WITIO-PCI64
EXTENDED

EDP No.: A-461800

64 TTL Inputs via 74LS244
(8 Inputs interrupt capable)
64 TTL Outputs via 74ABT273
3 * 16-bit Counter - interrupt capable

wasco®

user‘s guide

Registered Trademarks

IBM PC, PC/XT and PC/AT are registered trademarks of International Business Machines (IBM).
BASIC is registered trademark of Dartmouth College.
Turbo Pascal, Turbo C are registered trademarks of Borland. Quickbasic is registered trademark of Microsoft. Powerbasic is registered trademark of Robert S. Zale.
wasco® is registered trademark.

Disclaimer

The company Messcomp Datentechnik GmbH assumes no liability for the use of the interface card WITIO-PCI64EXTENDED and this documentation, neither for direct nor indirect damages.

wasco®

Table of Contents

1. Introduction

WITIO-PCI64EXTENDED features 64 digital inputs and 64 digital outputs, all of which are TTL compatible. The maximum loading capacity amounts to 20 mA. This board is suitable for input and output applications not requiring galvanic isolation. The internal data bus of this board is organized 32 bit, each reading or writing access to the inputs and outputs is implemented by double word access. You can trigger interrupts via eight out of the 64 inputs or time dependently via a timer chip combined with a quartz oscillator. Two 68-pin SCSI-II sockets enable connection to periphery. 32 inputs and 32 outputs are led to each of the two sockets. One of the sockets is mounted to the board‘s slot bracket, the other one is placed directly on-board. An optional available set of flat ribbon cable and appropriate plugs enables to connect to a SCSI-II socket with slot bracket.

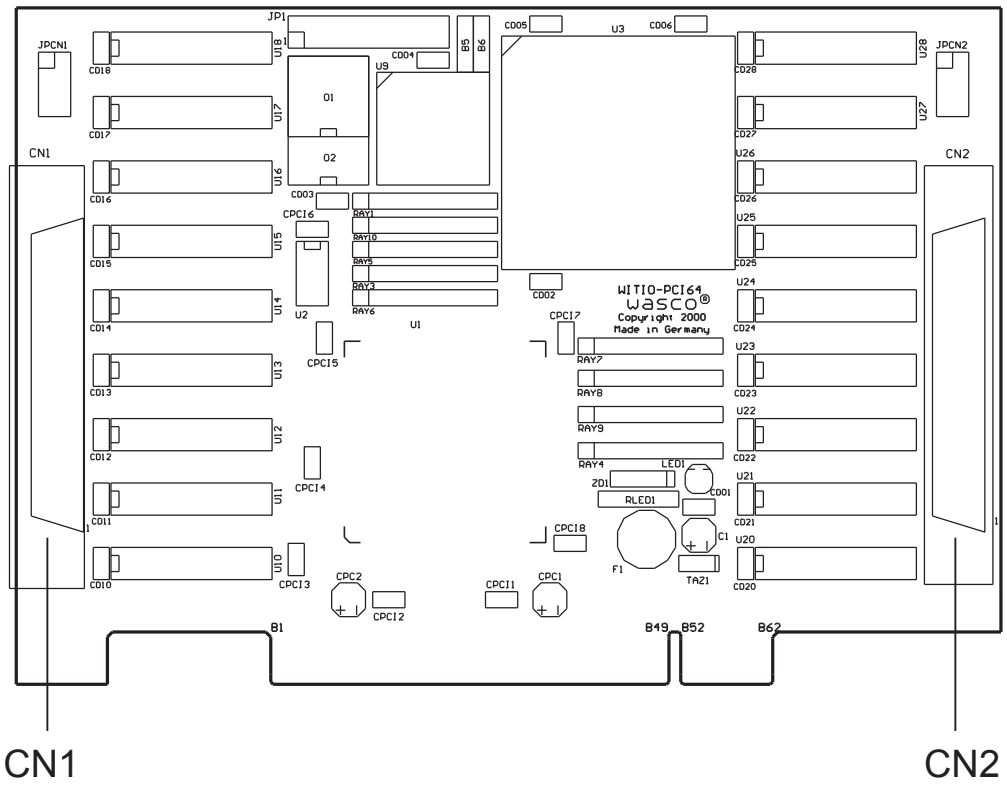
2. Installation of the WITIO-PCI64EXTENDED

2.1 How to install the circuit board into your system

Before you insert the WITIO-PCI64 unplug the power plug or at least turn off the computer to make sure, there is no current to/in the computer. Inserting the interface card in a running system may cause damaging or destroying the WITIO-PCI64 and even other previously installed boards.
Select an empty PCI slot of your computer to insert the board. Please refer to the computer‘s manual for support. Secure the WITIO-PCI64 from loss of connection by screwing the mounting bracket to the casing of your computer.

3. Connectors

3.1 Onboard position of the connector plugs

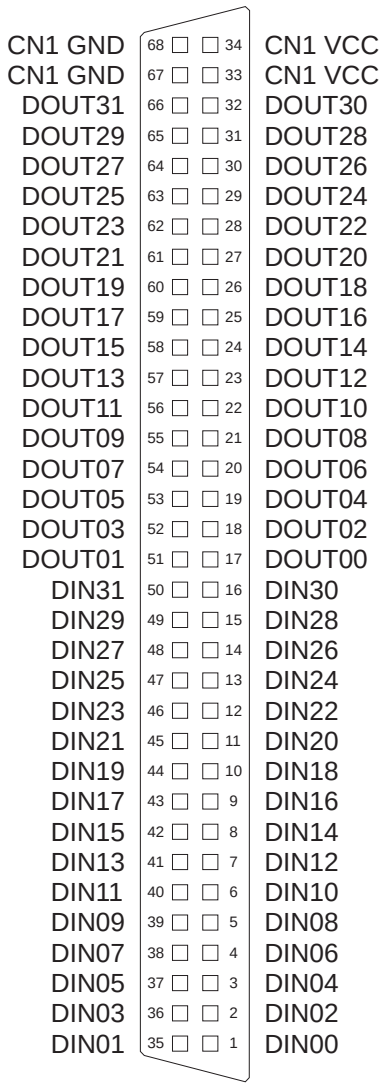


CN1 CN2

CN1: TTL inputs DIN00....DIN31
TTL outputs DOUT00....DOUT31

CN2: TTL inputs DIN32....DIN63
TTL outputs DOUT32....DOUT63

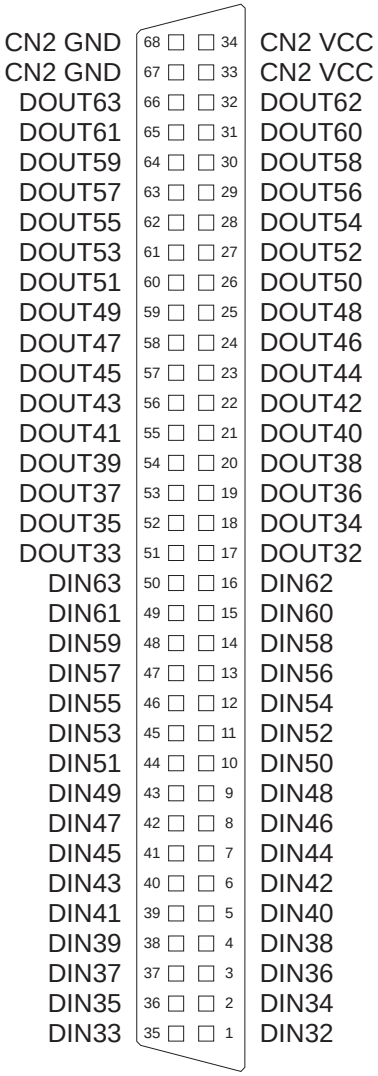
3.2 Pin assignment of CN1



Vcc:
Connector for internal power supply (+ 5V). **Never apply an external voltage across this pin.**

GND:
Ground connection

3.3 Pin assignment of CN2

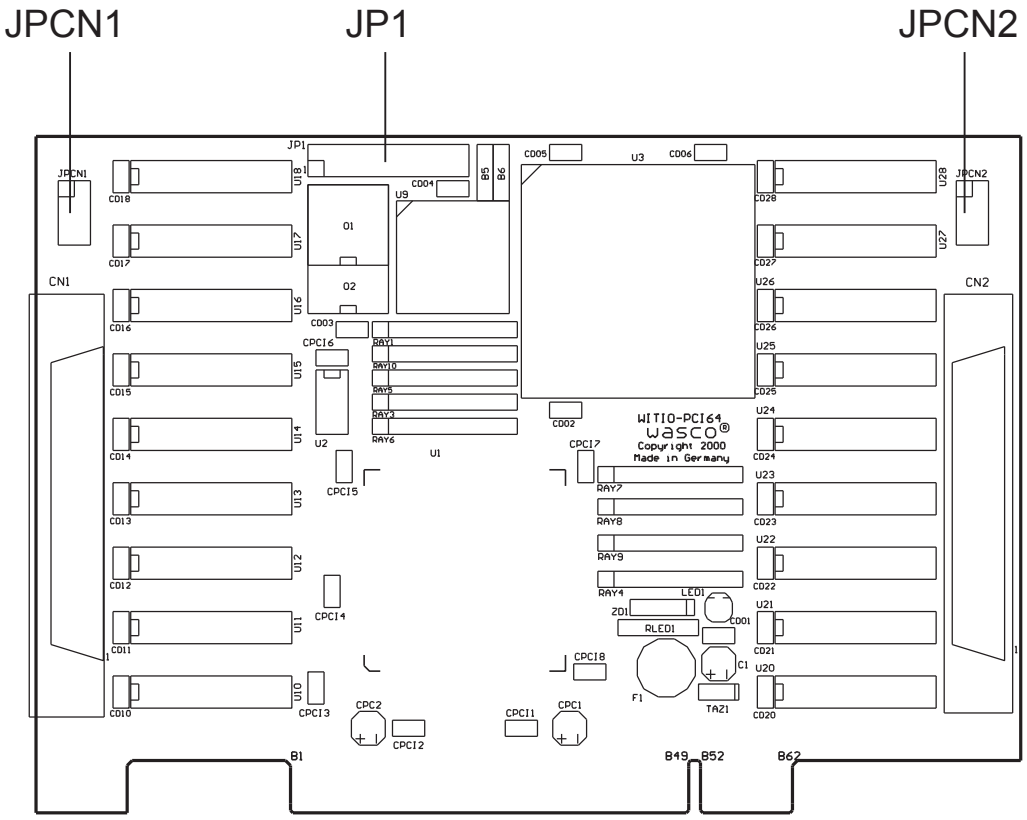


Vcc:
Connector for internal power supply (+ 5V). **Never apply an external voltage across this pin.**

