User Manual USB-COM
USB-COM PRO

Edition: June 2009

Tel: +49 40 528 401 0
Fax: +49 40 528 401 99
Web: www.visionsystems.de
Support: service@visionsystems.de

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

Disclaimer

Vision Systems reserves the right to make changes and improvements to its product without providing notice.

Vision Systems provides this document "as is", without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Vision Systems reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Vision Systems assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Contents

List of Figures

List of Tables

1 Overview

The VScom USB-COM are serial port adapters, which are connected to the USB port. The VScom USB-COM PRO series are industrial strength devices, providing flexible serial ports. The ports may operate in RS232, RS422 or RS485 mode. This is completely configured by software, no jumpers or DIP switches are required. Further the USB-COM PRO come in a metal case of IP30, and provide the option for DIN Rail mounting.

The supplied drivers for Windows operating system install the ports into the Windows API, so they appear as the usual Com ports. All applications capable to handle a serial port control the connected hardware the same way as they control the built-in ports Com1 and Com2. Similar drivers exist for other operating systems.

The driver allows to configure the serial ports for a bitrate of up to 3 Mbps. Of course RS232 can not reliably transport very high bitrates, even under best conditions 1 Mbit/s is the limit. As the result of the very high maximum speed, the USB-COM will generate many different bitrates in lower ranges. In fact every single bitrate from 185 bps up to 480000 bps is a possible transmission speed. And above this range most bitrates up to 1 Mbit/s are also possible, e.g. 500000 bps.

The interface to the host PC is USB 2.0, Full Speed or High Speed where required.

2 Introduction

This manual describes the hardware of VScom USB-COM PRO serial port adapters. Also the Windows driver for the USB-COM is described in detail.

2.1 Features

- USB 2.0
- Jumperless configuration
- Serial port interface RS232, RS422 or RS485, by software
- Termination and BIAS resistors for RS422 or RS485, by software
- Huge FIFO buffers 128 Byte
- Max 3.0 Mbit/s
- Drivers for Windows™ operating system
- Supported by Linux, Mac OS X, and Windows CE on several platforms.

2.2 Product Specifications

The VScom USB-COM PRO all have similar properties. Most important difference is the number of serial ports. The maximum speed is 3 Mbit/s, which may be used in RS422 or RS485 mode, RS232 is limited.

Host interface	USB 2.0	Full Speed or High Speed
	Connector	Type B
Serial Ports	1, 2, 4, 8	RS232 / RS422 / RS485
Signals	RS232	RxD, TxD, RTS, CTS, DTR, DSR, DCD, RI, GND
	RS422 / RS485	Tx±, Rx±, GND
	RS485	Data±
Max. Bitrates	3.0 Mbit/s	RS422 and RS485
	500 kbit/s	RS232
Serial configurations	Data bits	7, 8
	Parity	None, Odd, Even, Mark, Space
	Stop bits	1, 2
FIFO size	128 Bytes	At least, for Transmit and Receive each
LED	Data	Transmit and Receive
	Mode	3-Color LED
	Power	Red LED

Table 1: Features of VScom USB-COM PRO Series

Each of the serial ports may be used in 8 different operation modes as listed below. The Mode LED related to each port is light in a certain color to indicate the mode used.

Mode	Comment	LED
RS485 “4w” RTS	Full-Duplex, Transmitter controlled by RTS signal	Orange
RS485 “2w” RTS no Echo	Half-Duplex, Transmitter controlled by RTS signal, sent data is suppressed	Orange
RS485 “2w” RTS Echo	Half-Duplex, Transmitter controlled by RTS signal, sent data is received	Orange
RS485 “4w” auto	Full-Duplex, Transmitter activates automatic	Orange
RS485 “2w” auto no Echo	Half-Duplex, Transmitter activates automatic, sent data is suppressed	Orange
RS485 “2w” auto Echo	Half-Duplex, Transmitter activates automatic, sent data is received	Orange
RS422		Green
RS232		Red

Table 2: Serial Port Operation Modes

RS422 is a Full-Duplex mode by definition, RS485 may be used in Half- or Full-Duplex. Half-Duplex is also known as “2 wire” and denoted as “2w” here, while “4w” specifies the “4 wire” Full-Duplex mode. Since in Half-Duplex the wires to transmit and receive data are the same, it is possible to receive the data while they are sent out of the port. This option is named Echo, and it can be deselected.

The transmitter for Tx lines in RS485 mode may be controlled by the application via RTS signal, as well as by an internal automatic hardware circuitry.

Option	Result
BIAS	Polarize Tx+/Rx+ to +5V, Tx−/Rx− to GND
TxTerm	120Ω resistor from Tx+ to Tx−
RxTerm	120Ω resistor from Rx+ to Rx−

Figure 1: Serial Port Hardware Options

The modes of RS422 and RS485 allow to configure BIASing of the data lines, as well as connecting 120Ω termination resistors between the positive and negative wire. These three options can be selected per port, they are disabled if the port is in RS232 mode.

2.2.1 VScom USB-COM PRO

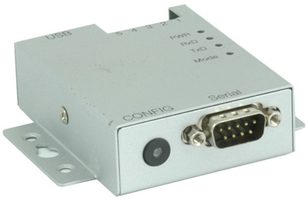


Figure 2: VScom USB-COM PRO

Serial Ports	1	
Connectors	1×DB9 male	RS232, RS422, RS485 (6.1)
	1×Terminal block	RS422, RS485 (6.3)
Power	Bus powered	
FIFO size	128 Bytes	Transmit
	256 Bytes	Receive
LED	Power	On Top: Red LED
	2×Data	On Top: Transmit Green, Receive Yellow
	1×Mode	On Top: 3-Color

Table 3: Features of VScom USB-COM PRO Single Port

2.2.2 VScom USB-2COM PRO

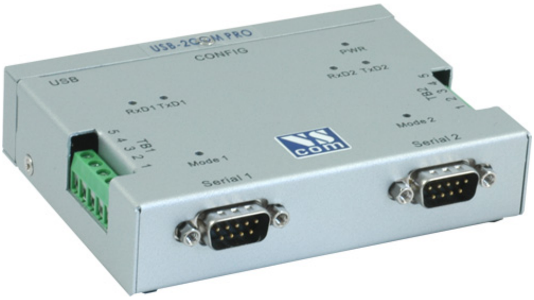


Figure 3: VScom USB-2COM PRO

Serial Ports	2	
Connectors	2×DB9 male	RS232, RS422, RS485 (6.1)
	2×Terminal block	RS422, RS485 (6.3)
Power	Bus powered	
FIFO size	128 Bytes	Transmit
	384 Bytes	Receive
LED	Power	On Top: Red LED
	4×Data	On Top: Transmit Green, Receive Yellow per port
	2×Mode	On Top: 3-Color

Table 4: Features of VScom USB-2COM PRO Dual Port

2.2.3 VScom USB-4COM PRO

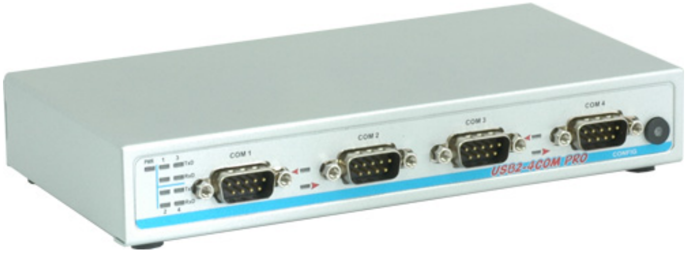


Figure 4: VScom USB-4COM PRO

Serial Ports	4	
Connectors	4×DB9 male	RS232, RS422, RS485 (6.1)
Power	Self powered	Shipped with 5V DC adapter
FIFO size	128 Bytes	Transmit
	384 Bytes	Receive
LED	Power	On Front: Red LED
	8×Data	In Block on Front: Transmit Green, Receive Yellow
	4×Mode	Near Port: 3-Color

Table 5: Features of VScom USB-4COM PRO Quad Port

2.2.4 VScom USB-8COM PRO

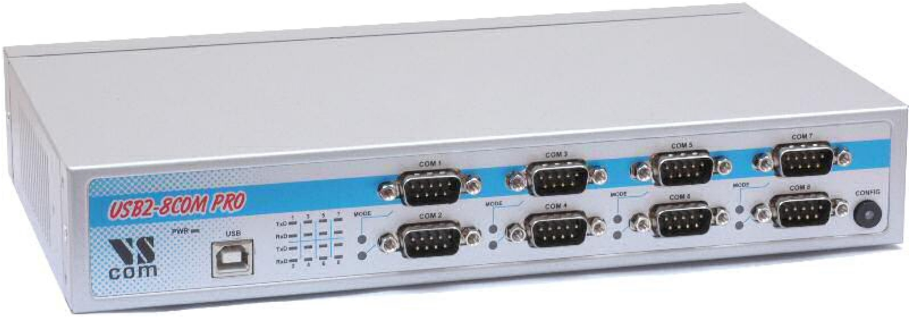


Figure 5: VScom USB-8COM PRO

Serial Ports	8	
Connectors	8×DB9 male	RS232, RS422, RS485 (6.1)
Power	Self powered	110V - 240V AC, 48 - 67 Hz
FIFO size	128 Bytes	Transmit
	384 Bytes	Receive
LED	Power	On Front: Red LED
	16×Data	In Block on Front: Transmit Green, Receive Yellow
	8×Mode	Near Port: 3-Color

Table 6: Features of VScom USB-8COM PRO Octal Port

2.3 Packing List

All models are shipped with a driver CD-ROM, which includes the documentation and the driver software.

