxlen` and `-txlen` parameter pair where `rxlen > txlen`) resulted in a 'transfer stalled' timeout error during its first half cycle.

For this first Windows test run, its first half-cycle's `write()` method invocation transmits a total of one (1) byte, per the `-txlen` parameter, but the corresponding initial `read()` requests two (2) bytes, per the `-rxlen` parameter. Note that the read timeout is set to 50 ms, per the `-rxtmo` value. Also the initial `read()` is invoked precisely 100 ms after its matching half-cycle `write()` completes, per the `-posttxdelaysms` parameter. This sets up the situation where the initial `read()` should find already received data available upon entry – precisely one byte. Since the `-rxtmo` value wasn't 0, this run doesn't result in the problematic indefinite `read()` blocking scenario. Note the port baud rates are set to 200,000 baud. But this initial half-cycle will not successfully fulfill its expected total read count of two(2), and therefore will eventually trigger a transfer stalled timeout after one second, per its `-xfrstalledtmo` parameter of 1000 ms. Please note that each subsequent `read()` invocation while the half-cycle is active will reduce its `read()` buffer request size by the accumulatively received data (to this point) in the active half-cycle.

Note that in all the logs shown herein which use the existing crate version, the initial half-cycles will all error or block indefinitely, but this was arranged intentionally (via the supplied execution parameters) for the purposes of this explanatory document.

```
2023-07-16 19:11:59.2347468 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 19:11:59.2351164 INFO receive_timing_info - Test setup: Platform='windows', Baud=200000, rxtmo=50 ms, posttxdelaysms=100 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldbgs=false
2023-07-16 19:11:59.2378638 INFO receive_timing_info - Test Logfile Name: 'D:\Users\ricej\windows_receive_timing_info.txt'
2023-07-16 19:11:59.29698 INFO receive_timing_info -
2023-07-16 19:11:59.2969872 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 19:11:59.2969881 INFO receive_timing_info - Cycle 1 first phase -> Rx port = 'COM7', Tx port = 'COM6' .
2023-07-16 19:11:59.65374 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 548 us. Read() invoked 100412 us after write(), rxport.read(2) returned 1 bytes while blocked for 50255 us.
2023-07-16 19:11:59.7129727 INFO receive_timing_info - Read() invoked 150719 us after write(), rxport.read(1) returned 0 bytes while blocked for 59176 us. Rcv timeout.
2023-07-16 19:11:59.7629789 INFO receive_timing_info - Read() invoked 209949 us after write(), rxport.read(1) returned 0 bytes while blocked for 49959 us. Rcv timeout.
2023-07-16 19:11:59.8131091 INFO receive_timing_info - Read() invoked 259953 us after write(), rxport.read(1) returned 0 bytes while blocked for 50050 us. Rcv timeout.
2023-07-16 19:11:59.8729912 INFO receive_timing_info - Read() invoked 310088 us after write(), rxport.read(1) returned 0 bytes while blocked for 59832 us. Rcv timeout.
2023-07-16 19:11:59.923074 INFO receive_timing_info - Read() invoked 369984 us after write(), rxport.read(1) returned 0 bytes while blocked for 50024 us. Rcv timeout.
2023-07-16 19:11:59.9730056 INFO receive_timing_info - Read() invoked 420051 us after write(), rxport.read(1) returned 0 bytes while blocked for 49882 us. Rcv timeout.
2023-07-16 19:12:00.0301661 INFO receive_timing_info - Read() invoked 469980 us after write(), rxport.read(1) returned 0 bytes while blocked for 57127 us. Rcv timeout.
2023-07-16 19:12:00.0802684 INFO receive_timing_info - Read() invoked 527131 us after write(), rxport.read(1) returned 0 bytes while blocked for 50065 us. Rcv timeout.
2023-07-16 19:12:00.1401687 INFO receive_timing_info - Read() invoked 577248 us after write(), rxport.read(1) returned 0 bytes while blocked for 59835 us. Rcv timeout.
2023-07-16 19:12:00.190347 INFO receive_timing_info - Read() invoked 637145 us after write(), rxport.read(1) returned 0 bytes while blocked for 50135 us. Rcv timeout.
2023-07-16 19:12:00.2403654 INFO receive_timing_info - Read() invoked 687319 us after write(), rxport.read(1) returned 0 bytes while blocked for 49979 us. Rcv timeout.
2023-07-16 19:12:00.2992358 INFO receive_timing_info - Read() invoked 737337 us after write(), rxport.read(1) returned 0 bytes while blocked for 58831 us. Rcv timeout.
2023-07-16 19:12:00.3492649 INFO receive_timing_info - Read() invoked 796206 us after write(), rxport.read(1) returned 0 bytes while blocked for 49992 us. Rcv timeout.
2023-07-16 19:12:00.3992507 INFO receive_timing_info - Read() invoked 846241 us after write(), rxport.read(1) returned 0 bytes while blocked for 49941 us. Rcv timeout.
2023-07-16 19:12:00.458996 INFO receive_timing_info - Read() invoked 896227 us after write(), rxport.read(1) returned 0 bytes while blocked for 59714 us. Rcv timeout.
2023-07-16 19:12:00.5091288 INFO receive_timing_info - Read() invoked 955960 us after write(), rxport.read(1) returned 0 bytes while blocked for 50082 us. Rcv timeout.
```

```

2023-07-16 19:12:00.5690609 INFO receive_timing_info - Read() invoked 1006104 us after write(), rxport.read(1) returned 0 bytes while blocked for 59874 us. Rcv timeout.
2023-07-16 19:12:00.6191622 INFO receive_timing_info - Read() invoked 1066021 us after write(), rxport.read(1) returned 0 bytes while blocked for 50077 us. Rcv timeout.
2023-07-16 19:12:00.6696142 INFO receive_timing_info - Read() invoked 1116135 us after write(), rxport.read(1) returned 0 bytes while blocked for 50413 us. Rcv timeout.
2023-07-16 19:12:00.6696732 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect+verify the serial connections. Aborting.

```

Next is the second Windows log example.

For this (second) Windows test run, its first half-cycle's write() invocation again transmits a total of one (1) byte, per the `-txlen` parameter, and the corresponding read() again requests two (2) bytes, per its `-rxlen` parameter. Note that the read timeout is set to 1 ms, per its `-rxtmo` value. Also the initial read() is invoked immediately following its matching write() completes, per the `-posttxdelaysms` parameter of 0. This sets up the situation where our initial read() will likely not find already received data available upon entry. Since the `-rxtmo` value wasn't 0 for this run, it doesn't result in the problematic indefinite read() blocking scenario. Rather, since the total accumulated read will not never fulfill the total requested read() count of 2 (per `-rxlen`), the half cycle will eventually error with a transfer stalled timeout after 1000 ms (1 second). Finally, note the port baud rates are set to 200,000 baud.

```

2023-07-16 19:13:30.8416232 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 19:13:30.84176 INFO receive_timing_info - Test setup: Platform='windows', Baud=200000, rxtmo=1 ms, posttxdelaysms=0 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldbg=false
2023-07-16 19:13:30.8458403 INFO receive_timing_info - Test Logfile Name: 'D:\Users\rice\windows_receive_timing_info.txt'
2023-07-16 19:13:30.8690723 INFO receive_timing_info -
2023-07-16 19:13:30.8690797 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 19:13:30.8690811 INFO receive_timing_info - Cycle 1 first phase -> Rx port = 'COM7', Tx port = 'COM6'.
2023-07-16 19:13:31.0837107 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 427 us. Read() invoked 25 us after write(), rxport.read(2) returned 0 bytes while blocked for 9449 us. Rcv timeout.
2023-07-16 19:13:31.0935963 INFO receive_timing_info - Read() invoked 9520 us after write(), rxport.read(2) returned 1 bytes while blocked for 9827 us.
2023-07-16 19:13:31.1037244 INFO receive_timing_info - Read() invoked 19400 us after write(), rxport.read(1) returned 0 bytes while blocked for 10087 us. Rcv timeout.
2023-07-16 19:13:31.1137371 INFO receive_timing_info - Read() invoked 29555 us after write(), rxport.read(1) returned 0 bytes while blocked for 9945 us. Rcv timeout.
2023-07-16 19:13:31.1237386 INFO receive_timing_info - Read() invoked 39547 us after write(), rxport.read(1) returned 0 bytes while blocked for 9953 us. Rcv timeout.
2023-07-16 19:13:31.1337384 INFO receive_timing_info - Read() invoked 49549 us after write(), rxport.read(1) returned 0 bytes while blocked for 9951 us. Rcv timeout.
2023-07-16 19:13:31.1435955 INFO receive_timing_info - Read() invoked 59552 us after write(), rxport.read(1) returned 0 bytes while blocked for 9806 us. Rcv timeout.
2023-07-16 19:13:31.1537175 INFO receive_timing_info - Read() invoked 69407 us after write(), rxport.read(1) returned 0 bytes while blocked for 10074 us. Rcv timeout.
2023-07-16 19:13:31.1637297 INFO receive_timing_info - Read() invoked 79529 us after write(), rxport.read(1) returned 0 bytes while blocked for 9966 us. Rcv timeout.
2023-07-16 19:13:31.1737385 INFO receive_timing_info - Read() invoked 89542 us after write(), rxport.read(1) returned 0 bytes while blocked for 9963 us. Rcv timeout.
2023-07-16 19:13:31.1837062 INFO receive_timing_info - Read() invoked 99549 us after write(), rxport.read(1) returned 0 bytes while blocked for 9922 us. Rcv timeout.
2023-07-16 19:13:31.1936502 INFO receive_timing_info - Read() invoked 109518 us after write(), rxport.read(1) returned 0 bytes while blocked for 9898 us. Rcv timeout.
2023-07-16 19:13:31.2037372 INFO receive_timing_info - Read() invoked 119456 us after write(), rxport.read(1) returned 0 bytes while blocked for 10045 us. Rcv timeout.
2023-07-16 19:13:31.2137275 INFO receive_timing_info - Read() invoked 129550 us after write(), rxport.read(1) returned 0 bytes while blocked for 9944 us. Rcv timeout.
2023-07-16 19:13:31.223706 INFO receive_timing_info - Read() invoked 139539 us after write(), rxport.read(1) returned 0 bytes while blocked for 9932 us. Rcv timeout.
2023-07-16 19:13:31.2337312 INFO receive_timing_info - Read() invoked 149519 us after write(), rxport.read(1) returned 0 bytes while blocked for 9978 us. Rcv timeout.
2023-07-16 19:13:31.2436519 INFO receive_timing_info - Read() invoked 159544 us after write(), rxport.read(1) returned 0 bytes while blocked for 9870 us. Rcv timeout.
2023-07-16 19:13:31.2537485 INFO receive_timing_info - Read() invoked 169468 us after write(), rxport.read(1) returned 0 bytes while blocked for 10046 us. Rcv timeout.
2023-07-16 19:13:31.263729 INFO receive_timing_info - Read() invoked 179559 us after write(), rxport.read(1) returned 0 bytes while blocked for 9937 us. Rcv timeout.
2023-07-16 19:13:31.2760295 INFO receive_timing_info - Read() invoked 189541 us after write(), rxport.read(1) returned 0 bytes while blocked for 12259 us. Rcv timeout.
2023-07-16 19:13:31.2860841 INFO receive_timing_info - Read() invoked 201835 us after write(), rxport.read(1) returned 0 bytes while blocked for 10016 us. Rcv timeout.
2023-07-16 19:13:31.2961383 INFO receive_timing_info - Read() invoked 211896 us after write(), rxport.read(1) returned 0 bytes while blocked for 10006 us. Rcv timeout.
2023-07-16 19:13:31.3143309 INFO receive_timing_info - Read() invoked 221949 us after write(), rxport.read(1) returned 0 bytes while blocked for 18144 us. Rcv timeout.
2023-07-16 19:13:31.3243362 INFO receive_timing_info - Read() invoked 240141 us after write(), rxport.read(1) returned 0 bytes while blocked for 9960 us. Rcv timeout.