GND:
Ground connection

4. Jumper Blocks

4.1 Position of the jumper on the board

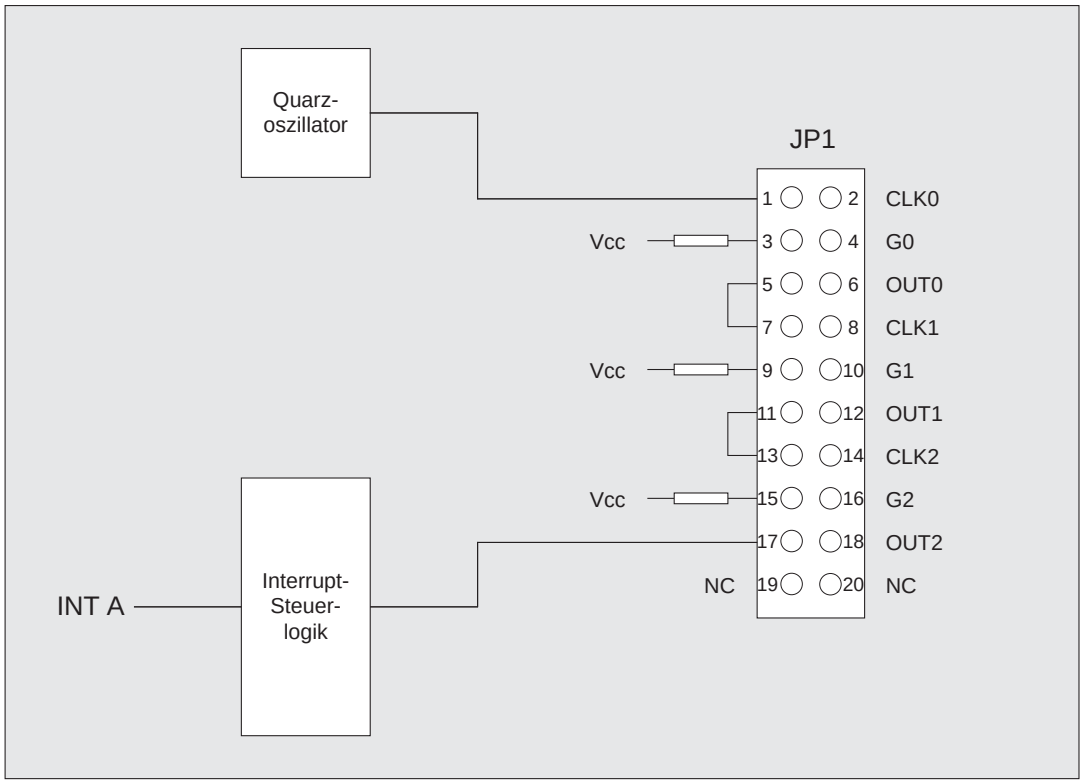


JP1: Jumper block for time dependent interrupt triggering by the timer chip

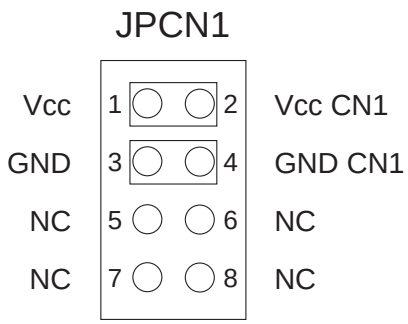
JPCN1: Connection of the PC‘s internal power supply to Sub-D socket CN1.

JPCN2: Connection of the PC‘s internal power supply to Sub-D socket CN2.

4.2 Jumper block assignment JP1

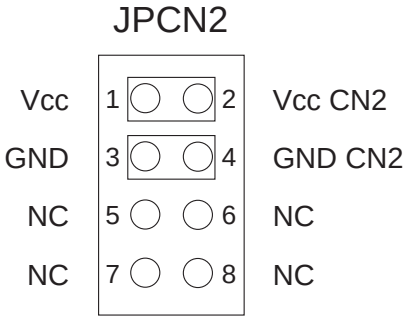


4.3 Jumper block assignment JPCN1



Plugging the jumpers over JPCN1/1-2 or 3-4 the internal power supply (+ 5V) as well as the PC‘s ground can be fed to the Sub-D socket CN1. Jumper setting at delivery: JPCN1/5-6, 7-8

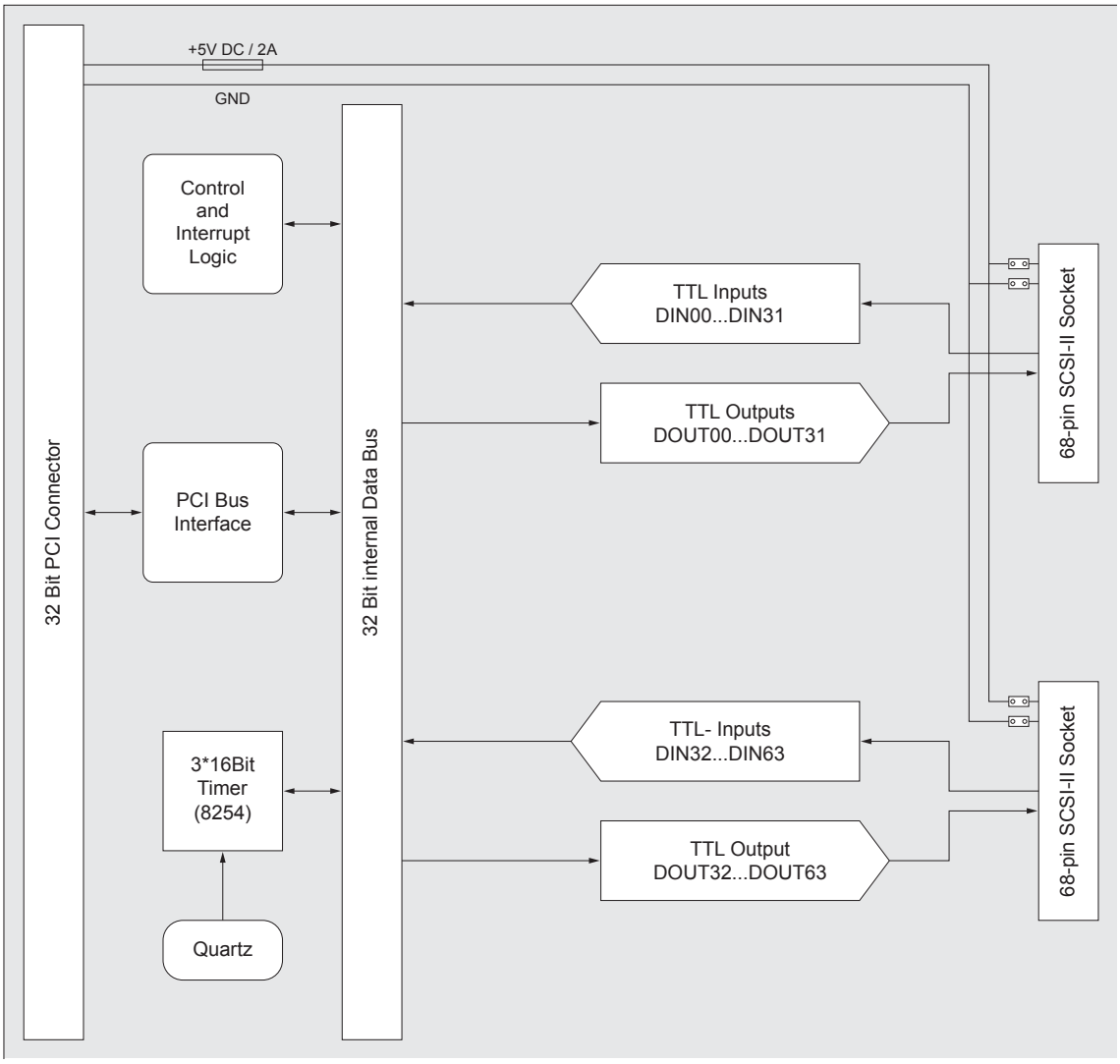
4.4 Jumper block assignment JPCN2



Plugging the jumpers over JPCN2/1-2 or 3-4 the internal power supply (+ 5V) as well as the PC‘s ground can be fed to the Sub-D socket CN2. Jumper setting at delivery: JPCN2/5-6, 7-8