- USB 2.0 High Speed cable
- CD-ROM with drivers, configuration software and manuals
- Metal plates for Wall mounting or DIN Rail clips
- Power supply hardware if required

2.4 About this Manual

This manual describes the hardware of VScom USB-COM PROs, as well as signal assignments of the serial connectors. The screen shots in here are made on an English language version of Windows XP. It is not difficult to find the appropriate part in any other language version of Windows. The current version of the driver is 2.04.16, the version of the firmware in VScom USB-COM PRO is 1.0.0.

Usually any hardware configuration is described before the software, including drivers. However the configuration of VScom USB-COM PRO requires the serial port drivers installed, so these are covered first.

3 The Drivers for Windows

This chapter of the manual documents the drivers for Windows Operating Systems. The driver is the same for all USB-COM available, independent of the model. As one consequence there is only one driver for download, linked to from all the products. It does not matter which link you use. The USB-COM appear in Windows as several components, depending on the specific model. This can be seen in the Device Manager, while or after installation of the drivers.

- Single port models appear as a “USB FAST SERIAL ADAPTER”, which finally installs as a “USB Serial Converter” plus a “USB Serial Port”.
- Dual port models appear as a “USB Composite Device” first, which installs USB Serial Converter A and B. These then install the serial ports.
- Some quad port models also appear as a “USB Composite Device” first, which installs USB Serial Converter A to D. These four converters then install the serial ports.
- Multiport models with 4 to 16 serial ports first appear as one or two “Generic USB Hub”, plus some USB Composite Devices. These install USB Serial Converter A and B or A to D each. The converters then install all the serial ports.

Windows already has a driver for a USB Hub, and it also handles USB Composite Devices without configuration and help by the user. The specific drivers are for the USB Serial Adapters, and the USB Serial Ports installed by them.

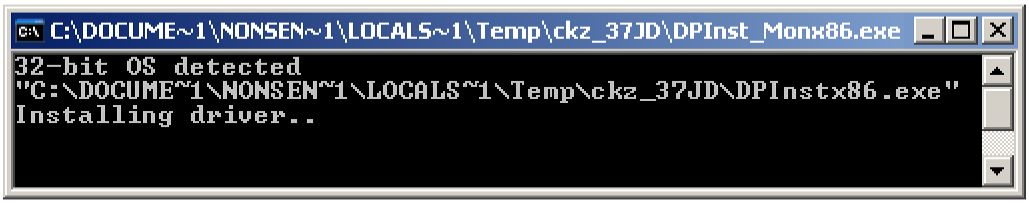
The manual first describes the process of installing the driver software, followed by instructions for removing the drivers in the unlikely case you wish to do this. After this the configuration of the serial ports is described. There is no option to configure the other drivers installed for the USB-COM, so the options are the same for all different models.

Many USB-COM may be installed simultaneously. In fact the driver does not limit the number of USB-COM installed, Windows does. In Windows the maximum number of serial ports installed is 256, no matter which driver installed and controls them. The current version of the driver is 2.04.16.

With the current driver version there are two methods of installation. There is the classical variant, where Windows detects the new hardware as soon as it is connected to the PC. Then Windows requests the driver from the user, and installs them. The modern way of installation uses a Pre-Installation Executable to put the drivers onto the system, when the hardware is not yet connected to the PC. When it is connected later, Windows already knows about the drivers, and installs them without further interaction by the user.

3.1 Install Drivers by Pre-Installation Executable

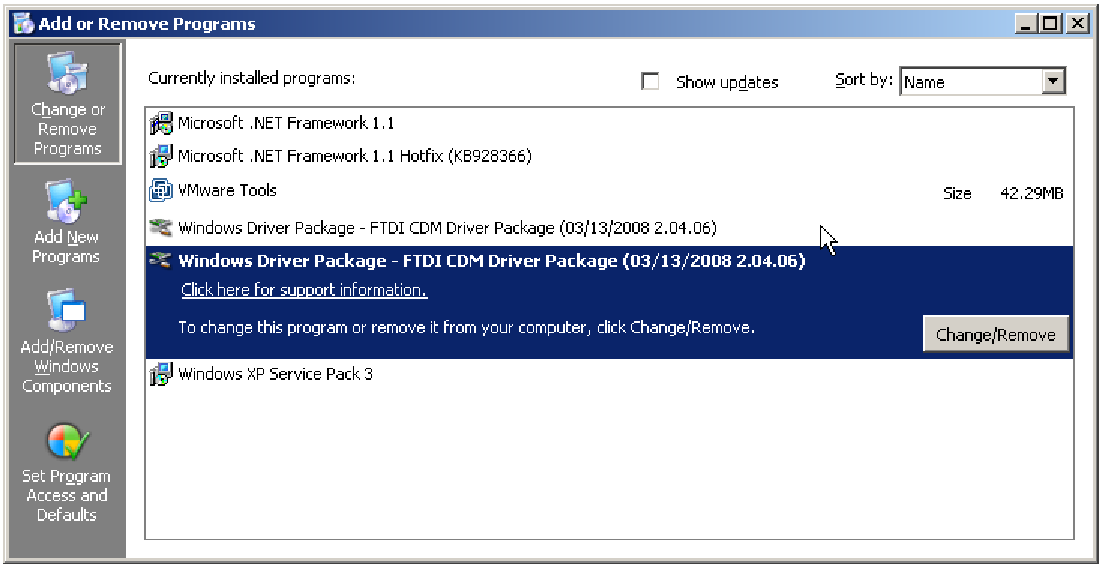
Do not connect the USB-COM to your PC. The driver installation program is usually named as USBCom-CDM_20416.exe, depending on the version number. This file is found on the driver CD-ROM, in the top folder. Just log on to Windows with Administrator privileges, and start the program. It is a self-extracting archive, which in turn executes the included installation software. The installation is a text-mode Console Application, which pretty much looks like a DOS box (see figure 6).


C:\DOCUME~1\NONSEN~1\LOCALS~1\Temp\ckz_37JD\DPInst_Monx86.exe

```
32-bit OS detected
"C:\DOCUME~1\NONSEN~1\LOCALS~1\Temp\ckz_37JD\DPInstx86.exe"
Installing driver..
```

Figure 6: Starting Pre-Installation

When the operation is finished, this box closes. No interaction is required. The result of this process is a Windows system which has the USB-COM added as know hardware. This installation process also added an entry in "Control Panel" (actually two entries), in the "Add or Remove Programs" applet. The drivers may be uninstalled by this entry.


Add or Remove Programs
Change or Remove Programs
Add New Programs
Add/Remove Windows Components
Set Program Access and Defaults
Currently installed programs:
☐ Show updates
Sort by: Name
Change/Remove

Microsoft .NET Framework 1.1
Microsoft .NET Framework 1.1 Hotfix (KB928366)
VMware Tools — Size 42.29MB
Windows Driver Package - FTDI CDM Driver Package (03/13/2008 2.04.06)
Windows Driver Package - FTDI CDM Driver Package (03/13/2008 2.04.06)
Click here for support information.
To change this program or remove it from your computer, click Change/Remove.
Windows XP Service Pack 3

Figure 7: Uninstall Option

Now connect your USB-COM to a USB port on your PC. Windows will detect the new hardware, and attempts to install the drivers. Probably it offers the option to search the Internet for more recent drivers, just say "No, not this time" to this question. In the next steps select the Automatic option, and click on Next and Finish as requested. Windows knows about the drivers, and installs all components and serial ports without further interaction.

3.1.1 Check Installation of the Drivers

Open the Device Manager, and see if all serial ports are available in the "Ports" Device class. Open the properties of each serial port. Open "Control Panel", and the "Add or Remove Programs" applet. The FTDI CDM Driver Package appears twice in the list of components.

3.1.2 Uninstall Drivers from Pre-Installation Executable

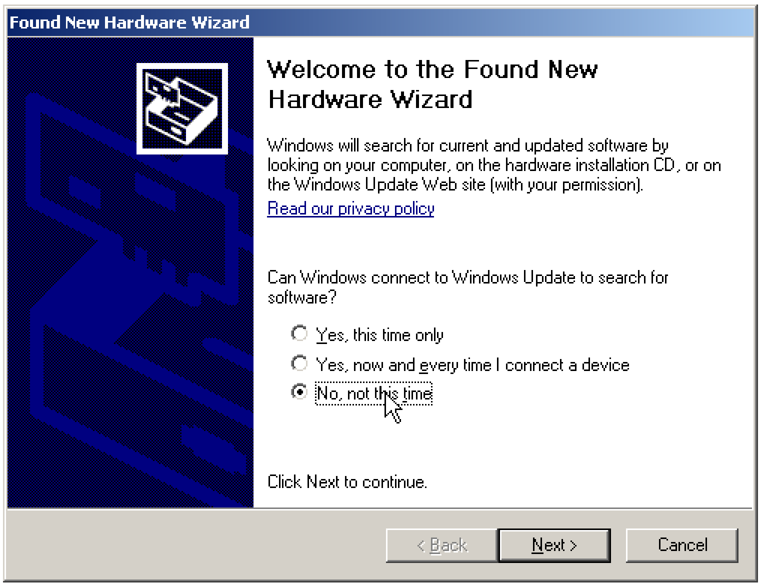
First disconnect all USB-COM from your system. Then open "Control Panel", and in that start the "Add or Remove Programs" applet. The FTDI CDM Driver Package appears twice in the list of components. On both entries, click the Change/Remove button to remove the software from your system. Once both entries are done, the drivers no longer exist.

This procedure not only removes all entries of USB-COM from the Device Manager, as the final step also the files are removed from the hard disk. As a result reconnecting a USB-COM causes Windows to start the “New Hardware” Wizard again.

3.2 Install Drivers the Classical Way

Start Windows, and log on to the system with administrative privileges. The connect your USB-COM PRO to a free USB port of your computer. Windows will detect the new hardware, and identify it as an unknown type. Of course “unknown” only if this is the first time drivers are installed to the system. The second USB-COM PRO is recognized automatically, without further interaction.