```

2023-07-16 19:13:31.3343353 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3442805 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3543142 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3643089 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3743395 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3842602 INFO receive\_timing\_info -  
2023-07-16 19:13:31.3959133 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4146801 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4245156 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4346106 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4446121 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4546299 INFO receive\_timing\_info -  
2023-07-16 19:13:31.464612 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4746493 INFO receive\_timing\_info -  
2023-07-16 19:13:31.4846173 INFO receive\_timing\_info -  
2023-07-16 19:13:31.49463 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5045199 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5146423 INFO receive\_timing\_info -  
2023-07-16 19:13:31.524661 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5346001 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5445119 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5546218 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5646134 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5746458 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5845786 INFO receive\_timing\_info -  
2023-07-16 19:13:31.5946506 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6046229 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6146285 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6246623 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6345675 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6446325 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6546681 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6645206 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6746493 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6845833 INFO receive\_timing\_info -  
2023-07-16 19:13:31.6946451 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7044569 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7146347 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7246317 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7347149 INFO receive\_timing\_info -  
2023-07-16 19:13:31.74461 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7545724 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7646463 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7746377 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7844717 INFO receive\_timing\_info -  
2023-07-16 19:13:31.7946455 INFO receive\_timing\_info -  
2023-07-16 19:13:31.8046542 INFO receive\_timing\_info -  
2023-07-16 19:13:31.8145803 INFO receive\_timing\_info -  
2023-07-16 19:13:31.8246227 INFO receive\_timing\_info -

Read() invoked 250148 us after write(), rxport.read(1) returned 0 bytes while blocked for 9948 us. Rcv timeout.  
Read() invoked 260143 us after write(), rxport.read(1) returned 0 bytes while blocked for 9903 us. Rcv timeout.  
Read() invoked 270112 us after write(), rxport.read(1) returned 0 bytes while blocked for 9969 us. Rcv timeout.  
Read() invoked 280127 us after write(), rxport.read(1) returned 0 bytes while blocked for 9949 us. Rcv timeout.  
Read() invoked 290121 us after write(), rxport.read(1) returned 0 bytes while blocked for 9984 us. Rcv timeout.  
Read() invoked 300151 us after write(), rxport.read(1) returned 0 bytes while blocked for 9871 us. Rcv timeout.  
Read() invoked 310072 us after write(), rxport.read(1) returned 0 bytes while blocked for 11609 us. Rcv timeout.  
Read() invoked 321722 us after write(), rxport.read(1) returned 0 bytes while blocked for 18720 us. Rcv timeout.  
Read() invoked 340484 us after write(), rxport.read(1) returned 0 bytes while blocked for 9796 us. Rcv timeout.  
Read() invoked 350328 us after write(), rxport.read(1) returned 0 bytes while blocked for 10048 us. Rcv timeout.  
Read() invoked 360415 us after write(), rxport.read(1) returned 0 bytes while blocked for 9962 us. Rcv timeout.  
Read() invoked 370427 us after write(), rxport.read(1) returned 0 bytes while blocked for 9969 us. Rcv timeout.  
Read() invoked 380441 us after write(), rxport.read(1) returned 0 bytes while blocked for 9937 us. Rcv timeout.  
Read() invoked 390424 us after write(), rxport.read(1) returned 0 bytes while blocked for 9989 us. Rcv timeout.  
Read() invoked 400462 us after write(), rxport.read(1) returned 0 bytes while blocked for 9920 us. Rcv timeout.  
Read() invoked 410434 us after write(), rxport.read(1) returned 0 bytes while blocked for 9962 us. Rcv timeout.  
Read() invoked 420442 us after write(), rxport.read(1) returned 0 bytes while blocked for 9842 us. Rcv timeout.  
Read() invoked 430334 us after write(), rxport.read(1) returned 0 bytes while blocked for 10075 us. Rcv timeout.  
Read() invoked 440453 us after write(), rxport.read(1) returned 0 bytes while blocked for 9974 us. Rcv timeout.  
Read() invoked 450494 us after write(), rxport.read(1) returned 0 bytes while blocked for 9869 us. Rcv timeout.  
Read() invoked 460414 us after write(), rxport.read(1) returned 0 bytes while blocked for 9864 us. Rcv timeout.  
Read() invoked 470323 us after write(), rxport.read(1) returned 0 bytes while blocked for 10064 us. Rcv timeout.  
Read() invoked 480432 us after write(), rxport.read(1) returned 0 bytes while blocked for 9945 us. Rcv timeout.  
Read() invoked 490426 us after write(), rxport.read(1) returned 0 bytes while blocked for 9987 us. Rcv timeout.  
Read() invoked 500456 us after write(), rxport.read(1) returned 0 bytes while blocked for 9869 us. Rcv timeout.  
Read() invoked 510385 us after write(), rxport.read(1) returned 0 bytes while blocked for 10027 us. Rcv timeout.  
Read() invoked 520465 us after write(), rxport.read(1) returned 0 bytes while blocked for 9922 us. Rcv timeout.  
Read() invoked 530438 us after write(), rxport.read(1) returned 0 bytes while blocked for 9959 us. Rcv timeout.  
Read() invoked 540438 us after write(), rxport.read(1) returned 0 bytes while blocked for 9964 us. Rcv timeout.  
Read() invoked 550476 us after write(), rxport.read(1) returned 0 bytes while blocked for 9853 us. Rcv timeout.  
Read() invoked 560379 us after write(), rxport.read(1) returned 0 bytes while blocked for 10019 us. Rcv timeout.  
Read() invoked 570443 us after write(), rxport.read(1) returned 0 bytes while blocked for 9987 us. Rcv timeout.  
Read() invoked 580486 us after write(), rxport.read(1) returned 0 bytes while blocked for 9799 us. Rcv timeout.  
Read() invoked 590332 us after write(), rxport.read(1) returned 0 bytes while blocked for 10083 us. Rcv timeout.  
Read() invoked 600462 us after write(), rxport.read(1) returned 0 bytes while blocked for 9889 us. Rcv timeout.  
Read() invoked 610393 us after write(), rxport.read(1) returned 0 bytes while blocked for 10019 us. Rcv timeout.  
Read() invoked 620456 us after write(), rxport.read(1) returned 0 bytes while blocked for 9755 us. Rcv timeout.  
Read() invoked 630267 us after write(), rxport.read(1) returned 0 bytes while blocked for 10135 us. Rcv timeout.  
Read() invoked 640446 us after write(), rxport.read(1) returned 0 bytes while blocked for 9952 us. Rcv timeout.  
Read() invoked 650443 us after write(), rxport.read(1) returned 0 bytes while blocked for 10037 us. Rcv timeout.  
Read() invoked 660524 us after write(), rxport.read(1) returned 0 bytes while blocked for 9847 us. Rcv timeout.  
Read() invoked 670422 us after write(), rxport.read(1) returned 0 bytes while blocked for 9911 us. Rcv timeout.  
Read() invoked 680383 us after write(), rxport.read(1) returned 0 bytes while blocked for 10028 us. Rcv timeout.  
Read() invoked 690457 us after write(), rxport.read(1) returned 0 bytes while blocked for 9948 us. Rcv timeout.  
Read() invoked 700448 us after write(), rxport.read(1) returned 0 bytes while blocked for 9784 us. Rcv timeout.  
Read() invoked 710282 us after write(), rxport.read(1) returned 0 bytes while blocked for 10129 us. Rcv timeout.  
Read() invoked 720458 us after write(), rxport.read(1) returned 0 bytes while blocked for 9957 us. Rcv timeout.  
Read() invoked 730471 us after write(), rxport.read(1) returned 0 bytes while blocked for 9870 us. Rcv timeout.  
Read() invoked 740391 us after write(), rxport.read(1) returned 0 bytes while blocked for 9998 us. Rcv timeout.

```

2023-07-16 19:13:31.8345487 INFO receive_timing_info -
2023-07-16 19:13:31.8445683 INFO receive_timing_info -
2023-07-16 19:13:31.8562955 INFO receive_timing_info -
2023-07-16 19:13:31.8664263 INFO receive_timing_info -
2023-07-16 19:13:31.8763461 INFO receive_timing_info -
2023-07-16 19:13:31.8864335 INFO receive_timing_info -
2023-07-16 19:13:31.8962505 INFO receive_timing_info -
2023-07-16 19:13:31.9150854 INFO receive_timing_info -
2023-07-16 19:13:31.9250768 INFO receive_timing_info -
2023-07-16 19:13:31.9349086 INFO receive_timing_info -
2023-07-16 19:13:31.9450669 INFO receive_timing_info -
2023-07-16 19:13:31.9550776 INFO receive_timing_info -
2023-07-16 19:13:31.9650426 INFO receive_timing_info -
2023-07-16 19:13:31.9750956 INFO receive_timing_info -
2023-07-16 19:13:31.9849724 INFO receive_timing_info -
2023-07-16 19:13:31.9949931 INFO receive_timing_info -
2023-07-16 19:13:32.0050653 INFO receive_timing_info -
2023-07-16 19:13:32.0149805 INFO receive_timing_info -
2023-07-16 19:13:32.0250305 INFO receive_timing_info -
2023-07-16 19:13:32.0350116 INFO receive_timing_info -
2023-07-16 19:13:32.0450704 INFO receive_timing_info -
2023-07-16 19:13:32.0549696 INFO receive_timing_info -
2023-07-16 19:13:32.0649347 INFO receive_timing_info -
2023-07-16 19:13:32.074916 INFO receive_timing_info -
2023-07-16 19:13:32.0849049 INFO receive_timing_info -
2023-07-16 19:13:32.0949135 INFO receive_timing_info -
2023-07-16 19:13:32.0949308 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect+verify the serial connections. Aborting.
Read() invoked 750434 us after write(), rxport.read(1) returned 0 bytes while blocked for 9876 us. Rcv timeout.
Read() invoked 760363 us after write(), rxport.read(1) returned 0 bytes while blocked for 9988 us. Rcv timeout.
Read() invoked 770367 us after write(), rxport.read(1) returned 0 bytes while blocked for 11700 us. Rcv timeout.
Read() invoked 782097 us after write(), rxport.read(1) returned 0 bytes while blocked for 10095 us. Rcv timeout.
Read() invoked 792237 us after write(), rxport.read(1) returned 0 bytes while blocked for 9871 us. Rcv timeout.
Read() invoked 802153 us after write(), rxport.read(1) returned 0 bytes while blocked for 10048 us. Rcv timeout.
Read() invoked 812242 us after write(), rxport.read(1) returned 0 bytes while blocked for 9779 us. Rcv timeout.
Read() invoked 822056 us after write(), rxport.read(1) returned 0 bytes while blocked for 18794 us. Rcv timeout.
Read() invoked 840901 us after write(), rxport.read(1) returned 0 bytes while blocked for 9943 us. Rcv timeout.
Read() invoked 850888 us after write(), rxport.read(1) returned 0 bytes while blocked for 9783 us. Rcv timeout.
Read() invoked 860720 us after write(), rxport.read(1) returned 0 bytes while blocked for 10108 us. Rcv timeout.
Read() invoked 870873 us after write(), rxport.read(1) returned 0 bytes while blocked for 9967 us. Rcv timeout.
Read() invoked 880889 us after write(), rxport.read(1) returned 0 bytes while blocked for 9920 us. Rcv timeout.
Read() invoked 890853 us after write(), rxport.read(1) returned 0 bytes while blocked for 9978 us. Rcv timeout.
Read() invoked 900902 us after write(), rxport.read(1) returned 0 bytes while blocked for 9835 us. Rcv timeout.
Read() invoked 910784 us after write(), rxport.read(1) returned 0 bytes while blocked for 9976 us. Rcv timeout.
Read() invoked 920798 us after write(), rxport.read(1) returned 0 bytes while blocked for 10033 us. Rcv timeout.
Read() invoked 930876 us after write(), rxport.read(1) returned 0 bytes while blocked for 9870 us. Rcv timeout.
Read() invoked 940788 us after write(), rxport.read(1) returned 0 bytes while blocked for 10008 us. Rcv timeout.
Read() invoked 950839 us after write(), rxport.read(1) returned 0 bytes while blocked for 9940 us. Rcv timeout.
Read() invoked 960822 us after write(), rxport.read(1) returned 0 bytes while blocked for 9988 us. Rcv timeout.
Read() invoked 970876 us after write(), rxport.read(1) returned 0 bytes while blocked for 9862 us. Rcv timeout.
Read() invoked 980779 us after write(), rxport.read(1) returned 0 bytes while blocked for 9928 us. Rcv timeout.
Read() invoked 990727 us after write(), rxport.read(1) returned 0 bytes while blocked for 9973 us. Rcv timeout.
Read() invoked 1000709 us after write(), rxport.read(1) returned 0 bytes while blocked for 9977 us. Rcv timeout.
Read() invoked 1010697 us after write(), rxport.read(1) returned 0 bytes while blocked for 10000 us. Rcv timeout.