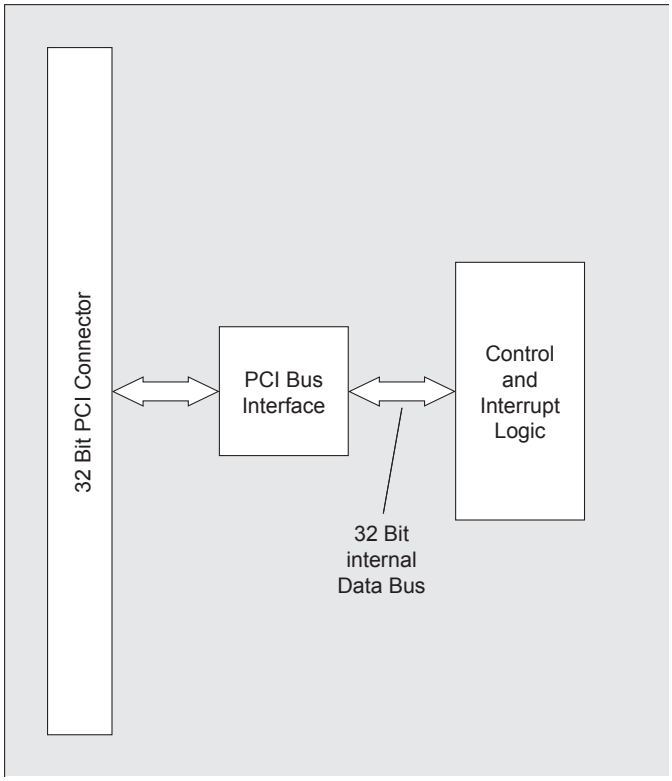
5. Sytem Components

5.1 Block diagram



5.2 Access to the system components

WITIO-PCI64 hardware components access by reading or writing in port adresses via library functions. The relevant port adresses for the WITIO-PCI64 result from the basic address, which is issued by the PCI BIOS. The port access of the WITIO-PCI64 is implemented exclusively in double-word accesses (32 bit), byte and word access is not applicable. (Please find more details in the chapter „programming“ and the sample programs on the enclosed CD).



6. 64 TTL Inputs via 74LS244

The WITIO-PCI64 features 64 TTL compatible input channels. You can trigger interrupts via eight out of the 64 inputs. Not connected or open inputs present level HIGH and should principally be fed to the ground.

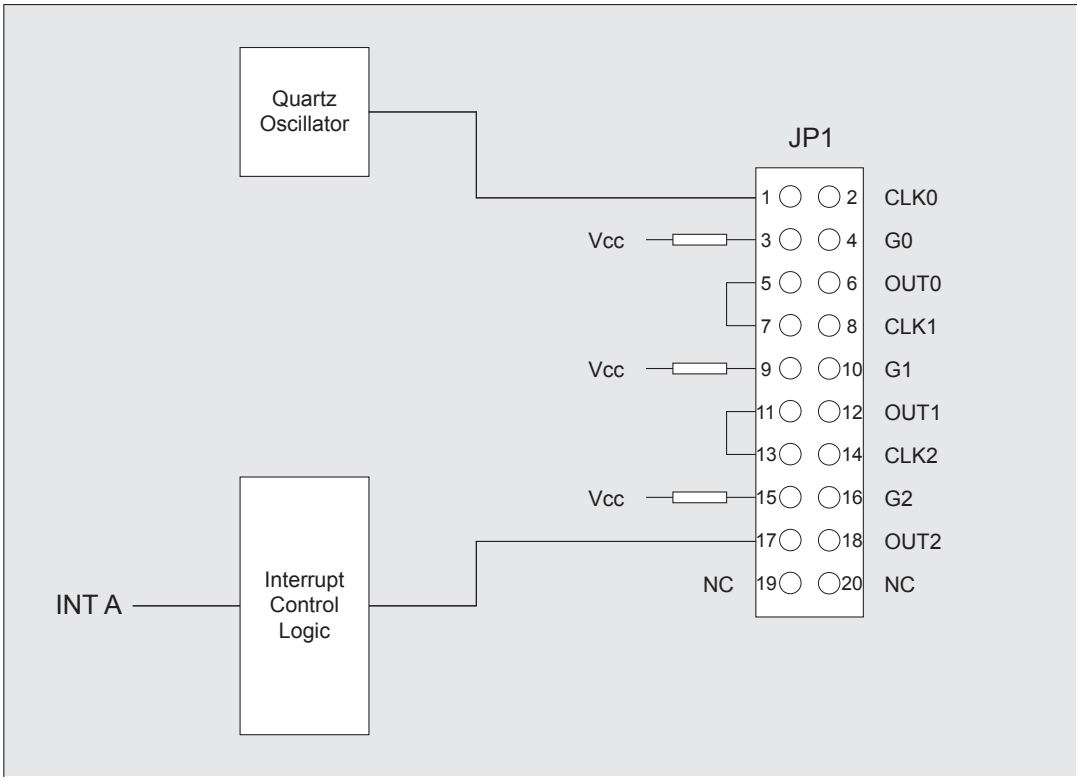
7. 64 TTL Outputs via 74ABT273

The WITIO-PCI64 features 64 TTL compatible output channels. The maximum loading capacity of each of the channels amounts to 20 mA.

8. 3 * 16-Bit Counter (Timer 8254)

WITIO-PCI64 generates time controlled interrupt triggers using a combination of 8254 timer IC and quartz oscillator. All interrupt requests are forwarded to one of the computer‘s IRQs via the PCI bus INT A. The PCI-PNP-BIOS of the computer creates a relating IRQ number by itself.

If you want the timer to generate time controlled interrupt triggers, you have to plug the relating jumpers over jumper block JP1. So please close all jumper connections 1-2 through 17-18 on JP1.
If you program the counter appropriately you can generate the signal for interrupt triggering (output OUT2 of counter 2) and forward it to the interrupt logic. The transition of logic level HIGH to logic level LOW at OUT2 induces interrupt requests.



8.1 Counters 0, 1 and 2

Each of the three counters is a 16-bit down counter, optionally usable for either decimal or dual counting. All three counters are entirely identical in operation. The counters can perform their tasks completely independently and you can use it in different operation modes simultaneously.
The control word defines the function of GATE, input and data output.
The control word‘s format contains special codes for loading the counter‘s initial value. You can read the counter content with a simple read command if used as an event counter.

8.2 Control Word Register

The control word register can only be written to. Selecting "High" at the address lines A0 and A1 the register is selected. The Control Word Register imports incoming data via data bus interface. The programmed information is used to define how the counter logically works (dual, decimal), the counter‘s mode of operation and the loading of the counter register.

8.3 Read/Write Logic

The microprocessor sends control signals via the system‘s data bus. The read/write logic accepts the signals and generates the internal control signals necessary for proper functional sequence. Setting /CS you can either release or latch read/write logic. To change functions via system software you must select the timer prior.

8.4 Read/Write Logic

/CS	/RD	/WR	A1	A0	
0	1	0	0	0	Write into counter 0
0	1	0	0	1	Write into counter 1
0	1	0	1	0	Write into counter 2
0	1	0	1	1	Write control word
0	0	1	0	0	Read from counter 0
0	0	1	0	1	Read from counter 1
0	0	1	1	0	Read from counter 2
0	0	1	1	1	No-operation, data bus interface with high impedance
1	X	X	X	X	Timer latched, data bus interface with high impedance
0	1	1	X	X	No-operation, data bus interface with high impedance

X = don‘t care

You can determine the mode of operation of each counter via software by simple output commands. Each of the three counters has to be initialized individually by a control word written into the Control Word Register. To trigger interrupt time-controlled the three counters are to be programmed in operation mode 3.
As the timer used on this board only is intended for time-controlled interrupt triggering we do not expand on the other potential operation modes!

8.5 Control Word Format

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

8.6 Counter Selection

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	read back

8.7 Modes of operation

M2	M1	M0	Mode
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5

8.8 Counting dual or decimal

BCD	Count mode
0	dual (16-bit binary counter)
1	decimal (4-digit BCD counter)

8.9 Read/Load

RW1	RW0	Read/Load
0	0	Counter Latch Command
0	1	least significant bytes
1	0	most significant bytes
1	1	firstly least significant bytes, then most significant bytes

8.10 Mode 3

Square Wave Generator

If started with even initial count values the counter output remains „High“ until half the initial count has been completed. During the second half-cycle the output goes "Low".
If the initial value (n) is odd for (n+1)/2 counts the output goes "High" and for (n-1)/2 counts it goes "Low". While current counting when you load a new initial value the counting reloads to start from the new value on the next transition of logic level.

8.11 Overview GATE input

Mode	GATE Signal		
	L level or falling to L level	**Rising Edge**	**H level**
0	disable counter	-	enable counter
1	-	1. start counter 2. set output to L at next clock cycle	-
2	1. disable counter 2. set output to H immediately	start counter	enable counter
3	1. disable counter 2. set output to H immediately	start counter	enable counter
4	disable counter	-	enable counter
5	-	start counter	-

8.12 Programming of the Timer

You can load the count register by setting a control word, which determines the operation mode and the initial counter value by its number of bytes sent.