First Windows may detect some of the standard components mentioned above (3). The Windows finds the “USB FAST SERIAL ADAPTER”, and requests a driver for this.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only

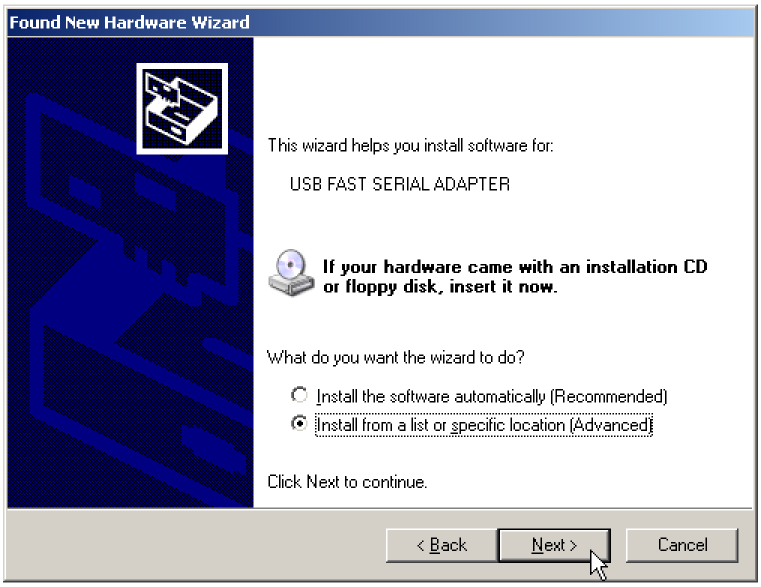
○ Yes, now and every time I connect a device

● No, not this time

Click Next to continue.

Figure 8: Wizard found new hardware

The “Found New Hardware Wizard” first offers to check on the Internet for latest drivers. Since the latest drivers are already on the CD-ROM or downloaded and stored on your hard disk, select the “No, not this time” option, and click on the Next button.



Found New Hardware Wizard

This wizard helps you install software for:

USB FAST SERIAL ADAPTER

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

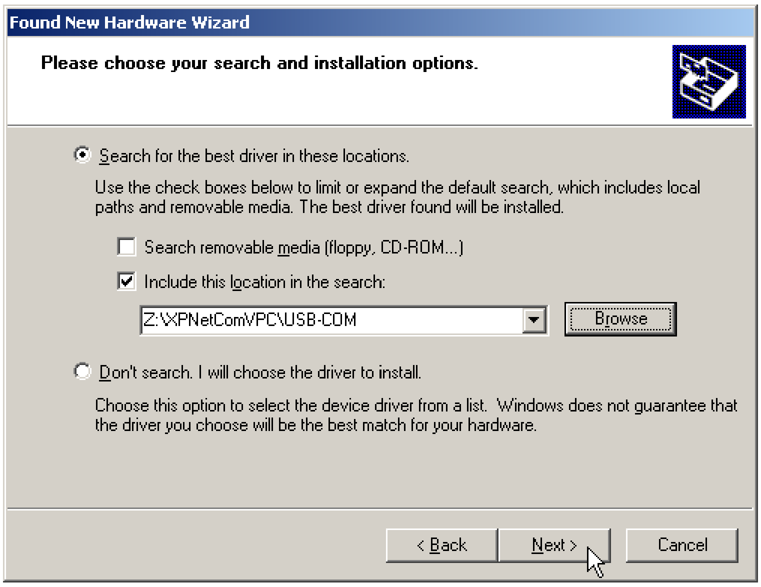
○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.

Figure 9: Help the Wizard

Usually Windows can not locate the best software automatically. Even if the driver is saved on a CD-ROM, the automatic option uses a long time to search. So instead select the "(Advanced)" option, and click on Next.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

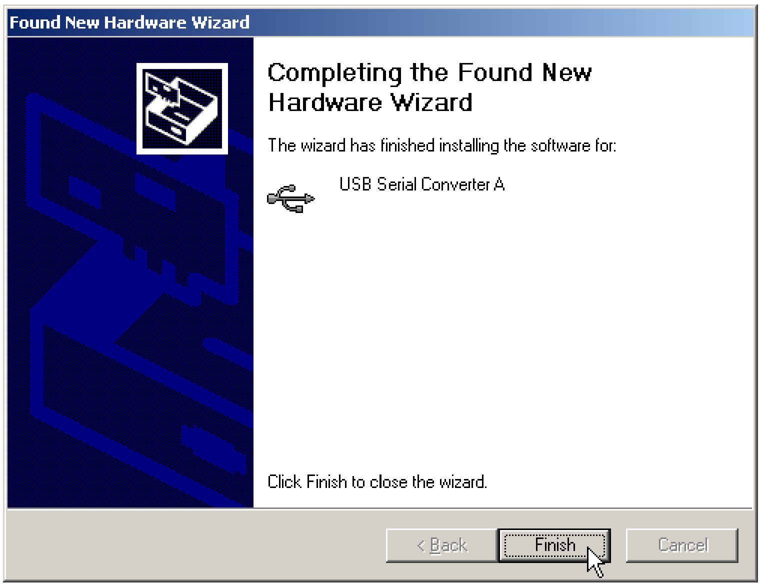
Z:\XPNetComVPC\USB-COM

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Figure 10: Select appropriate driver

Guide Windows to the directory where you saved the latest version or to the CD-ROM, and again click on Next.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

USB Serial Converter A

Click Finish to close the wizard.

Figure 11: Driver installed

Windows copies the drivers to the system directory, and loads them. After that the installation is done, so click on the Finish button.

But this was the installation of the "USB FAST SERIAL ADAPTER", not the serial ports themselves. These are also detected as new hardware, and the process repeats for these drivers. Follow the steps above a second time, and then you are done. Now Windows knows the required drives for all components, and installs them without further interaction.

3.2.1 Check Installation of the Drivers

Open the Device Manager, and see if all serial ports are available in the "Ports" Device class. Open the properties of each serial port.

3.2.2 Uninstall Drivers from Classical Way

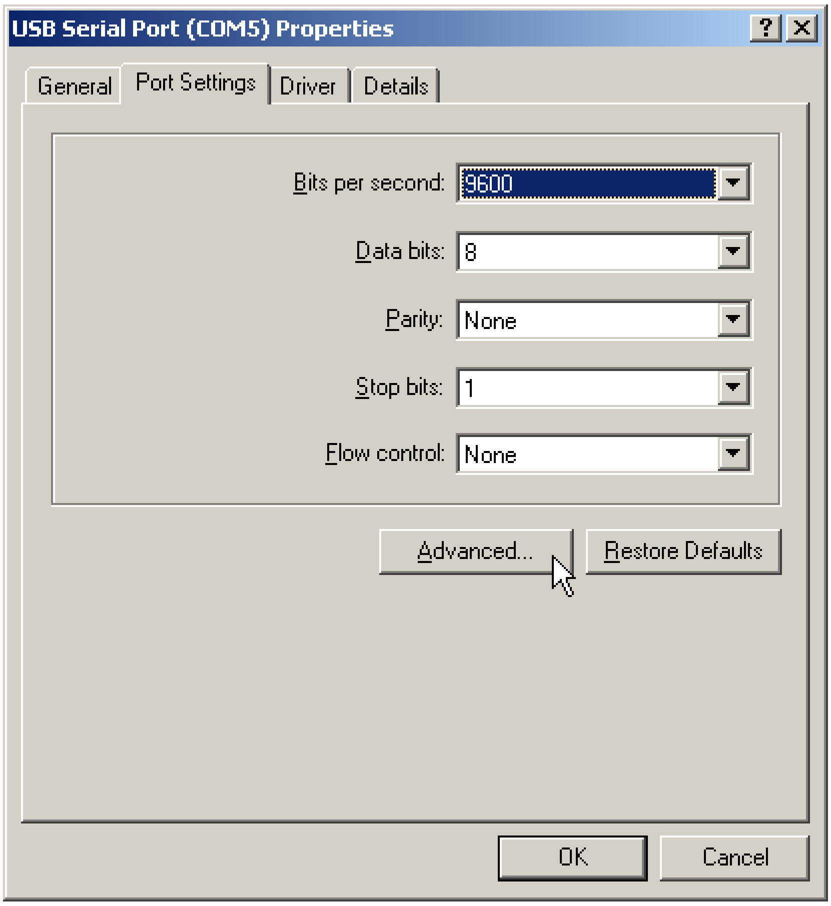
First disconnect all USB-COM from your system. On the CD-ROM or in the downloaded files find the folder named "FTClean". Inside this you'll find the program FTClean.exe, which is designed to remove all USB-COM entries in the system as well as all related files from the hard disk. Just start the software, and follow the instructions on the screen.

This method of driver removal is also valid for uninstalling the drivers, when they have been installed by the Pre-Installation Executable software.

3.3 Configure the Serial Ports

Open the Device Manager. In the Device Manager open the device class of "Ports (COM & LPT)"[1]. In this class you'll see all the entries of "USB Serial Port". Now open the properties of the serial port to configure.

