SerialSniffer manual
Version 6.01

SerialSniffer Manual

Printed: September 2012 in (whereever you are located)

Table of Contents

1 Introduction

1.1 Introduction

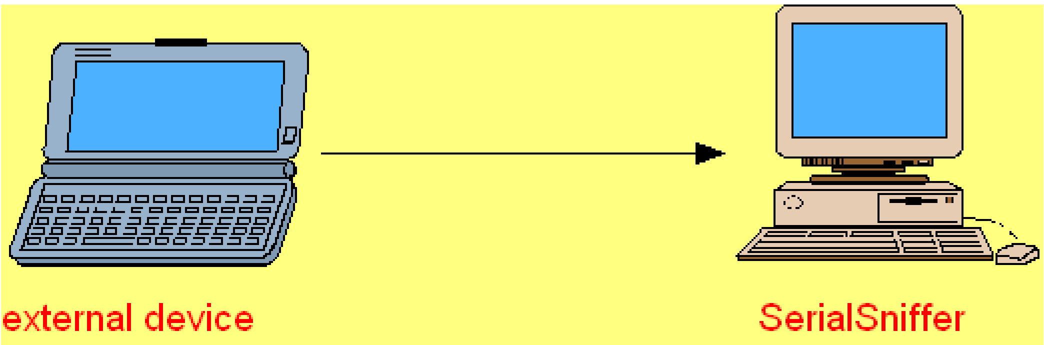
SerialSniffer is a tool to get more information about data, which is transmitted via a serial link.

First of all, there are **two different editions** of SerialSniffer available:

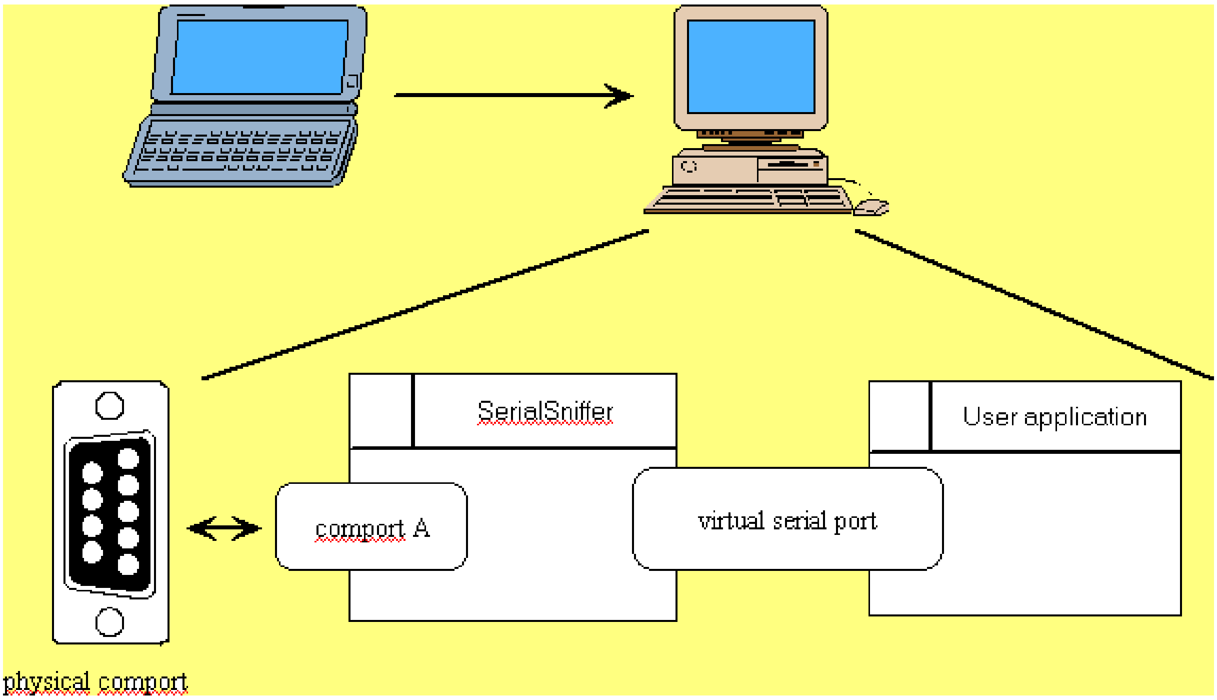
- Standard Edition: will not support the virtual serial port, but therefore does not need the installation of any drivers. Can be directly copied to the harddisk and used without installation
- VSP Edition: support the virtual serial port but needs the installation of the corresponding drivers.

Both editions can be found in the installation archive, which also contains the needed drivers and updates for Microsoft Windows, which may be needed for some Windows versions.
More information about the installation of SerialSniffer can be found here.

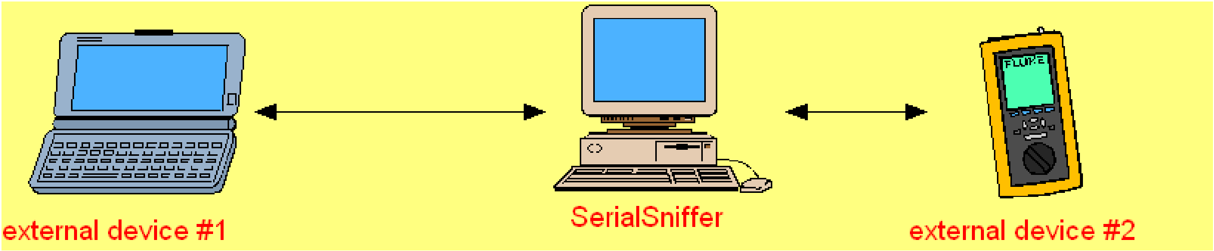
You can either connect SerialSniffer to a single device, which transmits serial data and SerialSniffer will print out the data, which is received, in the ASCII an HEX-Display.
In this mode, SerialSniffer works as a terminal software.



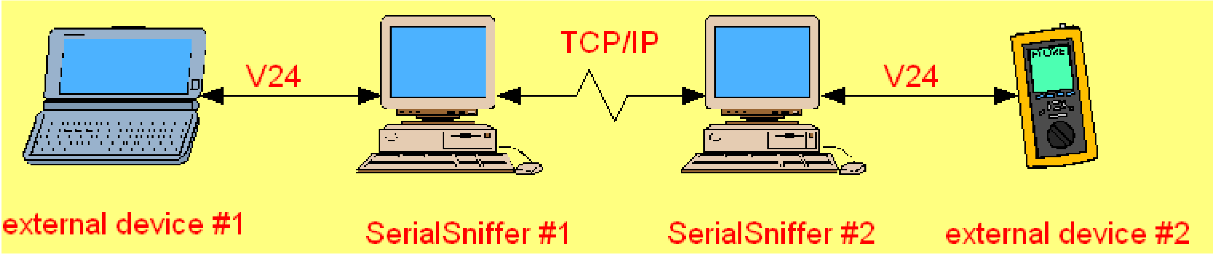
SerialSniffer also adds a virtual serial port to your system: connect your external device to the first port of SerialSniffer. Choose a free comport and open it as a virtual serial port (you will see the port added to the device manager after opening it). Now you can use any other program (i.e. the programm you are typically using with your external device) with "the other side" of the virtual serial port. SerialSniffer will pass all the data from one side to the other. Your standard application will work with the device as if there is no SerialSniffer (except from a short delay).