```

Next is the third Windows example log, which demonstrates the indefinite blocking issue with read(), where the read timeout setpoint is 0 and all requested receive data doesn't arrive. It also uses the `-fulldb` flag enable setting.

Summarizing this run in more detail, the initial half-cycle's write() invocation again transmits a total of (only) 1 byte, but the corresponding read() requests two (2) bytes (per `-rxlen`). In this run the read timeout is 0 ms, per the specified `-rxtmo` value. The initial read() is invoked immediately after its matching write() completes, per its `-posttxdelays` parameter value of 0. This sets up the situation where our initial read() will likely find no received data available upon initial entry, but this is irrelevant as more significantly it never receives the full two (2) bytes requested by the initial read() call. Consequently, since the `-rxtmo` value is 0, this results in the problematic infinite blocking read() situation. Baud rates are set to 200,000 baud. Note the `-fulldb` flag has resulted in explicit marker text preceding and following each write() and read() invocation, which as shown from the log clearly shows the initial read() never returns – it blocks indefinitely.

```

2023-07-16 19:17:03.334684 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 19:17:03.3348747 INFO receive_timing_info - Test setup: Platform='windows', Baud=200000, rxtmo=0 ms, posttxdelays=0 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldb=true
2023-07-16 19:17:03.3377227 INFO receive_timing_info - Test Logfile Name: 'D:\Users\ricej\windows_receive_timing_info.txt'
2023-07-16 19:17:03.4205881 INFO receive_timing_info -

```

```

2023-07-16 19:17:03.4205962 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 19:17:03.4205977 INFO receive_timing_info - Cycle 1 first phase -> Rx port = 'COM7', Tx port = 'COM6' .
2023-07-16 19:17:03.62881 INFO receive_timing_info - Enter write(txbuf[0..1]).
2023-07-16 19:17:03.6291673 INFO receive_timing_info - Return from write().
2023-07-16 19:17:03.6292021 INFO receive_timing_info - Enter read(rxbuf[0..2]).

```

The above very short log-file content resulted from the test application once it blocked/suspended for nearly two (2) minutes prior to the test operator (myself), eventually terminating the obviously blocked test application from the keyboard.

Next are the equivalent Linux logs for the same three (3) test invocation argument setups, generated on Ubuntu Linux executing in a VMWare guest on the same laptop computer. Again, these are generated from running the currently release 4.2.1 crate. The first Linux log =>

```

2023-07-16 16:43:14.272409174 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 16:43:14.272426885 INFO receive_timing_info - Test setup: Platform='linux', Baud=200000, rxtmo=50 ms, posttxdelays=100 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldb=false
2023-07-16 16:43:14.272434667 INFO receive_timing_info - Test Logfile Name: '/home/ricelj/ubuntu_receive_timing_info.txt'
2023-07-16 16:43:14.895900214 INFO receive_timing_info -
2023-07-16 16:43:14.895949482 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 16:43:14.89595447 INFO receive_timing_info - Cycle 1 first phase -> Rx port = '/dev/ttyUSB1', Tx port = '/dev/ttyUSB0' .
2023-07-16 16:43:15.198883731 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 2425 us. Read() invoked 100269 us after write(), rxport.read(2) returned 1 bytes while blocked for 17 us.
2023-07-16 16:43:15.249000315 INFO receive_timing_info - Read() invoked 100344 us after write(), rxport.read(1) returned 0 bytes while blocked for 50064 us. Rcv timeout.
2023-07-16 16:43:15.299208892 INFO receive_timing_info - Read() invoked 150429 us after write(), rxport.read(1) returned 0 bytes while blocked for 50186 us. Rcv timeout.
2023-07-16 16:43:15.350425949 INFO receive_timing_info - Read() invoked 200662 us after write(), rxport.read(1) returned 0 bytes while blocked for 51169 us. Rcv timeout.
2023-07-16 16:43:15.400619098 INFO receive_timing_info - Read() invoked 251881 us after write(), rxport.read(1) returned 0 bytes while blocked for 50143 us. Rcv timeout.
2023-07-16 16:43:15.451641702 INFO receive_timing_info - Read() invoked 302069 us after write(), rxport.read(1) returned 0 bytes while blocked for 50977 us. Rcv timeout.
2023-07-16 16:43:15.502401107 INFO receive_timing_info - Read() invoked 353102 us after write(), rxport.read(1) returned 0 bytes while blocked for 50699 us. Rcv timeout.
2023-07-16 16:43:15.553152011 INFO receive_timing_info - Read() invoked 403870 us after write(), rxport.read(1) returned 0 bytes while blocked for 50684 us. Rcv timeout.
2023-07-16 16:43:15.603731655 INFO receive_timing_info - Read() invoked 454621 us after write(), rxport.read(1) returned 0 bytes while blocked for 50516 us. Rcv timeout.
2023-07-16 16:43:15.654415799 INFO receive_timing_info - Read() invoked 505188 us after write(), rxport.read(1) returned 0 bytes while blocked for 50633 us. Rcv timeout.
2023-07-16 16:43:15.705176217 INFO receive_timing_info - Read() invoked 555874 us after write(), rxport.read(1) returned 0 bytes while blocked for 50707 us. Rcv timeout.
2023-07-16 16:43:15.756190409 INFO receive_timing_info - Read() invoked 606634 us after write(), rxport.read(1) returned 0 bytes while blocked for 50962 us. Rcv timeout.
2023-07-16 16:43:15.806492464 INFO receive_timing_info - Read() invoked 657645 us after write(), rxport.read(1) returned 0 bytes while blocked for 50252 us. Rcv timeout.
2023-07-16 16:43:15.857302959 INFO receive_timing_info - Read() invoked 707978 us after write(), rxport.read(1) returned 0 bytes while blocked for 50730 us. Rcv timeout.
2023-07-16 16:43:15.908021805 INFO receive_timing_info - Read() invoked 758762 us after write(), rxport.read(1) returned 0 bytes while blocked for 50666 us. Rcv timeout.
2023-07-16 16:43:15.958705464 INFO receive_timing_info - Read() invoked 809474 us after write(), rxport.read(1) returned 0 bytes while blocked for 50637 us. Rcv timeout.
2023-07-16 16:43:16.009378698 INFO receive_timing_info - Read() invoked 860161 us after write(), rxport.read(1) returned 0 bytes while blocked for 50621 us. Rcv timeout.
2023-07-16 16:43:16.059899261 INFO receive_timing_info - Read() invoked 910848 us after write(), rxport.read(1) returned 0 bytes while blocked for 50456 us. Rcv timeout.
2023-07-16 16:43:16.110283419 INFO receive_timing_info - Read() invoked 961356 us after write(), rxport.read(1) returned 0 bytes while blocked for 50332 us. Rcv timeout.
2023-07-16 16:43:16.160563977 INFO receive_timing_info - Read() invoked 1011715 us after write(), rxport.read(1) returned 0 bytes while blocked for 50254 us. Rcv timeout.
2023-07-16 16:43:16.211238623 INFO receive_timing_info - Read() invoked 1062019 us after write(), rxport.read(1) returned 0 bytes while blocked for 50625 us. Rcv timeout.
2023-07-16 16:43:16.211282556 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect+verify the serial connections. Aborting.

