

www.vskom.de

User Manual VScom PCI Cards VScom Industrial Card

Edition: July 2011



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

Copyright © 2009 Vision Systems. All rights reserved. Reproduction without permission is prohibited.

Trademarks

VScom is a registered trademark of Vision Systems GmbH. All other trademarks and brands are property of their rightful owners.

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

1	Overview	6
2	Introduction	6
2.1	Features	6
2.2	Product Specifications	6
2.2.1	VScom 200Ei PCIex	7
2.2.2	VScom 200Ei-SI PCIex	8
2.2.3	VScom 200I UPCI	9
2.2.4	VScom 200I-SI UPCI	10
2.2.5	VScom 400I UPCI	12
2.2.6	VScom 800I UPCI	13
2.3	Packing List	14
2.4	About this Manual	14
3	Hardware Configuration	14
3.1	VScom 200Ei PCIex	14
3.2	VScom 200Ei-SI PCIex	15
3.3	VScom 200I UPCI	16
3.4	VScom 200I-SI UPCI	17
3.5	VScom 400I UPCI	17
3.6	VScom 800I UPCI	18
4	Windows Driver	19
4.1	Installing the Drivers	19
4.1.1	Install in Windows 9x, 2000, XP and Vista	19
4.1.2	Install in Windows 7	20
4.1.3	Verify the Installation	22
	About assigned port numbers	22
4.1.4	Removing the Software in Windows 9x, 2000, XP and Vista	23
4.1.5	Removing the Software in Windows 7	23
4.2	Card and Port Configuration	24
4.2.1	Card and Ports in Device Manager	24
4.2.2	Card Advanced Settings	26
4.2.3	Ports Configuration	27
	VScom Standard Port Settings	28
	VScom Advanced Port Settings	29
	Enhanced Software Settings	33
	Automatic Software Flow Control	34
	Automatic Hardware Flow Control by RTS/CTS	36
	Automatic Hardware Flow Control by DTR/DSR	37
4.3	Windows NT 4.0	38
4.3.1	Installing in Windows NT 4.0	39
	Automatic Detection of PCI Cards	39
	Configuration via Control Panel	39
	System Configuration	40
	Card and Port Configuration	40
	Uninstall Drivers from Windows NT	41

Checking Installation in Windows NT	41
5 Linux Driver	42
5.1 Manual Installation	42
5.2 Automatic Installation	42
5.2.1 Known Cards	42
5.2.2 New Models	43
5.3 Problems since Kernel 2.6	43
5.3.1 Compile a new Kernel	43
5.3.2 Kernel Start Parameters	43
5.4 Card Hardware Parameters	44
6 Troubleshooting	46
7 Proper Cabling in RS 485 and RS 422	46
7.1 Transmission Technique	47
7.2 Termination	47
7.3 Polarization	47
7.4 2-Wire Scheme	48
7.5 4-Wire Scheme	49
8 Connector Definitions	50
8.1 DB9 male	50
8.2 DB37 female	50
8.3 DB78 female	52
9 History	53

List of Figures

1 VScom 200Ei PCIex	7
2 VScom 200Ei-SI PCIex	8
3 VScom 200I UPCI	9
4 VScom 200I-SI PCI	10
5 VScom 400I UPCI	12
6 VScom 800I UPCI	13
7 Windows 7 Installation	21
8 Driver Installation Target	21
9 Windows 7 Installing Drivers	22
10 Windows 7 Driver Removal	23
11 Windows 7 Drivers removed	24
12 Cards and Ports in Device Manager	25
13 Card Advanced Properties	26
14 Port Renaming	27
15 Standard Settings	28
16 Advanced Settings 16C950	29
17 Advanced Settings 16C550A	31
18 UART Types	31
19 Software Settings	33

20	Auto XON/XOFF	34
21	Auto RTS/CTS	36
22	Auto DTR/DSR	37
23	Auto-DTR Modes	38
24	NT Plug & Play	39
25	NT Device list	40
26	2-Wire Cabling Scheme	48
27	4-Wire Cabling Scheme	49
28	DB9 male Connector	50
29	DB37 female Connector	51
30	DB78 female Connector	52

List of Tables

1	Features of VScOm 200Ei PCIex	7
2	Features of VScOm 200Ei-SI PCIex	9
3	Features of VScOm 200I UPCI	10
4	Features of VScOm 200I-SI UPCI	11
5	Features of VScOm 400I UPCI	12
6	Features of VScOm 800I UPCI	13
7	Modes on 200Ei PCIex	15
8	Termination and BIAS on 200Ei PCIex	15
9	Modes on 200I UPCI	16
10	Termination and BIAS on 200I UPCI	16
11	JP1 for Speed	17
12	Modes on 400I UPCI	18
13	Kernel Compile Options	43
14	Hardware Parameters	45
15	DB9 male Connector	50
16	DB37 female Connector	51
17	DB78 female Connector RS 485	52
18	DB78 female Connector RS 232, RS 422	53

1 Overview

The VScom Industrial PCI cards offer three way serial ports, the electrical interfaces provided are RS 232, RS 422 and RS 485 in several operation modes. The cards are designed to control industrial hardware via asynchronous serial ports, as many software requires today. There are cards for the modern bus of PCI Express, and also cards to place in the previous Universal PCI bus.

When used in RS 485 modes, the serial ports detect the end of transmission by hardware, so the RS 485 driver is set in Off state. This feature is named as ART for Automatic Receive Transmit control. If required the user may also control the the RS 485 driver by his application, using the hardware signals RTS or DTR on some products.

The supplied drivers for Windows operating system install the serial ports into the Windows API, so they appear as the usual Com ports software requires. A rich set of driver options is configured via the Device Manager in Windows. The driver allows to configure the serial ports for a bitrate of up to 921,600 bps, some models allow up to 15.6 Mbps. Of course RS 232 can not reliably transport very high bitrates, but RS 422 and RS 485 are designed for 1 Mbps on typical length connection cables.

2 Introduction

This manual describes the hardware configuration of VScom PCI cards. PCI here especially means all kinds of PCI buses and the VScom cards designed for those. Namely this is for the classic PCI cards (5V signal level), Universal PCI cards (3.3V and 5V signal level, including the PCI-X bus), as well as the PCIex cards (PCI eXpress or PCIe bus).

Also the Windows driver for these cards is described in detail.

2.1 Features

- Universal PCI (5V and 3.3V) 32 bit 33MHz
PCI Express x1
- Three way serial port interfaces: RS 232, RS 422 and RS 485
- Max 921,600 bps, half- and full-duplex
- Drivers for Windows™ operating system

2.2 Product Specifications

The family of industrial cards consists of two cards for the PCI Express bus, or PCIe x1 for short. There are four other universal cards for the classic PCI bus slots.

All serial ports provide the option to operate with RS 232 electrical interface, as well as RS 422 and RS 485. In RS 485 the so called direction switching is performed by a hardware automatic named ART for Automatic Receive Transmit control. This automatic releases the transmission lines in less than one bit time after the transmission of data has ended.

Further in RS 485 mode the serial ports not only provide the most often used bus mode (“2-wire”, with or without Echo) as Half-Duplex, also Full-Duplex is possible (“4-wire”).

2.2.1 VScom 200Ei PCIe

The VScom 200Ei PCIe card features two serial ports, available via a PCI Express slot.

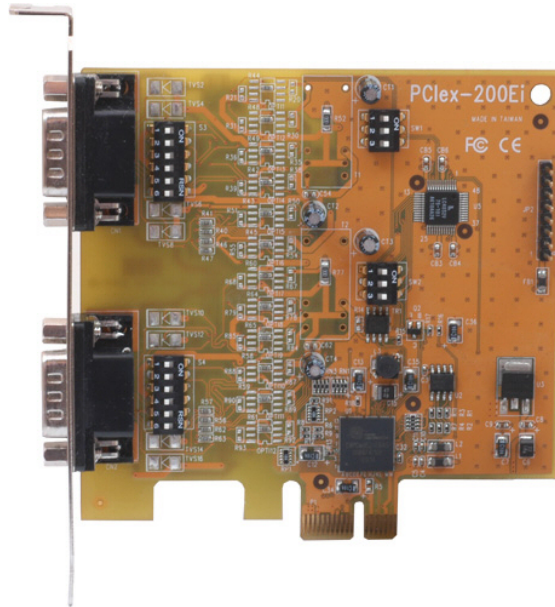


Figure 1: VScom 200Ei PCIe

The maximum speed is 15.625 Mbit/s, but this is theoretical because the electrical interface is not capable of such transmission rates.

Bus interface	PCIe x1	PCI Express, max. 500 MByte/s
Serial Ports	2	RS 232, RS 422 plus 3 modes of RS 485
Connectors	2×DB9 male	Standard for RS 232
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, DTR, DSR, DCD, RI, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
	RS 485 Half Duplex	Data±, GND
Max. Bitrates	Theoretical	15.625 Mbit/s
	RS 232	1 Mbit/s
	RS 422, RS 485	2 Mbit/s

Table 1: Features of VScom 200Ei PCIe

The operation mode is configured via the small DIP switches according to table 7. The RS 485 modes require the driver configured for Automatic DTR as “RS 485 Low Active”, as seen in 4.2.3.

Also resistors for Termination and BIAS option are built-in, configured by other switches as of table 8.

2.2.2 VScom 200Ei-SI PCIe

The VScom 200Ei-SI PCIe card features two optical isolated serial ports, available via a PCI Express slot. It is an optically isolated version of the VScom 200Ei PCIe card, the isolation is also effective in RS 232 mode.

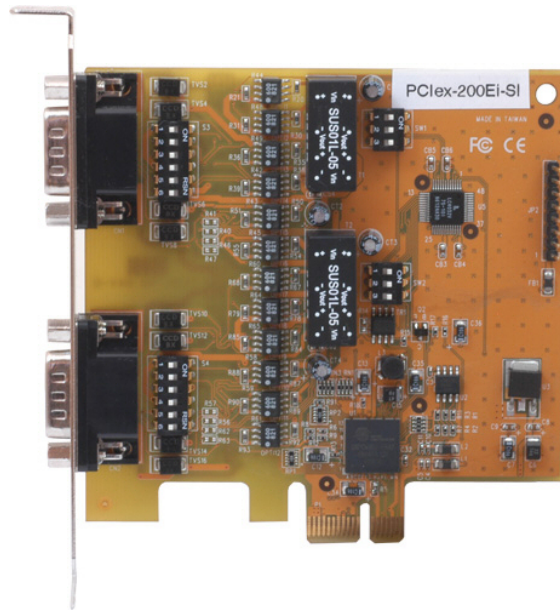


Figure 2: VScom 200Ei-SI PCIe

The maximum speed is 15.625 Mbit/s, but this is theoretical because the electrical interface is not capable of such transmission rates. The surge protection adds some load on the signal lines, so the maximum bitrates are reduced compared to the 200Ei card.

In a functional view the 200Ei-SI is very similar to the non-isolated 200Ei, especially the configuration of operation modes and Termination/BIAS is the same.

Bus interface	PCIe x1	PCI Express, max. 500 MByte/s
Serial Ports	2	RS 232, RS 422 plus 3 modes of RS 485
Connectors	2×DB9 male	Standard for RS 232
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
	RS 485 Half Duplex	Data±, GND
Max. Bitrates	Theoretical	15.625 Mbit/s
	RS 232	230,400 bit/s due to surge protection
	RS 422, RS 485	1 Mbit/s due to surge protection

Table 2: Features of VScOm 200Ei-SI PCIe

The operation mode is configured via the small DIP switches according to table 7. The RS 485 modes require the driver configured for Automatic DTR as “RS 485 Low Active”, as seen in 4.2.3. Also resistors for Termination and BIAS option are built-in, configured by other switches as of table 8.

2.2.3 VScOm 200I UPCI

The VScOm 200I UPCI card features two serial ports, available via a classic PCI slot.

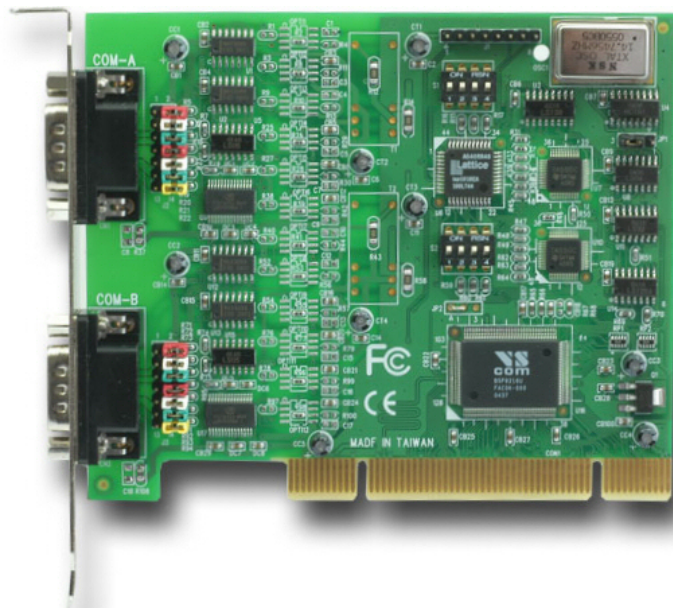


Figure 3: VScOm 200I UPCI

The maximum speed is 921.6 kbit/s. This can be reduced to 115,200 bit/s for compatibility issues, using jumper JP1.

Bus interface	Universal PCI	PCI 3.3V and 5V signals, 33 MHz
Serial Ports	2	RS 232, RS 422 plus 3 modes of RS 485
Connectors	2×DB9 male	Standard for RS 232
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, DTR, DSR, DCD, RI, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
Max. Bitrates	RS 485 Half Duplex	Data±, GND
		921,600 bit/s
	RS 232	requires short cables with low capacity
	RS 422, RS 485	921,600 bit/s

Table 3: Features of VScom 200I UPCI

The operation mode is configured via the small DIP switches according to table 9. Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers as of table 10. There is a jumper for compatibility mode, this is used to configure the maximum bitrate of the serial ports.