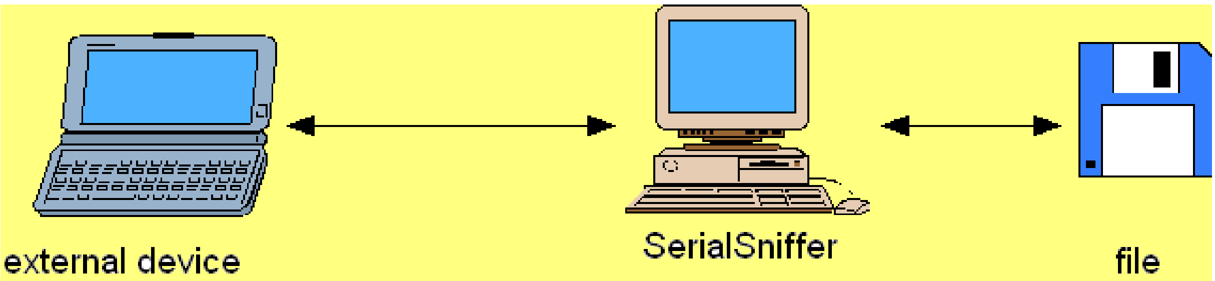
Another opportunity to use SerialSniffer is to use a PC with two serial ports and put the PC with SerialSniffer between the two devices, which are communicating with each other.
In this mode, Serial Sniffer displays the data from both devices, each in a separate color. So you can easily distinguish, which byte comes from which device.



Network functionality :
With this function, you will be able to connect two devices with each other, which are not in the same room. SerialSniffer connects them with each other using the TCP/IP protocol (see drawing) :



SerialRecorder :
Using this function, you can record the data, which is received by Comport A into a binary file together with the information about how many time was between the reception of each byte.



To minimze the influence of SerialSniffer, SerialSniffer can be connected "parallel" to the serial link instead of between the two devices (as shown in the picture above). You will find more information about the Hardware adapter or the V.24 tester in this documentation.

SerialSniffer is very easy to use :
Just enter the parameters for the serial link, you want to establish and click on the "Start"-Button. The serial port will be opend and all bytes, which are received, are displayed immediately. You will find detailed information here.

When you close SerialSniffer, the parameteres for the serial communication and the eXchange-function are stored in a data file, so you don't have to enter them again, when using this tool the next time.

Learn more about how to connect PCs and external devices by cables here.

You can get more information in the web. See About for more details.

You can start more than one SerialSniffer instance at the same time, but due to some drivers, this might cause serious problems.

See also :

- Installation
- About
- Terminal dialog
- Logging
- Trigger functions
- How to register
- Version history

Accessories :

- Hardware adapter

1.2 How to register

SerialSniffer is Shareware !!!

If you want to use this application permanently, please register it.

Shareware means, that you must evaluate the software before you register to make sure that the software will fit to your needs !

Please try, before you buy !!!

For restrictions, which are valid until you have registered, see below.

For detailled information how to register SerialSniffer and its models, please visit our website at:

http://www.serialsniffer.com/en/sites/serialsniffer_registrierung.php

The registration fee is 25 US$ (or 28 €) for one copy. For multilpe copies have a look into the following table :

amount of licenses	fee US$	fee €
1 - 2	25,-	28,-
3 - 4	22,50	25,-
5 - 9	20,-	22,50
10 - 20	15,-	17,-
21+	ask the author	ask the author

Any further updates will be free for registered users. Registered users will receive an email, when a new version is available for download.

The "SerialRecorder" function is a module ("module A") of SerialSniffer. This means, that you need a license to use this function. **The registration fee is 10 US$ (or 10 €) for one copy.** For multilpe copies have a look into the following table :

amount of licenses	fee US$	fee €
1 - 2	10,-	10,-
3 - 4	9,-	9,-
5 - 9	7,50	7,50
10 - 20	6,-	6,-
21+	ask the author	ask the author

The "virtual serial Port" (VSP) function is a module ("module B") of SerialSniffer. This means, that you need a license to use this function. **The registration fee is 10 US$ (or 10 €) for one copy.** For multilpe copies have a look into the following table :

amount of licenses	fee US$	fee €
1 - 2	10,-	10,-
3 - 4	9,-	9,-
5 - 9	7,50	7,50
10 - 20	6,-	6,-
21+	ask the author	ask the author

The prices above may vary because of the VAT, which is valid for your country.

This information was valid by the day, this documentation was written. For exact information about the current prices please visit our website at:

http://www.serialsniffer.com/en/sites/serialsniffer_gebuehren.php

After successfull registration, you will receive an email with a keyfile (named "SerialSniffer.key") attached to it within a few days. Simply copy this keyfile into the same directory as the application.

SerialSniffer has the following **restrictions**, which will disappear after you have installed the keyfile :

- Everytime you open a serial link (clicking the "open"-button) a message box will apear.
- Storage using SerialRecorder is limited to 64 characters.
- Unregistered copys are only valid for a limited period of time. After this period is over, you won't be able to open a serial link.
- The virtual serial port will be closed automatically after 60 seconds (virtual serial port is a addition modul and must be licensed separately)
- Re-opening of ports will be delayed with increasing period of time
- The overall usage time of SerialSniffer for Comport A is limited to 15 minutes. This will be tracked by our license server. Therefore, an Internet connection is needed in order to communicate with the server. Of course, licensed copies of SerialSniffer can be used unlimited and can also be used offline.

Remember that you must have at least two licenses to use the network functions, because otherwise the network connection will be close automatically after a short period of time. Have a look at Connections network for more details.

In case of an **invalid license**, which may happen, if the license is reported as not legaly purchased, a warning message occurs on program start. In this case, please contact the author (adress see above) and send the actual keyfile ("SerialSniffer.key") as an attachment. If you have a legal license, which is in the author's database, you will receive a new license file with a new serial number by email. The old license file with the old serial number will stay deactivated.
This error message may appear, if you have no write access on the actual directory (i.e. SerialSniffer is running from a CD-ROM). In this case, copy all files which belongs to SerialSniffer in a directory, where you have write access.

See also :

- Installation
- About

1.3 Version history

Version 1.00 16.04.01

- First Release

Version 1.01 21.04.01

- several smaller bugs processed

Version 1.10 03.05.01

- Online-Help : This page added
- ASCII-Output : every char < 0x32 is displayed as a bar now, resolving the problem with additonal Linefeeds
- Added "file" menu : "Open", "Save", "Saveas" & "Close"
- Added "Loggin" menu : "Start" & "Stop"
- Added "eXchange" menu : "Define"
- Added "?" menu : "Help" & "About"
- Main Dialog : Buttons "Close", "Help", "About" and "Logging-Buttons" deleted
- Added "eXchange" functions

Version 1.11 21.05.01

- Added short-cuts for menu
- The resource, which is used for the main dialog is now depending on the actual screen resolution : For resolutions which are 800x600 or smaller pixel, a smaller dialog appears, which has the configuration items for serial ports on an additional dialog
- Debug : DropDown Boxes for Baudrate are now updated when loading new parameters using menu "File -> Open"
- Main Dialog : Added "Enable/Disable" button
- Number of lines in Output-Boxes are now limited to 1000 lines
- Internal delay reduced from 25ms to 5ms
- Automatic deactivation of serial ports for non-registered version altered to 300 s

Version 1.12 28.05.01

- Added dialog "Configure Logging"
- More detailed logging functions, see **how to configure the logging**
- Added "Logging active!" button on main dialog in case of Logging activ

Version 1.20 25.06.01

- Not only the pre-defined baudrates from the dropdown-boxes are possible, the user can edit any baudrate, he wishes to
- Debug : eXchange parameters are now loaded immediately, before, they were only loaded after leaving the eXchange-Dialog with "OK"
- Debug : eXchange parameters with an empty "Replace" field caused the application to hang up
- EXchange function : Added Checkboxes "Add Enter" : User can now instruct SerialSniffer to add the characters "0x 0D 0A" at the end of the strings