```

Here's the second Linux log example (4.2.1 version crate). This full log is considerably lengthier than the equivalent Windows log, due to Linux more responsive (i.e. faster) read() timeout returns. For brevity I've excluded many uninteresting intermediate read() log entries soon after the initial byte is read, and resume with the final reads leading up to the terminating transfer stalled timeout error - around 1 second after the initial read() following the earlier matching write().

```

2023-07-16 16:36:48.733217257 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 16:36:48.733246792 INFO receive_timing_info - Test setup: Platform='linux', Baud=200000, rxtmo=1 ms, posttxdelays=50 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldb=false

```

```

2023-07-16 16:36:48.733254245 INFO receive_timing_info - Test Logfile Name: '/home/ricelj/ubuntu_receive_timing_info.txt'
2023-07-16 16:36:49.716092489 INFO receive_timing_info -
2023-07-16 16:36:49.716142723 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 16:36:49.716146655 INFO receive_timing_info - Cycle 1 first phase -> Rx port = '/dev/ttyUSB1', Tx port = '/dev/ttyUSB0'.
2023-07-16 16:36:49.969758513 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 2773 us. Read() invoked 50299 us after write(), rxport.read(2) returned 1 bytes while blocked for 23 us.
2023-07-16 16:36:49.971226136 INFO receive_timing_info - Read() invoked 50380 us after write(), rxport.read(1) returned 0 bytes while blocked for 1420 us. Rcv timeout.
2023-07-16 16:36:49.972910507 INFO receive_timing_info - Read() invoked 51818 us after write(), rxport.read(1) returned 0 bytes while blocked for 1662 us. Rcv timeout.
2023-07-16 16:36:49.974913961 INFO receive_timing_info - Read() invoked 53501 us after write(), rxport.read(1) returned 0 bytes while blocked for 1989 us. Rcv timeout.
2023-07-16 16:36:49.976142741 INFO receive_timing_info - Read() invoked 55495 us after write(), rxport.read(1) returned 0 bytes while blocked for 1223 us. Rcv timeout.
2023-07-16 16:36:49.977314849 INFO receive_timing_info - Read() invoked 56736 us after write(), rxport.read(1) returned 0 bytes while blocked for 1155 us. Rcv timeout.
2023-07-16 16:36:49.97928537 INFO receive_timing_info - Read() invoked 57904 us after write(), rxport.read(1) returned 0 bytes while blocked for 1957 us. Rcv timeout.
2023-07-16 16:36:49.98031803 INFO receive_timing_info - Read() invoked 59874 us after write(), rxport.read(1) returned 0 bytes while blocked for 1021 us. Rcv timeout.
2023-07-16 16:36:49.982303652 INFO receive_timing_info - Read() invoked 60906 us after write(), rxport.read(1) returned 0 bytes while blocked for 1974 us. Rcv timeout.
2023-07-16 16:36:49.984328611 INFO receive_timing_info - Read() invoked 62892 us after write(), rxport.read(1) returned 0 bytes while blocked for 2013 us. Rcv timeout.
2023-07-16 16:36:49.986306959 INFO receive_timing_info - Read() invoked 64909 us after write(), rxport.read(1) returned 0 bytes while blocked for 1975 us. Rcv timeout.
2023-07-16 16:36:49.988299764 INFO receive_timing_info - Read() invoked 66895 us after write(), rxport.read(1) returned 0 bytes while blocked for 1981 us. Rcv timeout.
2023-07-16 16:36:49.990303784 INFO receive_timing_info - Read() invoked 68887 us after write(), rxport.read(1) returned 0 bytes while blocked for 1993 us. Rcv timeout.
2023-07-16 16:36:49.992307476 INFO receive_timing_info - Read() invoked 70892 us after write(), rxport.read(1) returned 0 bytes while blocked for 1992 us. Rcv timeout.
2023-07-16 16:36:49.99349152 INFO receive_timing_info - Read() invoked 72895 us after write(), rxport.read(1) returned 0 bytes while blocked for 1172 us. Rcv timeout.
2023-07-16 16:36:49.995454175 INFO receive_timing_info - Read() invoked 74083 us after write(), rxport.read(1) returned 0 bytes while blocked for 1947 us. Rcv timeout.
2023-07-16 16:36:49.996552099 INFO receive_timing_info - Read() invoked 76043 us after write(), rxport.read(1) returned 0 bytes while blocked for 1086 us. Rcv timeout.

```

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

```

2023-07-16 16:36:50.94697694 INFO receive_timing_info - Read() invoked 1026199 us after write(), rxport.read(1) returned 0 bytes while blocked for 1348 us. Rcv timeout.
2023-07-16 16:36:50.948476102 INFO receive_timing_info - Read() invoked 1027599 us after write(), rxport.read(1) returned 0 bytes while blocked for 1448 us. Rcv timeout.
2023-07-16 16:36:50.949956468 INFO receive_timing_info - Read() invoked 1029098 us after write(), rxport.read(1) returned 0 bytes while blocked for 1430 us. Rcv timeout.
2023-07-16 16:36:50.951655698 INFO receive_timing_info - Read() invoked 1030575 us after write(), rxport.read(1) returned 0 bytes while blocked for 1652 us. Rcv timeout.
2023-07-16 16:36:50.953217239 INFO receive_timing_info - Read() invoked 1032274 us after write(), rxport.read(1) returned 0 bytes while blocked for 1515 us. Rcv timeout.
2023-07-16 16:36:50.954952595 INFO receive_timing_info - Read() invoked 1033836 us after write(), rxport.read(1) returned 0 bytes while blocked for 1689 us. Rcv timeout.
2023-07-16 16:36:50.956611695 INFO receive_timing_info - Read() invoked 1035569 us after write(), rxport.read(1) returned 0 bytes while blocked for 1615 us. Rcv timeout.
2023-07-16 16:36:50.958619843 INFO receive_timing_info - Read() invoked 1037231 us after write(), rxport.read(1) returned 0 bytes while blocked for 1961 us. Rcv timeout.
2023-07-16 16:36:50.960044618 INFO receive_timing_info - Read() invoked 1039235 us after write(), rxport.read(1) returned 0 bytes while blocked for 1382 us. Rcv timeout.
2023-07-16 16:36:50.962187024 INFO receive_timing_info - Read() invoked 1041244 us after write(), rxport.read(1) returned 0 bytes while blocked for 1515 us. Rcv timeout.
2023-07-16 16:36:50.963525653 INFO receive_timing_info - Read() invoked 1042804 us after write(), rxport.read(1) returned 0 bytes while blocked for 1294 us. Rcv timeout.
2023-07-16 16:36:50.965264495 INFO receive_timing_info - Read() invoked 1044147 us after write(), rxport.read(1) returned 0 bytes while blocked for 1690 us. Rcv timeout.
2023-07-16 16:36:50.966984191 INFO receive_timing_info - Read() invoked 1045881 us after write(), rxport.read(1) returned 0 bytes while blocked for 1676 us. Rcv timeout.
2023-07-16 16:36:50.968903071 INFO receive_timing_info - Read() invoked 1047599 us after write(), rxport.read(1) returned 0 bytes while blocked for 1876 us. Rcv timeout.
2023-07-16 16:36:50.970892558 INFO receive_timing_info - Read() invoked 1049520 us after write(), rxport.read(1) returned 0 bytes while blocked for 1945 us. Rcv timeout.
2023-07-16 16:36:50.970934005 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect-verify the serial connections. Aborting.

```

Here's the third Linux example log (version 4.2.1 version crate). Again for brevity this shows the leading section of the full log, followed by the section where the first byte is successfully read(), and third the final section where the transfer timeout error occurs. It that it obviously doesn't incur the indefinite blocking which Windows experiences with a read timeout setpoint of 0 and all requested receive data not arriving. Note also in this Linux test run (as compared to its corresponding Windows run) I didn't enable the `--fulldb` flag, since it merely increases the log size and skews certain timing data and makes it more difficult to interpret the results. To be clear, I've indeed executed this same test argument configuration on Linux with the `--fulldb` flag enabled, and it runs as above, but with many more log entries and skewed timing numbers. You may do so as well, if you're curious about this scenario.

```

2023-07-16 17:33:51.182958831 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-07-16 17:33:51.182994242 INFO receive_timing_info - Test setup: Platform='linux', Baud=200000, rxtmo=0 ms, posttxdelaysms=0 ms, xfrstalledtmo=1000 ms, txlen=1, rxlen=2, repeat=1, fulldbgs=false
2023-07-16 17:33:51.183009314 INFO receive_timing_info - Test Logfile Name: '/home/ricelj/ubuntu_receive_timing_info.txt'
2023-07-16 17:33:51.343628307 INFO receive_timing_info -
2023-07-16 17:33:51.343678069 INFO receive_timing_info - ** Start of cycle 1. **
2023-07-16 17:33:51.343682827 INFO receive_timing_info - Cycle 1 first phase -> Rx port = '/dev/ttyUSB1', Tx port = '/dev/ttyUSB0' .
2023-07-16 17:33:51.54727904 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 3010 us. Read() invoked 9 us after write(), rxport.read(2) returned 0 bytes while blocked for 12 us. Rcv timeout.
2023-07-16 17:33:51.547330689 INFO receive_timing_info - Read() invoked 68 us after write(), rxport.read(2) returned 0 bytes while blocked for 6 us. Rcv timeout.
2023-07-16 17:33:51.547337022 INFO receive_timing_info - Read() invoked 80 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547342551 INFO receive_timing_info - Read() invoked 86 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547348054 INFO receive_timing_info - Read() invoked 92 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547354325 INFO receive_timing_info - Read() invoked 97 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547360614 INFO receive_timing_info - Read() invoked 103 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547366008 INFO receive_timing_info - Read() invoked 110 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547372565 INFO receive_timing_info - Read() invoked 115 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.547378652 INFO receive_timing_info - Read() invoked 122 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.

```

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

```

2023-07-16 17:33:51.561370021 INFO receive_timing_info - Read() invoked 14112 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561376082 INFO receive_timing_info - Read() invoked 14119 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561382249 INFO receive_timing_info - Read() invoked 14125 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561388814 INFO receive_timing_info - Read() invoked 14131 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561394964 INFO receive_timing_info - Read() invoked 14138 us after write(), rxport.read(2) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561664472 INFO receive_timing_info - Read() invoked 14144 us after write(), rxport.read(2) returned 1 bytes while blocked for 264 us.
2023-07-16 17:33:51.561686314 INFO receive_timing_info - Read() invoked 14430 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-07-16 17:33:51.561687918 INFO receive_timing_info - Read() invoked 14435 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:51.561689288 INFO receive_timing_info - Read() invoked 14437 us after write(), rxport.read(1) returned 0 bytes while blocked for 0 us. Rcv timeout.
2023-07-16 17:33:51.561690846 INFO receive_timing_info - Read() invoked 14438 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:51.561692203 INFO receive_timing_info - Read() invoked 14440 us after write(), rxport.read(1) returned 0 bytes while blocked for 0 us. Rcv timeout.
2023-07-16 17:33:51.561694084 INFO receive_timing_info - Read() invoked 14441 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:51.561695649 INFO receive_timing_info - Read() invoked 14443 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.