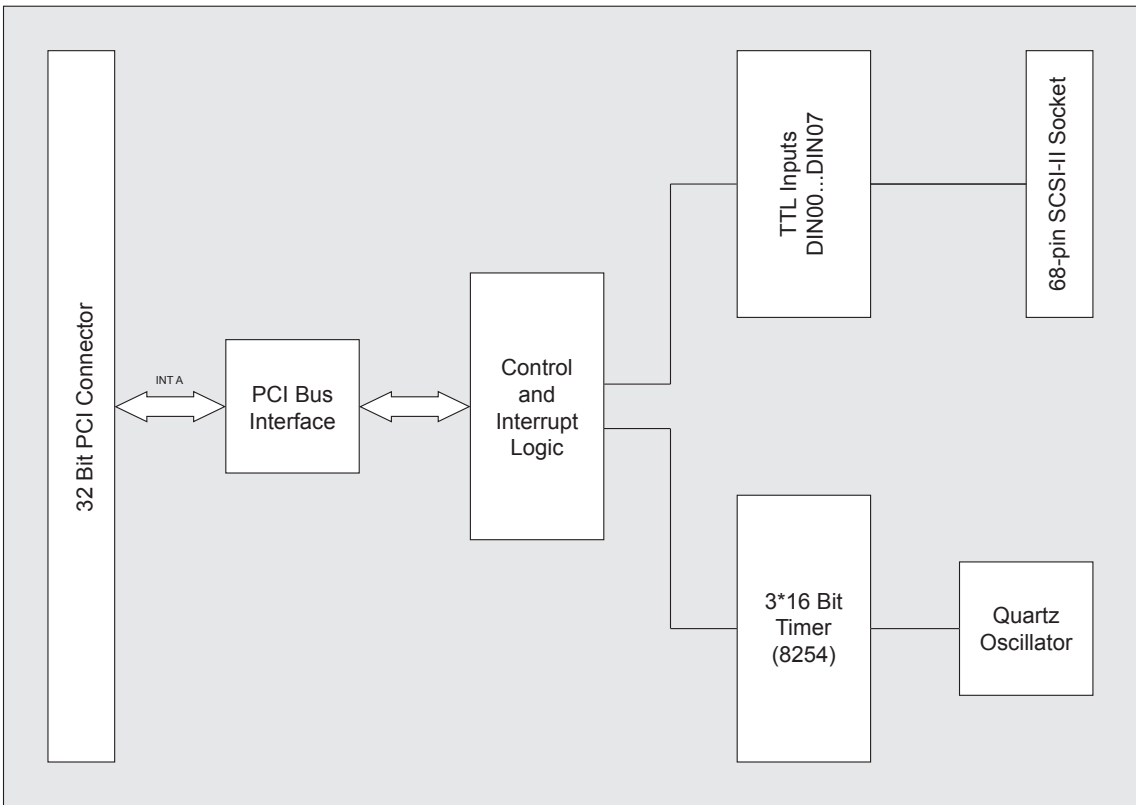
The control words can be written in any order since each of the counters has its own control register with individual address lines (SC0, SC1). You can load counter in any order as well. Please note, the number of bytes (most-significant bytes, MSB, and least-significant bytes, LSB) set in the control word must follow the number of bytes loaded into the counter. And it is necessary to pay attention to the proper order defined by RL0 and RL1 in the control word when loading the initial counter value. You can write the one or two bytes of the initial counter value into the counter register at any time, but not more than set in the control word. If the counter register only is filled with zeros the maximum of count cycles for count down results. This is 2^{16} cycles at dual counting and BCD 10^4 at decimal counting.

9. Interrupt

9.1 Interrupt requests

WITIO-PCI64 enables to generate interrupt triggers via eight TTL inputs DIN00...DIN07, as well as time-controlled triggering by combining the timer IC 8254 and a quartz crystal oscillator. All interrupt requests are forwarded to one of the PC‘s IRQs via PCI bus wire INT A. The PCI-PNP-BIOS of the computer issues the relevant IRQ number autonomously.

The following block diagram pictures the communication between the interrupt inputs of the WITIO-PCI64 and the PC:



9.2 Interrupt triggering via TTL inputs

Interrupt block:
[TTL inputs DIN00...DIN07]

-> Interrupt request when level goes from LOW to HIGH at at least one of the TTL inputs DIN00...DIN07

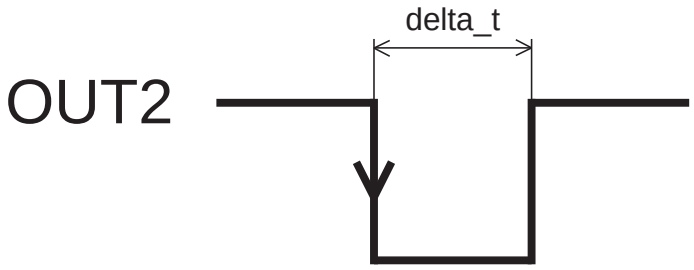
Software release in the interrupt control register as well as in the DIGIN interrupt mask register must be enabled!

9.3 Time controlled interrupt triggering

Interrupt block:
[Timer/Quartz oscillator]

-> Interrupt request when level goes from HIGH to LOW at OUT2 of counter 2 of IC 8254.

It is required to enable the software in the interrupt control register!



Interrupt at transition High ---> Low of OUT2
recommended: delta t = 5 µs

The interrupt requests of both of the interrupt blocks are handled in a shared interrupt service routine. You can detect the trigger‘s source reading the interrupt status register and the DIGIN interrupt input register, which one buffers the TTL input interrupt requests.
Interrupt requests remain held until they are reset by reading a source specific reset address!

9.4 Interrupt register

Following registers for control, read and write serve to program interrupts of the WITIO-PCI64:

- Interrupt control register (Port addresses BA + $A0, Write access):

Function: Enable or latch interrupt blocks
Format: Double word [c31...c00]

Bit c00: Enable (1) / Latch (0) DIGIN interrupt block
Bit c01: Enable (1) / Latch (0) Timer interrupt block
Bit c02...c31: any

int_contr_reg = $0000 -> both of the interrupt blocks latched (Status after PC‘S reset)

int_contr_reg = $0001 -> DIGIN interrupt block enabled*

int_contr_reg = $0002 -> DIGIN interrupt block latched
-> Timer interrupt block enabled

int_contr_reg = $0003 -> DIGIN interrupt block enabled
-> Timer interrupt block enabled

* at least one of the inputs of the DIGIN interrupt mask register has to be enabled additionally!

- DIGIN interrupt mask register (Port address BA + $A4, Write access):

Function: Enable or latch single TTL interrupt inputs
Format: Double word [m31...m00]

Bit m00: Enable (1) / Latch (0) TTL interrupt input DIN00
Bit m01: Enable (1) / Latch (0) TTL interrupt input DIN01
Bit m02: Enable (1) / Latch (0) TTL interrupt input DIN02
Bit m03: Enable (1) / Latch (0) TTL interrupt input DIN03
Bit m04: Enable (1) / Latch (0) TTL interrupt input DIN04
Bit m05: Enable (1) / Latch (0) TTL interrupt input DIN05
Bit m06: Enable (1) / Latch (0) TTL interrupt input DIN06
Bit m07: Enable (1) / Latch (0) TTL interrupt input DIN07

Bit m08...m31: any

digin_int_mask_reg = $00000000 -> all interrupt inputs latched
(Status after PC‘S reset)

digin_int_mask_reg = $00000001 -> Interrupt input
DIN00 enabled

- DIGIN interrupt input register (Port address BA + $B0, read access):

Function: Detection of the interrupt‘s source / TTL inputs
Format: Double word [e31...e00]

Bit e00: If 1 : Interrupt via TTL Interrupt input DIN00
Bit e01: If 1 : Interrupt via TTL Interrupt input DIN01
Bit e02: If 1 : Interrupt via TTL Interrupt input DIN02
Bit e03: If 1 : Interrupt via TTL Interrupt input DIN03
Bit e04: If 1 : Interrupt via TTL Interrupt input DIN04
Bit e05: If 1 : Interrupt via TTL Interrupt input DIN05
Bit e06: If 1 : Interrupt via TTL Interrupt input DIN06
Bit e07: If 1 : Interrupt via TTL Interrupt input DIN07

- DIGIN interrupt reset addresses (Port addresses BA + $C0 + 4* offs, read access):

Function: Reset of the TTL input interrupt
Format: double word [x31...x00] (x = undefined)

Reading a double word of the address BA + $C0 + 4* (offs 0...7) resets the via TTL input DIN00...DIN07 triggered interrupt.

Read address BA + $C0	:	Reset TTL interrupt DIN00
Read address BA + $C4	:	Reset TTL interrupt DIN01
Read address BA + $C8	:	Reset TTL interrupt DIN02
Read address BA + $CC	:	Reset TTL interrupt DIN03
Read address BA + $D0	:	Reset TTL interrupt DIN04
Read address BA + $D4	:	Reset TTL interrupt DIN05
Read address BA + $D8	:	Reset TTL interrupt DIN06
Read address BA + $DC	:	Reset TTL interrupt DIN07

- Timer interrupt reset (Port address BA + $BC, read access):

Function: Reset the timer‘s interrupt
Format: Double word [x31...x00] (x = undefined)

Reading a word of address BA + $BE resets the via timer triggered interrupt.

- interrupt status register (Port address LC + $4C, write access):

Function: Detection of the interrupt source
(TTL interrupt block / Timer interrupt)
Format: Double word [s31...s00]

Bit s02: if 1 : Interrupt via TTL interrupt block
Bit s05: if 1 : Interrupt via Timer

--> read double word "AND" $00000020 = $00000020 --> Timer interrupt

--> read double word "AND" $00000004 = $00000004 --> DIGIN interrupt

10. DOS® Programming

10.1 Programming of the WITIO-PCI64

You can find library functions and programming samples for access to the WITIO-PCI64 under DOS® with the accompanying software. Hardware components of the WITIO-PCI64 are programmed by access to port addresses. These result from the I/O basic address (and LC basic address) issued by the PCI BIOS for the WITIO-PCI64.
You can detect I/O basic address, LC basic address and actual port addresses of the single hardware components with help of initialization routines. Additionally you can access to more information such as IRQ number, localisation of the board in the bus system and the board‘s version. If you are working with a programming language not (yet) providing library functions, you can detect the PCI parameter of the WITIO-PCI64 with help of the program „WIT64SCA“ (-> in directory UTIL)

PCI Parameter:

- I/O basic address
- IRQ number
- LC basic address
- Bus number
- Device number
- Function number
- WITIO version

PCI Identification:

Device ID	=	$9050
Vendor ID	=	$10B5
Subsystem Vendor ID	=	$10B5
Subsystem ID	=	$11A2

10.2 Allocation of the port addresses

The port addresses of the single hardware components result from the I/O basic address (BA) and the LC basic address (LC) as follows:

Port/Register	BA + Offset	RD/WR
TTL input port A (DIN00...31)	BA + $40	RD
TTL input port B (DIN32...63)	BA + $44	RD
TTL output port A (DOUT00...31)	BA + $60	WR
TTL output port B (DOUT32...63)	BA + $64	WR
8254-Timer 0	BA + $80	RD/WR
8254-Timer 1	BA + $84	RD/WR
8254-Timer 2	BA + $88	RD/WR
8254-Timer control register	BA + $8C	RD/WR
Interrupt control register	BA + $A0	WR
DIGIN interrupt mask register	BA + $A4	WR
DIGIN interrupt input register	BA + $B0	RD
DIGIN interrupt reset base	BA + $C0	RD
Timer interrupt reset	BA + $BC	RD
Interrupt status register	LC + $4C	RD

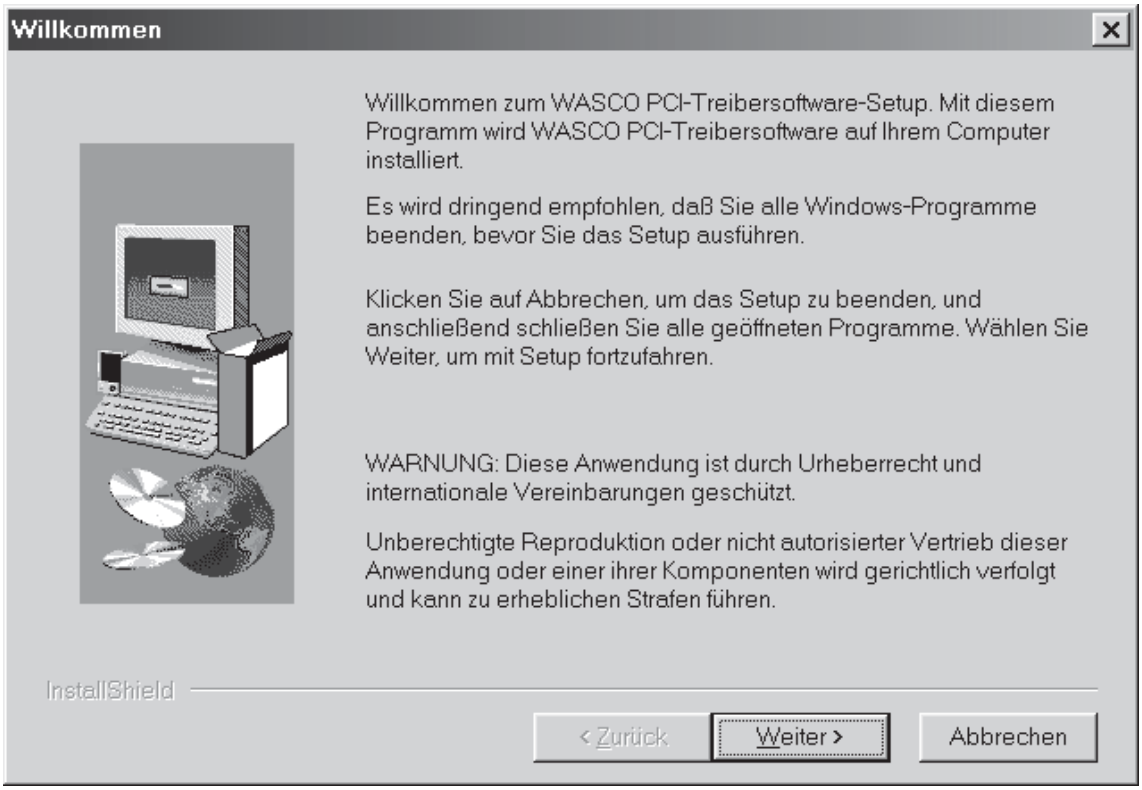
11. Windows® Programming

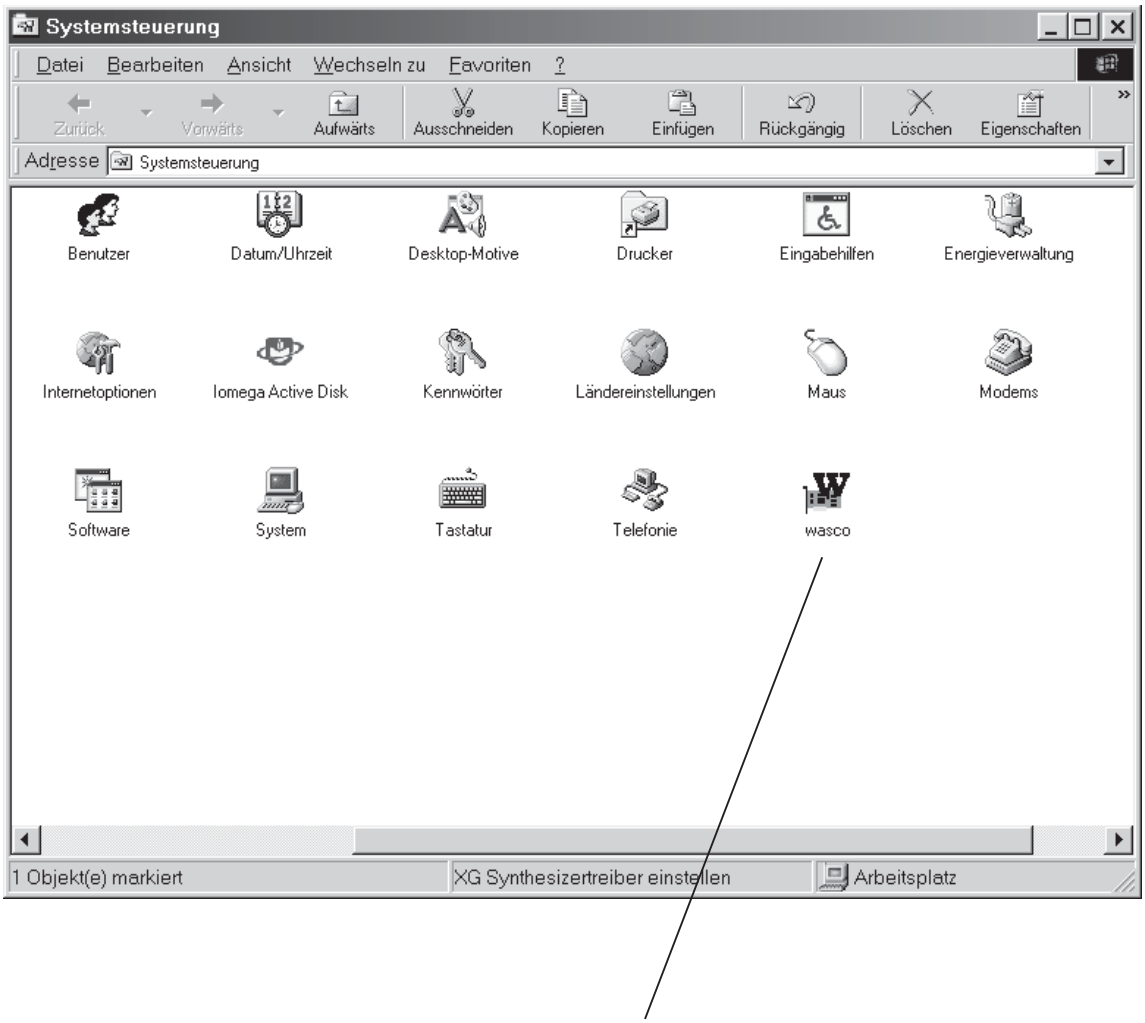
11.1 Programming of the WITIO-PCI64

To apply the board under Windows® it is necessary to install a special driver which allows the port access to the board.

11.2 Installation of the Windows® drivers

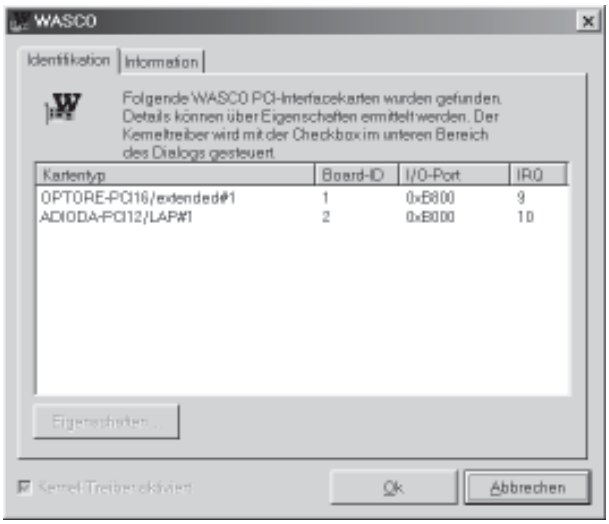
To install the Windows® driver please run the file "Setup.Exe" provided on the enclosed CD in folder driver and follow the installation instructions.





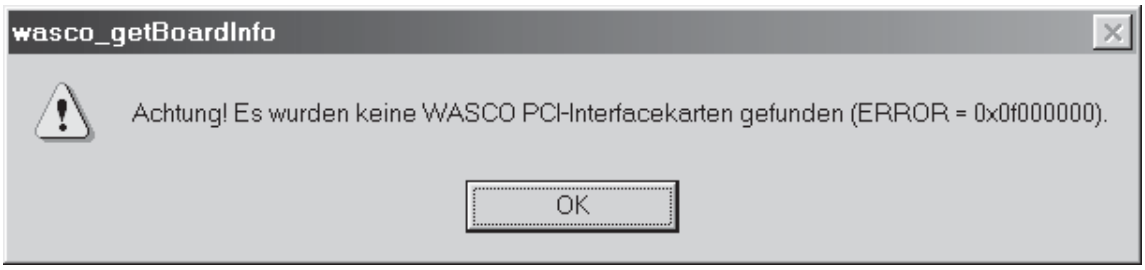
Once the driver software installation is completed you can find an icon in your system control panel for localisation of all **wasco**® PCI boards installed to your system.