[1] "Anschlüsse (COM und LPT)" in einem deutschen Windows



USB Serial Port (COM5) Properties

Bits per second: 9600

Data bits: 8

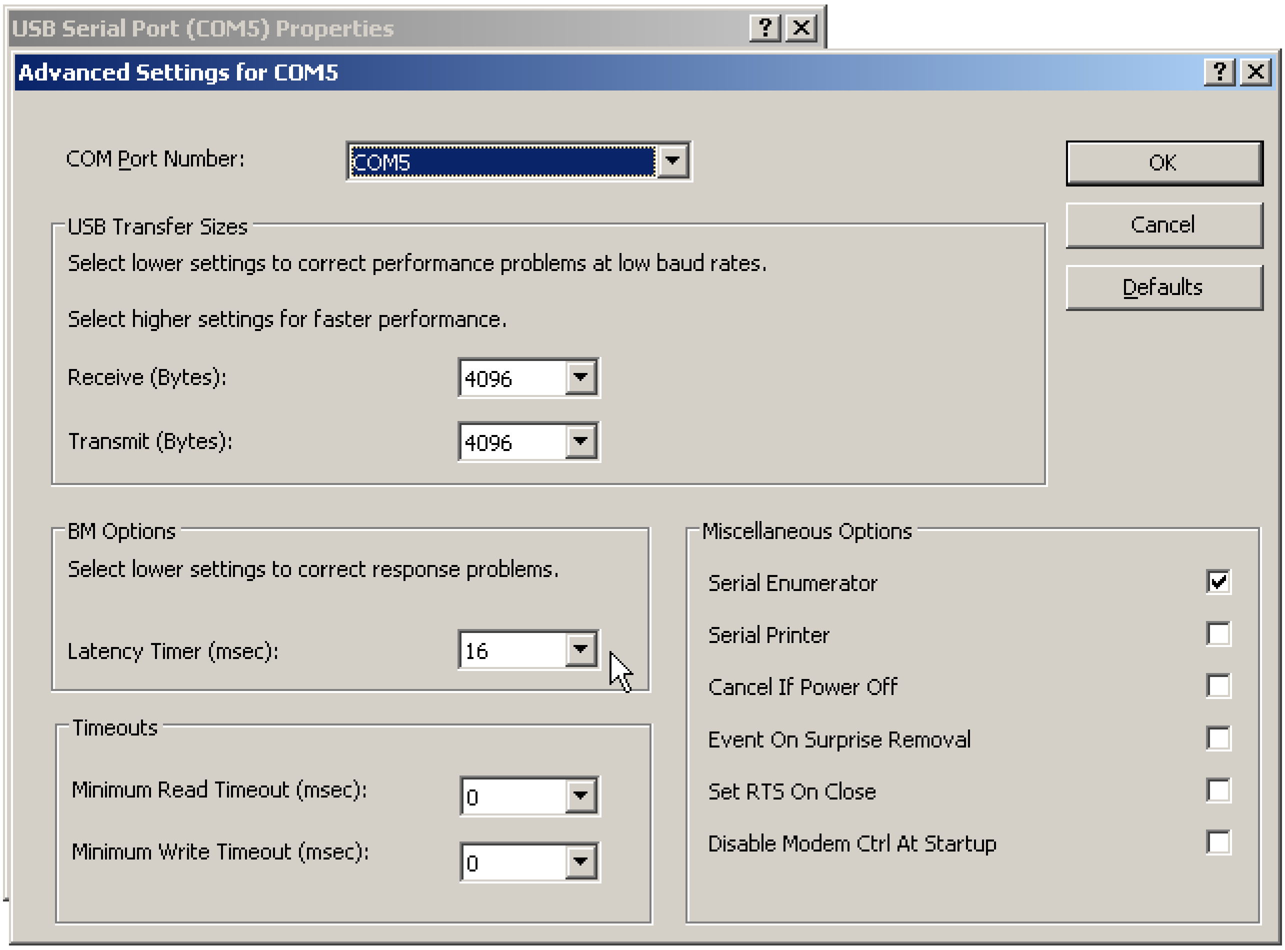
Parity: None

Stop bits: 1

Flow control: None

Figure 12: USB Serial Port Properties

Click on the button named "Advanced..." to see the special configuration options. They are available via panel 13.



USB Serial Port (COM5) Properties

Advanced Settings for COM5

COM Port Number: COM5

USB Transfer Sizes

Select lower settings to correct performance problems at low baud rates.

Select higher settings for faster performance.

Receive (Bytes): 4096

Transmit (Bytes): 4096

BM Options

Select lower settings to correct response problems.

Latency Timer (msec): 16

Timeouts

Minimum Read Timeout (msec): 0

Minimum Write Timeout (msec): 0

Miscellaneous Options

Serial Enumerator ☑

Serial Printer □

Cancel If Power Off □

Event On Surprise Removal □

Set RTS On Close □

Disable Modem Ctrl At Startup □

Figure 13: Advanced port options

3.3.1 Rename the Serial Port

The option “COM Port Number” on top of the advanced settings allows to rename the port. You may select any free port number. It is Windows which refuses a change, if the new port name is already occupied.

3.3.2 Miscellaneous Options

In the lower right corner of the Advance Options (figure 13) are some seldom changed options.

3.3.2.1 Enable faster Responses In the middle of the left part is the section named “BM Options”. The parameter “Latency Timer” can be reduced to 1 millisecond, if there are problems with certain applications. The default value is 16, to reduce the USB protocol overhead and bandwidth usage. Usually the serial port buffers incoming data for up to 16 ms, or until 62 bytes of data are received. When either of the criteria becomes true, the data is sent to the Host PC. Then the application software will recognize the received data. If a small amount of data is received, or the last part of the transmission does not trigger the 62 byte limit, this causes a delay in receiving the data. As the consequence the application can not operate on the data very early.

Usually this is not a problem, but with certain serial protocols this can cause strange effects. So a reduction to 1 ms is appropriate then, faster reactions are not possible. This is caused by the protocol structure of USB, not by the serial port.

3.3.2.2 Serial Enumerator In the lower right corner of the advanced options is a set of options for different purposes.

This option is enabled by default. When the driver is started, Windows tries to detect hardware attached to the serial port. There may be a mouse or a modem. Most often the attached device is neither of that, so it is not a problem to disable this option. Further some devices permanently send data to the Host, without special requests. Windows may falsely recognize such devices as a serial mouse. This causes your pointer to randomly jump over the screen. Disabling the option prevents this.

3.3.2.3 Serial Printer When flow control like RTS/CTS stops data from being sent to the serial port for a sufficient time, this will cause an error event for the application. However if the attached serial device is a printer, sometimes this will halt for paper feed, especially for manual interaction. This will cause the data flow to stop for a significant long time.

Enabling this option causes such errors never to be reported to the application. So printing will never cause erroneous error messages.

3.3.2.4 Cancel if Power Off Instead of a regular shut down the computer can be sent into sleep or hibernate mode. In this situation neither the serial port nor the driver can perform any requests. In rare situations this can cause problems.

Enabling this option dumps any open requests for driver or hardware, if the computer is powered off by any Suspend state.

3.3.2.5 Event On Surprise Removal Usually this option is disabled because software does not use it. When it is enabled, application software can request to be notified if the hardware of the serial port is removed from the system while it is in use. So instead of claiming a malfunction of the attached serial device it can correctly report the removal of the port.

3.3.2.6 Set RTS On Close Usually the signal RTS is disabled when the port gets closed. However some kind of serial hardware may require a permanently active RTS for correct operation. Otherwise there may be error messages when the port is opened again to use the device.

Enabling this option causes the RTS to stay active even when the application closes the serial port.

3.3.2.7 Disable Modem Ctrl At Startup Usually on startup the modem control signals RTS and DTR follow the behavior of standard ports (i.e. Com1). Due to the longer timing compared to built-in serial ports a very short enable or disable pulse on the control signals may become a comparably long pulse on the USB serial port. Such a long pulse can cause external hardware to malfunction. By correct configuration of the serial port application software can avoid that problem. However since it does not appear on Com1, most software does not care about that. This option will help to "heal" such problem.

Note that if the "Serial Enumerator" option is enabled, in the enumeration sequence Windows causes the control signals to change. So if it is necessary to select "Disable Modem Ctrl At Startup", it is likely that "Serial Enumerator" should be disabled.

3.4 Optimizing Data Throughput

Roughly speaking, the USB allows one operation between Host PC and attached USB device per millisecond. This is the basic reason why the Latency (see 3.3.2.1) can't be shorter than 1 ms. But there is a second effect causing slow transmissions.

Many software sends the data to the driver byte-by-byte. This is not a problem with built-in ports like Com1. Except for extreme high transmission rates the operation time of the driver is much shorter than the serial transmission time. So when the second and following byte are sent to the driver, the previous bytes did not completely leave the PC and its buffers. The serial port does not run out of data.

This is different with USB serial ports. Now matter how fast the byte is transmitted over the USB cable (12 Mbit/s or 48Mbit/s), it takes 1 ms to send the next byte. Calculated the other way this is 1000 byte per second, which is equivalent to 9600 bit/s. The user feels the transmission as sluggish. Raising the serial speed does not help, since this is not the problem.

However it requires nearly the same millisecond to send one byte or more. So if the application sends complete buffers instead byte-by-byte, the driver can send more than one byte per millisecond. This way the USB serial port is permanently fed with work to send.