```

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

... .. <intentionally omitted uninteresting read() log entries ... >

```

2023-07-16 17:33:52.561620759 INFO receive_timing_info - Read() invoked 1014367 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561622825 INFO receive_timing_info - Read() invoked 1014370 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561624658 INFO receive_timing_info - Read() invoked 1014372 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.56162675 INFO receive_timing_info - Read() invoked 1014373 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561629146 INFO receive_timing_info - Read() invoked 1014376 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561631284 INFO receive_timing_info - Read() invoked 1014378 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.

```



```

2023-07-16 17:33:52.561633332 INFO receive_timing_info - Read() invoked 1014380 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561635306 INFO receive_timing_info - Read() invoked 1014382 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561776316 INFO receive_timing_info - Read() invoked 1014384 us after write(), rxport.read(1) returned 0 bytes while blocked for 1 us. Rcv timeout.
2023-07-16 17:33:52.561778111 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect+verify the serial connections. Aborting.

```

## 5. Patch from Github PR #79 for v4.2.1 'serialport-rs' Crate's 'set\_timeout()' Method

This section provides comparative timing and behavior for the patch submitted in the Github pr #79, by Lars Christensen (<https://github.com/larsch>).

The logs in the prior sections were generated using the existing serialport-rs crate release v4.2.1, while the logs below was generated running the patched version of the crate per pr #79 (on my local development system).

This patch eliminates the 0 timeout settings infinite blocking wait issue. When a 0 ms timeout setting is active, it now returns immediately with or without data. If a non-zero timeout setting is active, it returns immediately once any data arrives. If no bytes are received and the timeout expires, a timeout error is returned.

Here's the Windows log produced by executing 'receive\_timing\_info' built with SerialPort-rs 4.2.1 with PR #79 patch - also using a 50 ms read timeout setting, where one (1) data byte is transmitted, and two (2) bytes are requested in the read. Note that the one (1) transmitted byte is in-fact received, but the first phase/loop cycle fails (intentionally) since not all requested data bytes are received. This is expected - nothing is in-fact wrong here.

```

2023-08-15 16:36:14.0356686 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-08-15 16:36:14.0360273 INFO receive_timing_info - Test setup: Platform='windows', Baud=200000, rxtno=50 ms, posttxdelays=100 ms, xfrstalledtmo=10000 ms, txlen=1, rxlen=2, repeat=1, fulldbgs=false
2023-08-15 16:36:14.0364714 INFO receive_timing_info - Test Logfile Name: 'D:/Users/ricelj/windows_receive_timing_info.txt'
2023-08-15 16:36:14.05045 INFO receive_timing_info -
2023-08-15 16:36:14.0504951 INFO receive_timing_info - ** Start of cycle 1. **
2023-08-15 16:36:14.050499 INFO receive_timing_info - Cycle 1 first phase -> Rx port = 'COM6', Tx port = 'COM7' .
2023-08-15 16:36:14.3596371 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 255 us. Read() invoked 100034 us after write(), rxport.read(2) returned 1 bytes while blocked for 8 us.
2023-08-15 16:36:14.4092284 INFO receive_timing_info - Read() invoked 100077 us after write(), rxport.read(1) returned 0 bytes while blocked for 49563 us. Rcv timeout.
2023-08-15 16:36:14.4692137 INFO receive_timing_info - Read() invoked 149663 us after write(), rxport.read(1) returned 0 bytes while blocked for 59953 us. Rcv timeout.
2023-08-15 16:36:14.5292016 INFO receive_timing_info - Read() invoked 209648 us after write(), rxport.read(1) returned 0 bytes while blocked for 59969 us. Rcv timeout.
2023-08-15 16:36:14.5892274 INFO receive_timing_info - Read() invoked 269636 us after write(), rxport.read(1) returned 0 bytes while blocked for 59998 us. Rcv timeout.
2023-08-15 16:36:14.6393568 INFO receive_timing_info - Read() invoked 329666 us after write(), rxport.read(1) returned 0 bytes while blocked for 50033 us. Rcv timeout.
2023-08-15 16:36:14.6993909 INFO receive_timing_info - Read() invoked 379817 us after write(), rxport.read(1) returned 0 bytes while blocked for 59945 us. Rcv timeout.
2023-08-15 16:36:14.759207 INFO receive_timing_info - Read() invoked 439855 us after write(), rxport.read(1) returned 0 bytes while blocked for 59742 us. Rcv timeout.
2023-08-15 16:36:14.8091999 INFO receive_timing_info - Read() invoked 499648 us after write(), rxport.read(1) returned 0 bytes while blocked for 49965 us. Rcv timeout.
2023-08-15 16:36:14.8595679 INFO receive_timing_info - Read() invoked 549634 us after write(), rxport.read(1) returned 0 bytes while blocked for 50349 us. Rcv timeout.
2023-08-15 16:36:14.9180136 INFO receive_timing_info - Read() invoked 600003 us after write(), rxport.read(1) returned 0 bytes while blocked for 58410 us. Rcv timeout.
2023-08-15 16:36:14.9779597 INFO receive_timing_info - Read() invoked 658465 us after write(), rxport.read(1) returned 0 bytes while blocked for 59910 us. Rcv timeout.
2023-08-15 16:36:15.0279326 INFO receive_timing_info - Read() invoked 718412 us after write(), rxport.read(1) returned 0 bytes while blocked for 49933 us. Rcv timeout.
2023-08-15 16:36:15.0820727 INFO receive_timing_info - Read() invoked 768383 us after write(), rxport.read(1) returned 0 bytes while blocked for 54095 us. Rcv timeout.
2023-08-15 16:36:15.1381059 INFO receive_timing_info - Read() invoked 822512 us after write(), rxport.read(1) returned 0 bytes while blocked for 56003 us. Rcv timeout.
2023-08-15 16:36:15.1981143 INFO receive_timing_info - Read() invoked 878541 us after write(), rxport.read(1) returned 0 bytes while blocked for 59973 us. Rcv timeout.
2023-08-15 16:36:15.2580858 INFO receive_timing_info - Read() invoked 938550 us after write(), rxport.read(1) returned 0 bytes while blocked for 59951 us. Rcv timeout.

```

2023-08-15 16:36:15.3080842 INFO receive\_timing\_info -  
2023-08-15 16:36:15.3587084 INFO receive\_timing\_info -  
2023-08-15 16:36:15.4113073 INFO receive\_timing\_info -  
2023-08-15 16:36:15.4683654 INFO receive\_timing\_info -  
2023-08-15 16:36:15.5283697 INFO receive\_timing\_info -  
2023-08-15 16:36:15.5883054 INFO receive\_timing\_info -  
2023-08-15 16:36:15.6439619 INFO receive\_timing\_info -  
2023-08-15 16:36:15.6984415 INFO receive\_timing\_info -  
2023-08-15 16:36:15.7582406 INFO receive\_timing\_info -  
2023-08-15 16:36:15.8082953 INFO receive\_timing\_info -  
2023-08-15 16:36:15.8682931 INFO receive\_timing\_info -  
2023-08-15 16:36:15.9238752 INFO receive\_timing\_info -  
2023-08-15 16:36:15.974037 INFO receive\_timing\_info -  
2023-08-15 16:36:16.0240515 INFO receive\_timing\_info -  
2023-08-15 16:36:16.0838891 INFO receive\_timing\_info -  
2023-08-15 16:36:16.1441451 INFO receive\_timing\_info -  
2023-08-15 16:36:16.2041518 INFO receive\_timing\_info -  
2023-08-15 16:36:16.2543594 INFO receive\_timing\_info -  
2023-08-15 16:36:16.3143435 INFO receive\_timing\_info -  
2023-08-15 16:36:16.3643662 INFO receive\_timing\_info -  
2023-08-15 16:36:16.4176731 INFO receive\_timing\_info -  
2023-08-15 16:36:16.4743 INFO receive\_timing\_info -  
2023-08-15 16:36:16.5342389 INFO receive\_timing\_info -  
2023-08-15 16:36:16.5844982 INFO receive\_timing\_info -  
2023-08-15 16:36:16.6444717 INFO receive\_timing\_info -  
2023-08-15 16:36:16.6945434 INFO receive\_timing\_info -  
2023-08-15 16:36:16.7445897 INFO receive\_timing\_info -  
2023-08-15 16:36:16.8044828 INFO receive\_timing\_info -  
2023-08-15 16:36:16.8644642 INFO receive\_timing\_info -  
2023-08-15 16:36:16.9193908 INFO receive\_timing\_info -  
2023-08-15 16:36:16.9793308 INFO receive\_timing\_info -  
2023-08-15 16:36:17.0294026 INFO receive\_timing\_info -  
2023-08-15 16:36:17.0893686 INFO receive\_timing\_info -  
2023-08-15 16:36:17.1394156 INFO receive\_timing\_info -  
2023-08-15 16:36:17.1893412 INFO receive\_timing\_info -  
2023-08-15 16:36:17.2396252 INFO receive\_timing\_info -  
2023-08-15 16:36:17.2897066 INFO receive\_timing\_info -  
2023-08-15 16:36:17.3497251 INFO receive\_timing\_info -  
2023-08-15 16:36:17.4127929 INFO receive\_timing\_info -  
2023-08-15 16:36:17.4698327 INFO receive\_timing\_info -  
2023-08-15 16:36:17.5198514 INFO receive\_timing\_info -  
2023-08-15 16:36:17.579974 INFO receive\_timing\_info -  
2023-08-15 16:36:17.6300878 INFO receive\_timing\_info -  
2023-08-15 16:36:17.6801226 INFO receive\_timing\_info -  
2023-08-15 16:36:17.739959 INFO receive\_timing\_info -  
2023-08-15 16:36:17.7999363 INFO receive\_timing\_info -  
2023-08-15 16:36:17.8619489 INFO receive\_timing\_info -  
2023-08-15 16:36:17.9183723 INFO receive\_timing\_info -  
2023-08-15 16:36:17.9698465 INFO receive\_timing\_info -  
2023-08-15 16:36:18.0199331 INFO receive\_timing\_info -  
2023-08-15 16:36:18.079917 INFO receive\_timing\_info -  
2023-08-15 16:36:18.1299194 INFO receive\_timing\_info -  
2023-08-15 16:36:18.1899203 INFO receive\_timing\_info -  
2023-08-15 16:36:18.2499261 INFO receive\_timing\_info -  
2023-08-15 16:36:18.2999099 INFO receive\_timing\_info -  
2023-08-15 16:36:18.3499379 INFO receive\_timing\_info -