2.2.4 VScom 200I-SI UPCI

The VScom 200I UPCI card features two optical isolated serial ports, available via a classic PCI slot. It is an optically isolated version of the VScom 200I UPCI card, the isolation is also effective in RS 232 mode.

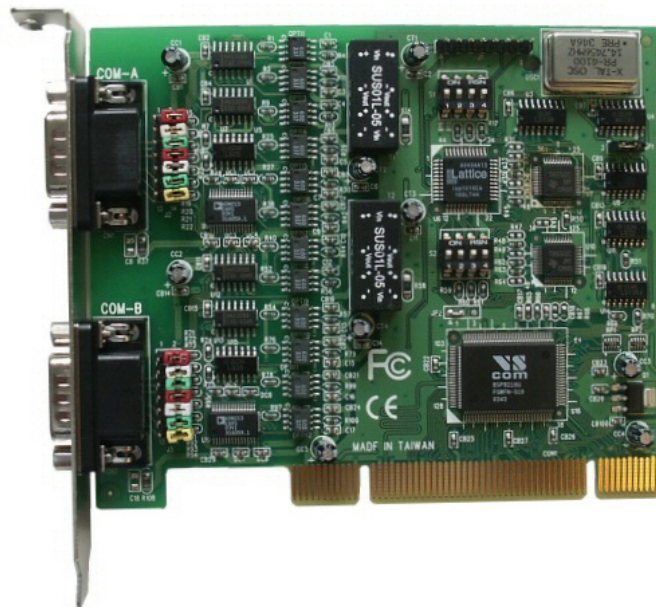


Figure 4: VScom 200I-SI PCI

The maximum speed is 921.6 kbit/s. This can be reduced to 115,200 bit/s for compatibility issues, using jumper JP1.

The surge protection adds some load on the signal lines, so the maximum bitrates are reduced compared to the 200I card.

Bus interface	Universal PCI	PCI 3.3V and 5V signals, 33 MHz
Serial Ports	2	RS 232, RS 422 plus 3 modes of RS 485
Connectors	2×DB9 male	Standard for RS 232
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
	RS 485 Half Duplex	Data±, GND
Max. Bitrates		921,600 bit/s
	RS 232	230,400 bit/s due to surge protection
	RS 422, RS 485	921,600 bit/s

Table 4: Features of VScom 200I-SI UPCI

Besides the consequences of optical isolation and added surge protection, the VScom 200I-SI UPCI card is configured and used in much the same way as the 200I card (2.2.3). The 200I-SI does not have DTR, DSR, DCD and RI signals.

The operation mode is configured via the small DIP switches according to table 9. Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers as of table 10. There is a jumper for compatibility mode, this is used to configure the maximum bitrate of the serial ports.

2.2.5 VScOm 400I UPCI

The VScOm 400I UPCI card features four serial ports, available via a classic PCI slot. All four serial ports share one common connector DB37-female, an adapter cable is included. For signal assignment see [16](#).

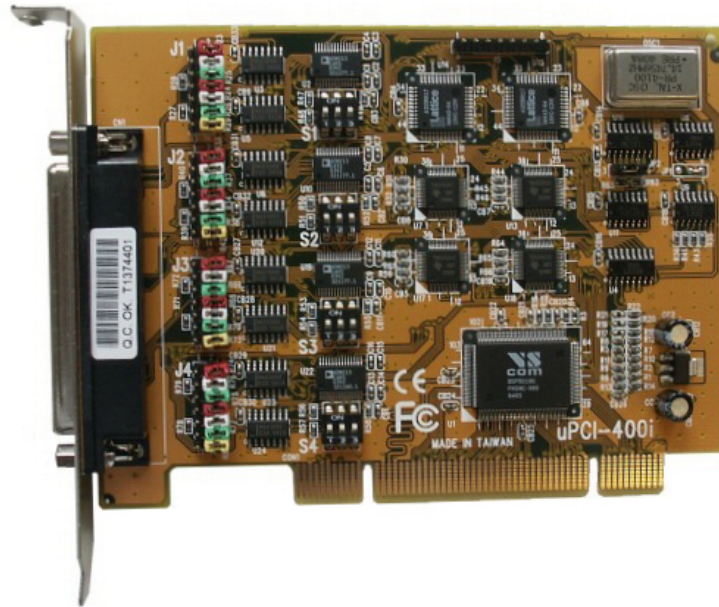


Figure 5: VScOm 400I UPCI

The maximum speed is 921.6 kbit/s. This can be reduced to 115,200 bit/s for compatibility issues, using jumper JP1.

Bus interface	Universal PCI	PCI 3.3V and 5V signals, 33 MHz
Serial Ports	4	RS 232, RS 422 plus 6 modes of RS 485
Connectors	1×DB37 female	Adapter to 4×DB9 male included
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, DTR, DSR, DCD, RI, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
	RS 485 Half Duplex	Data±, GND
Max. Bitrates		921,600 bit/s
	RS 232	requires short cables with low capacity
	RS 422, RS 485	921,600 bit/s

Table 5: Features of VScOm 400I UPCI

The operation mode is configured via the small DIP switches according to [table 12](#). Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers as of [table 10](#). There is a jumper for compatibility mode, this is used to configure the maximum bitrate of the serial ports.

2.2.6 VScom 800I UPCI

The VScom 800I UPCI card features eight serial ports, available via a classic PCI slot. All eight serial ports share one common connector DB78-female. An adapter cable to eight times DB9-male is provided. For signal assignment see 18.

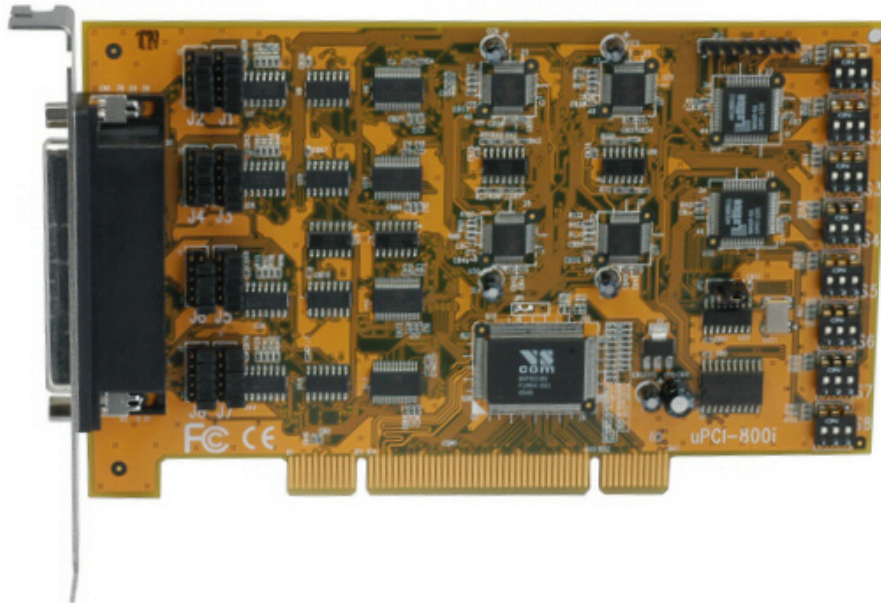


Figure 6: VScom 800I UPCI

The maximum speed is 921.6 kbit/s. This can be reduced to 115,200 bit/s for compatibility issues, using jumper JP1.

Bus interface	Universal PCI	PCI 3.3V and 5V signals, 33 MHz
Serial Ports	8	RS 232, RS 422 plus 6 modes of RS 485
Connectors	1×DB78 female	Adapter to 8×DB9 male included
Termination	Built-In	Configured for RS 422 and RS 485
BIAS	Built-In	Configured for RS 485 modes
Signals	RS 232	RxD, TxD, RTS, CTS, DTR, DSR, DCD, RI, GND
	RS 422	Rx±, Tx±, RTS±, CTS±, GND
	RS 485 Full Duplex	Rx±, Tx±, GND
	RS 485 Half Duplex	Data±, GND
Max. Bitrates		921,600 bit/s
	RS 232	requires short cables with low capacity
	RS 422, RS 485	921,600 bit/s

Table 6: Features of VScom 800I UPCI

The operation mode is configured via the small DIP switches according to table 12. Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers as of table 10. There is a jumper for compatibility mode, this is used to configure the maximum bitrate of the serial ports.

2.3 Packing List

All models are shipped with a driver CD-ROM, which includes the documentation and the driver software. The models VScOm 200I, 200I-SI, VScOm 400I and 800I UPCI also come with jumper caps placed on the pin headers. They are placed in the “Open” position. Further with models VScOm 400I and 800I UPCI a required adapter cable for DB9 male connectors is part of the package.

2.4 About this Manual

This manual describes the hardware configuration of VScOm Industrial Cards, as well as signal assignments. Also general information about RS 422 and RS 485 operation is given.

The part describing the driver is generic for all VScOm PCI cards, the same in all manuals. It will describe options not available on the VScOm Industrial Cards, mainly parallel ports. Just ignore such sections.

The screen shots in this manual 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 3.1.2.

3 Hardware Configuration

This section describes in detail how to configure the serial ports of the VScOm Industrial Card to perform the required functions. On all cards is a small block of DIP switches to configure the basic operation mode and also the electrical interface as RS 232, RS 422 or the selected mode of RS 485.

There is another Jumper block or DIP switches, which will configure the Termination and BIAS of the serial ports. These options must be deactivated for RS 232 operation, otherwise there will be transmission errors. BIAS is mostly used in RS 485 modes, but may be required for the Rx_{\pm} lines in RS 422 mode also. Please read section 7 with more details on Termination and BIAS.

Finally the UPCI bus cards offer a compatibility speed mode, configured by a separate jumper.

3.1 VScOm 200Ei PCIe

Connector CN1 is the first serial port, CN2 is the second port. The operation mode is configured via the small DIP switches SW1 and SW2 according to table 7. When the serial port is configured for any of the RS 485 modes, the transmitter has to be activated to transmit data, and disabled when no data is sent. In Half-Duplex mode the latter function is necessary to receive data. The RS 485 transmitter is controlled by the signal DTR.

Customers software may control the signal DTR, but this may be slow. Especially in modern operating systems the application may not have the priority to perform this function in a timely manner. Further, on Windows there is no option to definitely tell all data has really sent out of the port.

However the serial port itself¹ has an option to control the signal DTR exactly as required for RS 485 operation. This must be activated by configuration in the driver. Select Automatic DTR as “RS 485 Low Active”, as seen in 4.2.3






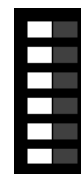
Operation Mode	Control	DIP Configuration	Driver Configuration
RS 232			
RS 422			
RS 485 4-wire	by DTR		Automatic DTR 4.2.3
RS 485 2-wire	by DTR		Automatic DTR 4.2.3
RS 485 2-wire with Echo	by DTR		Automatic DTR 4.2.3

Table 7: Modes on 200Ei PCIex

Also resistors for Termination and BIAS option are built-in, configured by other switches S3 and S4 as of table 8.

DIP	Function	Remark
1	BIAS Tx+	Use with DIP 3
2	Termination Tx+ Tx-	120Ω
3	BIAS Tx-	Use with DIP 1
4	BIAS Rx+	Use with DIP 6
5	Termination Rx+ Rx-	120Ω
6	BIAS Rx-	Use with DIP 4



DIP are numbered top down

Table 8: Termination and BIAS on 200Ei PCIex

3.2 VScom 200Ei-SI PCIex

In a functional view the 200Ei-SI is very similar to the non-isolated 200Ei, especially the configuration of operation modes and Termination/BIAS is the same. Configure the VScom 200EI-SI PCIex exactly as the VScom 200EI PCIex card, as described above.

¹The UART for Universal Asynchronous Receiver Transmitter, a 16C950 here

The operation mode is configured via the small DIP switches SW1 and SW2 according to table 7 above. The RS 485 modes require the driver configured for Automatic DTR as “RS 485 Low Active”, as seen in 4.2.3

Also resistors for Termination and BIAS option are built-in, configured by other switches S3 and S4 as of table 8 above.

3.3 VScom 200I UPCI

Connector COM-A is the first serial port, COM-B is the second port. The operation mode is configured via the DIP switches S1 and S2 according to table 9. The serial ports of this card have a special hardware function named ART to control the RS 485 transmitter. There is no need to configure anything in the driver.






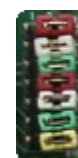
Operation Mode	DIP Configuration
RS 232	
RS 422	
RS 485 4-wire	
RS 485 2-wire with Echo	
RS 485 2-wire no Echo	

Table 9: Modes on 200I UPCI

Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers J2 and J3 as of table 10. A jumper must be closed (both pins connected by a cap) to activate the mentioned function.

Pins	Function	Remark
1-2	Termination Tx+ Tx-	120Ω
3-4	BIAS Tx+	750Ω to VCC, use with 5-6
5-6	BIAS Tx-	750Ω to GND, use with 3-4
7-8	Termination Rx+ Rx-	120Ω
9-10	BIAS Rx+	750Ω to VCC, use with 11-12
11-12	BIAS Rx-	750Ω to GND, use with 9-10
13-14	Termination CTS+ CTS-	120Ω



Pins are numbered left to right and top down

Table 10: Termination and BIAS on 200I UPCI

For BIAS there is a natural rule. The BIAS of positive (+) and negative (–) lines are either both activated, or both not used.

Right below the crystal is the compatibility speed header JP1. This is used to configure the maximum bitrate of the serial ports.