4 Hardware Configuration

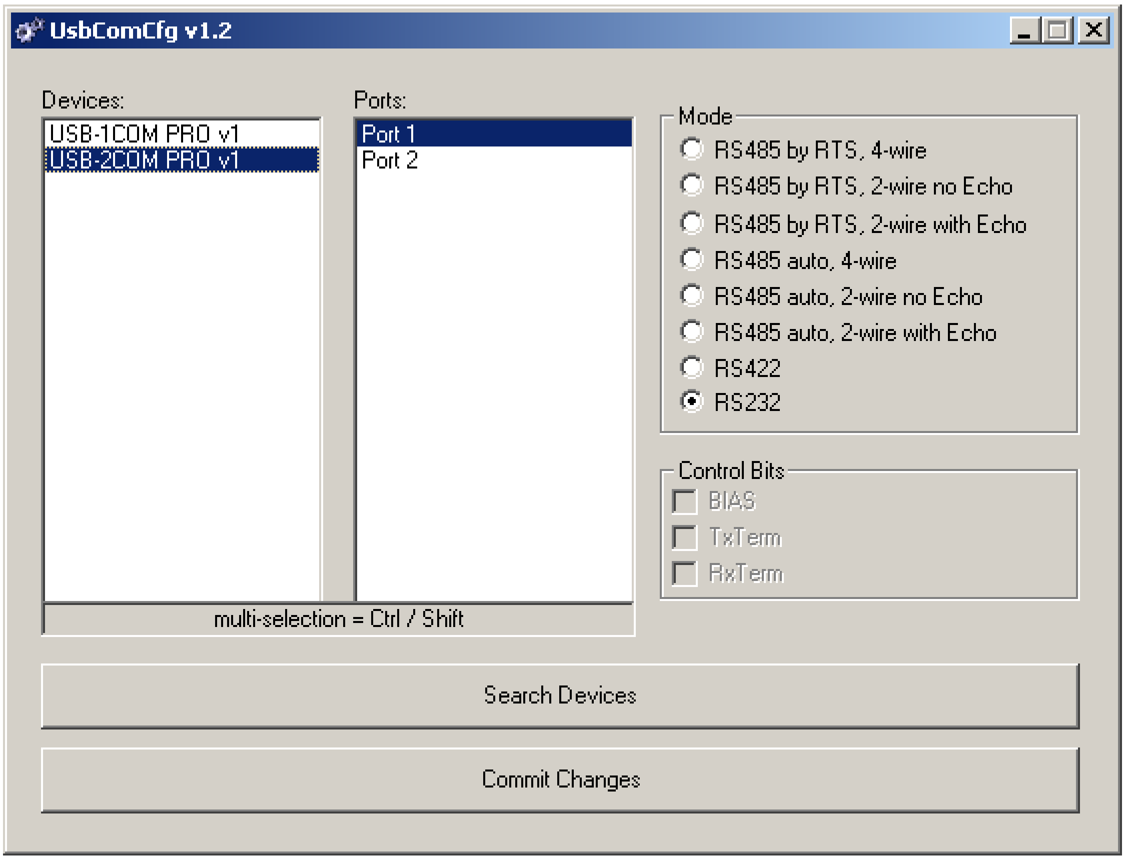
The hardware of VScom USB-COM PRO is not configured via Jumper blocks or DIP switches, as is mentioned above. Instead this is done by software. Each model has a single button named as “Config”. User press this button for at least 3 seconds, a blue LED inside the button will light. This shows the device is in configuration mode now. In configuration mode the serial port number 1 is disabled, the configuration is done via the Com Port associated with this serial port. Pressing the “Config” button again returns to normal operation, thus exiting from configuration mode.

This way of software configuration of course requires to have a driver for USB Virtual Serial Ports on the system. That is not a problem, because without such driver the VScom USB-COM PRO can not be used anyway. The configuration parameters are saved to a non-volatile memory inside the USB-COM PRO, so configuration and use may happen on different systems.

4.1 Configuration via Windows Software

If the drivers for Windows are not installed, please do that now (chapter 3). Then locate the program UsbComCfg.exe on your CD-ROM, or in the files downloaded from the FTP server. Close any application currently operating via the serial port(s) provided by your USB-COM PRO. During configuration the first (or only) serial port is disconnected from the external devices.

On your USB-COM PRO press the button named “Config” for at least 3 seconds, a LED in the button will light in blue color. The USB-COM PRO is in configuration mode now. Then start the UsbComCfg program, it searches and detects all USB-COM PRO in configuration mode.



Devices:
- USB-1COM PRO v1
- USB-2COM PRO v1

Ports:
- Port 1
- Port 2

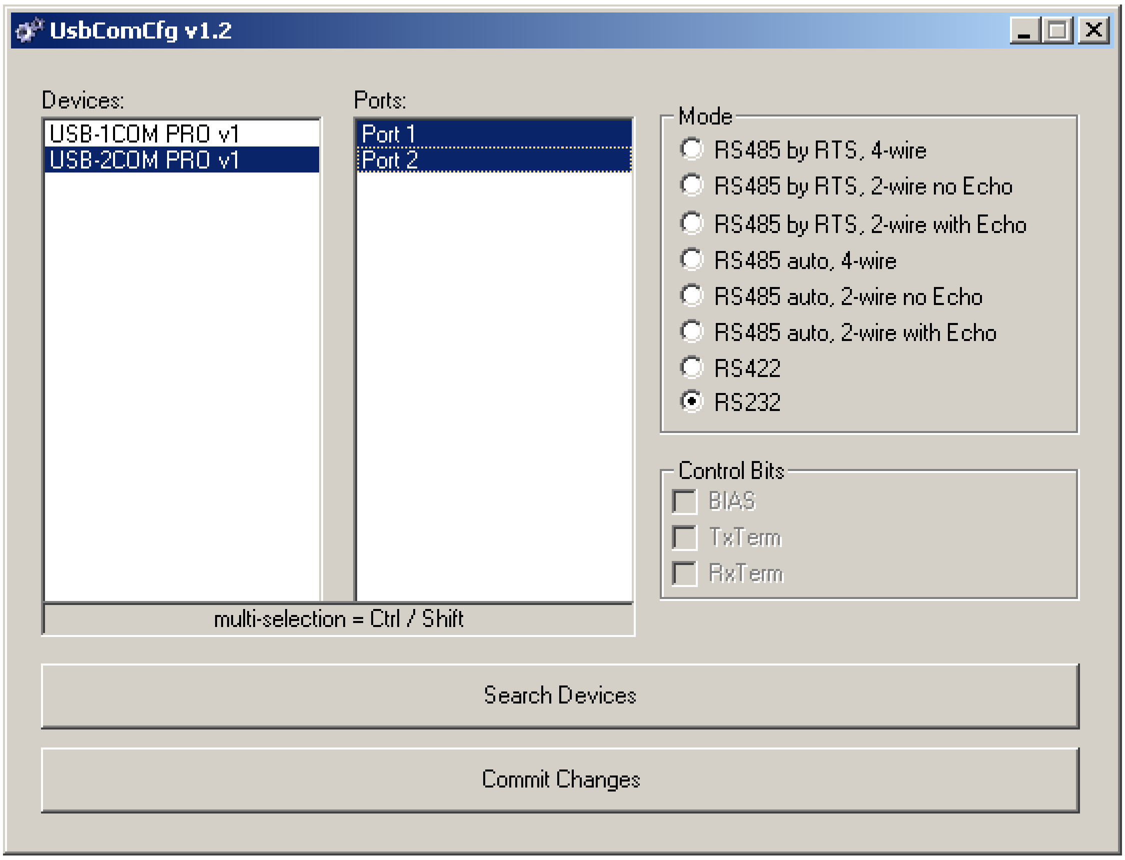
Mode
- ○ RS485 by RTS, 4-wire
- ○ RS485 by RTS, 2-wire no Echo
- ○ RS485 by RTS, 2-wire with Echo
- ○ RS485 auto, 4-wire
- ○ RS485 auto, 2-wire no Echo
- ○ RS485 auto, 2-wire with Echo
- ○ RS422
- ● RS232

Control Bits
- □ BIAS
- □ TxTerm
- □ RxTerm

Figure 14: UsbComCfg started

If a device is not in Configuration mode, it is not detected. It is also not possible to find a device, when the first serial port is used by an application software. UsbComCfg repeats the search when the appropriate button is clicked.

By clicking a device in the left pane, a port in the middle pane can be selected. The operation mode is then changed by the radio buttons and check boxes in the right pane. The button to “Commit Changes” transfers the required parameters to the selected device. They are stored in non-volatile RAM, and activated after that. The device remains in Configuration mode.



UsbComCfg v1.2

Devices:
- USB-1COM PRO v1
- USB-2COM PRO v1 (selected)

Ports:
- Port 1 (selected)
- Port 2 (selected)

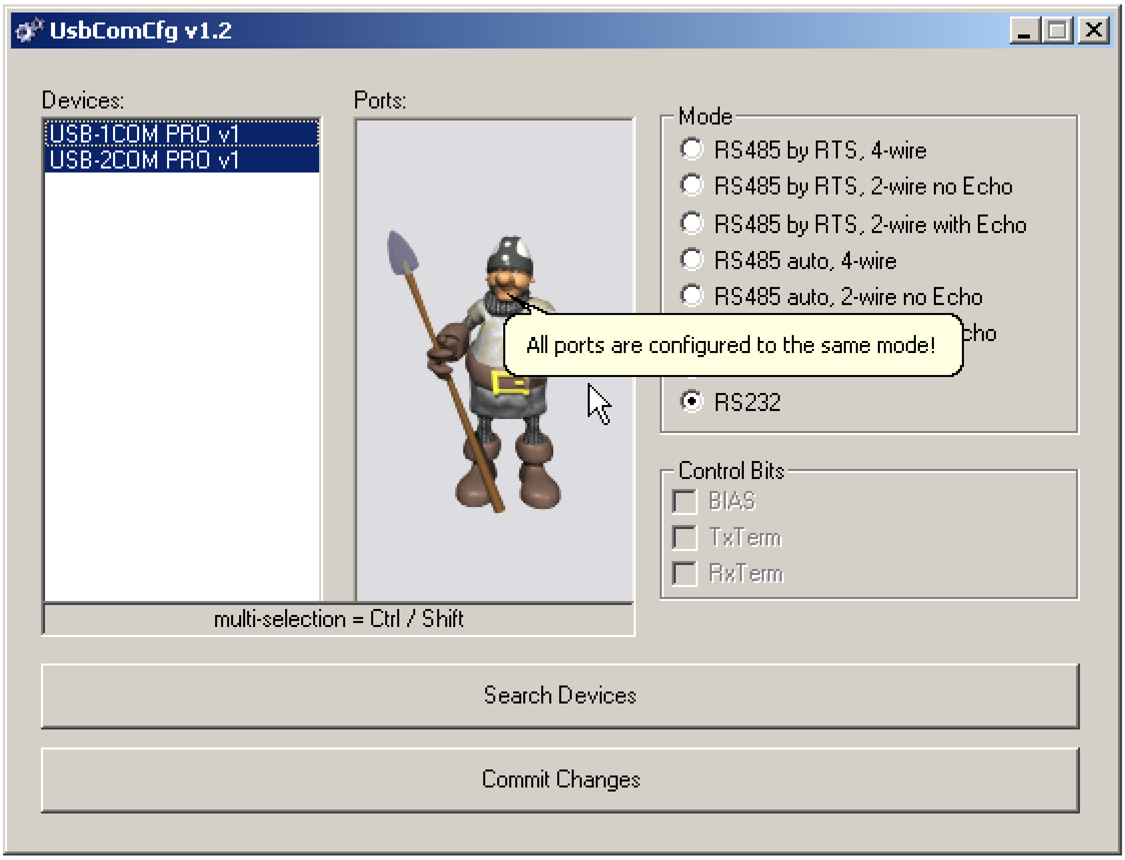
multi-selection = Ctrl / Shift

Mode
- ○ RS485 by RTS, 4-wire
- ○ RS485 by RTS, 2-wire no Echo
- ○ RS485 by RTS, 2-wire with Echo
- ○ RS485 auto, 4-wire
- ○ RS485 auto, 2-wire no Echo
- ○ RS485 auto, 2-wire with Echo
- ○ RS422
- ● RS232