Read() invoked 998520 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49949 us. Rcv timeout.  
Read() invoked 1048535 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50547 us. Rcv timeout.  
Read() invoked 1099170 us after write(), rxdport.read(1) returned 0 bytes while blocked for 52553 us. Rcv timeout.  
Read() invoked 1151754 us after write(), rxdport.read(1) returned 0 bytes while blocked for 57025 us. Rcv timeout.  
Read() invoked 1208800 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59971 us. Rcv timeout.  
Read() invoked 1268820 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59845 us. Rcv timeout.  
Read() invoked 1328746 us after write(), rxdport.read(1) returned 0 bytes while blocked for 55620 us. Rcv timeout.  
Read() invoked 1384418 us after write(), rxdport.read(1) returned 0 bytes while blocked for 54408 us. Rcv timeout.  
Read() invoked 1438907 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59729 us. Rcv timeout.  
Read() invoked 1498695 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50015 us. Rcv timeout.  
Read() invoked 1548737 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59971 us. Rcv timeout.  
Read() invoked 1608727 us after write(), rxdport.read(1) returned 0 bytes while blocked for 55562 us. Rcv timeout.  
Read() invoked 1664310 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50101 us. Rcv timeout.  
Read() invoked 1714498 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49927 us. Rcv timeout.  
Read() invoked 1764512 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59787 us. Rcv timeout.  
Read() invoked 1824324 us after write(), rxdport.read(1) returned 0 bytes while blocked for 60206 us. Rcv timeout.  
Read() invoked 1884594 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59974 us. Rcv timeout.  
Read() invoked 1944586 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50148 us. Rcv timeout.  
Read() invoked 1994838 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59899 us. Rcv timeout.  
Read() invoked 2054785 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49976 us. Rcv timeout.  
Read() invoked 2104832 us after write(), rxdport.read(1) returned 0 bytes while blocked for 53243 us. Rcv timeout.  
Read() invoked 2158114 us after write(), rxdport.read(1) returned 0 bytes while blocked for 56563 us. Rcv timeout.  
Read() invoked 2214743 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59875 us. Rcv timeout.  
Read() invoked 2274679 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50222 us. Rcv timeout.  
Read() invoked 2324934 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59922 us. Rcv timeout.  
Read() invoked 2384913 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50046 us. Rcv timeout.  
Read() invoked 2434978 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49978 us. Rcv timeout.  
Read() invoked 2485033 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59844 us. Rcv timeout.  
Read() invoked 2544916 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59964 us. Rcv timeout.  
Read() invoked 2604913 us after write(), rxdport.read(1) returned 0 bytes while blocked for 54893 us. Rcv timeout.  
Read() invoked 2659825 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59915 us. Rcv timeout.  
Read() invoked 2719769 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50049 us. Rcv timeout.  
Read() invoked 2769836 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59928 us. Rcv timeout.  
Read() invoked 2829802 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50027 us. Rcv timeout.  
Read() invoked 2879850 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49879 us. Rcv timeout.  
Read() invoked 2929774 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50266 us. Rcv timeout.  
Read() invoked 2980059 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50017 us. Rcv timeout.  
Read() invoked 3030168 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59922 us. Rcv timeout.  
Read() invoked 3090188 us after write(), rxdport.read(1) returned 0 bytes while blocked for 63007 us. Rcv timeout.  
Read() invoked 3153234 us after write(), rxdport.read(1) returned 0 bytes while blocked for 57013 us. Rcv timeout.  
Read() invoked 3210267 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49989 us. Rcv timeout.  
Read() invoked 3260299 us after write(), rxdport.read(1) returned 0 bytes while blocked for 60076 us. Rcv timeout.  
Read() invoked 3320416 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50042 us. Rcv timeout.  
Read() invoked 3370544 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49953 us. Rcv timeout.  
Read() invoked 3420584 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59778 us. Rcv timeout.  
Read() invoked 3480401 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59938 us. Rcv timeout.  
Read() invoked 3540390 us after write(), rxdport.read(1) returned 0 bytes while blocked for 61956 us. Rcv timeout.  
Read() invoked 3602393 us after write(), rxdport.read(1) returned 0 bytes while blocked for 56391 us. Rcv timeout.  
Read() invoked 3658820 us after write(), rxdport.read(1) returned 0 bytes while blocked for 51410 us. Rcv timeout.  
Read() invoked 3710280 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50067 us. Rcv timeout.  
Read() invoked 3760377 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59936 us. Rcv timeout.  
Read() invoked 3820351 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49982 us. Rcv timeout.  
Read() invoked 3870353 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59979 us. Rcv timeout.  
Read() invoked 3930354 us after write(), rxdport.read(1) returned 0 bytes while blocked for 59984 us. Rcv timeout.  
Read() invoked 3990360 us after write(), rxdport.read(1) returned 0 bytes while blocked for 49965 us. Rcv timeout.  
Read() invoked 4040344 us after write(), rxdport.read(1) returned 0 bytes while blocked for 50009 us. Rcv timeout.

2023-08-15 16:36:18.4098398 INFO receive\_timing\_info -  
2023-08-15 16:36:18.4703631 INFO receive\_timing\_info -  
2023-08-15 16:36:18.5203458 INFO receive\_timing\_info -  
2023-08-15 16:36:18.5802795 INFO receive\_timing\_info -  
2023-08-15 16:36:18.6438742 INFO receive\_timing\_info -  
2023-08-15 16:36:18.7012008 INFO receive\_timing\_info -  
2023-08-15 16:36:18.7515089 INFO receive\_timing\_info -  
2023-08-15 16:36:18.8135231 INFO receive\_timing\_info -  
2023-08-15 16:36:18.8757722 INFO receive\_timing\_info -  
2023-08-15 16:36:18.9377031 INFO receive\_timing\_info -  
2023-08-15 16:36:18.9908439 INFO receive\_timing\_info -  
2023-08-15 16:36:19.0446821 INFO receive\_timing\_info -  
2023-08-15 16:36:19.106923 INFO receive\_timing\_info -  
2023-08-15 16:36:19.1601463 INFO receive\_timing\_info -  
2023-08-15 16:36:19.2200495 INFO receive\_timing\_info -  
2023-08-15 16:36:19.2800657 INFO receive\_timing\_info -  
2023-08-15 16:36:19.3300914 INFO receive\_timing\_info -  
2023-08-15 16:36:19.3812208 INFO receive\_timing\_info -  
2023-08-15 16:36:19.4420058 INFO receive\_timing\_info -  
2023-08-15 16:36:19.5047837 INFO receive\_timing\_info -  
2023-08-15 16:36:19.5563313 INFO receive\_timing\_info -  
2023-08-15 16:36:19.6063488 INFO receive\_timing\_info -  
2023-08-15 16:36:19.6661929 INFO receive\_timing\_info -  
2023-08-15 16:36:19.7261125 INFO receive\_timing\_info -  
2023-08-15 16:36:19.776124 INFO receive\_timing\_info -  
2023-08-15 16:36:19.8261097 INFO receive\_timing\_info -  
2023-08-15 16:36:19.8760251 INFO receive\_timing\_info -  
2023-08-15 16:36:19.9260818 INFO receive\_timing\_info -  
2023-08-15 16:36:19.9826482 INFO receive\_timing\_info -  
2023-08-15 16:36:20.042566 INFO receive\_timing\_info -  
2023-08-15 16:36:20.1007342 INFO receive\_timing\_info -  
2023-08-15 16:36:20.1694121 INFO receive\_timing\_info -  
2023-08-15 16:36:20.222572 INFO receive\_timing\_info -  
2023-08-15 16:36:20.2842683 INFO receive\_timing\_info -  
2023-08-15 16:36:20.3411795 INFO receive\_timing\_info -  
2023-08-15 16:36:20.4040529 INFO receive\_timing\_info -  
2023-08-15 16:36:20.4676781 INFO receive\_timing\_info -  
2023-08-15 16:36:20.530869 INFO receive\_timing\_info -  
2023-08-15 16:36:20.5921979 INFO receive\_timing\_info -  
2023-08-15 16:36:20.6509975 INFO receive\_timing\_info -  
2023-08-15 16:36:20.7057508 INFO receive\_timing\_info -  
2023-08-15 16:36:20.7684572 INFO receive\_timing\_info -  
2023-08-15 16:36:20.8205606 INFO receive\_timing\_info -  
2023-08-15 16:36:20.8708291 INFO receive\_timing\_info -  
2023-08-15 16:36:20.9309035 INFO receive\_timing\_info -  
2023-08-15 16:36:20.9886754 INFO receive\_timing\_info -  
2023-08-15 16:36:21.0486902 INFO receive\_timing\_info -  
2023-08-15 16:36:21.0999442 INFO receive\_timing\_info -  
2023-08-15 16:36:21.1590857 INFO receive\_timing\_info -  
2023-08-15 16:36:21.2142165 INFO receive\_timing\_info -  
2023-08-15 16:36:21.2651691 INFO receive\_timing\_info -  
2023-08-15 16:36:21.3151572 INFO receive\_timing\_info -  
2023-08-15 16:36:21.3755395 INFO receive\_timing\_info -  
2023-08-15 16:36:21.4326519 INFO receive\_timing\_info -  
2023-08-15 16:36:21.497852 INFO receive\_timing\_info -  
2023-08-15 16:36:21.5578308 INFO receive\_timing\_info -