Pins	max. Bitrate	Remark
2–3	921,600	Default configuration
1–2	115,200	Compatibility option

Table 11: JP1 for Speed

The mentioned pins have to be connected by a cap to configure the operation mode. Make sure there is a cap placed in any of the two allowed positions, otherwise the port will not operate. When the configuration has been changed, the driver for Windows has to learn this. In the Advanced panel click on the “Check” button (4.2.2).

Check

3.4 VScom 200I-SI UPCI

Besides the consequences of optical isolation and added surge protection, the VScom 200I-SI UPCI card is configured and used in much the same way as the 200I card (3.3).

The operation mode is configured via the DIP switches S1 and S2 according to table 9 above.

Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers J2 and J3 as of table 10 above. A jumper must be closed (both pins connected by a cap) to activate the mentioned function.

Right below the crystal is the compatibility speed header JP1. This is used to configure the maximum bitrate of the serial ports, see table 11 above. When the configuration has been changed, the driver for Windows has to learn this. In the Advanced panel click on the “Check” button (4.2.2).

Check

3.5 VScom 400I UPCI

The connector CN1 harbors all signals of the serial ports, see section 8.2. The operation mode is configured via the DIP switches S1 to S4 according to table 12. In contrast to the 2-port cards, here are six modes of RS 485. The new modes allow to control the RS 485 transmitter using signal RTS² instead of the automatic ART.

²This means by customers application









Operation Mode	Control	DIP Configuration
RS 232		
RS 422		
RS 485 4-wire	by ART	
RS 485 2-wire	by ART	
RS 485 2-wire with Echo	by ART	
RS 485 4-wire	by RTS	
RS 485 2-wire	by RTS	
RS 485 2-wire with Echo	by RTS	

Table 12: Modes on 400I UPCI

Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers J1 to J4 as of table 10 above. A jumper must be closed (both pins connected by a cap) to activate the mentioned function. These are the same jumpers as on the VScom 200I UPCI card.

Right below the crystal is the compatibility speed header JP1, configure as of table 11 above. When the configuration has been changed, the driver for Windows has to learn this. In the Advanced panel click on the “Check” button (4.2.2).

Check

3.6 VScom 800I UPCI

The connector CN1 harbors all signals of the serial ports, see section 8.3. Besides the other connector the VScom 800I UPCI card features the same functions as the VScom 400I UPCI, now for eight ports. So it is configured in the same way, via the DIP switches S1 to S8 according to table 12 above.

Also resistors for Termination and BIAS option are built-in, configured by a block of jumpers J1 to J8 as of table 10 above. A jumper must be closed (both pins connected by a cap) to activate the mentioned function.

Near the small crystal is the compatibility speed header JP2, configure as of table 11 above. When the configuration has been changed, the driver for Windows has to learn this. In the Advanced panel click on the “Check” button (4.2.2).

Check

4 Windows Driver

This chapter of the manual documents the drivers for Windows Operating Systems. The driver is basically the same for all the VScom PCI cards 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.

When Windows loads the driver, the type of card is identified. The driver adapts to the options available on the certain card, and disables or hides other options. Because of the great similarity of the products, users find the options always at the same positions, they do not have to learn each product individually.

The following text is common to all cards, important differences are mentioned when necessary. When a described option is not available on your product, this is not an error. This option is just not offered by your hardware.

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. The drivers still support very old phased out products, as well as old Windows versions 95 to ME (9x strand) and NT. The old Windows systems are no longer supported by Microsoft, so there may be hidden problems on your computer. These may cause the VScom cards to malfunction in your system. Said that, we can not guarantee the old products operate reliably, and also modern products may have unexpected behaviour in old operating systems. The old operating systems are just mentioned in the following.

Many VScom cards may be installed simultaneously. In fact the driver does not limit the number of cards installed, Windows does. In Windows 9x the maximum number of serial or parallel ports installed is 127, in Windows NT or later the limit is 256 ports of serial or parallel kind.

The current version of the driver is 3.1.2.

4.1 Installing the Drivers

Windows Systems perform the hardware detection by Plug & Play, you do not have to configure the card in any way. Special DIP switches as described above of course are an exception.

Windows NT performs different, this is described later (section [4.3](#)).

4.1.1 Install in Windows 9x³, 2000, XP and Vista

1. Install the card in your system and boot Windows.
2. Log on to an account with administrative privileges.
In Windows 95/98/98SE/ME each account is appropriate,
Windows NT is handled extra below.
3. Windows will automatically detect the VScom Industrial Card.
4. The “Add New Hardware Wizard” dialog box appears and searches for new drivers for the VScom Industrial Card.

³Windows 95 up to Windows ME

5. Click on “Next”.
6. From the listed box, choose “Search for the best driver for your device”.
7. Click on “Next”.
8. If not done already, insert the driver CD into the CD-ROM drive.
9. From the generated choices, choose “CD-ROM drive”.
10. Windows searches for the driver, and detects the .inf-file on the CD. It is ready to install the driver for VScom Industrial Card.
Click on “Next”.
11. On some systems Windows may claim about a “Missing digital signature”. This is not an error, just “Continue”.
12. Click on Finish.
13. Windows now detects the COM ports on the VScom Industrial Card.
14. If Windows requests the driver disk again, point to the directory named after your operating system, e.g. “Windows XP (32 bit)”.
15. Windows will copy and install the driver for the new PCI serial ports.
16. You might be asked to restart your computer in the process. We suggest to wait with until all components (cards and serial ports) are completely installed, and restart then.

4.1.2 Install in Windows 7

Windows 7 looks different when installing drivers. It seems it is very different from the other Plug & Play enabled versions. But this is a deception.

When detecting new hardware Windows first visits the website of Windows Update to find a suitable driver, which shall be installed. If none is found, Windows searches the local system for the required driver, and loads it. If also there is none, this would be the time to pop-up the “Add New Hardware Wizard”, so the user can supply a driver. And this step does not happen in Windows 7, the Wizard does not appear⁴.

Instead Microsoft decided to have the user run an installation program for the hardware. This software will install all the driver files to the system, and then initiates a hardware detection process. This installation program in version 3.3.0.15 is available for all cards.

The following assumes the card is already installed in a slot.

1. Start the installation program. On the CD-ROM this is named vscompci.exe, and vscomp-cix64.exe for the x64 Edition of Windows 7. For download from [download](#) it is available as a self-extracting file, which detects the system automatically. This is the program, click on “Install”.

⁴It is still available using some tricks



Figure 7: Windows 7 Installation

2. The software requests a location to install the required files.

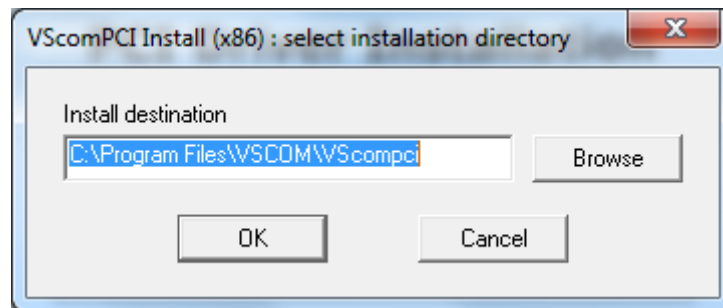


Figure 8: Driver Installation Target

Confirm the location.

3. The installation proceeds.



Figure 9: Windows 7 Installing Drivers

When this is completed, the hardware is recognized by Windows 7, and the drivers are loaded. This may take some time, especially for many serial ports.

It is also possible to first run the Installation program. This will install the drivers in the same way, but they are not loaded. The system is prepared to detect a card. After shut down and installation of the hardware the serial ports are installed without further user intervention. The same happens when more cards are added to an existing system.

4.1.3 Verify the Installation

You can now verify the installation by looking at the “Multi-port serial adapters” section of the Device Manager (Go there by Start - Settings - Control Panel - System - Hardware - Device Manager). There you will find the new device “VScom Industrial Card” listed.

Select this, and open the properties. In the “Advanced” tab, click the “Check” button to verify the speed setting from the hardware installation. You may also rename the ports here.



About assigned port numbers In some systems the ports will receive numbers customers do not expect. For example a card with two serial ports may receive the names ‘Com5’ and ‘Com8’, but

the expected ports 'Com3' and 'Com4' do not appear in the Device Manager. This is neither an error of the card or the driver, it is just how Windows manages port numbers.

Windows maintains an internal database⁵ of port numbers which have been assigned to some hardware in the past. These numbers are skipped when assigning port names to new hardware. The numbers may be changed after installation of VScom hardware.

4.1.4 Removing the Software in Windows 9x⁶, 2000, XP and Vista

To remove installed files and Windows registry information

1. Go to the directory where you installed the driver from as described [above](#).
2. Double click the "VSCLEAN.EXE", and reboot your computer when asked to.

4.1.5 Removing the Software in Windows 7

To remove installed files and Windows registry information you may use the same way as in the past (above [4.1.4](#)). Or you may use the installation software.

1. Start the installation program, and use the "Remove" option.



Figure 10: Windows 7 Driver Removal

⁵A Binary value in the registry: HKLM\SYSTEM\CurrentControlSet\Control\COM Name Arbiter

⁶Windows 95 up to Windows ME

2. Short time later the removal is done.



Figure 11: Windows 7 Drivers removed

4.2 Card and Port Configuration

Each card is organized in two types of components in the system. One type is for the card itself, i.e. the basic PCI interface. The other type is for the ports. This is reflected in Windows and the driver software.

4.2.1 Card and Ports in Device Manager

The category where the VScom cards are installed is 'Multi-port serial adapters'. And as expected there is the category of 'Ports (COM & LPT)' for the serial ports. Both categories are standard on Windows.

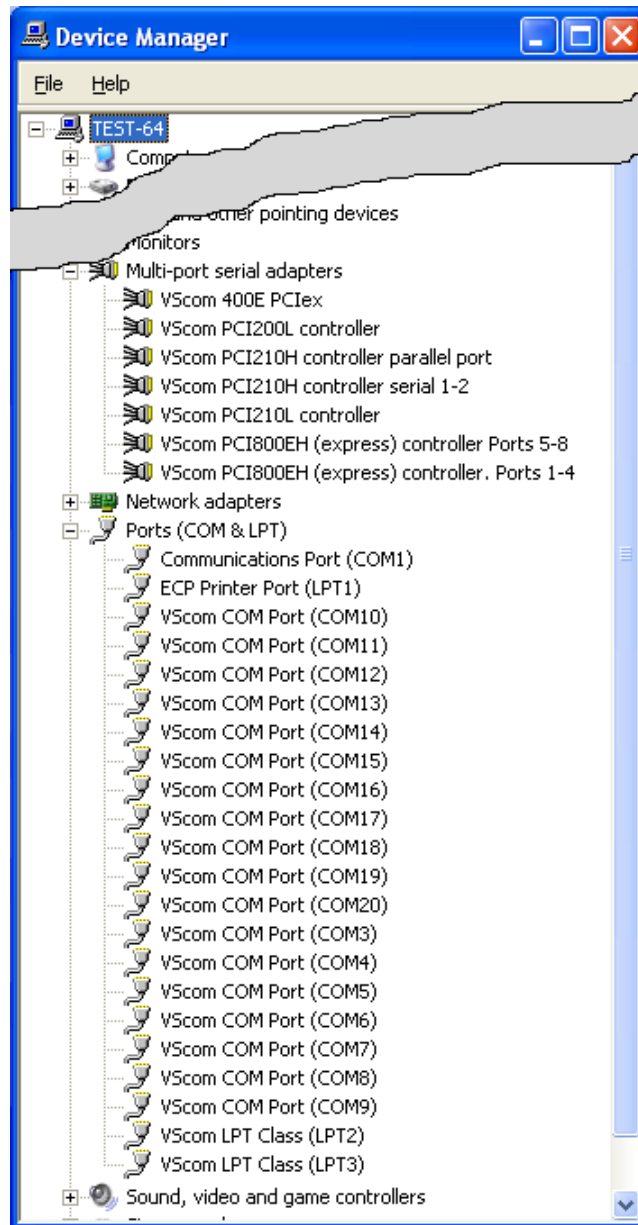


Figure 12: Cards and Ports in Device Manager

Shown here are the data of some serial and parallel V5com cards in the Device Manager. There are many cards installed simultaneously, just to show this is possible.

The 800EH is displayed as two items in 'Multi-port serial adapters'. Because the H-series cards support two functions each card is listed as two devices. The 800EH card appears as two groups of serial ports. The same happens for the 210H card, with two serial ports in one group, and the parallel port in the other.

L-series cards like 200L or 210L put all ports in one component, so serial and parallel ports may be mixed.

4.2.2 Card Advanced Settings

Open the properties of a card via the Device Manager, and select the “Advanced” tab. These are the advanced settings for a VScom card, a VScom 210L UPCI in this case.