Version 1.21 08.07.01

- Debug : There were some problems (exceptions) with V1.20 when opening a serial comport
- Debug : On Opening the comports the actual baudrate in the "Edit" Dialogboxes will be taken over
- Debug : endless loop in eXchange function, when search="a" and replace="abc"

- Added "Trigger" function

Version 1.30 10.09.01

- Device driver for serial ports completely updated, the problem with "(char) 0x00" upcoming in the output dialogs should be solved now
- Terminal dialog added
- Handling for hardware handshaking added
- Parameter "0 Stopbits" for serial communication deleted
- Output to the logging file increased from 100 to 1024 characters
- Radiobuttons for Com1..4 exchanged by ComboBox for Com1..32
- Debug : Parameter "AutoDeactivate" for Trigger function was not correctly initialized and saved in parameter file
- Debug : eXchange function created a recursive loop under special conditions
- Debug : Comport B was detected as open, even in case of an unseccessful Init for the serial port
- Debug : size of dialog "eXchange-Define" is now small enough to fit for 640x480 resolution
- Debug : The checkboxes for resolutions which are 800x600 or smaller had been ignored

Version 1.31 13.05.02

- Online help completely updated
- Added PatchProtection to avoid illegal patches
- Logging : Replacing Hex-values by user definded symbols added
- Logging : Function "Disable Linefeed" added
- Logging : Function "Single Hex char" added
- Logging : default filename has now date/time in it
- Priority : Dialog to change the priority added
- Terminal : "Backspace" should now work correct
- Terminal : Function "Hex input" added
- Terminal : Function "Raw upload" added
- Terminal : Function "Raw download" added
- Terminal : font changed to "courier"
- Configure network : dialog can only be opened, when no connection is made
- Configure network : serialnumbers of both applications must now be different
- Configure network : only one network connection is possible at the same time
- Menu item "define symbols" can now be found at "Extras"
- Function "Use ASCII symbols in display" added
- small dialog will now be used at resolutions of 1024x768 and smaller
- debug : special baudrates for serial communication should now work better
- debug : closing of serial ports may hang up
- debug : blocking connections were possible, when both checkboxes "Network" were marked
- several smaller bugs are done

Version 1.32 23.06.02

- debug : In V1.31 method "CancelIO()" was added, therefore SerialSniffer was not able to run with Windows95. Method removed.

Version 1.40 16.02.03

- Onlinehelp : spelling mistakes corrected
- Onlinehelp : picture V.24 tester exchanged
- Onlinehelp : circuit diagramm of hardware adapter corrected
- Onlinehelp : URL to website corrected
- Onlinehelp : explanation, why SerialSniffer may hang up due to hardware handshaking added
- Onlinehelp : explanation for software handshaking added
- Onlinehelp : explanation for cabeling added

- main dialog : counter for transmitted bytes (below ASCII output) added
- main dialog : terminal window cannot be opened, if network connection for comport A is not connected
- main dialog : comport A cannot be openend, if terminal window is open
- main dialog : small dialog will now be used at resolutions of 800x600 and smaller
- main dialog : to prevent scrolling in hex output, size of characters are now being set and hex output is bigger
- incorrect entries in parameter file for define symbols will now be exchanged by defaults
- main menu : "Use hex symbols in display" added
- main menu : "Local echo comport x" added
- Define char value dialog : empty entries are no longer possible
- complete redesign of logging
- network connection : partly redesigned due to bugs in transmission
- network connection : no data will be send unless the connection is established
- terminal dialog : now 5 different files can be uploaded to support sending pre-defined sequences
- terminal dialog : files for raw upload will now be transmitted correctly
- terminal dialog : key "v" was interpreted as "CTRL-V" and therefore pasted the clipboard
- debug : in case of changing the actual directory, the parameter file was saved in the wrong directory

Version 1.41 11.08.04

- Enddate for Evaluation period changed to 06/30/2005
- new Version of License key encryption added

Version 1.50 02.07.05

- Enddate for Evaluation period changed to 06/30/2006
- debug : comports > "Com9:" can now be used
- added parameter "identifier"
- Logging : added the system name of the comport (i.e. "Com1:")
- Logging : added function "SingleLine"
- Logging : debug : in case of ASCII output and when the first character of the line is a space, the whole line was shifted
- debug : spaces in hex output (main window) were missing in case of using hex symbols
- Terminal : added funtions to send pre-defined strings by the special function keys
- Terminal : added abort windows in case of raw upload
- main dialog : the number of characters per line can now be configured
- changed restrictions for not registered shareware : ports will now be closed after 3 min, log files are restricted to 256 characters

Version 2.00 09.11.05

- Enddate for Evaluation period changed to 12/31/2006
- project changed to Visual Studio.NET 2003
- dialogs changed to RichEdit2
- file upload using terminal dialog: debug: exception on closing abort dialog fixed
- number of usable comports increased to 256
- added function "SerialRecorder" (SerialSniffer modul A)

Version 2.10 20.04.06

- Enddate for Evaluation period changed to 06/30/2007
- added checkboxes for (de-)activating of eXchange function on main dialog
- added use of wildcards for eXchange function
- debug: fixed problems during writing of parameter file ("SerialSniffer.dat"
- debug: "number of bytes per row" was sometimes empty or not updated
- debug: corrupt keyfiles may cause exceptions

Version 2.20 05.10.07
- Enddate for Evaluation period changed to 12/31/2008

Version 2.21 06.12.07
- Bugfix in serial communication: loss of complete received data fixed

Version 3.00 release candidate 1 09.12.08
- added virtual serial port
- Bugfix in serial communication: some user still have problems receiving data, should be fixed now
- Bugfix terminal dialog: "backspace" did not work starting with line 2 of the editor window

Version 3.00 release candidate 2 09.12.08
- added second virtual port for comport A

Version 3.00 release candidate 3 15.12.08
- debug of file dialogs: sometimes, SerialSniffer crashes upon opening a file dialog

Version 3.00 Final 25.01.09
- debug: logging was not able to handle more than 48 bytes per line
- Enddate for Evaluation period changed to 12/31/2009

Version 4.00 25.11.10
Major:
- SerialSniffer now supports two different editions:
  - Standard edition: without support for virtual serial port but therefore with no need for installing drivers
  - VSP edition: with support for virtual serial ports
- fitted for Windows 7
- search for available updates upon start of application added
- Enddate for Evaluation period changed to 12/31/2011
- support for screen resolutions smaller than 1024x768 deleted
- added symbolic names for use of comport names in logging
- added possibility to log only one comport or both
- logfile will no longer be locked by SerialSniffer during logging, only for short time intervall of writing data to the file

Minor:
- Terminal dialog:
  - re-arranged buttons "Clear" & "Copy" according to main dialog
  - re-arranged edut fields for "special functions keys"
  - filenames and path for uploading files right-justified for better reading of filename
  - spaces can now be used during input in hex mode
  - auto-update for "special function keys" upon leavin the edit field ("kill focus"), no more update button
  - each "special function keys" can now add "enter"
  - all characters should now be displayed (e.g. the euro sign was not correctly displayed)
- eXchange define dialog:
  - other font used for better visibility of content of edit fields
  - spaces can now be used during input in hex mode

Version 5.00 28.02.12
Major:
- Stand-Along setup rogram as well as De-Installer included in package
- changing user rights in installation folder by setup in order to have write access
- upgrade included virtual serial port drivers to V7.0
- added support for netbooks with screen resolution smaller than 1024x768