Read() invoked 4090372 us after write(), rxport.read(1) returned 0 bytes while blocked for 59829 us. Rcv timeout.  
Read() invoked 4150282 us after write(), rxport.read(1) returned 0 bytes while blocked for 60497 us. Rcv timeout.  
Read() invoked 4210797 us after write(), rxport.read(1) returned 0 bytes while blocked for 49962 us. Rcv timeout.  
Read() invoked 4260780 us after write(), rxport.read(1) returned 0 bytes while blocked for 59901 us. Rcv timeout.  
Read() invoked 4320731 us after write(), rxport.read(1) returned 0 bytes while blocked for 63538 us. Rcv timeout.  
Read() invoked 4384313 us after write(), rxport.read(1) returned 0 bytes while blocked for 57265 us. Rcv timeout.  
Read() invoked 4441661 us after write(), rxport.read(1) returned 0 bytes while blocked for 50261 us. Rcv timeout.  
Read() invoked 4491943 us after write(), rxport.read(1) returned 0 bytes while blocked for 61991 us. Rcv timeout.  
Read() invoked 4553960 us after write(), rxport.read(1) returned 0 bytes while blocked for 62216 us. Rcv timeout.  
Read() invoked 4616208 us after write(), rxport.read(1) returned 0 bytes while blocked for 61877 us. Rcv timeout.  
Read() invoked 4678142 us after write(), rxport.read(1) returned 0 bytes while blocked for 53114 us. Rcv timeout.  
Read() invoked 4731280 us after write(), rxport.read(1) returned 0 bytes while blocked for 53814 us. Rcv timeout.  
Read() invoked 4785126 us after write(), rxport.read(1) returned 0 bytes while blocked for 62212 us. Rcv timeout.  
Read() invoked 4847355 us after write(), rxport.read(1) returned 0 bytes while blocked for 53150 us. Rcv timeout.  
Read() invoked 4900603 us after write(), rxport.read(1) returned 0 bytes while blocked for 59814 us. Rcv timeout.  
Read() invoked 4960491 us after write(), rxport.read(1) returned 0 bytes while blocked for 59989 us. Rcv timeout.  
Read() invoked 5020500 us after write(), rxport.read(1) returned 0 bytes while blocked for 50004 us. Rcv timeout.  
Read() invoked 5070525 us after write(), rxport.read(1) returned 0 bytes while blocked for 51111 us. Rcv timeout.  
Read() invoked 5121655 us after write(), rxport.read(1) returned 0 bytes while blocked for 60763 us. Rcv timeout.  
Read() invoked 5182440 us after write(), rxport.read(1) returned 0 bytes while blocked for 62742 us. Rcv timeout.  
Read() invoked 5245220 us after write(), rxport.read(1) returned 0 bytes while blocked for 51490 us. Rcv timeout.  
Read() invoked 5296790 us after write(), rxport.read(1) returned 0 bytes while blocked for 49934 us. Rcv timeout.  
Read() invoked 5346811 us after write(), rxport.read(1) returned 0 bytes while blocked for 59784 us. Rcv timeout.  
Read() invoked 5406636 us after write(), rxport.read(1) returned 0 bytes while blocked for 59893 us. Rcv timeout.  
Read() invoked 5466546 us after write(), rxport.read(1) returned 0 bytes while blocked for 49992 us. Rcv timeout.  
Read() invoked 5516558 us after write(), rxport.read(1) returned 0 bytes while blocked for 49954 us. Rcv timeout.  
Read() invoked 5566553 us after write(), rxport.read(1) returned 0 bytes while blocked for 49887 us. Rcv timeout.  
Read() invoked 5616458 us after write(), rxport.read(1) returned 0 bytes while blocked for 50038 us. Rcv timeout.  
Read() invoked 5666516 us after write(), rxport.read(1) returned 0 bytes while blocked for 56508 us. Rcv timeout.  
Read() invoked 5723090 us after write(), rxport.read(1) returned 0 bytes while blocked for 59891 us. Rcv timeout.  
Read() invoked 5783001 us after write(), rxport.read(1) returned 0 bytes while blocked for 58125 us. Rcv timeout.  
Read() invoked 5841169 us after write(), rxport.read(1) returned 0 bytes while blocked for 68638 us. Rcv timeout.  
Read() invoked 5909855 us after write(), rxport.read(1) returned 0 bytes while blocked for 53130 us. Rcv timeout.  
Read() invoked 5963016 us after write(), rxport.read(1) returned 0 bytes while blocked for 61646 us. Rcv timeout.  
Read() invoked 6024703 us after write(), rxport.read(1) returned 0 bytes while blocked for 56870 us. Rcv timeout.  
Read() invoked 6081614 us after write(), rxport.read(1) returned 0 bytes while blocked for 62822 us. Rcv timeout.  
Read() invoked 6144485 us after write(), rxport.read(1) returned 0 bytes while blocked for 63580 us. Rcv timeout.  
Read() invoked 6208117 us after write(), rxport.read(1) returned 0 bytes while blocked for 63160 us. Rcv timeout.  
Read() invoked 6271306 us after write(), rxport.read(1) returned 0 bytes while blocked for 61291 us. Rcv timeout.  
Read() invoked 6332641 us after write(), rxport.read(1) returned 0 bytes while blocked for 58754 us. Rcv timeout.  
Read() invoked 6391433 us after write(), rxport.read(1) returned 0 bytes while blocked for 54723 us. Rcv timeout.  
Read() invoked 6446184 us after write(), rxport.read(1) returned 0 bytes while blocked for 62659 us. Rcv timeout.  
Read() invoked 6508892 us after write(), rxport.read(1) returned 0 bytes while blocked for 52083 us. Rcv timeout.  
Read() invoked 6561006 us after write(), rxport.read(1) returned 0 bytes while blocked for 50218 us. Rcv timeout.  
Read() invoked 6611262 us after write(), rxport.read(1) returned 0 bytes while blocked for 60013 us. Rcv timeout.  
Read() invoked 6671347 us after write(), rxport.read(1) returned 0 bytes while blocked for 57673 us. Rcv timeout.  
Read() invoked 6729133 us after write(), rxport.read(1) returned 0 bytes while blocked for 59919 us. Rcv timeout.  
Read() invoked 6789169 us after write(), rxport.read(1) returned 0 bytes while blocked for 51136 us. Rcv timeout.  
Read() invoked 6840403 us after write(), rxport.read(1) returned 0 bytes while blocked for 59058 us. Rcv timeout.  
Read() invoked 6899547 us after write(), rxport.read(1) returned 0 bytes while blocked for 55043 us. Rcv timeout.  
Read() invoked 6954656 us after write(), rxport.read(1) returned 0 bytes while blocked for 50926 us. Rcv timeout.  
Read() invoked 7005603 us after write(), rxport.read(1) returned 0 bytes while blocked for 49967 us. Rcv timeout.  
Read() invoked 7055591 us after write(), rxport.read(1) returned 0 bytes while blocked for 60336 us. Rcv timeout.  
Read() invoked 7116003 us after write(), rxport.read(1) returned 0 bytes while blocked for 57022 us. Rcv timeout.  
Read() invoked 7173115 us after write(), rxport.read(1) returned 0 bytes while blocked for 65133 us. Rcv timeout.  
Read() invoked 7238309 us after write(), rxport.read(1) returned 0 bytes while blocked for 59935 us. Rcv timeout.

2023-08-15 16:36:21.6177446 INFO receive\_timing\_info -  
2023-08-15 16:36:21.677775 INFO receive\_timing\_info -  
2023-08-15 16:36:21.7278126 INFO receive\_timing\_info -  
2023-08-15 16:36:21.7779585 INFO receive\_timing\_info -  
2023-08-15 16:36:21.837842 INFO receive\_timing\_info -  
2023-08-15 16:36:21.8978067 INFO receive\_timing\_info -  
2023-08-15 16:36:21.9478241 INFO receive\_timing\_info -  
2023-08-15 16:36:22.0023277 INFO receive\_timing\_info -  
2023-08-15 16:36:22.062289 INFO receive\_timing\_info -  
2023-08-15 16:36:22.1123193 INFO receive\_timing\_info -  
2023-08-15 16:36:22.1624921 INFO receive\_timing\_info -  
2023-08-15 16:36:22.2224825 INFO receive\_timing\_info -  
2023-08-15 16:36:22.2824931 INFO receive\_timing\_info -  
2023-08-15 16:36:22.3347643 INFO receive\_timing\_info -  
2023-08-15 16:36:22.3929101 INFO receive\_timing\_info -  
2023-08-15 16:36:22.4529368 INFO receive\_timing\_info -  
2023-08-15 16:36:22.5090426 INFO receive\_timing\_info -  
2023-08-15 16:36:22.5591819 INFO receive\_timing\_info -  
2023-08-15 16:36:22.6091692 INFO receive\_timing\_info -  
2023-08-15 16:36:22.659081 INFO receive\_timing\_info -  
2023-08-15 16:36:22.7191373 INFO receive\_timing\_info -  
2023-08-15 16:36:22.7691467 INFO receive\_timing\_info -  
2023-08-15 16:36:22.829191 INFO receive\_timing\_info -  
2023-08-15 16:36:22.88547 INFO receive\_timing\_info -  
2023-08-15 16:36:22.9478704 INFO receive\_timing\_info -  
2023-08-15 16:36:23.0095628 INFO receive\_timing\_info -  
2023-08-15 16:36:23.0665456 INFO receive\_timing\_info -  
2023-08-15 16:36:23.1165447 INFO receive\_timing\_info -  
2023-08-15 16:36:23.1664869 INFO receive\_timing\_info -  
2023-08-15 16:36:23.2215136 INFO receive\_timing\_info -  
2023-08-15 16:36:23.2715149 INFO receive\_timing\_info -  
2023-08-15 16:36:23.321533 INFO receive\_timing\_info -  
2023-08-15 16:36:23.3820909 INFO receive\_timing\_info -  
2023-08-15 16:36:23.4418919 INFO receive\_timing\_info -  
2023-08-15 16:36:23.5018992 INFO receive\_timing\_info -  
2023-08-15 16:36:23.5579807 INFO receive\_timing\_info -  
2023-08-15 16:36:23.6211601 INFO receive\_timing\_info -  
2023-08-15 16:36:23.6799298 INFO receive\_timing\_info -  
2023-08-15 16:36:23.739918 INFO receive\_timing\_info -  
2023-08-15 16:36:23.7899988 INFO receive\_timing\_info -  
2023-08-15 16:36:23.8497246 INFO receive\_timing\_info -  
2023-08-15 16:36:23.8998561 INFO receive\_timing\_info -  
2023-08-15 16:36:23.9599324 INFO receive\_timing\_info -  
2023-08-15 16:36:24.0098813 INFO receive\_timing\_info -  
2023-08-15 16:36:24.0663109 INFO receive\_timing\_info -  
2023-08-15 16:36:24.1262503 INFO receive\_timing\_info -  
2023-08-15 16:36:24.1764724 INFO receive\_timing\_info -  
2023-08-15 16:36:24.2266545 INFO receive\_timing\_info -  
2023-08-15 16:36:24.2766775 INFO receive\_timing\_info -  
2023-08-15 16:36:24.3366769 INFO receive\_timing\_info -  
2023-08-15 16:36:24.3866781 INFO receive\_timing\_info -  
2023-08-15 16:36:24.386719 INFO receive\_timing\_info -  
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect+verify the serial connections. Aborting.