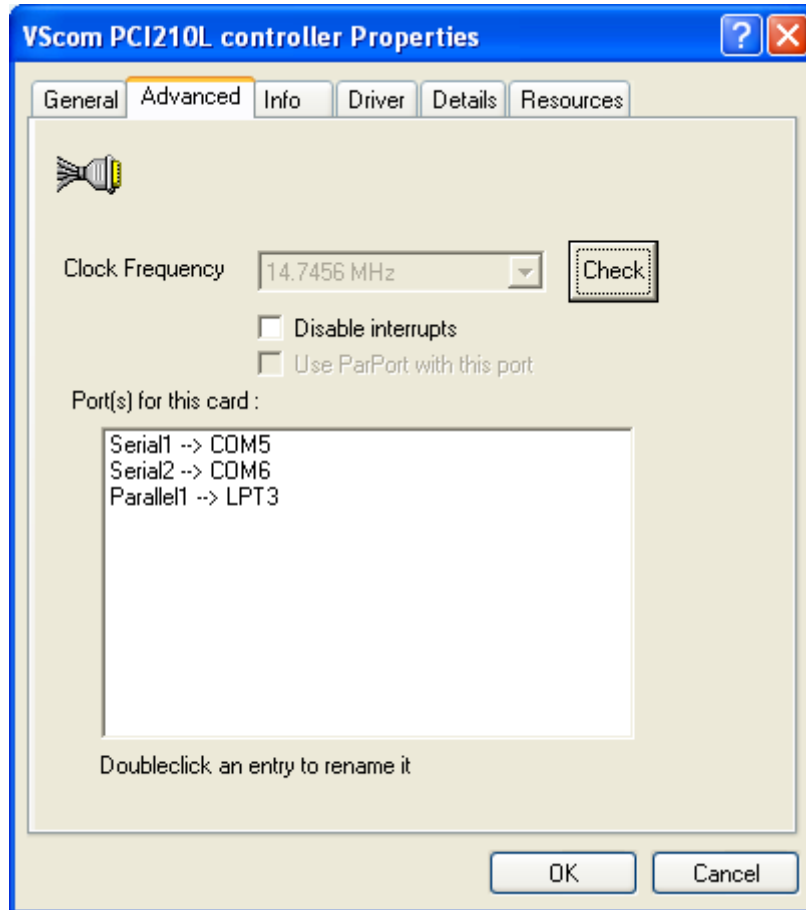


Figure 13: Card Advanced Properties

This panel displays the ports of the card, and their assigned names in Windows.

For serial (COM) ports there is the “Clock Frequency”. Some models allow to change this clock, which is the same for all ports on a card. When installing a card the driver attempts to detect the clock of the serial ports. If the system is very busy while installing the driver, the detection may yield a wrong result.

If you have communication problems after installation, click the “Check” button. The driver will measure the actual speed again, and change the entry as needed. There is no option to manually enter the clock value. Unless you have very special requests you should leave the hardware at top speed. This offers higher speed and a wider range of bitrates, especially unusual configurations.

The display of parallel (LPT) ports is similar.

Check

The COM- or LPT-assignments of each port may be changed from the initial values after installing the drivers. To change the assignment just double-click the entry in the “Advanced” panel.

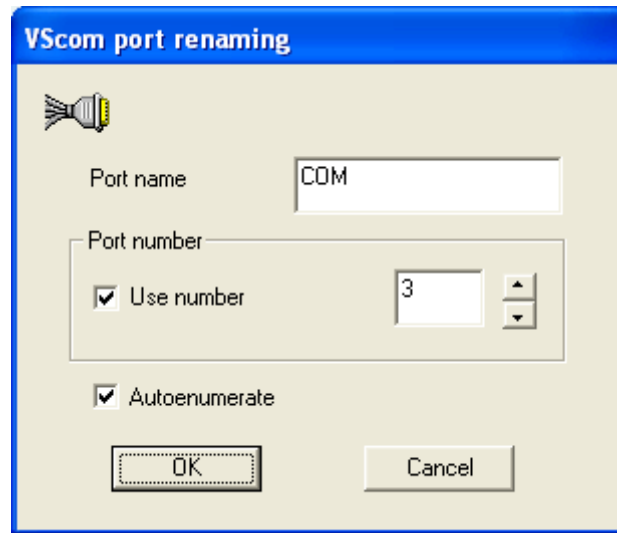


Figure 14: Port Renaming

Then this panel is opened for renaming a port.

In some cases it may be useful to change the name from COM (or LPT) to something else. Do this only if you exactly know what you are doing. Usually you will change the number of the first port only. With the ‘Autoenumerate’ feature enabled the subsequent ports receive the next numbers. However it is possible to assign port numbers with gaps in between.

4.2.3 Ports Configuration

The available functions of the serial ports depend on the type of UART implemented on the card. Cards with 16C550A type serial ports provide less configuration than the 16C950 UART. As a consequence some panels are just missing. This is not a sign of an error.

VScom Standard Port Settings

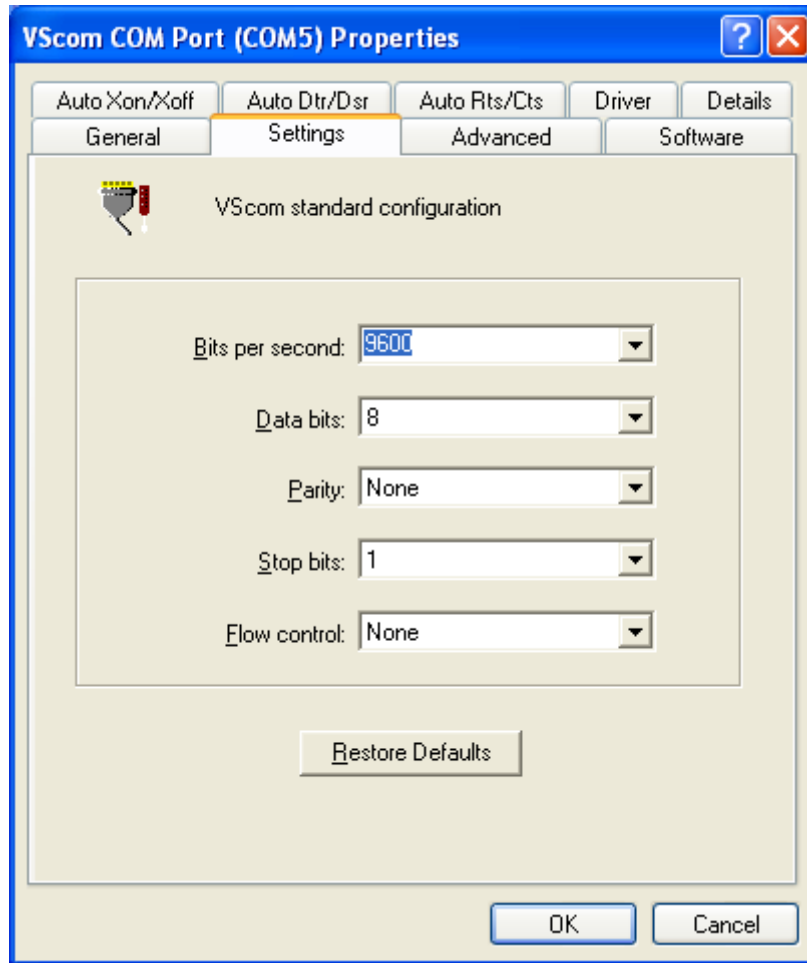


Figure 15: Standard Settings

This panel is well known to most users, it is similar to the panel used by the Microsoft driver for Com1/2. Please note the option of Flow control. To enable the automatic features of software or hardware flow control, the appropriate selection must be done here. This does *not* require to change the configuration in the user application.

VScom Advanced Port Settings The Microsoft driver for Com1/2 provides some control of FIFO configuration. The Advanced Settings are for similar purposes, but with extended functions.

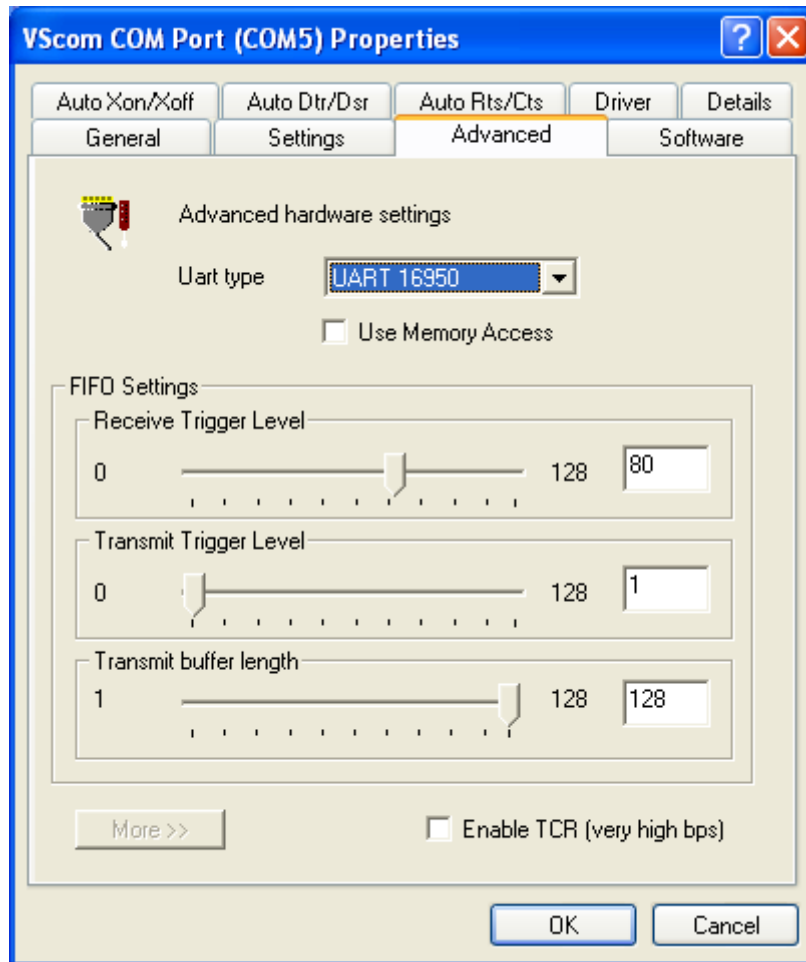


Figure 16: Advanced Settings 16C950

The VScom H- and E-series cards use a 16C950 UART to transmit data. This chip and the driver can emulate a variety of other UARTS. We suggest to use the standard value of 16950, the driver is optimized for this. If the driver is used for other VScom series, a different UART may be implemented. The available emulations depend on this. At least a 16550 is possible with all series. The possible settings for FIFO control depend on the chosen emulation.

Here the elements one by one

UART type This is to choose the emulation. The UARTS of type 16450 and 8250 operate without FIFO. We suggest to use the highest possible setting for best operation.

Use Memory Access Instead of operating the UART via I/O, it is possible to operate it via Memory Access. In critical systems this can save a little amount of performance. It may be worth a try, however VScom recommends to stay with the box unchecked.

Receive Trigger Level The time when the UART generates an interrupt for received data. It defines the amount of data necessary to do this. With the 16950 you may select any value from 1 to 128. Other types allow only certain values. Higher values increase performance, lower values decrease possibility for loss of data. As a reference: the usual value for a 16C550A is 8, leaving space for 8 more bytes to receive.

Transmit Trigger Level The time when the UART generates an interrupt for sent data. When the transmit FIFO contains less data than this, the interrupt is issued. This value is not available with all emulations. Higher values make a smoother transmission of data, but increase the system load.

Transmit Buffer Length The amount of data sent to the UART when the transmit FIFO is empty. This is a complete software item. The value ranges from 1 to the FIFO size of the emulation. Higher values increase transmission performance. The typical value here is 16, the FIFO size of a 16550. This value must not be less than the Transmit trigger level. CAUTION: external hardware - such as a modem - must be able to handle this blocksize, very old systems may require to reduce this parameter to '1'.

Enable TCR The 16C950 UART used on the H-series cards has some hardware option to get very high bit rates, up to 3.6 Mbps. Of course these bit rates can not be used on RS 232, but there are other advantages. The maximum serial speed is raised, so there are more possible bit rates available in the usual range of RS 232 transmission. This allows to use some more rather unusual settings for transmission rates, some special hardware may require. Possible products with RS 422/485 line drivers can use such high bit rates. Since this option may reduce the transmission accuracy in certain environments, it is disabled by default.

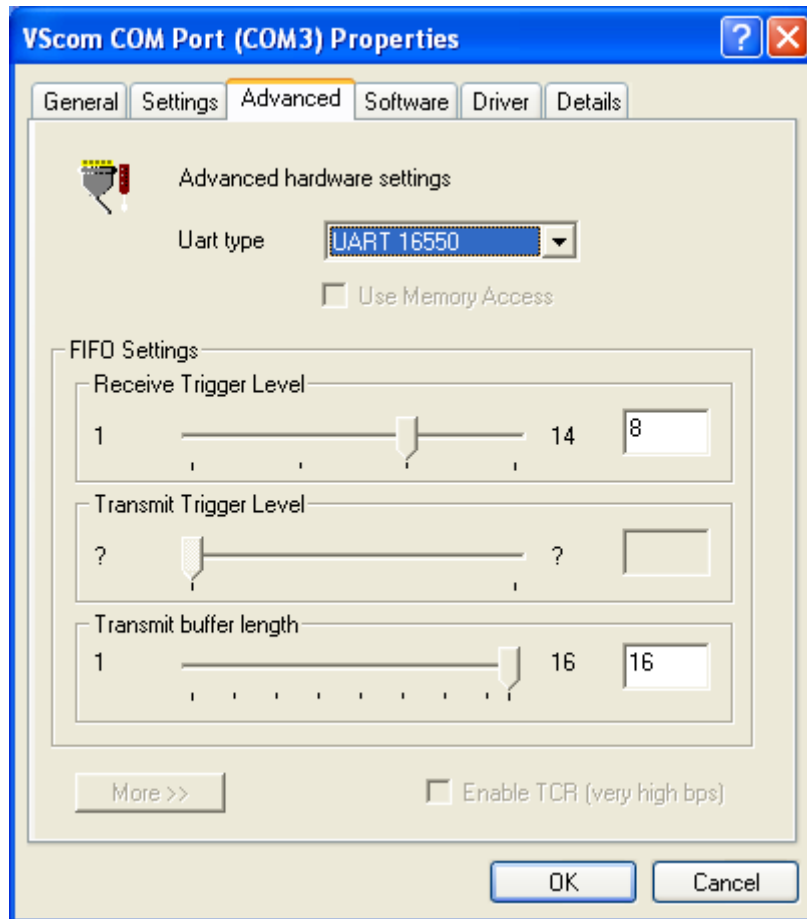


Figure 17: Advanced Settings 16C550A

The VScom L-series cards have a 16C550A compatible UART for the serial ports. The Receive Trigger Level only provides the four steps 1, 4, 8 and 14. There is no Transmit Trigger Level, hence this slider is disabled. The Transmit Buffer Length ranges from 1 to 16 only. The serial port can emulate the 16C450 and the ancient 8250 UART.