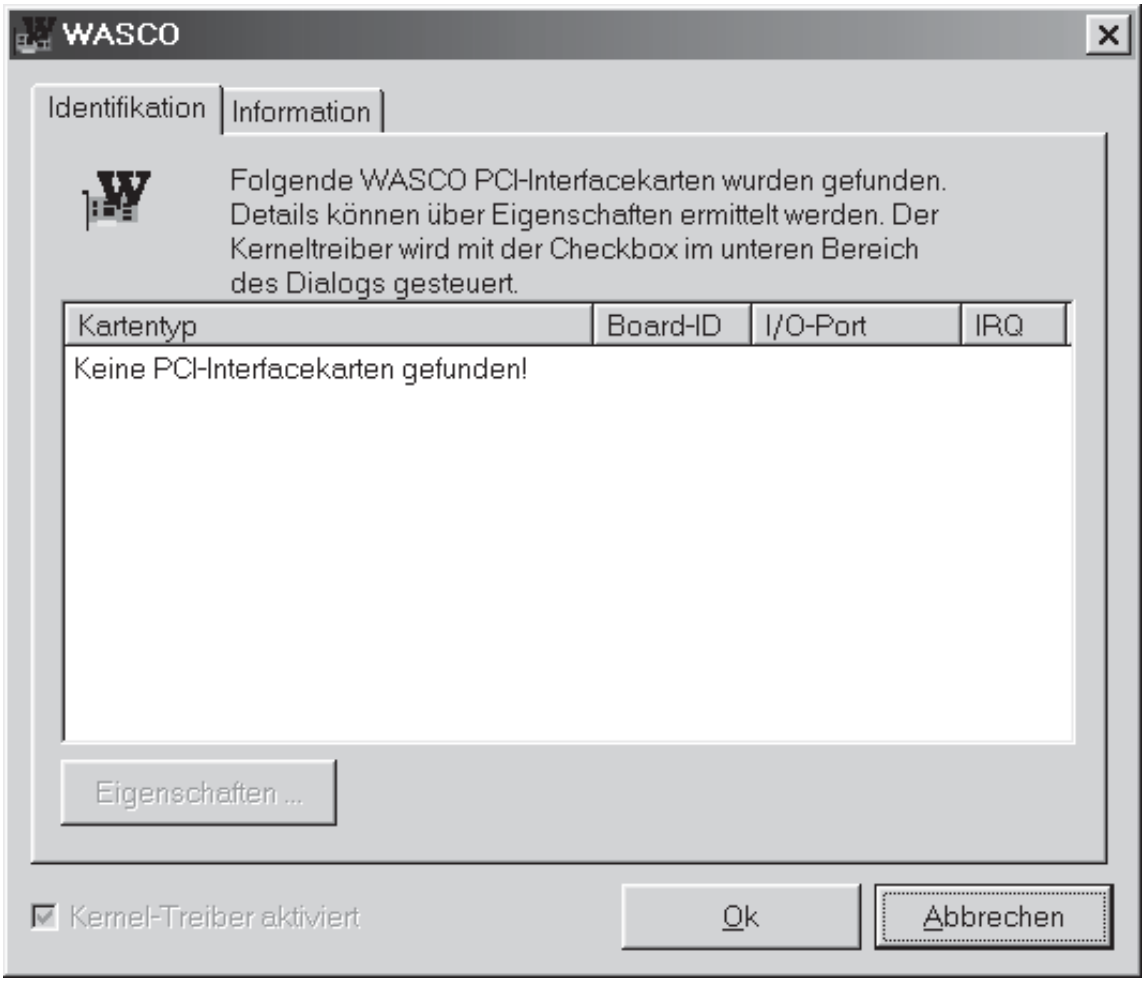
To monitor the boards start by double-clicking the "**wasco®**" Icon. Following window pops up: (An OPTORE-PCI16 and an ADIODA-PCI12 may be used here as an example)



This window presents card name, Board ID, I/O address and possible interrupt number of each board, if it was detected correctly. Furthermore you can find information about the driver version or localisation of the driver file clicking the tab „Information“.

If your card was not detected correctly, following error messages may pop-up:





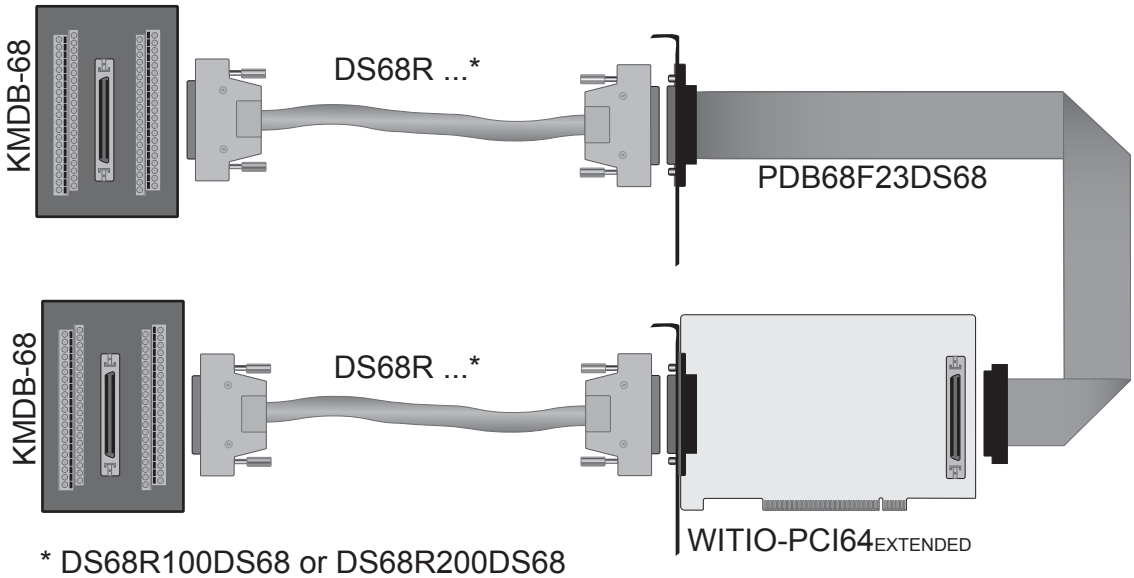
Please check chapter Troubleshooting for possible causes.