Control Bits
- □ BIAS
- □ TxTerm
- □ RxTerm

Figure 15: UsbComCfg Multi-Port Selection

You may select several ports of one device at the same time. This is done by the usual Windows method, i.e. clicking with Shift- or Ctrl-Key pressed. The selected configuration then applies to all selected ports, “Commit Changes” configures all of them at the same time.



UsbComCfg v1.2

Devices:
- USB-1COM PRO v1
- USB-2COM PRO v1

Ports:

Mode
- ○ RS485 by RTS, 4-wire
- ○ RS485 by RTS, 2-wire no Echo
- ○ RS485 by RTS, 2-wire with Echo
- ○ RS485 auto, 4-wire
- ○ RS485 auto, 2-wire no Echo
- cho
- ● RS232

Control Bits
- □ BIAS
- □ TxTerm
- □ RxTerm

All ports are configured to the same mode!

Figure 16: UsbComCfg Multi-Device Selection

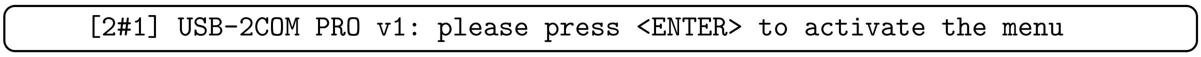
Finally you may select multiple devices at the same time. The configuration then applies to all ports of all selected devices. Watching the LEDs on each device you can see how the configuration proceeds from device to device.

When all USB-COM PRO are configured, they are still in Configuration mode, shown by the blue LED. Shortly press the “Config” button put the devices back to normal operation. The LED in the button will go dark.

4.2 Configuration via Terminal Program

Not all systems operate under Windows, and not all users have the UsbComCfg program available when they want to reconfigure the USB-COM PRO. Even then the configuration is possible via the virtual serial port implemented by the driver. If no driver is available on the system, the USB-COM PRO will be of little use anyway. If the driver is not installed, please do that now (chapter 3 for Windows).

Then open your terminal emulation software, and open the first (only) serial port provided by the USB-COM PRO. Configure the serial port for 38400 bit per second, eight bit per character without parity, and one stop bit (38400,8N1). Flow control is not used. Select the terminal emulation as a very dumb terminal. Incoming <CR> (Carriage Return characters) shall not result in a line feed, but <BS> (Backspace) should position the cursor one space to the left.

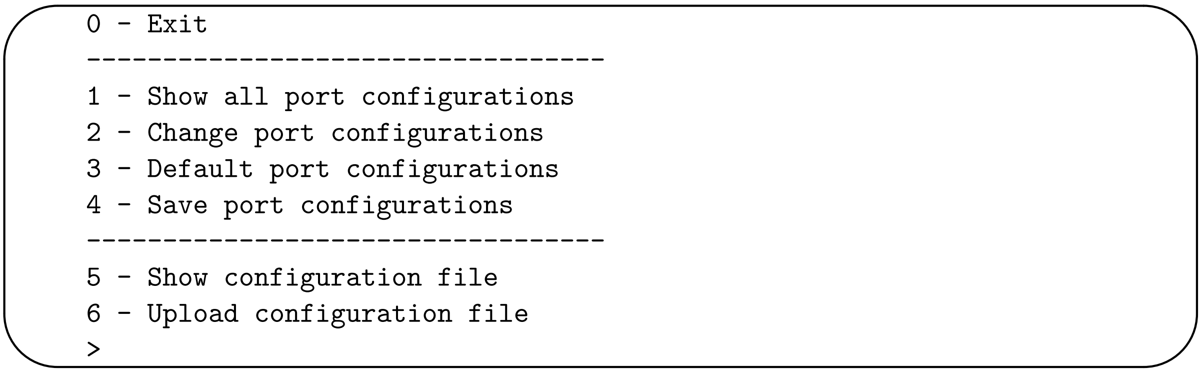


```
[2#1] USB-2COM PRO v1: please press <ENTER> to activate the menu
```

Figure 17: Open Terminal Configuration

Now press the “Config” button to enter the configuration mode. In your terminal screen a line similar to the above will appear. If you carefully watch the software, you may notice the line is written again and again. If your terminal application translates a <CR> to a <CR><LF> sequence, it adds a Line Feed character to each line. So you’ll see the text quickly scroll in your Windows. This is intentional, the Configuration mode redraws the screen, so you have a chance to change the parameters of your terminal application. The line lists the USB-COM PRO found using this serial port.

This redraw happens until you press the <Enter> key. The firmware in USB-COM PRO now recognizes interactive input, so a redraw is no longer required. If you need a redraw later, just press the <Enter> key.



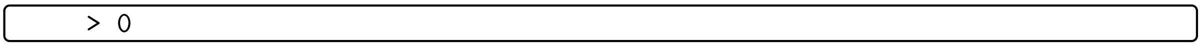
```
0 - Exit
----------------------------------
1 - Show all port configurations
2 - Change port configurations
3 - Default port configurations
4 - Save port configurations
----------------------------------
5 - Show configuration file
6 - Upload configuration file
>
```

Figure 18: Terminal Main Menu

The Main Menu appears. In this and the other menus just type the character given at the beginning of the line, to get the function described there. Hit the Enter key to activate the command. The optional result as well as the menu will be written on the screen. Just test this by hitting ’0’ and Enter to exit from the configuration. The blue Config LED will go off.

In the Main Menu (figure 18) the upper section provides for status information and manual configuration of the serial ports. The lower section provides a method to save the configuration, especially for a later upload of these parameters.

4.2.1 Exit the Configuration (0)



```
> 0
```

Figure 19: Exit Terminal Configuration

Enter a ’0’ in the Main Menu, this will terminate the Configuration. The blue LED in the “Config” button will be off, unsaved changes to the configuration are discarded. The serial port is disconnected from the Configuration hardware, and re-attached to the serial device.

4.2.2 Show Port Configurations (1)

```
> 1
Port 1: RS232
Port 2: RS232
```

Figure 20: Show all port configurations

The command '1' in the Main Menu lists the current configuration of all serial ports in the USB-COM PRO. This example is for a USB-2COM PRO with both ports in RS232 mode. The list is followed by the Main Menu again.

4.2.3 Change Port Configuration (2)

```
> 2
 0 - Back
---------------
 1 - Port 1
 2 - Port 2
---------------
 3 - All Ports
>
```

Figure 21: Change port configurations

Command '2' allows to change the configuration of some ports. The ports may be selected individually, or all at once for common configuration.

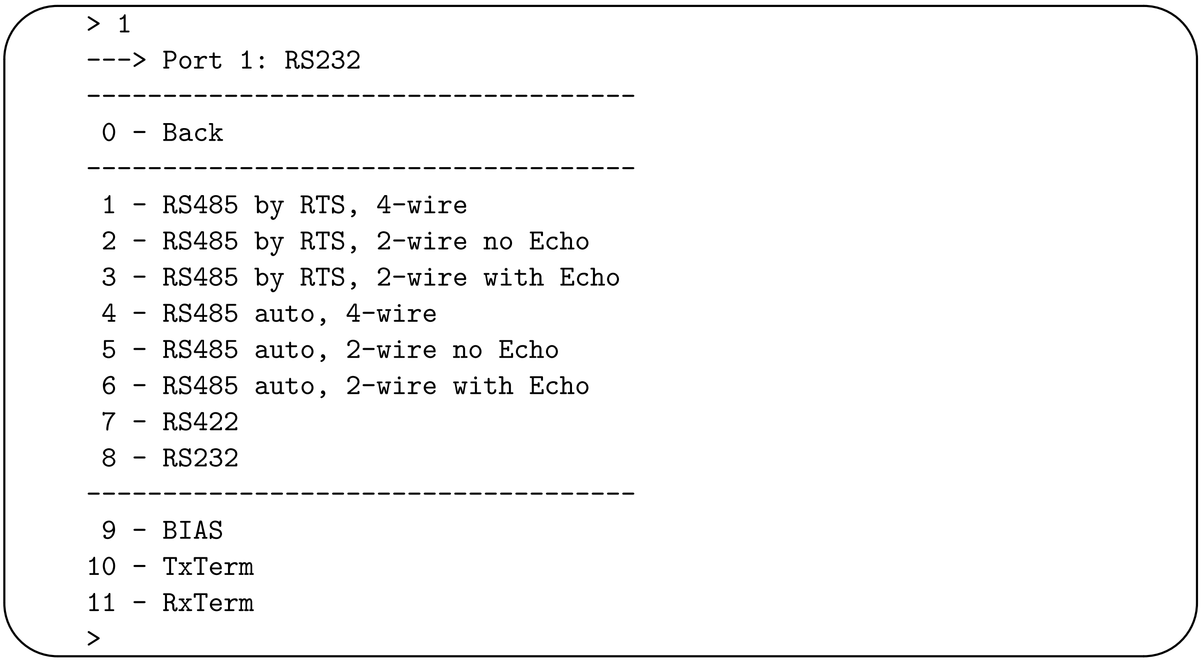
0 Selects the Main Menu again.