- enddate for Evaluation period changed to 12/31/2014
- Terminal dialog: integrated possibility to delay file raw upload in given packets

Minor:

- added startup screen
- Main screen: deleted "update" button for "Number of Bytes per Row"
- Trigger dialog: debug: default values were not loaded

Version 5.01 11.04.12

Major:

- debug: Setup and De-Installer updated: update of DLL needed for virtual serial ports on 64 bit machines

Minor:

- debug: playback of SerialRecorder is now also possible on virtual Comport A

Version 6.00 11.09.12

Major:

- actual version is useable until 30.06.2012
- complete rework of restrictions of un-registered Shareware Version: see http://www.serialsniffer.com/en/sites/serialsniffer_lizenz.php for details

  -> most important change: the left comport (Comport A) can only be used for a given time throughout the whole livetime of SerialSniffer not only per session. The remaining lifetime will be handled by our server. Therefore, SerialSniffer needs an online access as long as it is not regisered! Of course, these restriction disappear with the registered version and offline usage is possible!
- in case of high amount of quickly incoming data, incoming stream became mixed up after a while due to packet handling used too small packets, additionally, SerialSniffer seems to "hang", as the data processing took too much time and the screen wasn't refreshed
- debug: writing data to serial ports seems to block under certain circumstances
- TerminalDialog: depending on actual screen resolution, different sizes of the dialog will be used
- SerialRecorder: data is now displayed both in ASCII and Hex display during playback
- Shareware-Version: added increasing delay for re-opening Comports

Minor:

- added display of actual version in splashscreen
- logging: added "use milliseconds": switch on/off display of milliseconds in logfile (usefull to switch off in case of later import to Microsoft Excel)
- logging: debug: upon using paragraphing on a fixed number of characters per line, the timestamp was not updated
- registry handling: check for completeness of entries
- storing installation path in registry for usage on re-install and de-install

Version 6.01 20.09.12

Major:

- memory for receiving data via serial ports no longer dynamically but static
- output of serially received characters per second, if exceeding thrshold, loss of data may happen -> displaying warning

Minor:

- counter for processed characters will continue to count, even if output to main window is disabled
- number of lines in output windows reduced to 100

1.4 Installation

As from V5.00 of SerialSniffer, a setup program will be delivered. You will find this "Setup.exe" in the main folder of the installation archive. Please only use this setup and do not try to install SerialSniffer by your own. This may cause trouble on your system and may inhibit the full functionality of SerialSniffer.
Also, a de-installation software is available which will be installed automatically.

There are **two different editions** of SerialSniffer available:

- Standard Edition: will not support the virtual serial port, but therefore does not need the installation of any drivers. Can be directly copied to the harddisk and used without installation
- VSP Edition: support the virtual serial port but needs the installation of the corresponding drivers.

The setup program let the user choose, which edition he wants to install.

Starting with V4.00 of SerialSniffer, each time the application starts, SerialSniffer takes a short look on our webserver in order to check, whether a newer version is available.
If not, nothing more happens.
If yes, SerialSniffer will show a notice to the user informing him about the availabled update.
During this update check, only one website will be downloaded. No information from the customer to our servers will be transmitted.

Starting with V5.00 of SerialSniffer, the setup program will transmit your public IP adress together with a unique identifier to our servers in order to help us improve SerialSniffer.
Also, as long as you are using the shareware version (= having no license), information about the duration of your usage will be transmitted to our servers. As soon, as you obtain a license, no more data will be transmitted!
This information will be kept strictly confidential and no third party will get access to this data. In case, you have more questions on this, please contact us.

1.5 Cabeling

How to connect PCs and external devices which each other

It is always a little bit tricky to find out, which type of cable has to be used : Sometimes you need a 1:1 cable and sometimes a cable with crossed connections. This page is trying to clarify this.

You always have to cross the cable, when you are using devices of the same type. This means :
- connect two PCs : cross the cable
- connect two external devices, which can be connect to a PC using a 1:1 cable : cross the cable

But in case of a PC and an external device, you have to use a 1:1 cable.

A 1:1 cable means :
- connect RXD with RXD
- connect TXD with TXD

A crossed cable means :
- connect RXD with TXD
- connect TXD with RXD

There must be always a connection of both GND connectors.

There is no risk of using the wrong type of cable. So in case, you are not sure, try it out. There can be not damage to the PC or the external device. (Otherwise, please don't harm me !)

2 How To Use

2.1 About

See the Version history for more information.

Download of most recent version :
http://www.serialsniffer.com/en/sites/download.php

More information (english version) :
http://www.serialsniffer.com/en/index.php

Weitere Informationen in deutsch :
http://www.serialsniffer.com/de/index.php

Warning :
Due to changes in the data file ("SerialSniffer.dat") you should **always** delete the old data file when you are upgrading to a new version !

Copyright for SerialSniffer :

Entwicklungsbuero Dipl. Phys. Stefan Flache
Tiroler Ring 2
24147 Kiel, Germany
shareware@flache.de

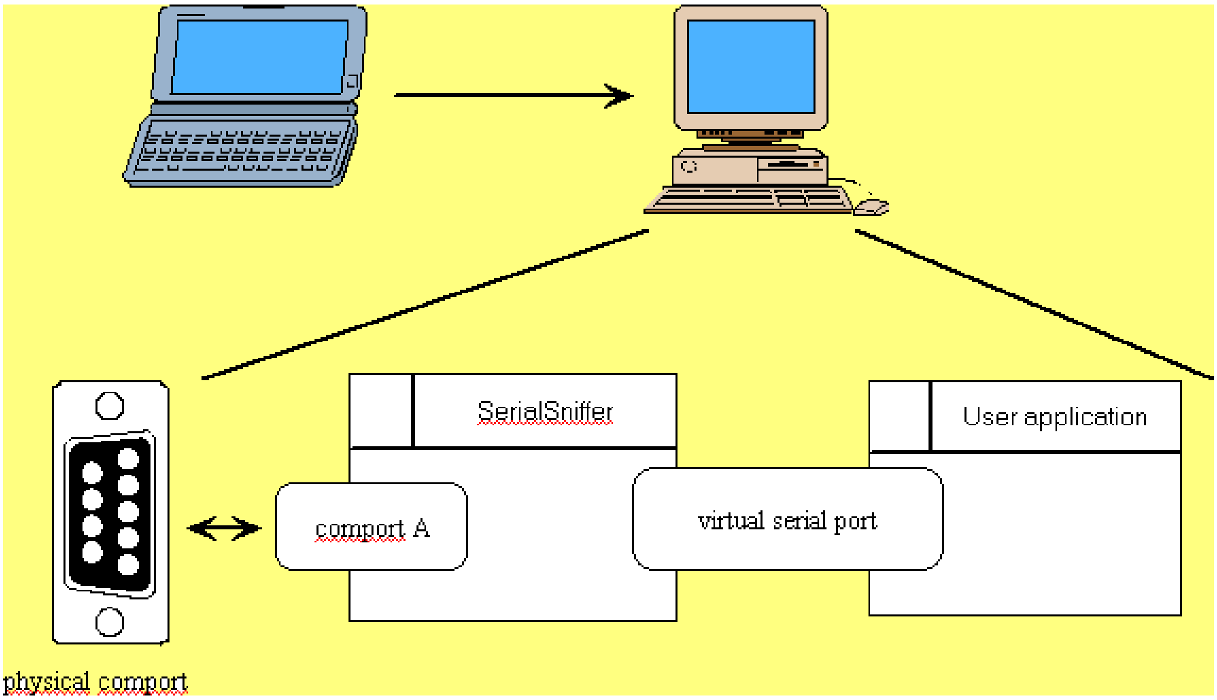
See also :

- How to register

2.2 VirtualSerialPort

Starting with version 3.00, SerialSniffer provides a virtual serial port. This saves the need for additional serial ports to connect "the second side of SerialSniffer" (= Comport B) with your standard application. You can use your application as you are used to it and SerialSniffer will do all of its work.