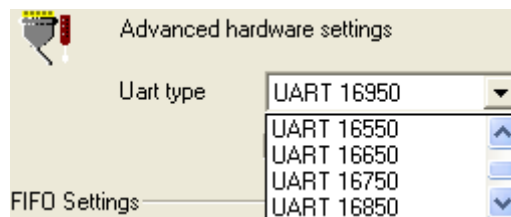


Figure 18: UART Types

Remarks on the UART Emulation Modes Usually the best choice for the UART type emulation is just the best available, i.e. the standard 16C950 model. However some hardware options may be configured to match the functions of less advanced types. This may be required for rare and very

special situations, though in some of those it is even better to just change the configuration of the other parameters.

8250 The FIFO is totally switched off, as are all other options. The real 8250 UART had a speed limitation of 57600 bps, this is maintained by software here.

16450 The most common UART in the late 1980 decade. There still is no FIFO available, the maximum bitrate is raised to 115200 bps. Sometimes this Emulation may be useful, if the connected device is overrun by the transmitted data. But usually it is better to restrict the Transmit Buffer length to 1.

16550 The most common UART until today. There are 16 Byte FIFO for transmit and receive each, and four trigger levels for receive. The built-in serial ports of modern computers use this UART model, with a maximum speed of 115200 bps, which is 921600 bps on VScom cards.

16650 This UART basically has 32 bytes FIFO for transmit and receive each, and also four trigger levels for receive. However this emulation uses 128 Bytes FIFO for receive. Additional function is automatic Flow Control by RTS/CTS or XON/XOFF.

16750 This is another UART with extended FIFO compared to the 16550 model. In basic configuration there are 16/16 Byte FIFO, which can be increased to 64/64 Byte. However as an emulation it uses 128/128 byte FIFO. Also automatic Flow Control by RTS/CTS and Xon/Xoff is possible.

16850 This is another UART model with 128/128 Byte FIFO.

16950 This is the native mode the ports on the VScom cards are used. In most situations it is better to use this mode, and restrict the options to a suitable configuration.

Enhanced Software Settings

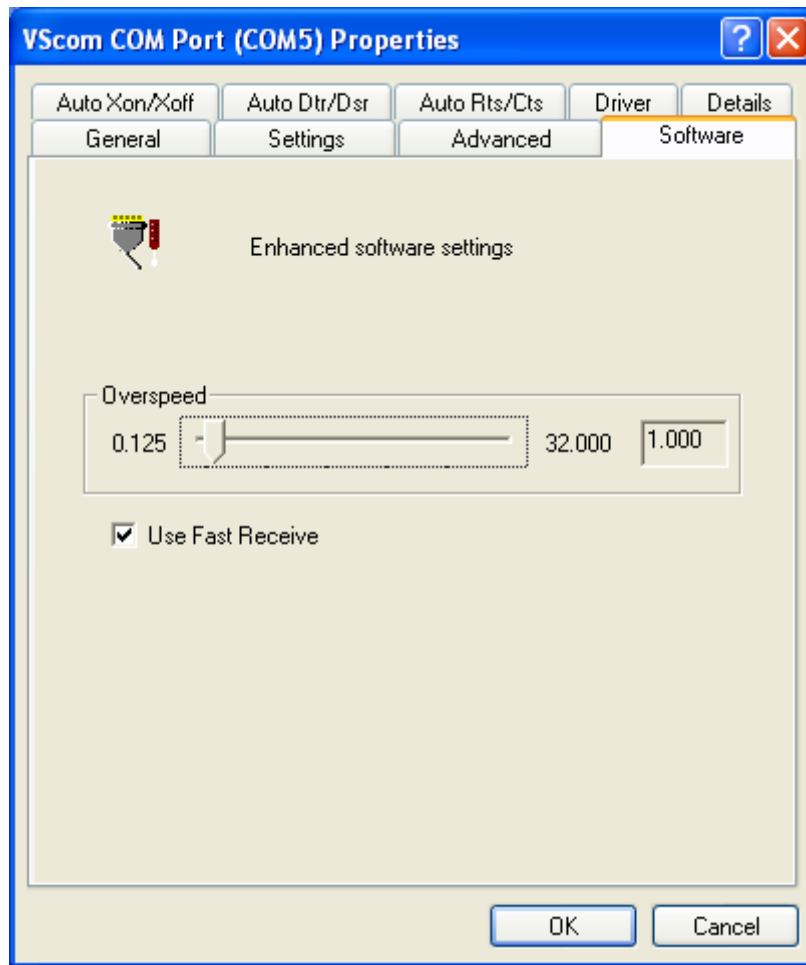


Figure 19: Software Settings

This is a special feature of the VScOm driver. In older driver versions it was simply called Overspeed.

Overspeed opens the possibility to get other speeds than the application requested for. The application requests a certain speed; this is multiplied with the value defined in here. The result is set as the real speed. Possible factors range from 0.125 (1 divided by 8) up to 32 in steps of 0.125. Of course you can not get a speed which is higher than the maximum speed of the port (921600 bps / 3.6 Mbps). The normal value is 1 of course, and it should be left there. This is a feature with some risk. With a modem you very easy get a speed the modem can not handle.

Examples:

- Wrong: Overspeed is set to 8. The modem has a maximum interface rate of 460800 Baud. When the application requests 115200, the result is 921600 Baud. The modem can not handle this.

- Wrong: Overspeed is set to 2. The application requests 38400, the result is 76800. Even when this is below the maximum speed of the modem, it does not recognize this speed in many cases. Operation is not possible.
- Wrong: Windows searches for Modems at certain speeds. Modem detection may fail, because of wrong settings.
- Correct: The application is limited to 19200. Overspeed may be 6 or 12 to get 115200 or even 230400 with a modern modem.

Automatic Software Flow Control

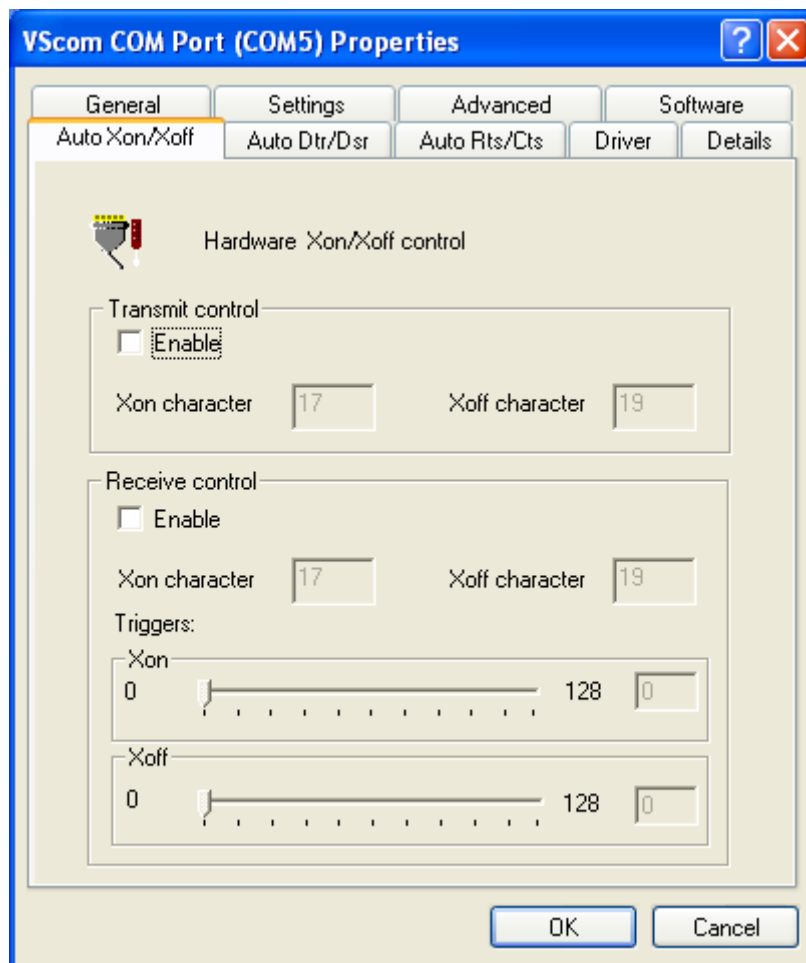


Figure 20: Auto XON/XOFF

This panel configures the On-Chip Software Flow Control provided by the UART. Only some UART types support the Flow Control directly on the chip. Use the 16950 emulation to allow rich possibilities.

To get access to this panel configure Software Flow Control in the Standard Port Settings (see 4.2.3), and leave it enabled in the driver.

Transmit control Enable This enables the automatic control when sending data. If the receiving device sends a stop (Xoff) signal, transmission is halted. When a continue (Xon) signal is received, transmission continues. This reduces the probability of sending too much data to the receiver. This does not need any processing by the main processor.

Transmit control Xon/Xoff Character The special characters acting as continue and stop signals as above. Normally stop (Xoff) is a ^S (dec: 19, hex: \$13), continue (Xon) is a ^Q (dec: 17, hex: \$11). You should not change these values.

Receive control Enable This enables the automatic control when receiving data. When too much data is received and not read by the processor, the UART generates a stop signal to the other side. When enough data is read by the processor, the UART generates a continue signal.

Receive control Xon/Xoff Character Same function as Transmit control above. Even when it is possible to use other values than above, we strongly recommend to use the standard values.

Receive control Triggers Xon The level when the UART generates the continue signal. When the processor reads the received data, the amount remaining in the UART will eventually fall below this limit. At that time the continue signal is issued to the other side.

Receive control Triggers Xoff The level when the UART generates the stop signal. When this amount of data is received and stored in the FIFO, the UART issues the stop signal to hold further transmission by the other side. This value must be higher than the Xon Trigger. Depending on the global situation (processor speed, system load, transmission parameters, transmitting device, ...) possible values can be: 32 (quarter of FIFO) for Xon and 64 (half) for Xoff.

Automatic Hardware Flow Control by RTS/CTS

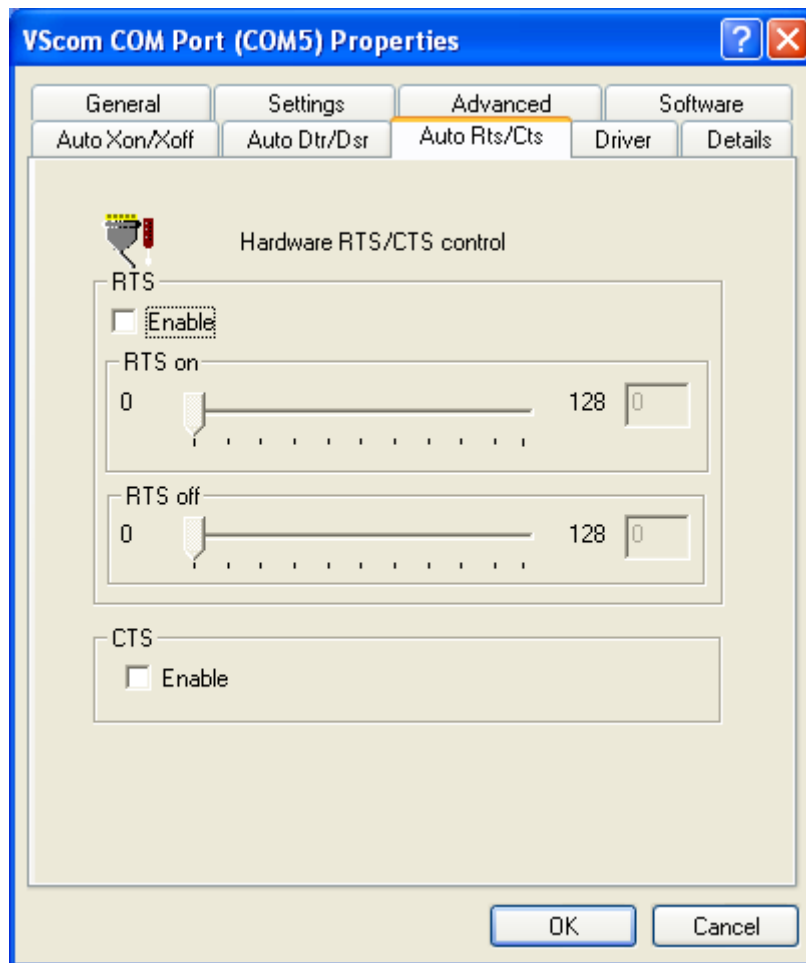


Figure 21: Auto RTS/CTS

This panel configures the On-Chip RTS/CTS Hardware Flow Control provided by the UART. The function of this panel is similar to the software Flow control panel. However, the stop and start signals are provided by the RTS and CTS signals. RTS controls the input (receive) direction, while CTS controls output.

To get access to this panel configure Hardware Flow Control in the Standard Port Settings (see 4.2.3), and leave it enabled in the driver.

Basics of hardware flow control: When CTS is asserted by the other side, transmission is allowed. If deasserted, transmission is halted. When the system is ready to receive data, it asserts the RTS line to the other side. If it needs a pause, it deasserts RTS.

RTS Enable Selects the On-Chip receive Flow control.

RTS on Function is similar to the Xon slider from [above](#).

RTS off Function is similar to the Xoff slider from [above](#).