1 Selects the first serial port.

2 Selects the second serial port.

3 Selects all serial ports at once. This number changes depending on the number of available serial ports.

For this example port 1 is selected for configuration. To get details about the available modes please read section 2.2.



```
> 1
---> Port 1: RS232
------------------------------------
 0 - Back
------------------------------------
 1 - RS485 by RTS, 4-wire
 2 - RS485 by RTS, 2-wire no Echo
 3 - RS485 by RTS, 2-wire with Echo
 4 - RS485 auto, 4-wire
 5 - RS485 auto, 2-wire no Echo
 6 - RS485 auto, 2-wire with Echo
 7 - RS422
 8 - RS232
------------------------------------
 9 - BIAS
10 - TxTerm
11 - RxTerm
>
```

Figure 22: Configure a serial port

The options are selected by entering the number, followed by the Enter key. '0' Returns to the selection of the serial ports, but so far the configuration is not activated. This requires an explicit command, see section 4.2.5.

4.2.4 Default Port Configuration (3)

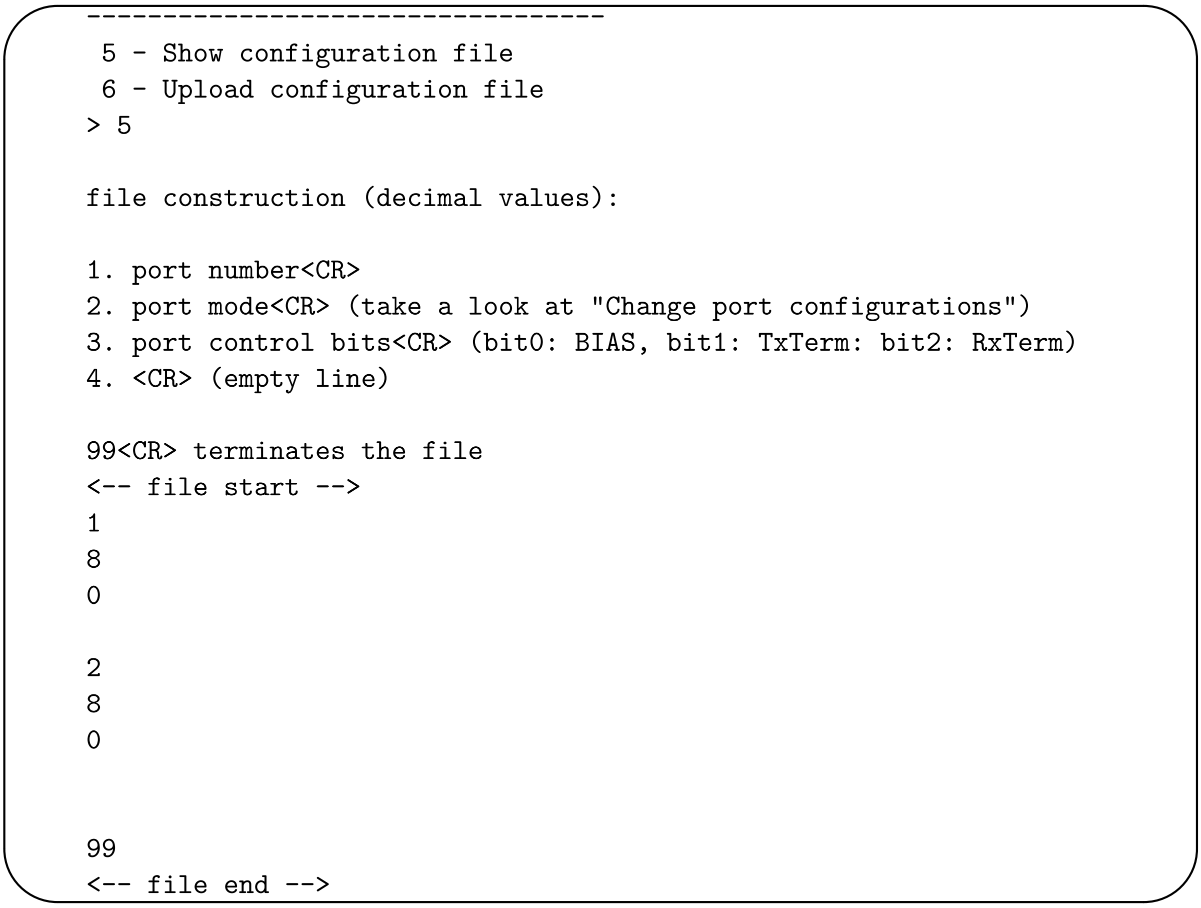
Option '3' in the Main Menu (figure 18) loads a standard configuration for all ports. This is RS232, Termination and BIAS is disabled.

4.2.5 Save Port Configurations (4)

Each change in Port configuration is temporary, only saved in volatile memory. It needs explicit confirmation for using the new configuration. Option '4' in the Main Menu (figure 18) performs this operation. The configuration data is stored to non-volatile memory inside the USB-COM PRO, and becomes active.

4.2.6 Show Configuration File (5)

Configuring each serial port (and each option of it) in a USB-COM PRO can become a long process. And this has to be repeated on the next device, and so on. To make the handling much easier the current configuration can be changed to a file, for later usage. Option '5' in the Main Menu (figure 18) is prepared for this task.



```
----------------------------------
 5 - Show configuration file
 6 - Upload configuration file
> 5

file construction (decimal values):

1. port number<CR>
2. port mode<CR> (take a look at "Change port configurations")
3. port control bits<CR> (bit0: BIAS, bit1: TxTerm: bit2: RxTerm)
4. <CR> (empty line)

99<CR> terminates the file
<-- file start -->
1
8
0

2
8
0



99
<-- file end -->
```

Figure 23: Saving Configuration file

The configuration is displayed on the terminal screen. It is preceded by a short description of the file format. Using the Terminal Emulation software this output can be captured and saved to a file. It depends on the particular software how this is done, so please check your software documentation.

The configuration of each serial port is written on three distinct lines, followed by an empty line. Port number and operation mode are already described. The definition of the control options is a bit-oriented value. Basically it is the sum of the active options, where ’1’ is for BIAS, ’2’ is for Termination of Tx Lines, and ’4’ is for Termination of Rx Lines. So this control value ranges from ’0’ to ’7’.

4.2.7 Upload Configuration File (6)

When the Terminal Emulation software allows to save the configuration of the device, it will also allow to send a text file to the USB-COM PRO. This way a previously saved configuration can be loaded to the USB-COM PRO. It can also be loaded to another device, for ease of configuring a series of USB-COM PRO. The last option ’6’ in the Main Menu (figure 18) provides this function.

5 Technical Background on RS485

This chapter will provide a little bit of theory about RS422 and RS485 data transmission. It is necessary to have this basic knowledge, to avoid or find errors in data transmission. Failures in cabling are responsible for the vast majority of transmission problems.

5.1 Transmission Technique

RS422 and RS485 use the same balanced transmission method. Signals are not transmitted by voltage on a single wire, as RS232 does. Instead two wires are used; when one carries high voltage, the other one carries low voltage. The signal is defined by the difference in voltage between those two wires. This hardens the transmission against noise. Usually twisted pairs are used, which further reduces the sensitivity for noise.

Typical voltages are +4V as high, about +0.5V as low. These voltages are generated and defined against the GND signal of the transmitter. The minimum *differential* voltage is required as ±200mV by the specifications of RS422 and RS485. The receiver detects the polarity of the differential voltage, and thus gets a Zero (negative) or a One (positive) difference.

To do this detection the voltages on the receivers side have to be inside the bounds of the common voltage range defined as -7V through +12V, as measured against the local GND of the *receiver*. The specification of the common voltage range has been given to make the design of receiver circuits more simple and cost effective.

To make sure the signals meet the common voltage range, the GND of sender and receiver must be connected somehow, otherwise the voltages are undefined and may have any value. To assure the correct range RS485 (RS422) usually requires an extra wire for GND[2], which is often forgotten. The connection may also happen by protective ground, or by other means. If the connection of GND is bad, it may be impossible to receive correct data.

5.2 Termination

When transmitted signals arrive at the end of a cable, they get reflected. They travel on the cable back and forth some more times, which is called ringing. This can cause false reading of transmitted data. When the reflections travel on the cable several times, they are damped and do no longer cause errors. This happens earlier if the cables are short.

For long cables Termination Resistors are required. These increase the damping of reflections. The value of the resistor must match the impedance of the cable, typically 120Ω for twisted pair.

As a rule of thumb[3], when the cables are longer than $\frac{1000000}{Bitrate}$ (one million divided by the bit rate) in meters, you should consider Termination Resistors.

[2] with an optional resistor

[3] Assuming group speed of 100.000km/s, 10 travels to damp out, and 10% of bit time

5.3 Polarization

In RS485 the sender must activate the transmitter before sending data, and deactivate it when all data is sent. At times when no devices send data *all* transmitters are inactive. As the result the data lines are floating, and the differential voltage is undefined. It may happen the next data is not correctly recognized, because the change from undefined to data signals is not detected.

To avoid such problems the data lines should be polarized by resistors. These insure the differential voltage to be above +200mV. Typically the positive line is pulled to +5V, while the negative line is pulled to GND. When Termination Resistors are applied, the voltage on these must be +200mV or above. When the termination resistors are of the 120Ω types, the polarization resistors shall have about 750 to 1000Ω.