Only 1 physical serial port is needed for SerialSniffer and your own application at the same time !!!



How it works:
SerialSniffer includes an active X element, which provides a virtual serial port. You have to install and register this element before using it. Please look at the installation chapter for details. Remember that SerialSniffer comes with two editions, the Standard one and the VSP edition. **This functionality is only available in the VSP edition!**
One comport can only be used as a physical comport or a virtual one, not both at the same time.

The virtual serial port will appear in the device manager **after** openining it in SerialSniffer and of course, it can be opened only after creating it. So, please ensure, that you first open the virtual serial port in SerialSniffer and then in your user application!

In case, you do not have a valid license (meaning that you are using it for evaluation), the virtual serial port will close automatically after 3 minutes.

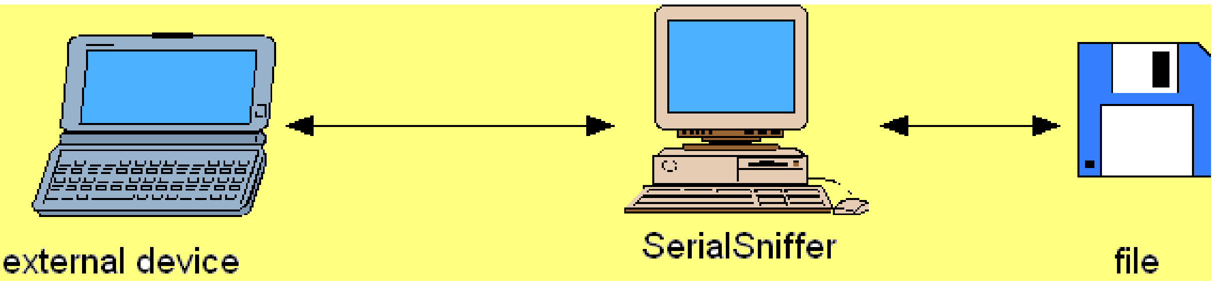
Step-by-step instruction:
- Connect your external device and open the physical comport on Comport A
- Choose a comport number which is not in use for the virtual port (using Comport B)
- Open Comport B -> the virtual serial port will appear in the device manager
- Open the virtual serial port in your user application (it is the same port number as in SerialSniffer - ComportB)

- Start working

ATTENTION: SerialSniffer will not start, in case the port drivers are not installed and registered!!!
Please read the Read-Me files in the SerialSniffer archive!!!

2.3 SerialRecorder

Did you ever have the problem to store a serial data stream to disc together with the time information about the amount of time between the reception of the data ?



SerialRecorder is the answer for this. For every character, which is received by comport A, the character itself together with the amount of milliseconds between this character and the one before is stored into a binary file. The maximum space between two characters can be 1 day.
When you playback the file, the data will be sent by the serial port of comport A (not via network) with the same distance between the single characters.

One application could be to save a data stream somewhere out of your office from a device, store all the information needed and restore the stream at home for testing purpose.

SerialRecorder is a "module" of SerialSniffer. This means, that you have to obtain a single license together with the license from SerialSniffer itself. Without a license, you can only use this function in testing mode, meaning that the amount of characters, which are stored in the file is limited. For more information, see How to register.

2.4 Main dialog

SerialSniffer is very simple to use :

In the two big **output windows** you will find the data, which is received from the serial port(s).
The left window shows you the content in ASCII-characters. If a received byte cannot be displayed as a ASCII-character, a space is displayed instead. The rigth window shows you the same content as two-byte Hex-values. The number of lines in both displays (ASCII & HEX) is limited to 1000 lines. Each time, the maximum value is reached, the topmost 50 lines are deleted. The **number of characters per line** can be configured by the user.
When you are using the scrollbars, both windows will be scrolled simultaneously.
Bytes, which are received by Comport A are displayed in **red**, bytes received by Comport B are displayed **green**.

The button **Clear** clears both output windows.

The button **Copy** copys the actual selection in the output window to the Windows clipboard.

The button **Enable** gives you the possibility to stop the output on the ASCII an HEX-Displays and the logging. The serial communication is not affected by this.

With the menu items Logging-**Start** und **Stop** you are able to select a text file, where everything, which is displayed in the ASCII-window is written to.

For information of how to configure the comports, have a look at this.

To open a serial port, simply select the correct communication parameters and click the **Start** button.
By clicking the **Stop** button you can close the serial port again.

The checkboxes "**DTR**" & "**RTS**" gives you control of the output lines for **hardware handshaking**.
As long as the checkbox "**Auto**" is not activated, SerialSniffer does not check its input lines (DSR & CTS) and transmitts every character immidiately, regardles of the actual status of the input lines. When you active the checkbox, SerialSniffer takes care of the input lines. When both comports (Comport A & B) are opened as serial ports (checkbox "serial" activetd), the actual status of the hardware handshaking lines one port is passed through to the other port. This means, that :

- Output line RTS of Comport A will follow Comport B CTS
- Output line DTR of Comport A will follow Comport B DSR
- Output line RTS of Comport B will follow Comport A CTS
- Output line DTR of Comport B will follow Comport A DSR

The total number of received bytes is printed out in a box below the ASCII output. The counter can be resetted by using the Clear-Button.

Remark :
When you are using software handshaking, SerialSniffer is "transparant" for it. This means, that SerialSniffer doesn't react in any matter if you are using software handshaking. The reason is, that SerialSniffer doesn't distinguish between any characters. Every character is handled in the same ways and passed to the other comport.
So you can use it for your application and don't have to worry about it.

Remark :
In case, you use one comport of SerialSniffer for network connection as a server and this server is not yet connected to a client, you will not be able to open the other comport. Otherwise SerialSniffer will not be able to process data, which is received by the other comport to the network connection, because this connection is not working yet.
Additionally, you can not open the terminal window of comport B as long as Comport A is not connected via network and you can not open Comport A for network connection as long as the terminal

window is open (because the network connection may not be connected immediately and the data could not be processed).

Remark :
If SerialSniffer appears to work not fluently, the PC might be not fast enough to process all the tasks simultaneously. Try to close other applications and avoid the terminal window, which needs a lot of resources.

Special functions :

- Menu "Tools -> SerialRecorder" : save and playback data on disc
- Priority : change the priority which is used for SerialSniffer
- Identifier : add a unique identifier to this instance of SerialSniffer, which is printed in the main window and the log file
- Menu "Extras → Use ASCII Symbols in Display" : When activated, the output of the ASCII display will not display the characters, which are received by the comports, but will exchange each character with its counterpart from "Extras → Define ASCII Symbols".
- Same for "Extras → Use Hex Symbols in Display".
- Menu "Extras → Local Echo Comport x" : Gives you the possibility to immediately echo all characters, which are received by the specified comport, back to the same port.

Warning :
When using **hardware handshaking**, SerialSniffer seems to "hang up", when the receiver, where SerialSniffer wants to transmit data to, is not ready. As soon as the receiver becomes ready, the charaters will be transmitted and the application will continue. This is due to the PC, which recognizes that the hardware handshaking lines indicates that the counterpart is not ready, stops the application, which is trying to send data. The best way to solve this problem is :

- avoid using hardware handshaking,
- connect only RXD, TXD and GND,
- when SerialSniffer "hangs" try to make the counterpart accept data again, maybe it helps to close it.