CTS Enable Selects the On-Chip transmit Flow control, similar to the [Transmit Control above](#).

Automatic Hardware Flow Control by DTR/DSR

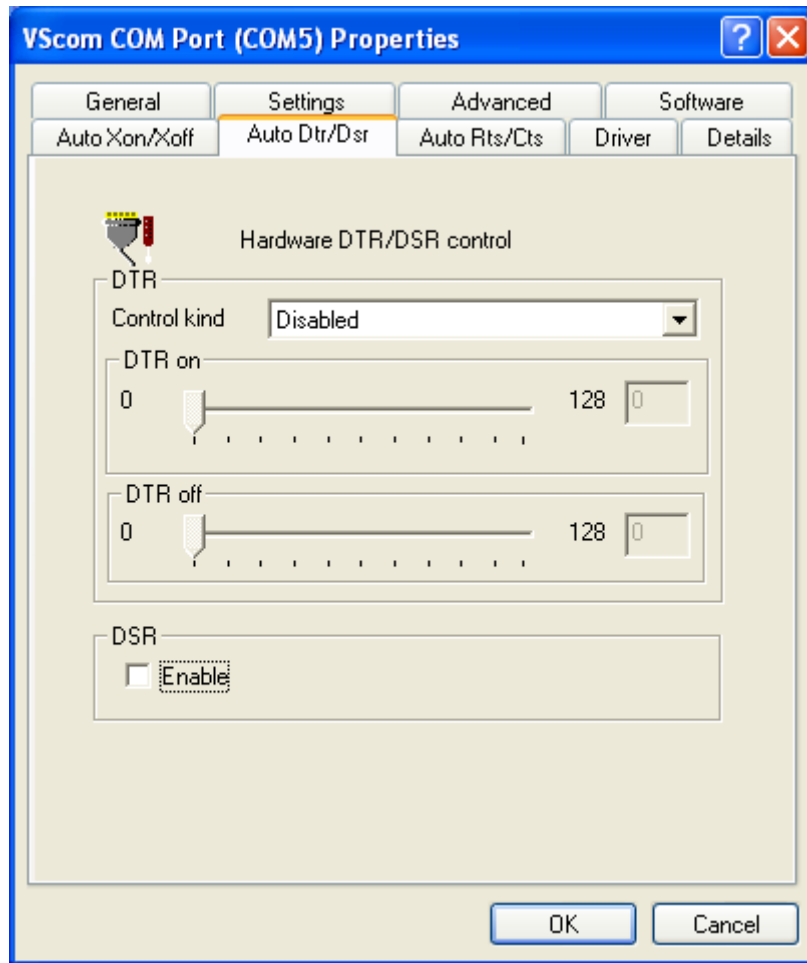


Figure 22: Auto DTR/DSR

This panel selects the function of DTR/DSR Hardware Flow control. Other signals than RTS or CTS may be used to control transmit and receive of data. This function of the UART offers more options.

To get access to this panel configure Hardware Flow Control in the Standard Port Settings (see 4.2.3), and leave it enabled in the driver.

Control kind How DTR is intended to operate.

Disabled: No Flow control for receive.

Triggered: Flow control similar to RTS Enable.

RS485 Low Active: This can be used to control a converter RS232 \Rightarrow RS485, see [below](#). DTR is low while data is transmitted. Required for 200Ei/200Ei-SI in RS 485 mode.

RS485 High Active: This can be used to control a converter RS232 \Rightarrow RS485, see [below](#). DTR is high while data is transmitted.

DTR on Compare the slider of RTS on.

DTR off Compare the slider of RTS off.

DSR enable Enables transmission control via the DSR input.

RS 485 operation RS 485 very often uses a shared medium (one pair of wires) to send and receive data. This requires the ability to disable the transmission unit. It must be off to receive data, and must be switched on when transmitting. After transmit is finished it must be off again as soon as possible.⁷ Very often this switching is controlled via signals like RTS or DTR. But in many environments it is difficult for an application to do this in a reasonable time. Now this option can help.

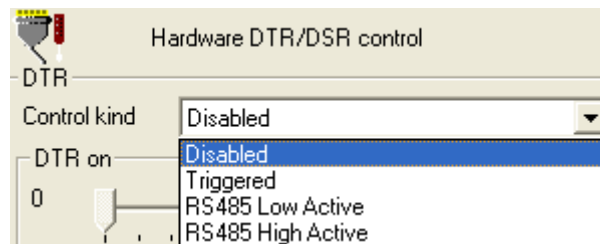


Figure 23: Auto-DTR Modes

The UART controls the DTR by itself. As long as data is in the transmission FIFO the DTR signal is held High (or Low). When the transmission FIFO is empty the DTR changes. An external converter may monitor this signal to act as needed.

RS 485 usage can be compared to communication over handheld transceiver, many people are familiar with this. To talk one has to press a button. If two or more persons talk at the same time, no one can hear clearly. For listening the button has to be released. Now the DTR operates like an “automatic” button, which presses itself as soon as one starts to talk. And it releases afterwards while the user is silent.

This feature is very useful for older applications. Often there are applications written for a half-duplex protocol⁸ over RS 232. Now this may have to operate over a longer distance. So RS 485 is chosen to transmit the data. But the application does not know about this, and does not control the RTS or DTR signal. The driver can do this invisible to the application.

4.3 Windows NT 4.0

Windows NT 4.0 is different from Windows 2000 and up, and also the no longer supported⁹ strand of 9x systems (Windows 95 up to ME). These other are all Plug & Play systems, meaning they will detect¹⁰ added hardware themselves. Users are asked to provide the required driver files. In Windows NT drivers are installed, and fail to load if the hardware is not available.

The VScom drivers for Windows NT implement the Plug & Play functions in a limited sense. There is a supervisor utility, restricted to detect VScom PCI cards.

⁷Think of it as talking by Walkie-Talkie

⁸XMODEM and YMODEM are good examples

⁹By Microsoft that is, they ceased support for all versions up to Windows 2000

¹⁰You may even configure the hardware via Windows

4.3.1 Installing in Windows NT 4.0

You need to have administrator privileges to install any new drivers under Windows NT 4.0. To install the driver or update the configuration please log on to Windows NT 4.0 as “Administrator” or ask your system administrator to install the VScom Industrial Card and driver. Please proceed with the following steps to install the driver:

1. Switch on the computer and start Windows NT 4.0.
2. Insert the supplied CD-ROM into the drive.
3. If your system is configured for Auto-start of CD-ROM, go to the products page, find your product and select driver. You’ll find a link to the NT4 drivers directory.
4. Otherwise double click on “My Computer”, and choose your CD-ROM drive.
5. Double Click “Windows NT 4.0”.
6. Double Click “PCI_Driver”.
7. Double click “vsinst.EXE”.
8. The Installer starts and requests confirmation of installation. You may change the target directory for the software. If there is a space in the name, please use the short name (e.g. PROGRA~1 for “Program Files”) instead.
9. Click Yes, and Reboot when requested.



Automatic Detection of PCI Cards The system detects a new VScom Card when an Administrator logs on to Windows NT 4.0



Figure 24: NT Plug & Play

The default option is to immediately install and configure the new card. However the upper checkbox allows to delay the configuration to the next logon. The lower check-box disables the check for new hardware until explicitly enabled again.

Configuration via Control Panel The installation program also adds the “VScom PCI” applet to the Control Panel of WindowsNT. Open it, to get access again to the configuration panel (figure 24).



System Configuration This panel opens when a new hardware is detected, or when the configuration is explicitly opened via the control panel.

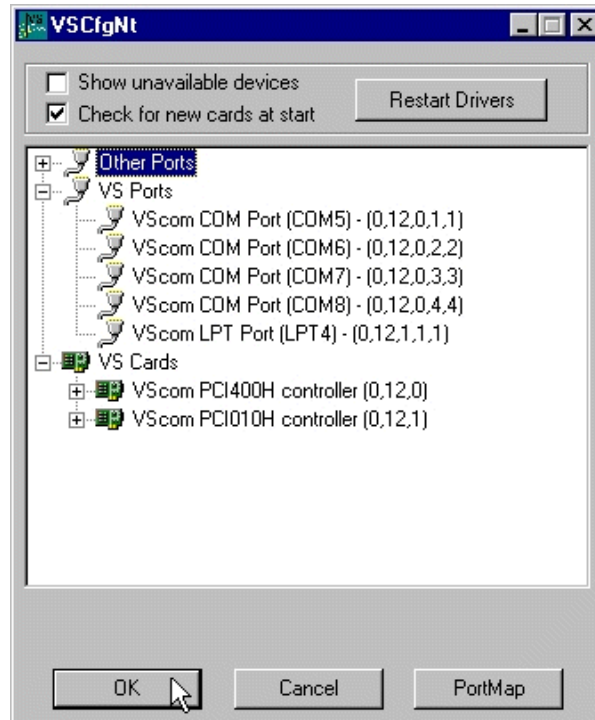


Figure 25: NT Device list

This is the basic configuration panel for the VScom PCI driver in Windows NT 4.0; the lower check-box re-enables the automatic check for new hardware. Here is the place to revoke the change done in the detection panel.

The upper check-box enables to uninstall the drivers for already removed cards, without a complete uninstall of the drivers. Different from other Windows, in NT all installed drivers load and check for there hardware. This is not efficient when the hardware is removed some time ago, and causes error messages in the event viewer. The properties of each installed VScom PCI card and its ports are associated with the PCI slot where the card is placed. When a card is removed or displaced, it is necessary to display the settings via this check-box. Then the board may be selected and removed.

Card and Port Configuration Open the panel as shown in figure 25. Select a VScom controller under “VScom Cards” to gain access to the card properties. Or select an individual port under “VScom Ports” to change its settings. The options are explained in the general sections of this documentation.

Uninstall Drivers from Windows NT To remove installed files and Windows registry information

1. Go to the Control Panel.
2. Open the “Add/Remove programs” applet.
3. Select “Vscm drivers for ...”.
4. Click “Remove”.
5. “VScm install” requests to confirm the uninstall, Click “Yes”.
6. Reboot the computer when requested.

However WindowsNT4.0 does not perform such an automated search. Users have to run the ‘VSINST.EXE’ program from the driver set, to install the cards. This installs the drivers to the system, and also a service for automatic detection of VScm PCI cards. Restricted to these cards, the usage is as handy as with modern Windows versions.

Checking Installation in Windows NT You can now verify the installation by looking at the “Administrative Tools” section of the “Windows NT Diagnostics (also called WinMSD)” (Go there by Start - Setting - Windows NT Diagnostics - Administrative Tools - Resources). There you will find new device “vscnt” listed.

Double click on it, you will see the IRQ and I/O information.

Click on the right tab, you can see all the detailed information of IRQ, Input/Output & Memory Range used by PCI Ports.

Open the “VScm PCI” applet (see 4.3.1) in the control panel (Go there by Start - Setting - Control panel). Open “VS Cards” and double click on the detected controller to open the properties. In the “advanced” tab, click the “Check” button to verify the speed setting from the hardware installation.



You may also rename the serial ports here.

5 Linux Driver

VScom PCI cards in Linux do not need a special driver. The driver 'serial' supplied with your Distribution is quite capable of controlling the serial ports on VScom cards. Driver 'serial' is configured to detect the old-fashioned ports known as Com1 to Com4 in DOS systems, and support them. Since the serial ports on VScom PCI cards use the same type of hardware, driver 'serial' is well capable to support the new serial ports. After installation of the card the driver 'serial' just needs to know about the new hardware.

Since Kernel 2.6 things are no longer as simple as written above. Some programmers invented a limit to the maximum number of serial ports. This is explained below in [5.3](#).

5.1 Manual Installation

Usually the command 'setserial' is used to provide the required information to the driver, thus adding the port.

```
setserial ttys4 port 0x100 irq 5 uart 16550A baud_base 921600
```

Such a command would be good for an ISA card, but with PCI it is not totally as easy as that. Because PCI is a Plug&Play system, the parameters 'port' and 'irq' are unknown in advance; they are assigned at boot time. Depending on your installation of Linux, you find the basic parameters in /proc/pci or via the lspci command. Given some hardware details about the card, port and irq for each serial port are calculated easy. These details are seen in [table 14](#).

This is just mentioned here, the better way of installation is automatic.

5.2 Automatic Installation

Depending on your version of the driver 'serial', some VScom PCI cards are already known. So the ports are detected, installed and usable. If the card in question is not already known to the driver (new model or old driver/kernel), the program 'vscardcfg' is supplied. This program scans /proc/pci or other information for parameters of VScom PCI cards.

5.2.1 Known Cards

PCI components are detected by BIOS and system routines at boot time. Later they are basically identified by the parameters Vendor- and Device-ID, on some hardware also by SubVendor- and SubDevice-ID. For identification of VScom PCI hardware, the Sub-parameters are not required. Not each specific hardware model has its unique identification, but at least for purposes of driver configuration different identification is supplied. So driver and configuration software just has to look up the identification in a table, and then controls the card perfectly.

If the VScom PCI card is known to the driver 'serial', users do not have to do any extra configuration. The system detects the card, identifies it and installs additional ports.

5.2.2 New Models

If the model of VScom PCI card is new to the driver, users add the program `vscardcfg` to their init scripts. This program detects the cards and the assigned parameters. Since this program knows about all VScom PCI cards, it also knows the required parameters for the `setserial` command mentioned in section 5.1. Then it performs the same tasks as users would do when manually invoking `setserial`. Especially the parameters `'port'`, `'irq'`, `'uart'` and `'baud_base'` are given according to the model and configuration.