The Polarization Resistors must not be too small, because they add current to the signals, and a transmitter has to act against that. A typical transmitter provides up to 60 mA of current in high and low level.

5.4 2-Wire Scheme

In many configurations a very simple cabling is required. RS485 allows for so called 2-wire cabling as shown below. Several devices are connected in parallel to the wires, which is called bus topology. Each device can either send own or receive foreign data at a given time, so it is operating in half-duplex mode.

Shown in figure 24 are three devices, RS485 specifies up to 32. The data lines are named as Data+ and Data-, a positive differential voltage is the state for a transmitted One. The GND is also connected between all devices as required, so the cabling is 3-wire in reality.

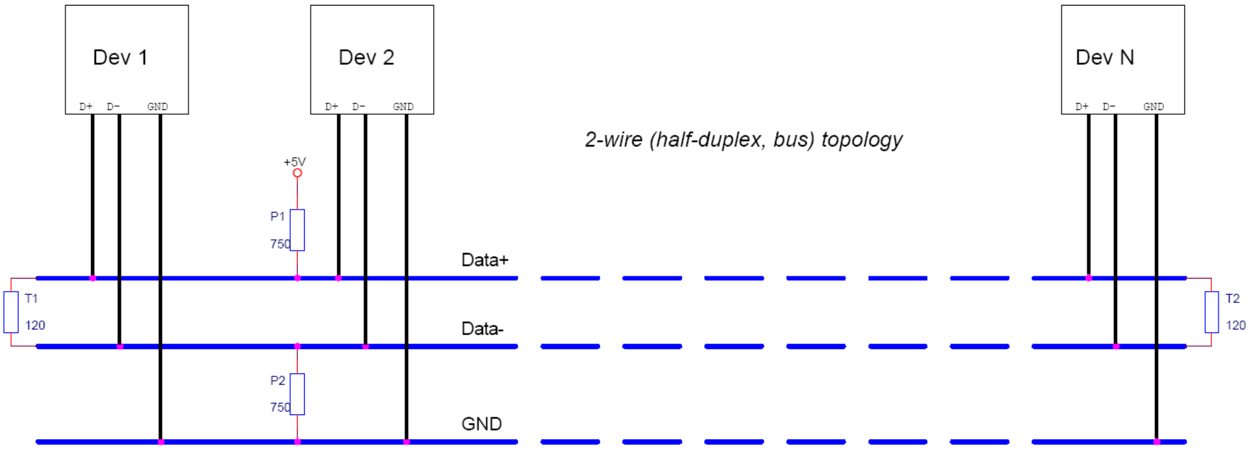


Figure 24: 2-wire cabling scheme

The resistors P1 and P2 are for polarization, T1 and T2 are for the termination function. Polarization of Data+ and Data- appears only once on this net, the termination is at the physical ends of the cable.

All devices appear the same on the cable, they have the same function. There is no Master or Slave defined by the hardware. Such functions are implemented by way of the data transmission protocol. Also RS485 addresses are defined by that protocol, as well as bus access.

5.5 4-Wire Scheme

RS422 requires dedicated wire pairs for transmit and receive. The transmit wires are used to send data to as many as 10 receivers, as stated in the specifications of RS422. Since the VScom products use RS485 line driver technology, up to 32 receivers are possible.

While one pair is used to transmit, a second pair is available to receive data at the same time. When only two devices are connected, this is a possible replacement of classic RS232 connections (PointtoPoint).

In RS485 4-wire mode the transmit wires may be shared between dedicated stations. As an example a second station can be a backup master for the network. Masters can send data and commands to one station, while they receive information from another device.

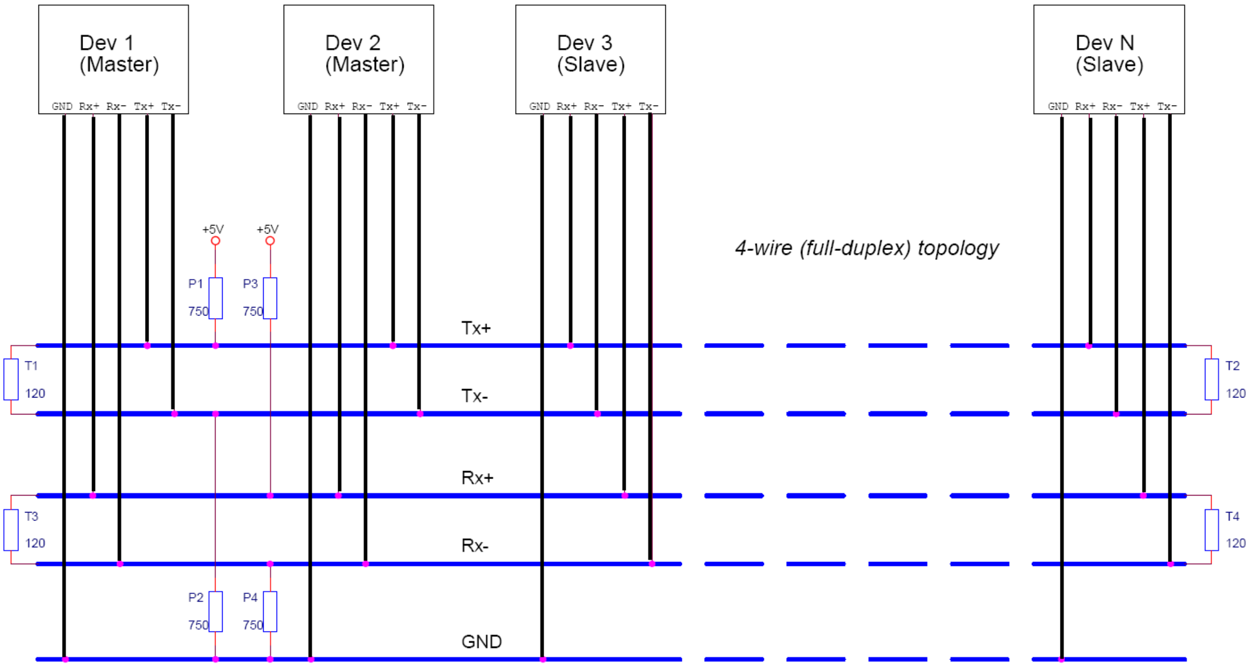


Figure 25: 4-wire cabling scheme

Figure 25 displays the wire pairs Tx± and Rx± as named for Devices 1 and 2. There are more slave devices, only two of them are shown. The slaves transmit on the Rx-lines, and receive on the Tx-lines. To implement the Common Voltage Range as specified the GND line is also connected.

Masters on the network are identified because they transmit on the Tx-lines. The two Masters in figure 25 have to synchronize their use of the Tx-lines be extra means. E.g. Dev 2 can be a backup master, which is manually activated.

Also indicated in figure 25 are polarization resistors P1/P2 for Tx± and P3/P4 for Rx±, only once per wire pair. Further Termination resistors T1/T2 for Tx± and T3/T4 for Rx± are added, one at each end of the cable. To ensure the Common Voltage Range the GND is also connected at each device, making this a 5-wire connection.

6 Connector Definitions

All the VScom USB-COM devices provide the today standard DB9 male connector for the signals of the serial ports. Some models may offer a DB25 male connector for RS232, other may provide an additional Terminal block for RS422 and RS485 signals.

The signal assignment for RS232 on DB9 male connectors is defined by the term “RS232” already. The same is for the DB25 male connector. Below is the signal definition for all connectors used by the USB-COMs.

6.1 DB9 male

Usually this type of connector is used by customers. “4w” denotes the Full-Duplex mode, usually named “4-wire”. Similarly “2w” denotes the Half-Duplex modes.

Pin	RS232	RS422	RS485 “4w”	RS485 “2w”
1	DCD	Tx−	Tx−	Data−
2	RxD	Tx+	Tx+	Data+
3	TxD	Rx+	Rx+	
4	DTR	Rx−	Rx−	
5	GND	GND	GND	GND
6	DSR	RTS−		
7	RTS	RTS+		
8	CTS	CTS+		
9	RI	CTS−		

Table 7: DB9 male Connector

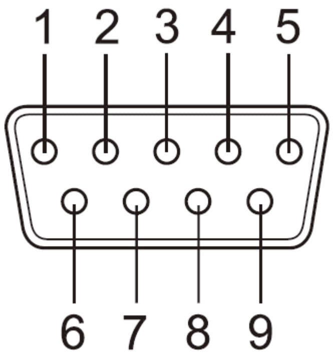


Figure 26: DB9 male Connector

6.2 DB25 male RS 232

This connector is only available on the VScom USB-COM 25 model. The pins not mentioned are a no-connect.

Pin	RS232	Pin	RS232
2	TxD	7	GND
3	RxD	8	DCD
4	RTS	20	DTR
5	CTS	22	RI
6	DSR		

Table 8: DB25 male RS232

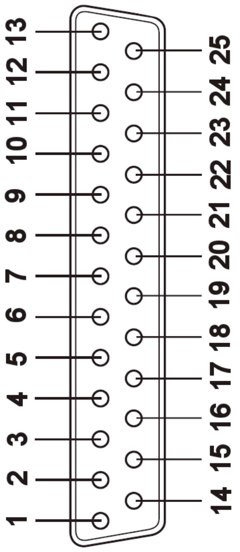


Figure 27: DB25 male RS232

6.3 Terminal Block

The Terminal Block connector is designed for connecting RS422 or RS485 signals. However it may be used for RS232, because the most important signals are available here. This type of connector is only available on some models.

Clamp	1	2	3	4	5
RS422	Tx−	Tx+	Rx+	Rx−	GND
RS485 “4w”	Tx−	Tx+	Rx+	Rx−	GND
RS485 “2w”	Data−	Data+			GND
RS232	DCD	RxD	TxD	DTR	GND

Table 9: Terminal Block