Learn more about how to connect PCs and external devices by cables here.

See also :

- Introduction
- Virtual Serial Port
- How To register (see also for details on invalid licenses)

2.5 Terminal dialog

The Terminal dialog gives you the opportunity to use a terminal dialog, similar to HyperTerm, directly inside of SerialSniffer.

To start the terminal dialog, simply click on the "**Terminal**" button in the parameter area for Comport B. The terminal window will be opened instead of opening a serial or ethernet communication port. Every character, which is received by Comport A will be display in the terminal window and you have the possibility to enter characters yourself, just like the other terminal programs you know.

When the "**local echo**" checkbox is marked, every character is displayed in the terminal dialog and is transmitted by Comport A. When the checkbox is unmarked, the characters will not be displayed.

The "**Hex Input**" checkbox enables SerialSniffer to interpret two characters which are typed in as one character in hexadecimal notation. This means, if the user types "31", the character 0x31 = dec 49 = "1" is sent to the other comport.

With "**Send 'Enter' as :**", you can control, which characters are transmitted via Comport A when you are pressing the "enter" key :

- CR-LF means characters 0x0D & 0x0A
- CR means only the character 0x0D
- CR means only the character 0x0A

The **Clear**- und **Copy**- buttons are working in the same manner as in the main dialog.

With the "**Raw Upload**" function, you have the possibility to define up to 5 files, which can be uploaded by one click on the "Upload"-button. With the help of this capability, you are able to first edit the files with a hex editor and then send the files as pre-defined sequences. Use this function to send sequences, you need often. While the upload is active, an abort button enables you to abort the transfer.
The **Delay** function gives the user the possibility to insert a delay of X milliseconds every Y bytes. Otherwise, the content of the upload will be transmitted with maximum speed. This may be helpful, if the receiver is not able to collect all the data at maximum speed and needs some breaks to do so.

In reverse, the button "**Raw Download**" opens a file, in which all characters, received by Comport A are written to.

You can pre-define strings which are sent on pressing a **special function key**. SerialSniffer can add an "Enter" as defined above to each string by activating the corresponding checkbox. Usually, you may want to add ASCII statements for use with the special functions key. But you can also enter the string to be sent as hex input. In this case, spaces between the 2-byte blocks (describing each one character) will be ignored.

See the remark on the main dialog page about SerialSniffer not working fluently.

See also :

- Introduction

2.6 Configure logging

To use the logging feature, you have to configure the logging mechanism and then "start" it.
When you are starting the logging, you will have to specify the name of the textfile, the output will be printed to. **During logging** (= until you have "stopped" it and therefore closed the file), **you are not able to open the logfile externally** (i.e. by the Windows Explorer) **nor to change the logging parameters.**

The Logging can be enabled or disabled. This means, that, even when the logging file is open, no characters will be printed out, when the logging is disabled. This function gives you the possibility to stop the logging for a while without closing the file.
To activate/deactivate the logging, you can either use the **Enable/Disable button** on the main dialog (the same button, whichs enables/disables the output on the main dialog) or the "**Enable**" checkbox in the logging configure dialog.

There are 3 different types of logging output :

- **Single Hex characters**
  Only one character is printed out as 2-digit Hexcode per row, followed by the comport, by which it was received ("A" or "B") and the time in milliseconds since the previous character was printed out (in case of the first character, it is the time since opening the logfile).
  Example :
```
31     B     926
32     B     194
33     B     232
34     B     238
35     B     233
61     A     7939
62     A     232
63     A     271
```

- **Plain ASCII**
  All the characters, which are received are printed out to the logfile. No timestamp or source of the characters will be remarked. There can be as much characters per row, as you like.
  Example :
```
This is a test of Plain ASCII logging.
Next line
```

- **Complex Output**
  This is the most powerfull way of logging. You have several parameters to take care of :
  - You have the possibility to decide, whether you want to log only one of the two comports or both by setting the checkboxes.
  - **SingleLine** : the complete output is printed in the same line including timestamp and portname to enable the user to import the logfile somewhere else (i.e. Microsoft Excel ©)
  - **ASCII only**, **Both**, **Hex only** : The logging will be done similar to the main dialog : You can have the ASCII-output on the left side and the Hex-output on the right side or only ASCII or only Hex. Choose what you want.
  - When **Enable paragraph** is checked, a linefeed is inserted each time, characters are received from a comport from which the had not been received before. Additionally, you can print out the **Port Names** and a **Timestamp** at the beginning of each new paragraph.
  - **Use ASCII symbols** and **Use Hex symbols** behave in the same way as in the main dialog : the output characters are exchanged by their symbols defined in the Define Symbol dialog.
  - With **Bytes per row** you can determine, how much characters are printed per row. Values between 1 and 256 are allowed.
    Example :
```
05.02.2003 20:27:07:400     Comport B
12345678                           31 32 33 34 35 36 37 38
90123456                           39 30 31 32 33 34 35 36
```

```
7890                           37 38 39 30

05.02.2003 20:27:14:920     Comport A
abcdefgh                       61 62 63 64 65 66 67 68
ijklmnop                       69 6A 6B 6C 6D 6E 6F 70

05.02.2003 20:27:25:250     Comport B
xxx                            78 78 78
```

- **Paragraph after time out of** xxx **ms** gives you the possibility to insert additional paragraphs in case no characters are received for the specified period of time. This could be useful, if a transmitter always sends the same or similar strings periodically and you want to log the timing of it.
- n the log file, you can choose to either use the name of the comport (e.g. "Comport A") or a symbolic name. E.g.: You have an external device communicating with an already existing application on your PC (via the virtual serial port). In this case, it may be helpful not to have "Comport A" and "Comport A" written in the logfile but "my Device" and "Application".

If you receive characters, which can not be printed out, strange things may happen. In this case, you should use **Use ASCII symbols** to ensure that only printable characters are written to the logfile.

The default filename for the logging has got the following structure :

SerialSniffer_YYMMDD_hhmm.log

- YY : Year (without century)
- MM : Month
- DD : Day
- hh : Hour
- mm : Minute

If you have added an unique **identifier** from the main menu, it is added to the logfile.

There was a complete redesign of the logging in version 1.33 !

See also :

- Introduction

2.7 Define Symbols

With the help of this dialog, you can define the symbols (as strings), which are used to replace the ASCII- and the HEX-values in the logging output ("Use Hex Symbols"), if the checkbox "Replace Symbols" is activated in the configure logging dialog.

or the output in the ASCII dialog on the main window ("Define ASCII Symbols").
To use the ASCII symbols in the main dialog, you have to activate the menu item "Extras → Use ASCII Symbols in Display".

All values (from 0x00 to 0xFF) can be replaced by a symbol. Because there are 256 fields for editing the values necessary, the dialog is divided into four parts. Use the menu items to select the range, you want to edit. There are two datasets, one for ASCII-Symbols and one for Hex-Symbols

With the **Set default**- button, the values of the actual page are set to their default values. The default values for ASCII-Symbols are the ASCII-characterset, but the values for the characters 0..31 are replaced by the character 127 ("I").

There must be a symbol (even if it is a blank) in each edit field. Empty fields are not allowed and you are not able to close the edit dialogs with the OK-button.