5.3 Problems since Kernel 2.6

In Linux systems based on Kernel 2.4 or earlier the installation of VScom PCI cards has been quite easy. This has changed since kernel 2.6 came out. Not because the driver `'serial'` was removed or could no longer support PCI serial ports. This happened because some programmers invented a limit to the maximum number of serial ports installed by driver `'serial'`. And they selected a default value ridiculously low (typically 4), which caused problems again and again. So over time they have been forced to offer several methods of bypassing this limit.

5.3.1 Compile a new Kernel

In early versions the driver `'serial'` required a configuration with a new limit. Then it had to be compiled with this limit. Since usually the driver is not loaded as a module, this required to compile the kernel.

Following kernel compile settings have to be set		
<code>CONFIG_SERIAL_8250_NR_UARTS=32</code>	Maximum number of 8250/16550 UARTs	32
<code>CONFIG_SERIAL_8250_RUNTIME_UARTS=32</code>	Number of 8250/16550 serial ports to register at runtime	32
<code>CONFIG_SERIAL_8250_EXTENDED=y</code>	Extended serial port options	mark
<code>CONFIG_SERIAL_8250_MANY_PORTS=y</code>	Support more than 4 legacy serial ports	mark
<code>CONFIG_SERIAL_8250_SHARE_IRQ=y</code>	Support for sharing serial interrupts	mark

Table 13: Kernel Compile Options

Depending on your distribution there may be different methods to define these parameters. Menu-config should be a good option.

5.3.2 Kernel Start Parameters

Today the implementation of the limit is in a way which allows to raise the defined value at boot time. This is done via Kernel parameters. How this is done depends on the way you boot your system. Following is one example to do the task in Ubuntu 8.10 on AMD64 with Grub as boot loader, basically Debian should operate the same way. Other distributions or loaders may require modifications.

1. In “/boot/grub/menu.lst” is a (Comment-) line with “kopt=”-Entry and text “8250.nr_uarts”. This line shall remain there as a comment, the new value should be “8250.nr_uarts=32”. Or any number appropriate for the new configuration.
2. Run update-grub. After that check if the above parameter is available as Kernel Entry in the file “menu.lst”. The entries from the kopt-line shall appear as extension after running update-grub.
3. Copy “vscardcfg64” for 64 bit systems to “/usr/local/bin”.
4. Program “/usr/local/bin/vscardcfg64” is added to “/etc/rc.local”, to run it at boot time. You may add an “echo” to show an explaining text when the system boots.

On 32 bit systems the program is ‘vscardcfg’ of course. Additionally steps 3 and 4 are only required if ‘serial’ does not identify the card. Most models have been submitted as patch information to ‘serial’, later models will get direct support later.

5.4 Card Hardware Parameters

This is a list of all VScom cards available at time of writing. The given values will show how to identify the model, get the configuration of the card and thus extract the parameters of the serial ports.

For example in “/proc/pci” you’ll find information about a VScom 800L UPCI card like this:

```
Bus 0, device 11, function 0:
  Serial controller: Unknown vendor Unknown device (rev 0).
  Vendor id=14d2. Device id=8080.
  Medium devsel. Fast back-to-back capable. IRQ 10.
  I/O at 0xb000.      [BAR0]
  I/O at 0xa800.      [BAR1]
  I/O at 0xa400.      [BAR2]
  I/O at 0xa000.      [BAR3]
  I/O at 0x9c00.      [BAR4]
```

The text in brackets is added here as a comment. Please note Vendor- and Device-ID, as well as the IRQ. I/O addresses are given as BAR<n>, where <n> is from 0 to 5, depending on the model of the card. Serial ports are supported via I/O ranges, and all ports on one card share the same IRQ.

Each BAR may represent several serial ports, or none at all. If there is one serial port, the given address is the address of the port. If there are more serial ports, they are located at the given address, plus an offset of 8 from port to port. In this example of VScom 800L UPCI the serial ports are located at 0xA800, 0xA400, 0x9C00, 0x9C08, 0x9C10, 0x9C18, 0x9C20 and 0x9C28. The IRQ is 10, obviously.

In the table below this is given like BAR1, BAR2, BAR4:6. The table lists the card name, Vendor- and Device-ID (14d2:8080 here) and the uart model. For all VScom PCI cards baud_base is 921600, unless this is explicitly configured otherwise (chapter 3).

Card	Ident	addresses	uart	remark
200S PCI	10b5:1103	BAR1, BAR2	16550A	phased out
210S PCI	10b5:1078	BAR1, BAR2	16550A	phased out
400S PCI	10b5:1077	BAR1:4	16550A	phased out
800S PCI	10b5:1076	BAR1:8	16550A	phased out
100L	14d2:8010	BAR1	16550A	
110L	14d2:8011	BAR1	16550A	
200L	14d2:8020	BAR1, BAR2	16550A	
210L	14d2:8021	BAR1, BAR2	16550A	
400L	14d2:8040	BAR1, BAR2, BAR4:2	16550A	
420L	14d2:8042	BAR1, BAR2, BAR4:2	16550A	
430L	14d2:8043	BAR1, BAR2, BAR4:2	16550A	
800L	14d2:8080	BAR1, BAR2, BAR4:6	16550A	
100H	14d2:a001	BAR0	16950	
110H	14d2:a001	BAR0	16950	
100HV2	14d2:e010	BAR0	16950	
110HV2	14d2:e010	BAR0	16950	
200H	14d2:a005	BAR0:2	16950	
210H	14d2:a005	BAR0:2	16950	
200HV2	14d2:e020	BAR0, BAR1	16950	
210HV2	14d2:e020	BAR0, BAR1	16950	
400H	14d2:a003	BAR0:4	16950	
410H	14d2:a003	BAR0:4	16950	
800H	14d2:a003	BAR0:4	16950	ports 1-4
800H	14d2:a004	BAR0:4	16950	ports 5-8
200I	14d2:8028	BAR4:2	16550A	
200I-SI	14d2:8028	BAR4:2	16550A	
400I	14d2:8048	BAR4:4	16550A	
800I	14d2:8088	BAR4:8	16550A	

Table 14: Hardware Parameters

6 Troubleshooting

1. The board is not found during or after installation
 - a) The card is not seated properly in the PCI slot. Re-install the card in the PCI slot.
2. Made wrong selection of “Cancel” during installation. When re-install the driver Windows 95/98/2000 doesn't detect the card.
 - a) The driver installation is not complete. Install the card in another PCI slot to initialize the installation again.
 - b) Alternatively you may try to run VSCLEAN to remove the falsely installed driver.
3. During installation, NT4.0 shows blue screen (white-text-on-blue-background)
 - a) In most cases, there is an IRQ conflict. The IRQ assigned to the board is in conflict with other devices. The VScom PCI driver supports IRQ sharing. However, some devices like network card or SCSI controller don't share IRQ with other devices.
 - b) Be sure that your system has a free IRQ for the card. Enter “BIOS setup - PCI configuration”, and set all IRQs to “PCI/ISA PnP” except for those non-PnP ISA components. This will bring up the maximum number of free IRQs in the system.
 - c) Try to install the card to another PCI slot. To install the VScom PCI card to a different PCI slot will get a different IRQ assignment. Or, you can exchange the PCI slot with another card.
4. In the Event Viewer, error message shows “Another driver on the system, which did not report its resources, has already claimed the interrupt used by LPTx”
 - a) Please refer to symptom (3) about the IRQ conflict problem, and the solution.
5. Using RS 485, the data transmission has errors
 - a) Please check your BIAS and Termination resistors. Termination reduces signal reflection. BIAS helps in detecting the start of a new transmission.
 - b) Further check if the signal GND of transmitter and receiver are connected. RS 485 is often referred to as “2-wire”, however connecting the GND is required. RS 485 defines a so called “Common Voltage Range” of $-7V$ to $+12V$. The only way to ensure this parameter is by connecting pin 5 (GND) with the logic GND of the transmitter.

7 Proper Cabling in RS 485 and RS 422

This paragraph will provide a little bit of theory about RS 422 and RS 485 data transmission. It is necessary to have this basic knowledge, to avoid or find errors in data transmission. If you are familiar with RS 485 and RS 422, do *not* skip this section, it covers common misunderstandings.

Failures in cabling are responsible for the vast majority of transmission problems.

7.1 Transmission Technique

RS 422 and RS 485 use the same balanced transmission method. Signals are not transmitted as voltage on a single wire, as RS 232 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 pair cables are used, which further reduces the sensitivity for noise.

Typical voltages are +4V as high, about +0.5V as low. These voltages are defined against the GND signal of the transmitter. The minimum *differential* voltage is required as $\pm 200\text{mV}$ by the specifications of RS 422 and RS 485. The receiver detects the polarity of the differential voltage, and thus gets a Zero (negative) or a One (positive).

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*. To make sure the signals meet the common voltage range, the GND of sender and receiver must be connected somehow. To insure the signals are in the valid voltage range and the differential voltage can be correctly sensed by the receiver the GND lines of the transmitter and receiver must be connected. This is often forgotten. In these cases the GND connection can be indirectly established over protective ground or over "parasite ground loops", which usually yield a bad GND connection. For these situation the correct data receiving is not reliable or not possible.

7.2 Termination

When transmitted signals arrive at the end of a cable, they get reflected. They travel on the cable 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 . The VScom Industrial Card provides an option to activate such resistors by jumper or DIP switch, depending on the model.

As a rule of thumb¹¹, when the cables are longer than $\frac{1000000}{\text{Bitrate}}$ (one million divided by the bit rate) in meters, you should consider Termination Resistors.

7.3 Polarization

In RS 485 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. The VScom Industrial Card provides an option to activate such resistors. They are calculated for termination resistors of 120Ω .

¹¹Assuming group speed of 100.000 km/s, 10 travels to damp out, and 10% of bit time

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

With the isolated models the polarization option may have additional advantage. As said above RS 485 requires a connection of GND signals, otherwise the signals at the receiver may be out of bounds of the Common Voltage Range. Since on SI-models the driver and polarization voltage are isolated, the received signals are now *in* the required range. As a result only two wires are required for data transmission.

Attention: this is the situation if the isolated VScom Industrial Card is the only device with polarization active.

7.4 2-Wire Scheme

In many configurations a very simple cabling is required. RS 485 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 or receive data at a given time, so it is operating in half-duplex mode.

Shown in figure 26 are three devices, RS 485 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 required¹².

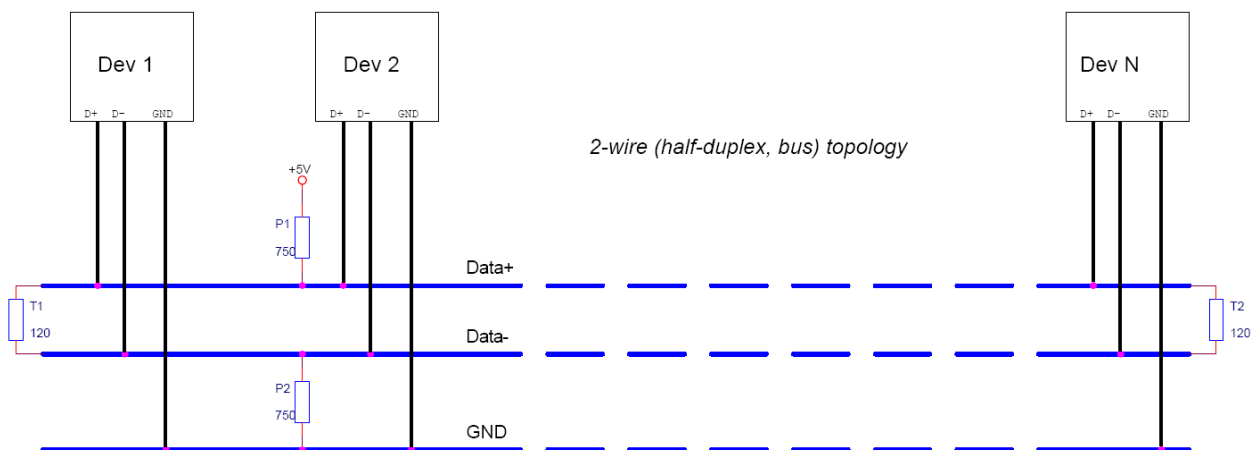


Figure 26: 2-Wire Cabling Scheme

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 RS 485 addresses are defined by that protocol, as well as bus access.

¹²See 7.3 for isolated VScom Industrial Card

7.5 4-Wire Scheme

RS 422 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 RS 422. Since the VScom Industrial Card uses RS 485 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 RS 232 connections.

In RS 485 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.