12. Accessories

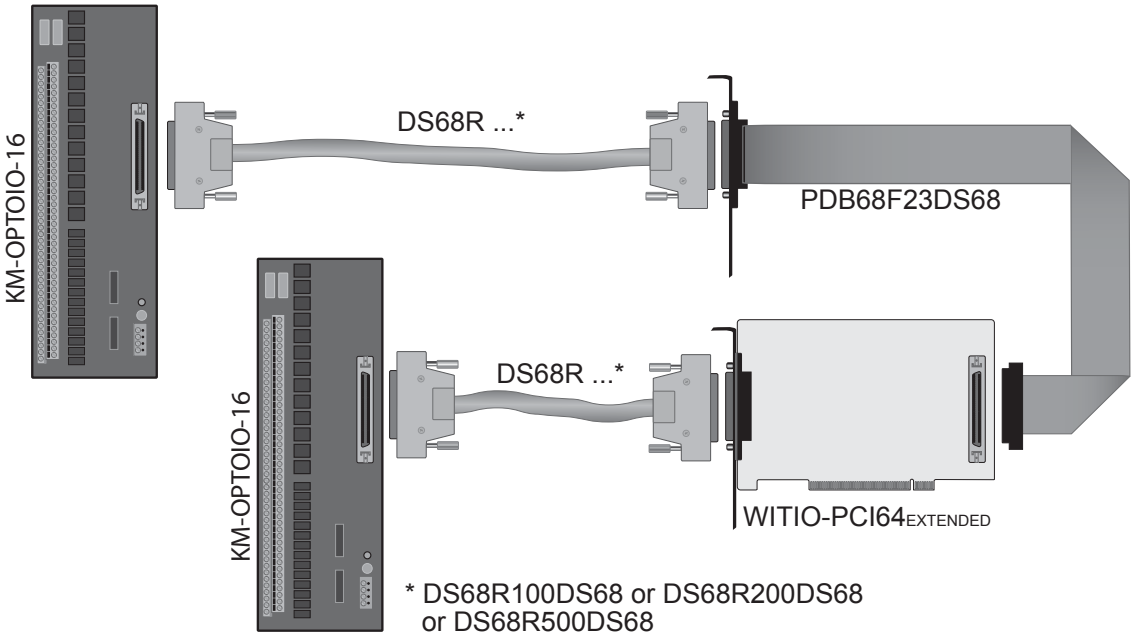
12.1 Suitable wasco® accessories

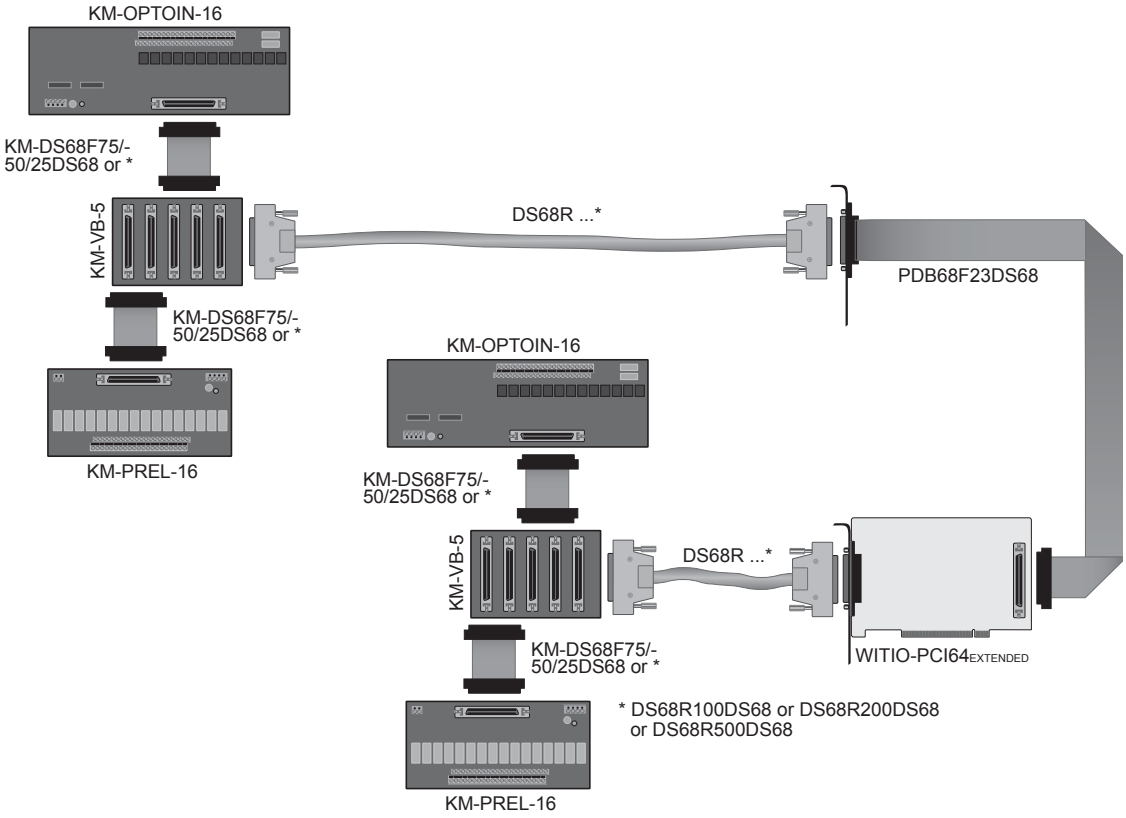
Connection parts	EDP No.
PDB68F23DS68 flat ribbon cable with connectors	A-498500
DS68R200DS68 connection cable	A-492400
DS68R500DS68 connection cable	A-492800
KMDB-68 Screw clamp module	A-494800
KM-OPTOIN-32 Optocoupler module	A-483600
KM-OPTOOUT-32 Optocoupler module	A-484600
KM-PREL-16 Relay module	A-485400
KM-REL-8 Relay module	A-486200
KM-VB-5 Connection module	A-488200

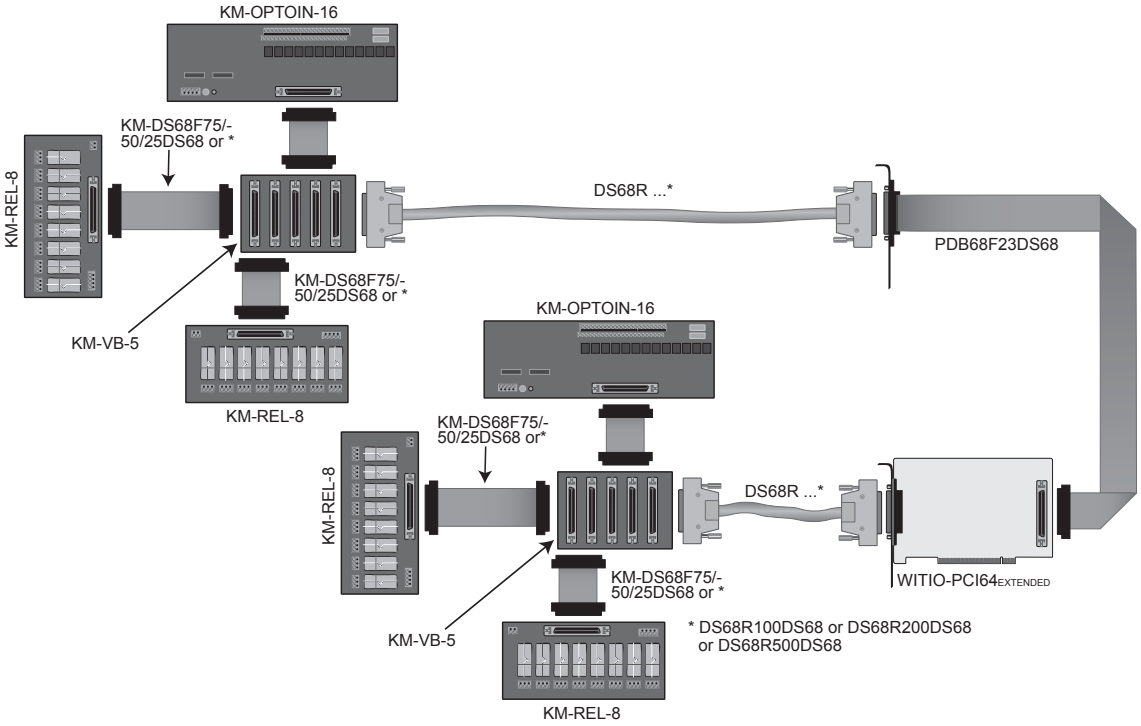
12.2 Connecting technique (application samples)

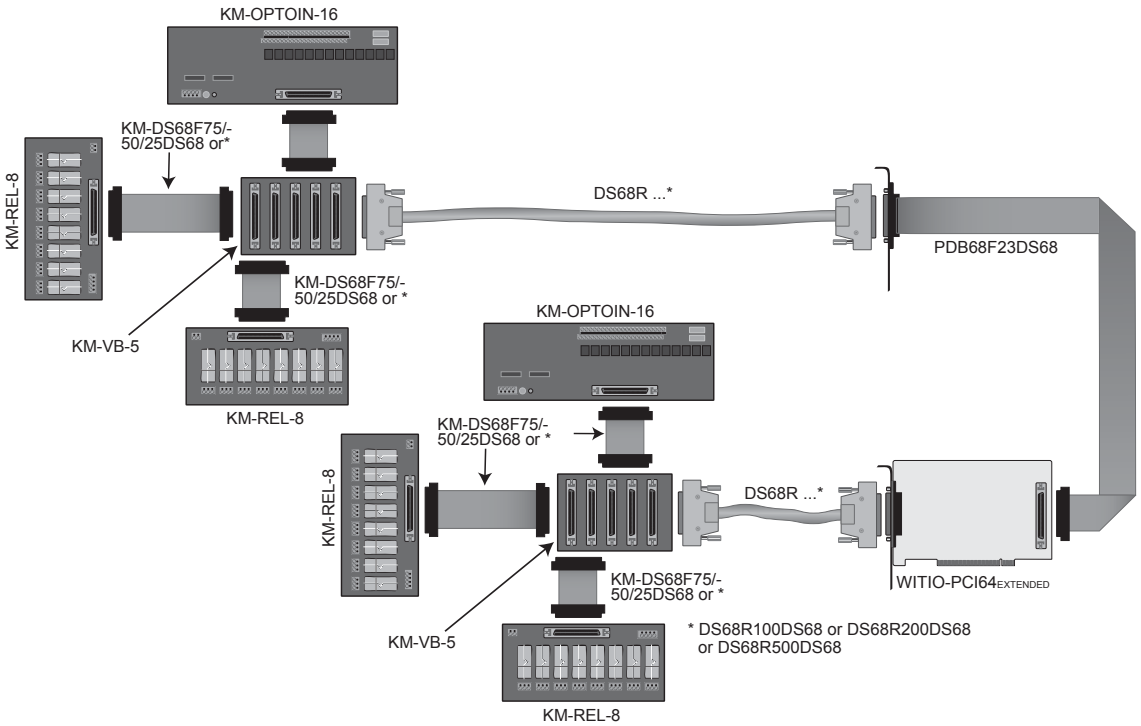


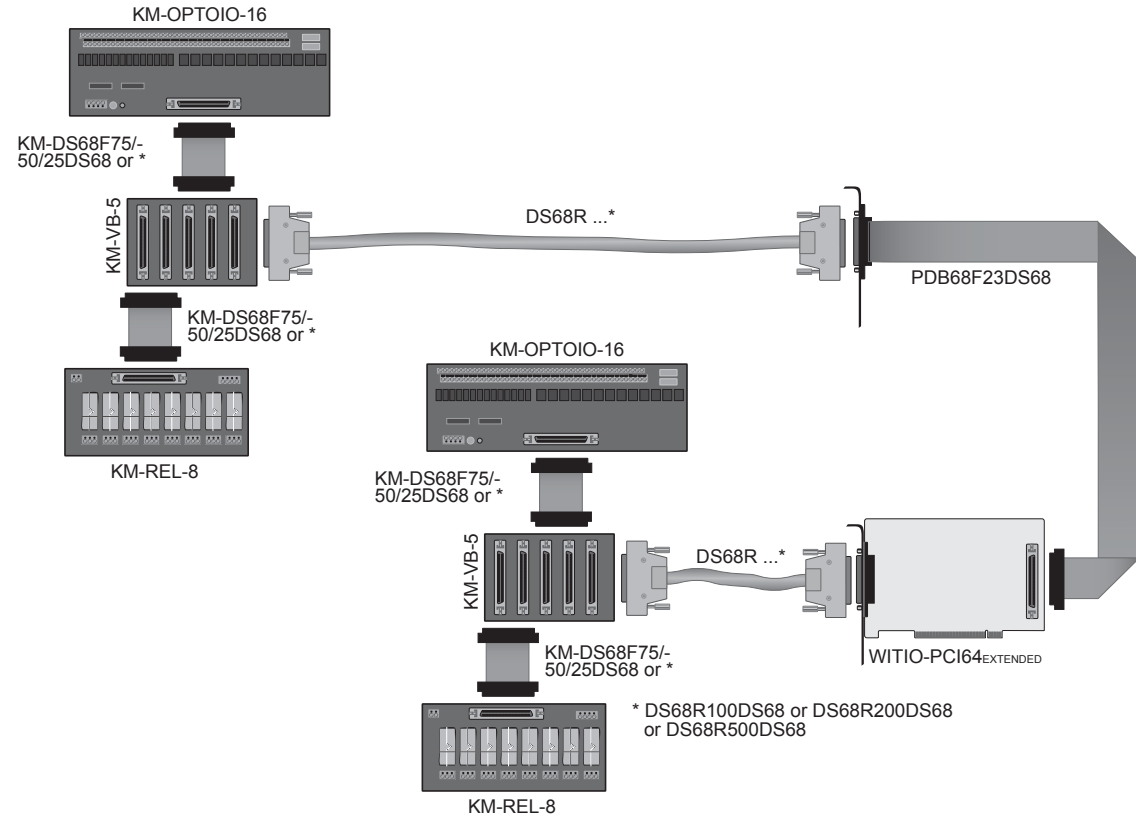
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68