The default values for the Hex-Symbols are listed in the following table :

Hex value	Symbol
0x00	NUL
0x01	SOH
0x02	STX
0x03	ETX
0x04	EOT
0x05	ENQ
0x06	ACK
0x07	BEL
0x08	BS
0x09	TAB
0x0A	LF
0x0B	VT
0x0C	FF
0x0D	CR
0x0E	SO
0x0F	SI
0x10	DLE
0x11	DC1
0x12	DC2
0x13	DC3
0x14	DC4
0x15	NAK
0x16	SYN
0x17	ETB
0x18	CAN
0x19	EM
0x1A	SUB
0x1B	ESC
0x1C	FS
0x1D	GS
0x1E	RS
0x1F	US
0x20 ... 0xFF	0x20 ... 0xFF

2.8 Configure comport

The user can configure the comports directly at the bottom of the main dialog.

You can select :

- Comports : 1..32
- Parity : None, Even, Odd
- Databits : 7, 8
- Stopbits : 1, 2
- Baudrate : 75 … 115200 Baud

You are able to enter any baudrate (>= 75 Baud) you want, in the edit field below the baudrate dropdown-box or you can use a pre-defined baudrate from the dropdown-box.

There must be a valid entry in the edit field (valid means a baudrate inside the range described above). Otherwise the programm will change the value to the default (9600 Bd).

If you discover problems with the correct order or the timing of the data (using log mechanism), you should disable the FIFO for your comports in the device manager.

See the **warning about hardware handshaking** and the **remark about software handshaking** on the main dialog page .

See also :

- Introduction

2.9 Define eXchange

The eXchange functionality gives you the possibility to define up to three datasets which contains strings which should be exchanged during serial communication.

For example :
You have defined :

- Search for : "**123**"
- Replace with : "**5678**"

then SerialSniffer will do as follows :

When receiving the following string on Comport A :

abcd12efg**123**hijk

the output will be :

abcd12efg**5678**hijk

There can be up to three datasets which are independet from each other. Each dataset can be activated or deactived. When deactivated, it won't be processed. You have the possibility to define the string as ASCII (like the sample above) or as Hex characters. This means that a string like :

6162

means :

ab (0x61 = 'a'; 0x62 = 'b' !)

There is a TimeOut of 250 ms for receiving the whole "Search For" string. If the string is not received completley within this time, it will not be processed.

With the checkboxes "Add Enter" you can instruct SerialSniffer to add the characters "0x 0D 0A" (meaning Carriage Return / Line Feed) at the end of the Search and/or Replace strings.

Wildcards : You can use the '?' as a wildcard for exact one character. In case of using hex input, use 0x"3F" as the wildcard (= ASCII code for '?'). You can not use the '?' in hex mode for a nibble

This function works only on Comport A !

See also :

- Introduction

2.10 Define trigger

The Trigger function lets SerialSniffer works like an "digital oscilloscope".

You can define a trigger string, which will "trigger" SerialSniffer. This string can be an ASCII or a HEX string (see also Define eXchange).

When this function is active, first the Box named "**Trigger:**" will be named "**Waiting...**". This means, that SerialSniffer is waiting for the Triggerstring to received.

Once this string is received, it will be displayed in **blue** characters in the display boxes. The Trigger-Box will change to "**Triggered**".

After the additional amount of characters, which are defined as "**Postscaler**" in the Define dialog are received, the Box will turn to "**Waiting...**". again (the postscaler value is the minimum value of characters, there can be some more characters, which will be displayed, too).

If you have activated "**Auto Deactivate**" in the Define dialog, the display boxes will be automaticly deactivated at this point of time (the serial communication will go on, of course).

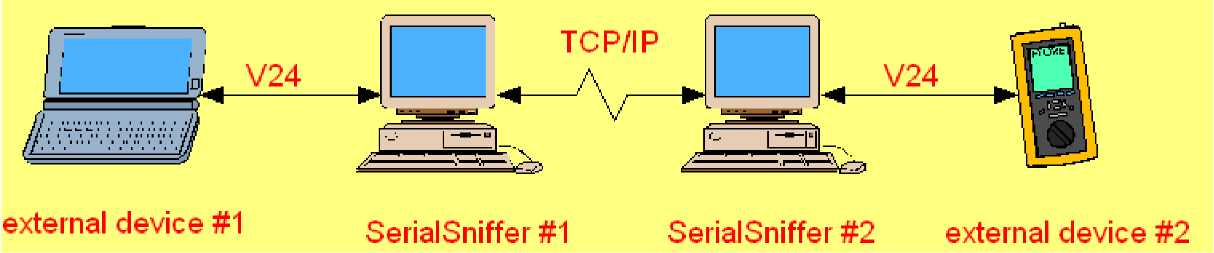
This function is "retriggerable". This means, that a secound trigger string, which is received before postscaler is reached, will reset the postscaler.

See also :

- Introduction

2.11 Connections network

You are able to use SerialSniffer as a converter between a serial communication port and the network, your computer is connected to (using TCP/IP) :



To do this, first you have to enter these parameters in the **configuration dialog** :

- **Client / Server** : Only two copies of SerialSniffer can work together via the network. One of them is the server and the other one is the client. Remeber that you have to start the server first, so that the client can connect to it.
- You have to specify a **Port**, which will be used to establish the communication. You will find more information about the usages of ports at the usual information databases about the TCP/IP protocol.
- If you configure SerialSniffer as a client, you must specify the **IP-Adress** of the machine, the SerialSniffer-Server is running on. You can do this, using real names (i.e. "localhost" for your own machine or "SerialSnifferServer@MyLan.com") or you use the IP-Adress (i.e. 192.168.1.1), if you know it.

Attention :
Due to changes in the communication protocoll, you have to use SerialSniffer at least version 1.31 on boh sides of the communication. Using an older version won't work !

Attention :
SerialSniffer is using "Blocking mode" when sending data on the network. This means, that the application seems to "hang" during a transmission is in progress. In case, you are transmitting a large amount of data. the application may "hang" quite a long time. Even the byte-counter in the main dialog may stop for some time, in case the application on the other side of the network is not able to accept the data as fast, as you are sending them. Immideately after the data, which is already sent, is processed and the input buffer of the other application is empty again, your SerialSniffer will restart sending the rest of the data.
So don't worry, if SerialSniffer doesn't react for some time, when in network mode.

Here is a typical configuration, you could use, as an example :

SerialSniffer-Server (running on "SerialSnifferServer@MyLan.com"). This machine works as a transmitter from serial com 1 to the network :
- Network-configuration :
  - This is a server
  - Port 5000
  - IP-Adress : left open
- Comport A : "Serial" only, "Network" unmarked
  - Set the serial parameters correct !
- Comport B : "Serial" unmarked, "Network" only
  - Serial parameters are not used

SerialSniffer-Client (running on another machine). This machine works as a receiver from the network to the serial comport 2 :
- Network-configuration :

- This is a client
    - Port 5000
    - IP-Adress : SerialSnifferServer@MyLan.com
- Comport A : "Serial" unmarked, "Network" only
    - Serial parameters are not used
- Comport B : "Serial" only, "Network" unmarked
    - Set the serial parameters correct !

In case you want to use SerialSniffer via network with another copy of SerialSniffer, the two copies must have **different serialnumbers** (see menu "?" → "About" and have a look at the "SerNo"). If the serialnumbers are identical, the network connection will be automatically after 5 minutes in order to give you a chance to try it out.
To obtain a further copy of SerialSniffer, have a look at How to register.

See the **remark** in the main dialog page for opening network connections as a server without connecting them to a client.

If you are not sure about the correct settings of the parameters, or if your client can not connect to your server, please ask you local supervisor. He will sure help you.

See also :

- Introduction

2.12 Priority

Sets the priority, the application is using.