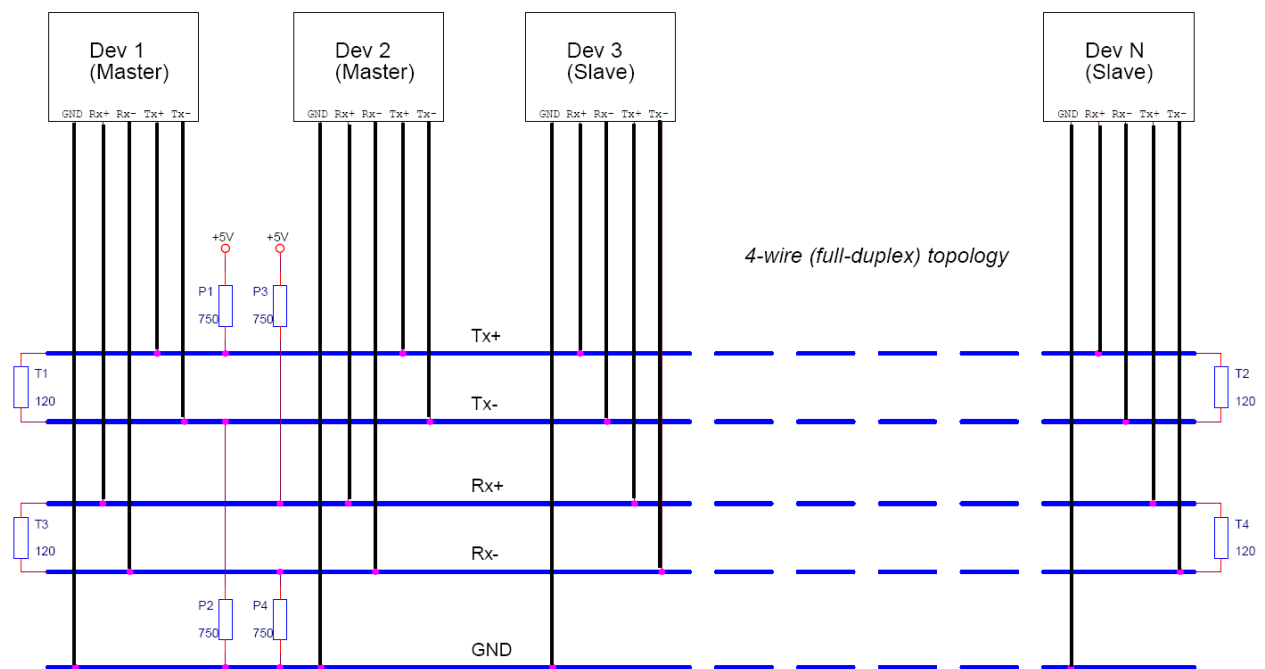


Figure 27: 4-Wire Cabling Scheme

Figure 27 displays the wire pairs Tx_{\pm} and Rx_{\pm} as named for Devices 1 and 2. There are more slave devices, two are shown. The slaves transmit on the Rx-lines, and receive on the Tx-lines. The GND line is also connected, to implement the Common Voltage Range as specified¹³.

Masters on the network are identified because they transmit on the Tx-lines. The two Masters in figure 27 have to synchronize their use of the Tx-lines by extra means. E.g. Dev 2 can be a backup master, which is manually activated.

Also indicated in figure 27 are polarization resistors P1/P2 for Tx_{\pm} and P3/P4 for Rx_{\pm} , only once per wire pair. Further Termination resistors T1/T2 for Tx_{\pm} and T3/T4 for Rx_{\pm} are added, one at each end of the cable. To ensure the Common Voltage Range the GND is also connected at each device.

¹³See 7.3 for isolated VScom Industrial Card

8 Connector Definitions

The VScOm Industrial cards provide various connectors for the signals of the serial ports. Either DB9 male connectors are directly available on the card, or a common connector for all ports is mounted on the bracket. This way each model only needs one slot to fit in.

The signal assignment for RS 232 on DB9 male connectors is defined by the term “RS 232” already. However there is no such standard for RS 422 or RS 485 exists. Even a de-facto standard for industrial connectors is not available. Some manufacturers selected similar assignments, some invented their own definition.

Below is the signal definition for all connectors used by the VScOm Industrial cards. RS 232 signals are mentioned for completeness. RTS and CTS signals are not given for RS 485 modes, even when they are available. They are not mentioned because they can not be used anyway, RS 485 usually connects more than two devices/stations with each other. In Half-Duplex mode (“2-wire”) the same pair of wires is used to transmit and receive. So these signals change their name to “Data”.

8.1 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	RS 232	RS 422	RS 485 “4w”	RS 485 “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 15: DB9 male Connector

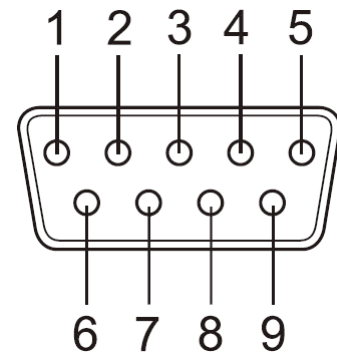


Figure 28: DB9 male Connector

8.2 DB37 female

This common connector is for four serial ports. So each signal name appears four times, it will have an index like (3) for the third serial port of the card. The serial ports are neither isolated nor isolated from each other, so GND is common and has no index. Pin 1 is a no-connect pin.

Pin	RS 232	RS 422	RS 485 “4w”	RS 485 “2w”
1	.nc.			
2	DCD (3)	Tx- (3)	Tx- (3)	Data- (3)
3	GND	GND	GND	GND
4	CTS (3)	CTS+ (3)		
5	RxD (3)	Tx+ (3)	Tx+ (3)	Data+ (3)
6	RI (4)	CTS- (4)		
7	DTR (4)	Rx- (4)	Rx- (4)	
8	DSR (4)	RTS- (4)		
9	RTS (4)	RTS+ (4)		
10	TxD (4)	Rx+ (4)	Rx+ (4)	
11	DCD (2)	Tx- (2)	Tx- (2)	Data- (2)
12	GND	GND	GND	GND
13	CTS (2)	CTS+ (2)		
14	RxD (2)	Tx+ (2)	Tx+ (2)	Data+ (2)
15	RI (1)	CTS- (1)		
16	DTR (1)	Rx- (1)	Rx- (1)	
17	DSR (1)	RTS- (1)		
18	RTS (1)	RTS+ (1)		
19	TxD (1)	Rx+ (1)	Rx+ (1)	
20	RI (3)	CTS- (3)		
21	DTR (3)	Rx- (3)	Rx- (3)	
22	DSR (3)	RTS- (3)		
23	RTS (3)	RTS+ (3)		
24	TxD (3)	Rx+ (3)		
25	DCD (4)	Tx- (4)	Tx- (4)	Data- (4)
26	GND	GND	GND	GND
27	CTS (4)	CTS+ (4)		
28	RxD (4)	Tx+ (4)	Tx+ (4)	Data+ (4)
29	RI (2)	CTS- (2)		
30	DTR (2)	Rx- (2)	Rx- (2)	
31	DSR (2)	RTS- (2)		
32	RTS (2)	RTS+ (2)		
33	TxD (2)	Rx+ (2)	Rx+ (2)	
34	DCD (1)	Tx- (1)	Tx- (1)	Data- (1)
35	GND	GND	GND	GND
36	CTS (1)	CTS+ (1)		
37	RxD (1)	Tx+ (1)	Tx+ (1)	Data+ (1)

Table 16: DB37 female Connector

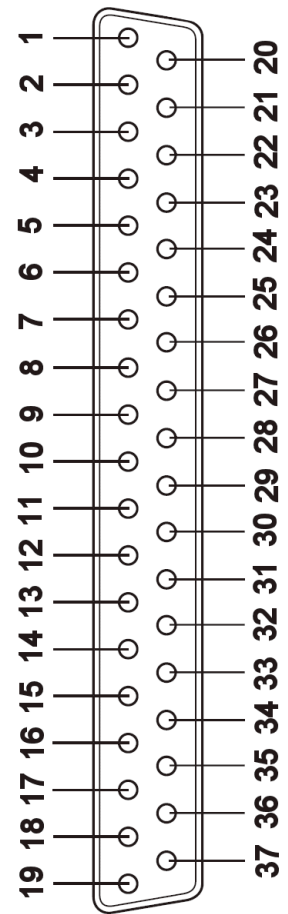


Figure 29: DB37 female Connector

8.3 DB78 female

This common connector is for eight serial ports. So each signal name appears eight times, it will have an index like (3) for the third serial port of the card. The serial ports are neither isolated nor isolated from each other, so GND is common and has no index. “.nc” is a no-connect pin.

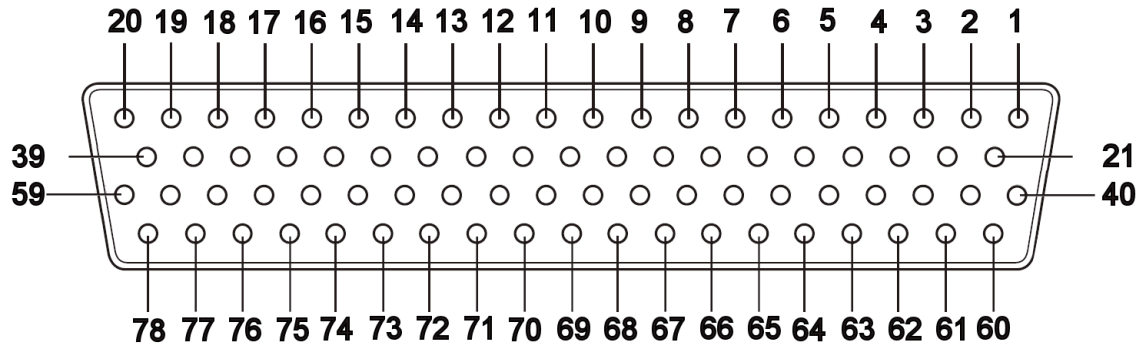


Figure 30: DB78 female Connector

The description is split in two tables, instead of one common. The first table 17 is for the RS 485 signals. “no-connect” and signals not useful in these modes are not shown to make the table shorter. The second table 18 defines the signal assignment for RS 232 and RS 422 modes, as well as GND and “no-connect”.

Pin	RS 485 “4w”	RS 485 “2w”	Pin	RS 485 “4w”	RS 485 “2w”
1	Rx- (8)		40	Rx+ (5)	
2	Rx+ (6)		43	Tx- (5)	Data- (5)
3	Rx- (6)		46	Tx+ (7)	Data+ (7)
6	Tx- (8)	Data- (8)	48	Tx- (7)	Data- (7)
8	Tx+ (6)	Data+ (6)	49	Rx- (1)	
10	Rx+ (4)		50	Rx+ (2)	
11	Rx+ (3)		52	Rx- (4)	
13	Rx- (3)		55	Tx+ (1)	Data+ (1)
17	Tx+ (2)	Data+ (2)	56	Tx+ (4)	Data+ (4)
18	Tx- (4)	Data- (4)	61	Rx- (7)	
22	Rx- (5)		63	Rx+ (7)	
23	Tx- (6)	Data- (6)	64	Rx+ (8)	
27	Tx+ (8)	Data+ (8)	68	GND	GND
28	Tx+ (5)	Data+ (5)	69	GND	GND
30	Rx+ (1)		70	GND	GND
32	Rx- (2)		71	GND	GND
33	Tx- (2)	Data- (2)	72	GND	GND
35	Tx- (1)	Data- (1)	73	GND	GND
37	Tx+ (3)	Data+ (3)	74	GND	GND
39	Tx- (3)	Data- (3)	75	GND	GND

Table 17: DB78 female Connector RS 485

Pin	RS 232	RS 422	Pin	RS 232	RS 422	Pin	RS 232	RS 422
1	DTR (8)	Rx- (8)	27	RxD (8)	Tx+ (8)	53	CTS (2)	CTS+ (2)
2	TxD (6)	Rx+ (6)	28	RxD (5)	Tx+ (5)	54	DSR (1)	RTS- (1)
3	DTR (6)	Rx- (6)	29	DSR (7)	RTS- (7)	55	RxD (1)	Tx+ (1)
4	CTS (6)	CTS+ (6)	30	TxD (1)	Rx+ (1)	56	RxD (4)	Tx+ (4)
5	DSR (5)	RTS- (5)	31	RTS (2)	RTS+ (2)	57	CTS (4)	CTS+ (4)
6	DCD (8)	Tx- (8)	32	DTR (2)	Rx- (2)	58	DSR (3)	RTS- (3)
7	RI (8)	CTS- (8)	33	DCD (2)	Tx- (2)	59	CTS (3)	CTS+ (3)
8	RxD (6)	Tx+ (6)	34	DSR (2)	RTS- (2)	60	RTS (8)	RTS+ (8)
9	CTS (7)	CTS+ (7)	35	DCD (1)	Tx- (1)	61	DTR (7)	Rx- (7)
10	TxD (4)	Rx+ (4)	36	RI (1)	CTS- (1)	62	RTS (7)	RTS+ (7)
11	TxD (3)	Rx+ (3)	37	RxD (3)	Tx+ (3)	63	TxD (7)	Rx+ (7)
12	RTS (3)	RTS+ (3)	38	DSR (4)	RTS- (4)	64	TxD (8)	Rx+ (8)
13	DTR (3)	Rx- (3)	39	DCD (3)	Tx- (3)	65	.nc.	
14	RTS (4)	RTS+ (4)	40	TxD (5)	Rx+ (5)	66	.nc.	
15	RI (2)	CTS- (2)	41	RTS (6)	RTS+ (6)	67	.nc.	
16	CTS (1)	CTS+ (1)	42	DSR (6)	RTS- (6)	68	GND	GND
17	RxD (2)	Tx+ (2)	43	DCD (5)	Tx- (5)	69	GND	GND
18	DCD (4)	Tx- (4)	44	RI (5)	CTS- (5)	70	GND	GND
19	RI (4)	CTS- (4)	45	CTS (8)	CTS+ (8)	71	GND	GND
20	RI (3)	CTS- (3)	46	RxD (7)	Tx+ (7)	72	GND	GND
21	RTS (5)	RTS+ (5)	47	RI (7)	CTS- (7)	73	GND	GND
22	DTR (5)	Rx- (5)	48	DCD (7)	Tx- (7)	74	GND	GND
23	DCD (6)	Tx- (6)	49	DTR (1)	Rx- (1)	75	GND	GND
24	RI (6)	CTS- (6)	50	TxD (2)	Rx+ (2)	76	.nc.	
25	CTS (5)	CTS+ (5)	51	RTS (1)	RTS+ (1)	77	.nc.	
26	DSR (8)	RTS- (8)	52	DTR (4)	Rx- (4)	78	.nc.	

Table 18: DB78 female Connector RS 232, RS 422

9 History

April 2009 First Release of this manual

February 2010 Installation in Windows 7 added

March 2011 Extended explanation on how to enable automatic handshaking

July 2011 Mistake corrected in configuration RS 485 2-wire on 200I/200I-SI