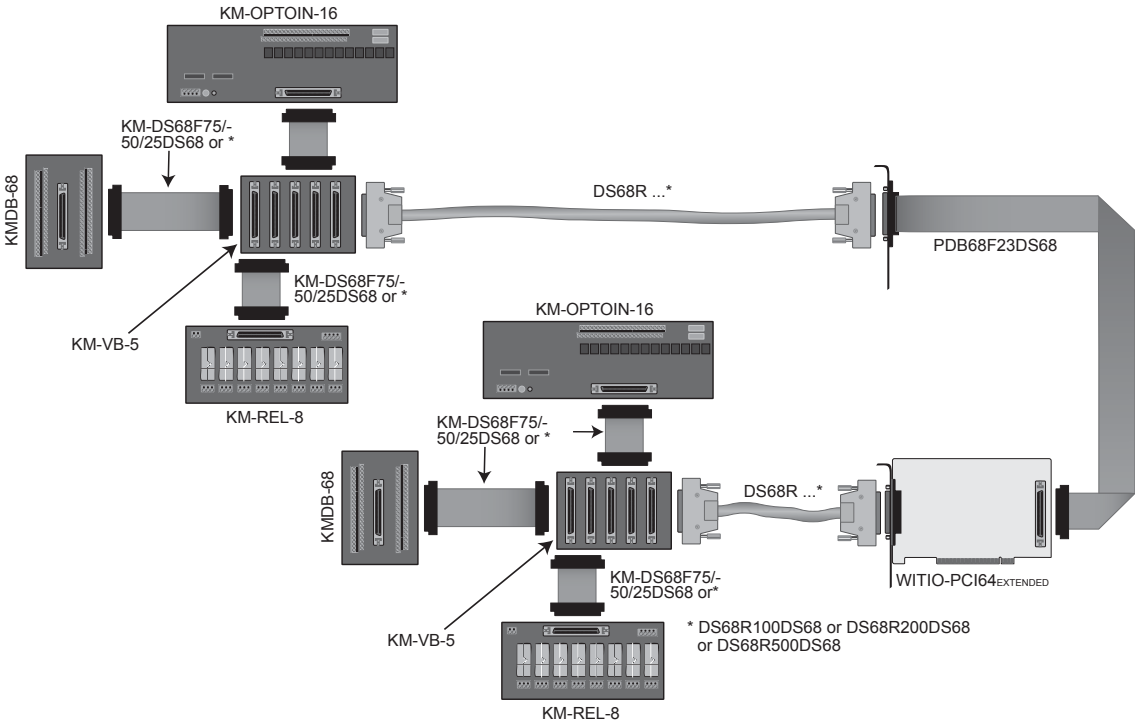

KM-OPTOIO-16
DS68R ...*
PDB68F23DS68
KM-OPTOIO-16
DS68R ...*
WITIO-PCI64EXTENDED
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68


KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-PREL-16
KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
DS68R ...*
KM-DS68F75/-
50/25DS68 or *
WITIO-PCI64EXTENDED
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-PREL-16


KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-REL-8
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
KM-REL-8
KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or*
KM-REL-8
DS68R ...*
KM-DS68F75/-
50/25DS68 or *
WITIO-PCI64EXTENDED
KM-VB-5
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-REL-8


KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or*
KM-REL-8
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
KM-REL-8
KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KM-REL-8
DS68R ...*
KM-DS68F75/-
50/25DS68 or *
WITIO-PCI64EXTENDED
KM-VB-5
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-REL-8


KM-OPTOIO-16
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-REL-8
KM-OPTOIO-16
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
DS68R ...*
KM-DS68F75/-
50/25DS68 or *
WITIO-PCI64EXTENDED
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-REL-8


KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KMDB-68
DS68R ...*
PDB68F23DS68
KM-DS68F75/-
50/25DS68 or *
KM-VB-5
KM-REL-8
KM-OPTOIN-16
KM-DS68F75/-
50/25DS68 or *
KMDB-68
DS68R ...*
KM-DS68F75/-
50/25DS68 or*
WITIO-PCI64EXTENDED
KM-VB-5
* DS68R100DS68 or DS68R200DS68
or DS68R500DS68
KM-REL-8

12.3 Individual components for customized assembly

Connection parts	EDP No.
SCSI-II plug 68-pin for flat ribbon cable wiring	A-553200
SCSI-II socket 68-pin for flat ribbon cable wiring	A-557200
Slot bracket with cut-out for 68-pin plug/socket	A-577800
Flat ribbon cable 68-pin	A-572800

13. Troubleshooting

Following find a short compilation of the most frequent known failure causes, which may occur during initialization or operation with WITIO-PCI64
Please check this list before you contact your dealer or distributor:

1. Is the WITIO-PCI64 inserted to the PCI slot properly?

2. Are all cable connections all right?

3. Is the fuse F1 blown?

4. Did your system detect the card correctly?
   Please therefore check all configurations of your computer or contact your system administrator (As this configurations are part of the BIOS system, we cannot expand on this issue here. We refer to your user‘s system manual)

5. Did you install the latest driver version of **wasco**® driver?

   Updates you can find here: http://www.messcomp.com
   http://www.wasco.de

14. Specifications

TTL Inputs
Channels: 64, TTL compatible
8 channels to be interrupt inputs

TTL Outputs
Channels: 64, TTL compatible

Loading capacity:	I_{OH} - 20 mA	2,0 V min.
	I_{OL} 20 mA	0,5 V max.

Quartz oscillator
4 MHz

Timer
IC: 8254 or 71054
3 * 16-bit down counter for time dependent interrupt triggering
Pulse cycles derived from Quartz oscillator

Connectors
2 * 68-pin SCSI-II socket

Bus System
32-bit PCI Bus (internal data bus 32 Bit)

Fuse
+ 5V 1 A Miniatur Fuse F1

Power Consumption
+ 5V typ. 950 mA

15. Product Liability Act

Information for Product Liability

The Product Liability Act (Act on Liability for Defective Products - ProdHaftG) in Germany regulates the manufacturer‘s liability for damages caused by defective products.

The obligation to pay compensation can be given, if the product’s presentation could cause a misconception of safety to a non-commercial end-user and also if the end-user is expected not to observe the necessary safety instructions handling this product.

It must therefore always be shown, that the non-commercial end-user was made familiar with the safety rules.

In the interest of safety, please always advise your non-commercial customer of the following safety instructions:

Safety instructions

The valid VDE-instructions must be observed, when handling products that come in contact with electrical voltage.

Especially the following instructions must be observed:
VDE100; VDE0550/0551; VDE0700; VDE0711; VDE0860.
The instructions are available from:
vde-Verlag GmbH
Bismarckstr. 33
10625 Berlin

* unplug the power cord before you open the unit or make sure, there is no current to/in the unit.

* You only may start up any components, boards or equipment, if they are installed inside a secure touch-protected casing before. During installation there must be no current to the equipment.

* Make sure that the device is disconnected from the power supply before using any tools on any components, boards or equipment. Any electric charges saved in components in the device are to be discharged prior.

* Voltaged cables or wires, which are connected with the unit, the components or the boards, must be tested for insulation defects or breaks. In case of any defect the device must be immediately taken out of operation until the defective cables are replaced.

* When using components or boards you must strictly comply with the characteristic data for electrical sizes shown in the corresponding description

* As a non-commercial end-user, if it is not clear whether the electrical characteristic data given in the provided description are valid for a component you must consult a specialist.

The compliance with building and safety instructions of all kinds (VDE, TÜV, industrial injuries corporation, etc.) are entirely the responsibility of the user/customer.

16. CE Confirmation

This is to certify, that the product

WITIO-PCI64EXTENDED
EDP Number A-461800

comply with the requirements of the relating EC directives. This declaration will lose its validity, if the instructions given in this manual for the intended use of the products are not fully complied with.

EN 5502 Class B
IEC 801-2
IEC 801-3
IEC 801-4
EN 50082-1
EN 60555-2
EN 60555-3

The following manufacturer is responsible for this declaration:

Messcomp Datentechnik GmbH
Neudecker Str. 11
83512 Wasserburg

given by

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 21.06.2005

Reference System for Intended Use

This PC extension board is not stand-alone device. The CE conformity only can be assessed when using additional computer components simultaneously. Thus the CE conformity only can be confirmed when using the following reference system for the intended use of the PC extension board:

Control Cabinet:	Vero IMRAK 3400	804-530061C 802-563424J 802-561589J
19" Casing:	Vero PC-Casing	145-010108L
19" Casing:	Additional Electronic	519-112111C
Motherboard:	GA-586HX	PIV 1.55
Floppy-Controller:	on Motherboard	
Floppy:	TEAC	FD-235HF
Grafic Card:	Advantech	PCA-6443
PC Card:	WITIO-PCI64$_{EXTENDED}$	A-461800