Value	Meaning
Realtime	Specify this class for a process that has the highest possible priority. The threads of the process preempt the threads of all other processes, including operating system processes performing important tasks. For example, a real-time process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive.
High Priority	Specify this class for a process that performs time-critical tasks that must be executed immediately. The threads of the process preempt the threads of normal or idle priority class processes. An example is the Task List, which must respond quickly when called by the user, regardless of the load on the operating system. Use extreme care when using the high-priority class, because a high-priority class application can use nearly all available CPU time.
Normal	Specify this class for a process with no special scheduling needs.
Idle	Specify this class for a process whose threads run only when the system is idle. The threads of the process are preempted by the threads of any process running in a higher priority class. An example is a screen saver. The idle-priority class is inherited by child processes.

The priority is **not** stored in the parameter file ! Each time, the user starts the application, the priority will be set to the default value (should be normal).

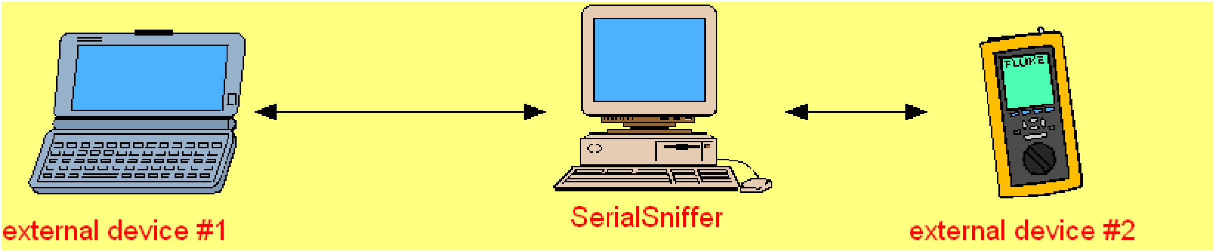
See also :

- Introduction

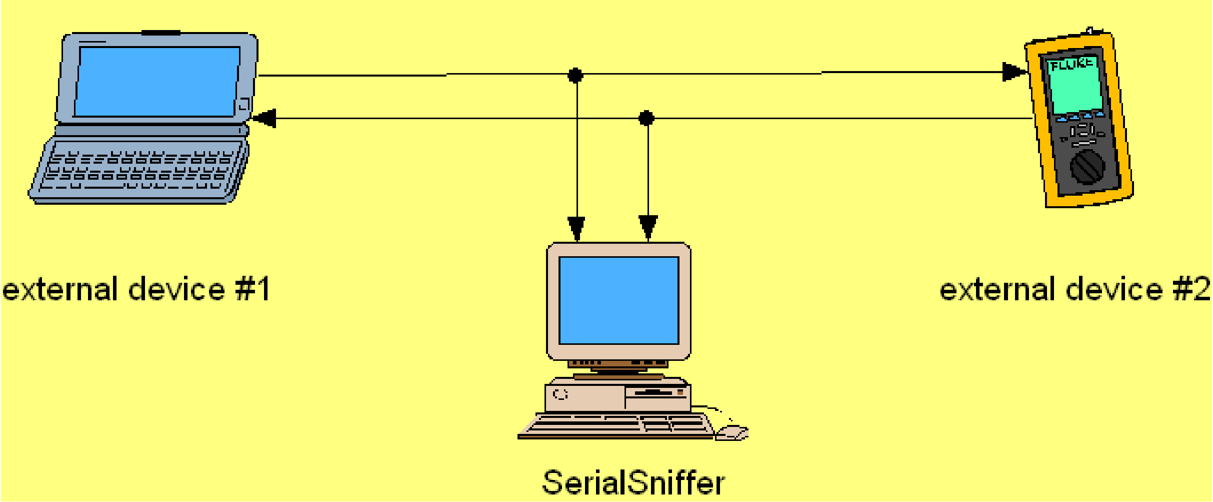
3 Accessories

3.1 Hardware Adapter

SerialSniffer is designed to record the data which is transferred on a serial link between two devices. The first possibility is to open up the serial link and connect SerialSniffer between the two devices :

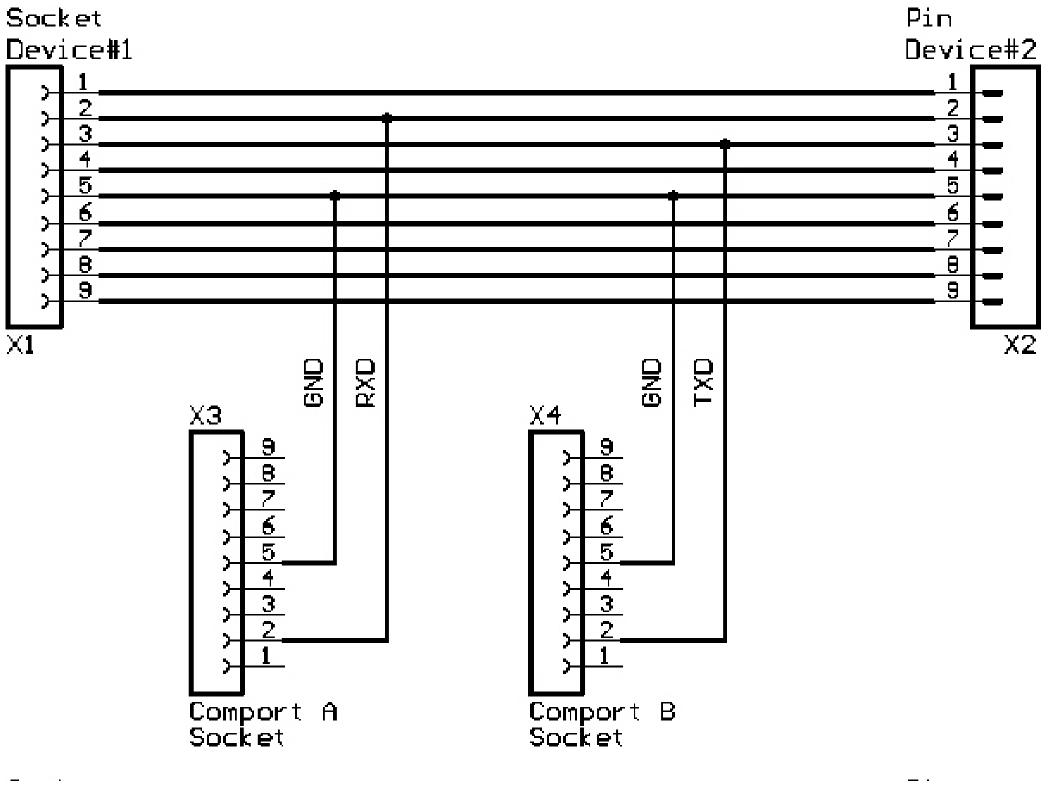


The second possibility is to connect SerialSniffer "parallel" to the existing serial link using a hardware adapter.
This solution prevents the serial connection from being influenced by SerialSniffer :



Using a hardware adapter may cause problems in displaying the correct order of the received data. This means, that it may be possible that two telegrams (one from each side to the other) are mixed up and that changes of the sending port are reported even in case that the second telegram is sent after the first telegram is fully received by the external device.
This is due to buffers, which are used internally by windows. The first choice should be to switch of the FIFO (by use of the device manager) and to reboot the PC. If it is possible, the user should try not to use a hardware adapter, which will force all data to go "through" SerialSniffer. This will ensure the correct order of the data.

The schematic diagram for a simple hardware adapter to be build by your own can be found below.



See also :

- Introduction
- Cabeling

3.2 V.24 tester

The V.24 tester is no longer available. Sorry.

Index