Read() invoked 7298265 us after write(), rxport.read(1) returned 0 bytes while blocked for 59879 us. Rcv timeout.  
Read() invoked 7358177 us after write(), rxport.read(1) returned 0 bytes while blocked for 60012 us. Rcv timeout.  
Read() invoked 7418209 us after write(), rxport.read(1) returned 0 bytes while blocked for 50015 us. Rcv timeout.  
Read() invoked 7468247 us after write(), rxport.read(1) returned 0 bytes while blocked for 50079 us. Rcv timeout.  
Read() invoked 7518415 us after write(), rxport.read(1) returned 0 bytes while blocked for 59826 us. Rcv timeout.  
Read() invoked 7578298 us after write(), rxport.read(1) returned 0 bytes while blocked for 59921 us. Rcv timeout.  
Read() invoked 7638241 us after write(), rxport.read(1) returned 0 bytes while blocked for 49997 us. Rcv timeout.  
Read() invoked 7688259 us after write(), rxport.read(1) returned 0 bytes while blocked for 54481 us. Rcv timeout.  
Read() invoked 7742762 us after write(), rxport.read(1) returned 0 bytes while blocked for 59941 us. Rcv timeout.  
Read() invoked 7802723 us after write(), rxport.read(1) returned 0 bytes while blocked for 50004 us. Rcv timeout.  
Read() invoked 7852755 us after write(), rxport.read(1) returned 0 bytes while blocked for 50153 us. Rcv timeout.  
Read() invoked 7902926 us after write(), rxport.read(1) returned 0 bytes while blocked for 59969 us. Rcv timeout.  
Read() invoked 7962928 us after write(), rxport.read(1) returned 0 bytes while blocked for 59979 us. Rcv timeout.  
Read() invoked 8022928 us after write(), rxport.read(1) returned 0 bytes while blocked for 52248 us. Rcv timeout.  
Read() invoked 8075200 us after write(), rxport.read(1) returned 0 bytes while blocked for 58090 us. Rcv timeout.  
Read() invoked 8133365 us after write(), rxport.read(1) returned 0 bytes while blocked for 59945 us. Rcv timeout.  
Read() invoked 8193399 us after write(), rxport.read(1) returned 0 bytes while blocked for 56039 us. Rcv timeout.  
Read() invoked 8249485 us after write(), rxport.read(1) returned 0 bytes while blocked for 50111 us. Rcv timeout.  
Read() invoked 8299627 us after write(), rxport.read(1) returned 0 bytes while blocked for 49958 us. Rcv timeout.  
Read() invoked 8349604 us after write(), rxport.read(1) returned 0 bytes while blocked for 49892 us. Rcv timeout.  
Read() invoked 8399514 us after write(), rxport.read(1) returned 0 bytes while blocked for 60039 us. Rcv timeout.  
Read() invoked 8459571 us after write(), rxport.read(1) returned 0 bytes while blocked for 49973 us. Rcv timeout.  
Read() invoked 8509580 us after write(), rxport.read(1) returned 0 bytes while blocked for 60027 us. Rcv timeout.  
Read() invoked 8569625 us after write(), rxport.read(1) returned 0 bytes while blocked for 56258 us. Rcv timeout.  
Read() invoked 8625914 us after write(), rxport.read(1) returned 0 bytes while blocked for 62349 us. Rcv timeout.  
Read() invoked 8688305 us after write(), rxport.read(1) returned 0 bytes while blocked for 61650 us. Rcv timeout.  
Read() invoked 8749999 us after write(), rxport.read(1) returned 0 bytes while blocked for 56941 us. Rcv timeout.  
Read() invoked 8806981 us after write(), rxport.read(1) returned 0 bytes while blocked for 49964 us. Rcv timeout.  
Read() invoked 8856979 us after write(), rxport.read(1) returned 0 bytes while blocked for 49865 us. Rcv timeout.  
Read() invoked 8906924 us after write(), rxport.read(1) returned 0 bytes while blocked for 55005 us. Rcv timeout.  
Read() invoked 8961948 us after write(), rxport.read(1) returned 0 bytes while blocked for 49981 us. Rcv timeout.  
Read() invoked 9011949 us after write(), rxport.read(1) returned 0 bytes while blocked for 49994 us. Rcv timeout.  
Read() invoked 9061969 us after write(), rxport.read(1) returned 0 bytes while blocked for 60496 us. Rcv timeout.  
Read() invoked 9122552 us after write(), rxport.read(1) returned 0 bytes while blocked for 59742 us. Rcv timeout.  
Read() invoked 9182334 us after write(), rxport.read(1) returned 0 bytes while blocked for 59981 us. Rcv timeout.  
Read() invoked 9242332 us after write(), rxport.read(1) returned 0 bytes while blocked for 56062 us. Rcv timeout.  
Read() invoked 9298415 us after write(), rxport.read(1) returned 0 bytes while blocked for 63130 us. Rcv timeout.  
Read() invoked 9361594 us after write(), rxport.read(1) returned 0 bytes while blocked for 58713 us. Rcv timeout.  
Read() invoked 9420391 us after write(), rxport.read(1) returned 0 bytes while blocked for 59896 us. Rcv timeout.  
Read() invoked 9480382 us after write(), rxport.read(1) returned 0 bytes while blocked for 49987 us. Rcv timeout.  
Read() invoked 9530463 us after write(), rxport.read(1) returned 0 bytes while blocked for 59636 us. Rcv timeout.  
Read() invoked 9590168 us after write(), rxport.read(1) returned 0 bytes while blocked for 50055 us. Rcv timeout.  
Read() invoked 9640322 us after write(), rxport.read(1) returned 0 bytes while blocked for 59974 us. Rcv timeout.  
Read() invoked 9700414 us after write(), rxport.read(1) returned 0 bytes while blocked for 49867 us. Rcv timeout.  
Read() invoked 9750335 us after write(), rxport.read(1) returned 0 bytes while blocked for 56370 us. Rcv timeout.  
Read() invoked 9806772 us after write(), rxport.read(1) returned 0 bytes while blocked for 59890 us. Rcv timeout.  
Read() invoked 9866685 us after write(), rxport.read(1) returned 0 bytes while blocked for 50200 us. Rcv timeout.  
Read() invoked 9916907 us after write(), rxport.read(1) returned 0 bytes while blocked for 50109 us. Rcv timeout.  
Read() invoked 9967115 us after write(), rxport.read(1) returned 0 bytes while blocked for 49934 us. Rcv timeout.  
Read() invoked 10017139 us after write(), rxport.read(1) returned 0 bytes while blocked for 59913 us. Rcv timeout.  
Read() invoked 10077138 us after write(), rxport.read(1) returned 0 bytes while blocked for 49910 us. Rcv timeout.

Here's the Windows log produced by executing 'receive\_timing\_info' with PR #79 patch and a 0 ms read timeout setting, where one (1) data byte is transmitted, and two (2) bytes are requested in the read. Note that the one transmitted byte is in-fact received, but the first phase/loop cycle fails (intentionally) since not all requested data bytes are received. This is expected - nothing is in-fact wrong here. I've abbreviated the log listing below as it grew too large to include the entire log content, due to the multitude of trailing immediate (0 time) return line entries prior to triggering the transaction level timeout.

```
2023-08-15 16:40:03.4407481 INFO receive_timing_info - 'receive_timing_info' cross platform dual RS-232 port null modem cable connected rcv+xmt+timeout test and characterization tool: v1.0
2023-08-15 16:40:03.4409159 INFO receive_timing_info - Test setup: Platform='windows', Baud=200000, rxtno=0 ms, postxttdelays=100 ms, xfrstalledtmo=10000 ms, txlen=1, rxlen=2, repeat=1, fulldbg=false
2023-08-15 16:40:03.4411003 INFO receive_timing_info - Test Logfile Name: 'D:/Users/ricelj/windows_receive_timing_info.txt'
2023-08-15 16:40:03.4736218 INFO receive_timing_info -
2023-08-15 16:40:03.4736429 INFO receive_timing_info - ** Start of cycle 1. **
2023-08-15 16:40:03.4736458 INFO receive_timing_info - Cycle 1 first phase -> Rx port = 'COM6', Tx port = 'COM7' .
2023-08-15 16:40:03.7753599 INFO receive_timing_info - txport.write() sent 1 bytes while blocked for 235 us. Read() invoked 100033 us after write(), rxport.read(2) returned 1 bytes while blocked for 5 us.
2023-08-15 16:40:03.7753919 INFO receive_timing_info - Read() invoked 100069 us after write(), rxport.read(1) returned 0 bytes while blocked for 10 us. Rcv timeout.
2023-08-15 16:40:03.7754004 INFO receive_timing_info - Read() invoked 100090 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:03.7754087 INFO receive_timing_info - Read() invoked 100098 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:03.7754182 INFO receive_timing_info - Read() invoked 100106 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:03.7754367 INFO receive_timing_info - Read() invoked 100116 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:03.7754501 INFO receive_timing_info - Read() invoked 100134 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:03.7754756 INFO receive_timing_info - Read() invoked 100148 us after write(), rxport.read(1) returned 0 bytes while blocked for 8 us. Rcv timeout.
2023-08-15 16:40:03.7754907 INFO receive_timing_info - Read() invoked 100174 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:03.7755129 INFO receive_timing_info - Read() invoked 100188 us after write(), rxport.read(1) returned 0 bytes while blocked for 6 us. Rcv timeout.
2023-08-15 16:40:03.7755307 INFO receive_timing_info - Read() invoked 100211 us after write(), rxport.read(1) returned 0 bytes while blocked for 6 us. Rcv timeout.
2
```

... .. <intentionally omitted uninteresting read() log entries ... >  
 ... .. <intentionally omitted uninteresting read() log entries ... >  
 ... .. <intentionally omitted uninteresting read() log entries ... >

```
2023-08-15 16:40:13.7747768 INFO receive_timing_info - Read() invoked 10099461 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7748555 INFO receive_timing_info - Read() invoked 10099490 us after write(), rxport.read(1) returned 0 bytes while blocked for 32 us. Rcv timeout.
2023-08-15 16:40:13.7748713 INFO receive_timing_info - Read() invoked 10099555 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7749368 INFO receive_timing_info - Read() invoked 10099601 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:13.7749518 INFO receive_timing_info - Read() invoked 10099636 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:13.775006 INFO receive_timing_info - Read() invoked 10099649 us after write(), rxport.read(1) returned 0 bytes while blocked for 27 us. Rcv timeout.
2023-08-15 16:40:13.7750209 INFO receive_timing_info - Read() invoked 10099706 us after write(), rxport.read(1) returned 0 bytes while blocked for 3 us. Rcv timeout.
2023-08-15 16:40:13.7750287 INFO receive_timing_info - Read() invoked 10099718 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7750398 INFO receive_timing_info - Read() invoked 10099726 us after write(), rxport.read(1) returned 0 bytes while blocked for 4 us. Rcv timeout.
2023-08-15 16:40:13.7750551 INFO receive_timing_info - Read() invoked 10099737 us after write(), rxport.read(1) returned 0 bytes while blocked for 10 us. Rcv timeout.
2023-08-15 16:40:13.7750628 INFO receive_timing_info - Read() invoked 10099753 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.775071 INFO receive_timing_info - Read() invoked 10099760 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7750789 INFO receive_timing_info - Read() invoked 10099769 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7750863 INFO receive_timing_info - Read() invoked 10099776 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7751934 INFO receive_timing_info - Read() invoked 10099784 us after write(), rxport.read(1) returned 0 bytes while blocked for 2 us. Rcv timeout.
2023-08-15 16:40:13.7753413 INFO receive_timing_info - Read() invoked 10099913 us after write(), rxport.read(1) returned 0 bytes while blocked for 42 us. Rcv timeout.
2023-08-15 16:40:13.7754058 INFO receive_timing_info -
TRANSFER STALLED TIMEOUT ERROR: 'rxport::read()' repeatedly timed-out without receiving its requested incoming data. If not induced, inspect-verify the serial connections. Aborting.
```